UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

A PROJECT REPORT

Submitted by

THAMARAI SELVAM T
THIRU MURUGAN A
YOGESHWARAN T
YUVARAJ SINGH T

In partial fulfilment for the award of the degree
of
BACHELOR OF TECHNOLOGY
in
INFORMATION TECHNOLOGY

MAHENDRA ENGINEERING COLLEGE

NAMAKKAL

ANNA UNIVERSITY CHENNAI 600035

NOVEMBER 2022

ANNA UNIVERSITY: CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report titled **"UNIVERSITY ADMIT ELIGIBILITY PREDICTOR"**

is the bonafide work of " **THAMARAI SELVAM T(**6113192071103**), THIRU MURUGAN A(**6113192071106**), YOGESHWARAN T(**611319207119**),YUVARAJ SINGH(**6113192071120) **"**who carried out the project work under my supervision.

SIGNATURE OF
INDUSTRIAL MENTOR
MR.P.SANDESH

SIGNATURE OF
MENTOR
Dr.K.RADHA

TABLE OF CONTENT

CHAPTER No. **TITLE**

ABSTRACT

Data is the most important asset for any organization, which is then processed to produce useful information. Machine learning and big data techniques are widely used in industrial sectors to create useful models to generate more profits and grow businesses. In recent years, much research work has been done using Big Data techniques on education data to improve education system. Machine Learning and Big Data can be useful in predicting student enrollment, teaching performance, student performance, identifying groups of students with similar behavior. However, the manual profile verification process is time-consuming, tedious, and error-prone; due to the inherent volume and complexity of the data. In this study, the combination of linear and non-linear machine learning algorithms; Logistic regression, decision tree, k-NN and Naïve Bayes were selected to perform target class prediction for unseen survey observation. Since the models built in this work predict the probability of students being admitted to any university based on student data collected by any marketing agency, the combined models are collectively known as admission predictors. Administrative managers at any academic institution can use this type of application to discover and analyze patterns affecting student admissions and recommend innovative strategies for improvement. improve admission. Such an application not only plays an important role in the management, but also helps the management to reframe the marketing criteria for the overall development of the academic institution.

CHAPTER-1

INTRODUCTION

1.1 Project Overview

We are currently students aiming for future studies, and we wanted to know the odds of getting admission in a university according to our scores before applying. It takes input from the user, namely, their GRE score, TOFEL score, current CGPA, SOP, LOP, and university rating and it outputs the result according to the trained Machine Learning model. In this hands-on guided project, we will train regression models to find the probability of a student getting accepted into a particular university based on their profile. This project could be practically used to get the university acceptance rate for individual students using web application and to understand regression and classification problems

- To grab insights from data through visualization.
- Applying different ML algorithms to determine the probability of acceptance in a particularuniversity.
- Evaluation metrics
- Build a web application using the Flask framework.

1.2 PURPOSE

- 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 to a particularuniversity.
- This analysis should also help students who are currently preparing or will be preparing to get abetter idea.

CHATER-2
LITERATURE SURVEY

2.1 EXISTING SYSTEM

In today's world many students are often worried about their chances of admission to university. The main moto of the project is help students to short listing universities with their marks. 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.

Given certain metrics of a student, our task is to predict the probability of the student getting accepted into graduate programs. Statistically, we have seen many students pursue their education away from their native countries. Generally, as the students don't have much of an idea about the procedures, requirements, and details of the universities, they seek help from education consultancy firms to help them successfully secure admission to the universities which are best suitable for their profiles. For this, they have to invest huge amounts of money in consultancy fees. The aim of this research is to develop a system using Applied Data Science.

University prediction would be the easiest mode to predict the university/college person is applicable for as well as it would be unbiased and totally transparent. Individually would no more need to depend upon the consultancies who may be slightly deviated from the list of colleges/universities that may be having contracts with them. Moreover, applying to only colleges/universities where the student has a genuine chance would reduce the application process. Additionally, living expense of the area where colleges/university is located would also be provided on the website.

2.2 REFERENCES

PAPER 1

TITLE: PREDICTION FOR UNIVERSITY ADMISSION USING MACHINE LEARNING

AUTHOR NAME: Chithara Apoorva D A

DESCRIPTION: Students who want to do masters in America have to write GRE and TOEFL/IELTS. Once they have attended the exams, they have to prepare their SOP and LOR which are one of the crucial factors they have to consider. These LOR and SOP play a vital role if the student was looking for any scholarship.

ADVANTAGES:

- Give more accurate options for foreign universities.
- Train more quickly, especially with bigger datasets.

DISADVANTAGES:

- Models, particularly those trained on CPUs, may be computationally costly and time-consuming.
-

PAPER 2

TITLE: GRADE MACHINE LEARNING SUPPOURT FOR GRADUATE ADMISSIONS

AUTHOR NAME: Austin Waters

DESCRIPTION: Waters and Mikkulainen proposed an astounding that asides in posting affirmation application as per the degree of acknowledgement and upgrades the presentation of inspecting applications utilizing measurable AI.

ADVANTAGE:

- GRADE has minimal software and hardware dependencies.

- It is implemented in Python with the Pandas and Skit-learn packages, which are open source and freely available.

DISADVANTAGE:

- It is used to display the result as a dotted graph.
- It is to predict the approximate value.

PAPER 3

TITLE:PREDICTING UNDERGRADUATE ADMISSION

AUTHOR NAME: Md. Portiku Zaman

DESCRIPTION: Here, the authors apply three machine learning algorithms XGBoost, lightGBM, and GBM on a collected dataset to estimate the probability of getting admission to the university after attending or before attending the admission test.

ADVANTAGES:

- Easy to implement evaluation of the conditional probability is simple.
- Individuals would no longer need to rely on consultancies that may have contracts with schools and universities that are somewhat off the list.

DISADVANTAGES:

- It is not always true that the conditional independence assumption is true. The feature often exhibits some sort of dependence.

PAPER 4

TITLE: GRATUATE ADMISSION PREDICTION USING

MACHINELEARNING

AUTHOR NAME: Sara Aljasami

DESCRIPTION: This will assist students to know in advance if they have a chance to get accepted. The machine learning models are multiple linear regression, K-nearest neighbour, random forest, and multilayer perceptron.

ADVANTAGES:

- It uses many algorithms like random forest, K-nearest neighbour, multi layer preception.
- The simplest way to determine whether a person is eligible for a university or college as well as being completely objective and transparent.

DISADVANTAGES:

- Contrarily, linear regression presumes that the relationship between the dependent and independent variables is linear. The implies that it considers their relationship to be linear. The independence of the qualities is assumed.

2.3 PROBLEM STATEMENT DEFINITION

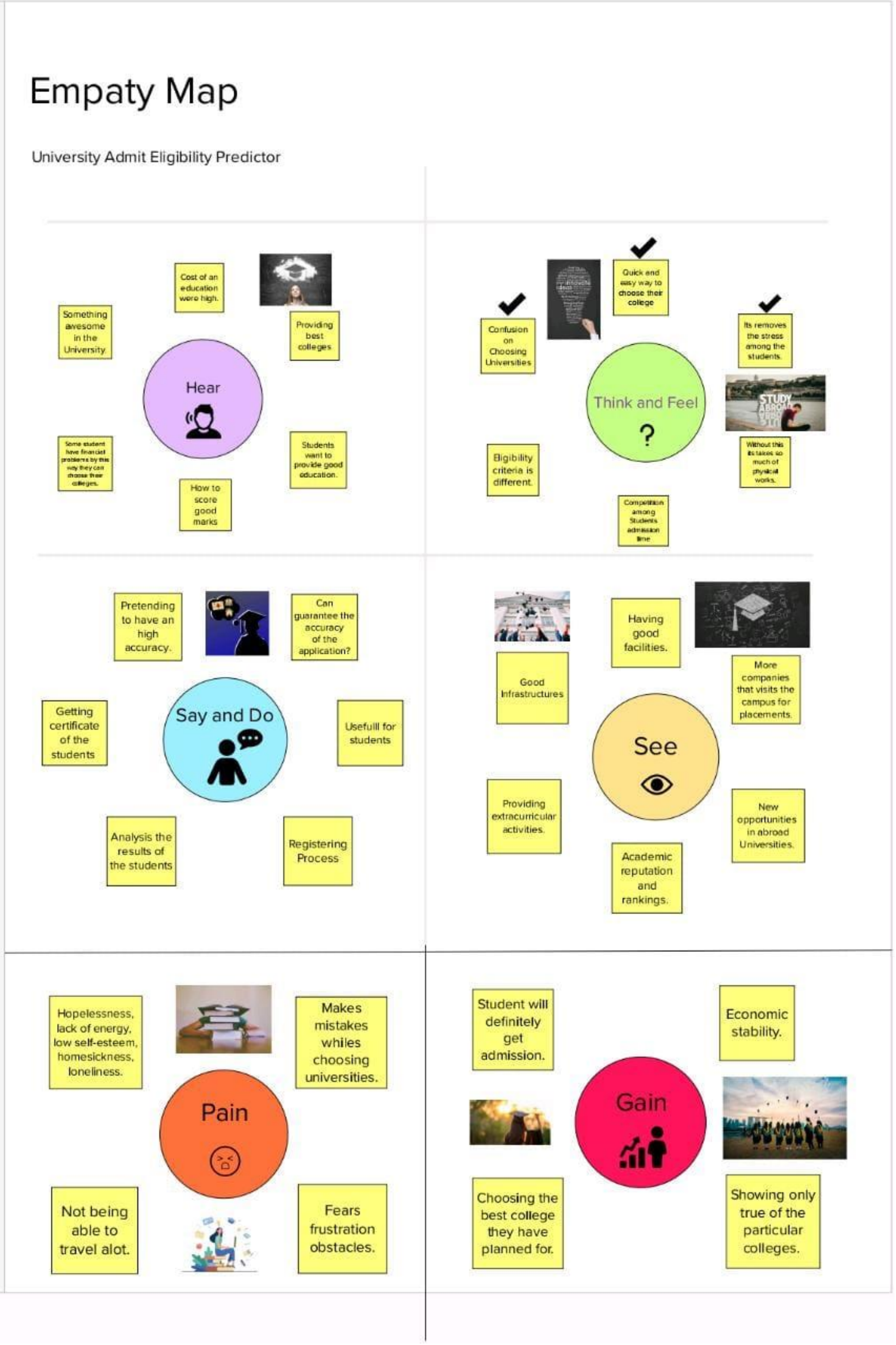
Problem Statement (PS)	**I am (Customer)**	**I'm trying to**	**But**	**Because**	**Which makes me feel**
PS-1	A student	Choose a good university	I am unaware about cutoff	I can't find people to guide me	Dejected
PS-2	A student	Enrol in a master degree	I am unaware about visa formalities	I can't find trusted websites	Frustrated
PS-3	A student	Enrol in a master	I am unaware about	I can't find helping	Depressed

		degree	scholarship opportunities	organisations	
PS -4	A student	Choose good university with well infrastructure	I am unaware about infrastructure and placement	I can’t find people to guide me	Anxious

CHAPTER-3

IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP



Empaty Map

University Admit Eligibility Predictor

Hear

Something awesome in the University.

Cost of an education were high.

Providing best colleges.

Some student have financial problems by this way they can choose their colleges.

How to score good marks

Students want to provide good education.

Think and Feel

?

✓ Confusion on Choosing Universities

✓ Quick and easy way to choose their college

✓ Its removes the stress among the students.

Eligibility criteria is different.

Competition among Students admission time

Without this its takes so much of physical works.

Say and Do

Pretending to have an high accuracy.

Can guarantee the accuracy of the application?

Getting certificate of the students

Usefull for students

Analysis the results of the students

Registering Process

See

Good Infrastructures

Having good facilities.

More companies that visits the campus for placements.

Providing extracurricular activities.

Academic reputation and rankings.

New opportunities in abroad Universities.

Pain

Hopelessness, lack of energy, low self-esteem, homesickness, loneliness.

Makes mistakes whiles choosing universities.

Not being able to travel alot.

Fears frustration obstacles.

Gain

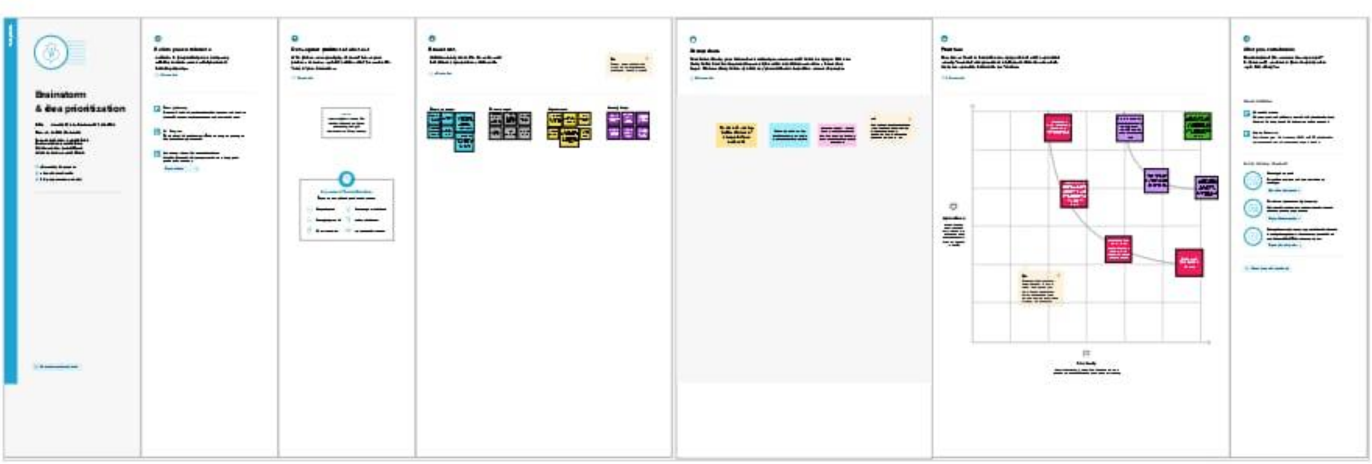
Student will definitely get admission.

Economic stability.

Choosing the best college they have planned for.

Showing only true of the particular colleges.

3.2 BRAINSTROM

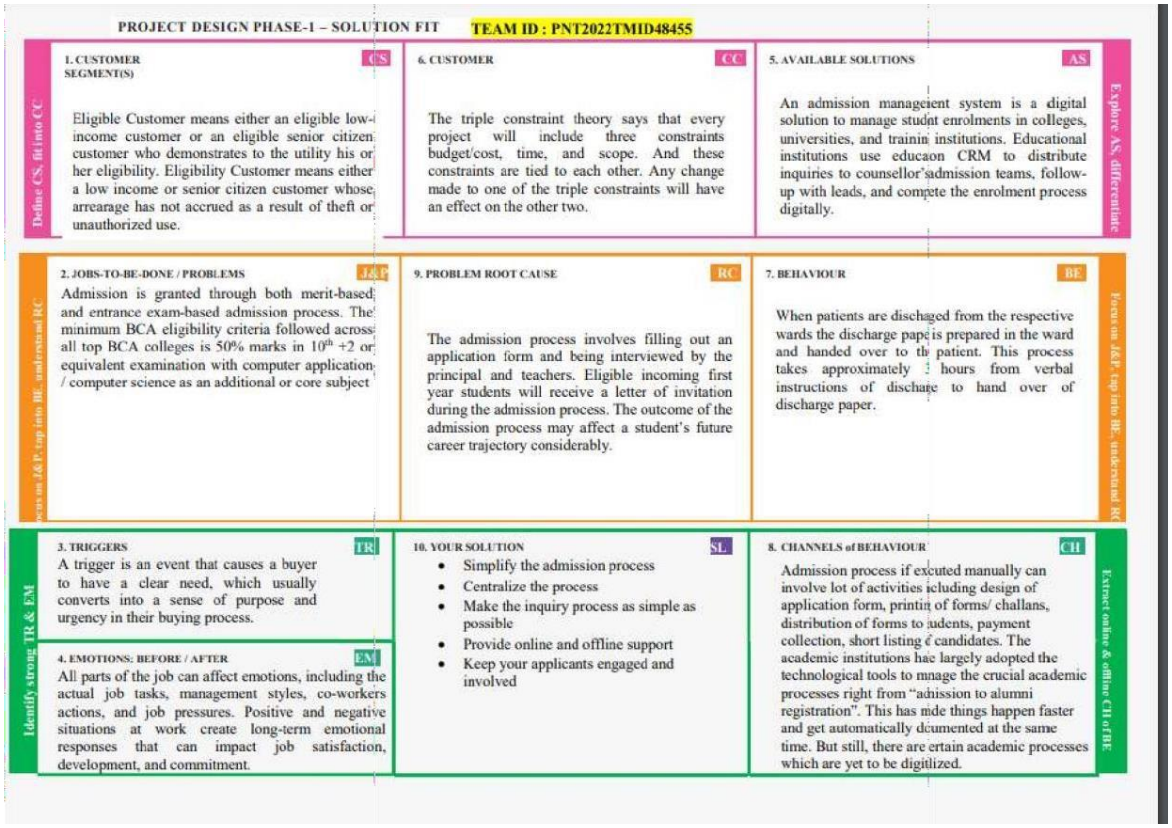


3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to besolved)	To predict the universities for the students towho is going to higher studies.
2.	Idea / Solution description	It will help the students to get admission for under graduate degrees from the top universitiesand well infrastructure laboratories. This analysis is helpful who is unaware of choosing UG degrees after secondary education. It will predict the students admission to the respected universities based on their secondary education marks.
3.	Novelty / Uniqueness	This website has to predict the universities inside India. And also give various informationabout the universities. Also to list the universities in theranking list.

4.	Social Impact / Customer Satisfaction	The website will reduce the panic and unawareness among students. It will reduce ourtime, travel, and costs. It will give the exact approximate prediction based on students secondary education marks.
5.	Business Model (Revenue Model)	Universities shall find the websites in order tomaintain it. This website will predict and display the exact results to the students.
6.	Scalability of the Solution	A future update shall have chat spacecomprising faculty, current students

3.4 PROBLEM SOLUTION FIT


Define CS, fit into CC
Explore AS, differentiate
ocus on J&P, tap into BE, understand RC
Focus on J&P, tap into BE, understand RC
Identify strong TR & EM
Extract online & offline CH of BE

PROJECT DESIGN PHASE-1 – SOLUTION FIT TEAM ID : PNT2022TMID48455

1. CUSTOMER SEGMENT(S) CS

Eligible Customer means either an eligible low-income customer or an eligible senior citizen customer who demonstrates to the utility his or her eligibility. Eligibility Customer means either a low income or senior citizen customer whose arrearage has not accrued as a result of theft or unauthorized use.

6. CUSTOMER CC

The triple constraint theory says that every project will include three constraints budget/cost, time, and scope. And these constraints are tied to each other. Any change made to one of the triple constraints will have an effect on the other two.

5. AVAILABLE SOLUTIONS AS

An admission manageient system is a digital solution to manage studnt enrolments in colleges, universities, and trainin institutions. Educational institutions use educaon CRM to distribute inquiries to counsellor'sadmission teams, follow-up with leads, and compete the enrolment process digitally.

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Admission is granted through both merit-based and entrance exam-based admission process. The minimum BCA eligibility criteria followed across all top BCA colleges is 50% marks in 10th +2 or equivalent examination with computer application / computer science as an additional or core subject

9. PROBLEM ROOT CAUSE RC

The admission process involves filling out an application form and being interviewed by the principal and teachers. Eligible incoming first year students will receive a letter of invitation during the admission process. The outcome of the admission process may affect a student's future career trajectory considerably.

7. BEHAVIOUR BE

When patients are dischaged from the respective wards the discharge pape is prepared in the ward and handed over to th patient. This process takes approximately hours from verbal instructions of dischage to hand over of discharge paper.

3. TRIGGERS TR

A trigger is an event that causes a buyer to have a clear need, which usually converts into a sense of purpose and urgency in their buying process.

4. EMOTIONS: BEFORE / AFTER EM

All parts of the job can affect emotions, including the actual job tasks, management styles, co-workers actions, and job pressures. Positive and negative situations at work create long-term emotional responses that can impact job satisfaction, development, and commitment.

10. YOUR SOLUTION SL

- Simplify the admission process
- Centralize the process
- Make the inquiry process as simple as possible
- Provide online and offline support
- Keep your applicants engaged and involved

8. CHANNELS of BEHAVIOUR CH

Admission process if excuted manually can involve lot of activities icluding design of application form, printin of forms/ challans, distribution of forms to udents, payment collection, short listing ć candidates. The academic institutions hae largely adopted the technological tools to mnage the crucial academic processes right from "adission to alumni registration". This has mde things happen faster and get automatically dcumented at the same time. But still, there are ertain academic processes which are yet to be digitlized.

CHAPTER-4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Calculate admission Prediction	Enter GPA, TOFEL, GRE Scores
FR-2	Check information about theuniversity	Visit the website of the respected university and to contact the alumni and faculties of those universities.
FR-3	Watch campus tour	Check guide for visa application and other procedures
FR-4	Check financial assistance lab	Check scholarship eligibility and application procedure
FR-5	Information about the universityand location	Refer official websites and contact directly to thefaculty.

4.2 NON FUNCTIONAL REQUIREMENTS

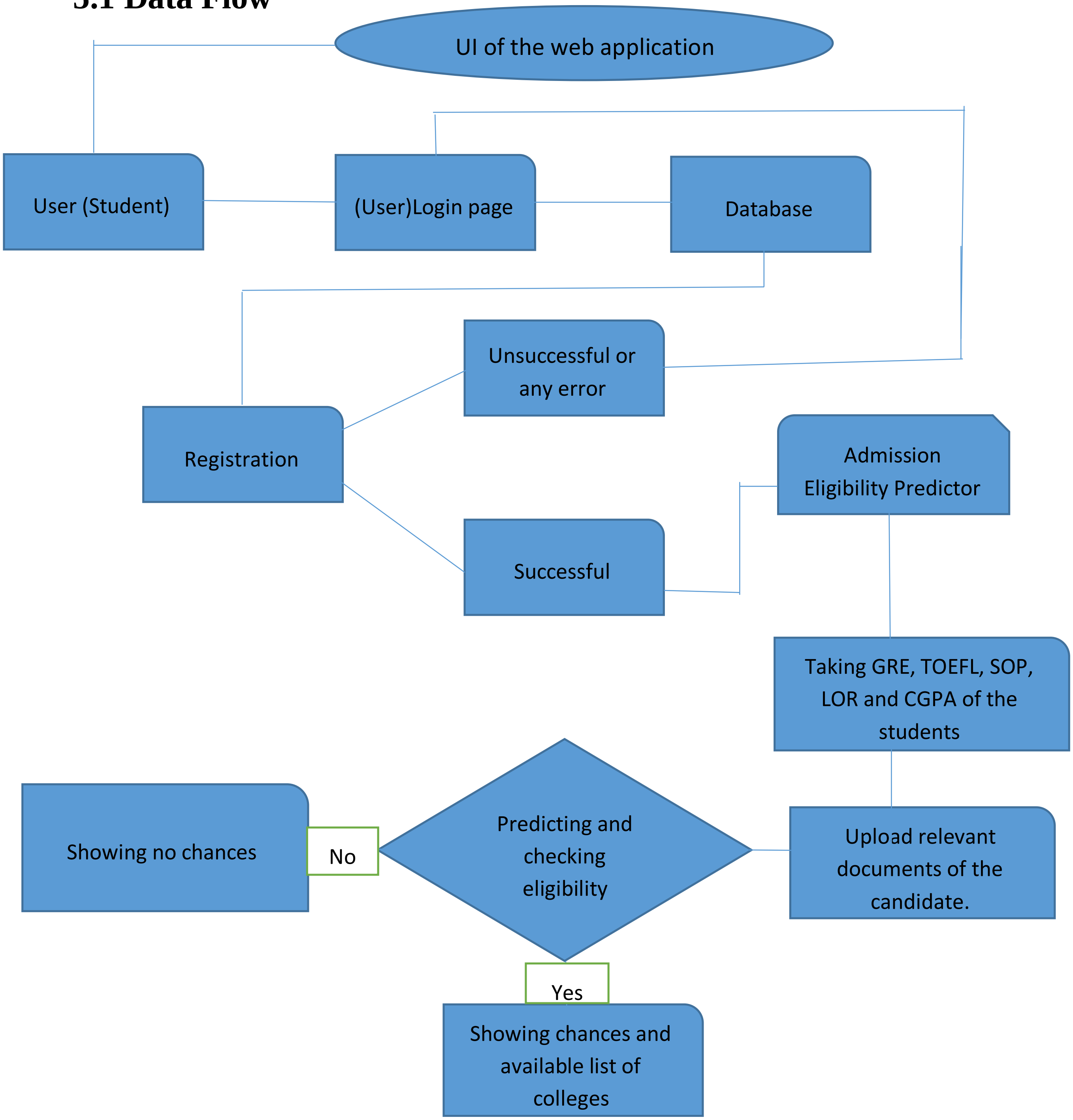
Following are the non-functional requirements of the proposed solution.

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	Usability	The UI/UX enhances the user experience. The entire journey of the customer throughout the application will be smooth and user-friendly approach to the user.
NFR-2	Security	It is the safest application you never used it beforebecause it doesn’t store your data.
NFR-3	Reliability	The system will give you to the most accurate andexact results.
NFR-4	Performance	The index page supporting more than 1000 users per hour must provide some seconds delay response timein a chrome desktop browser, including the rendering of text and images and over an LTE connection.
NFR-5	Availability	The admission predictor will be available to users 99.9 percent of the time every month
NFR-6	Scalability	The system must be scalable enough to support morethan one lacks visits at the same time while maintaining optimal performance.

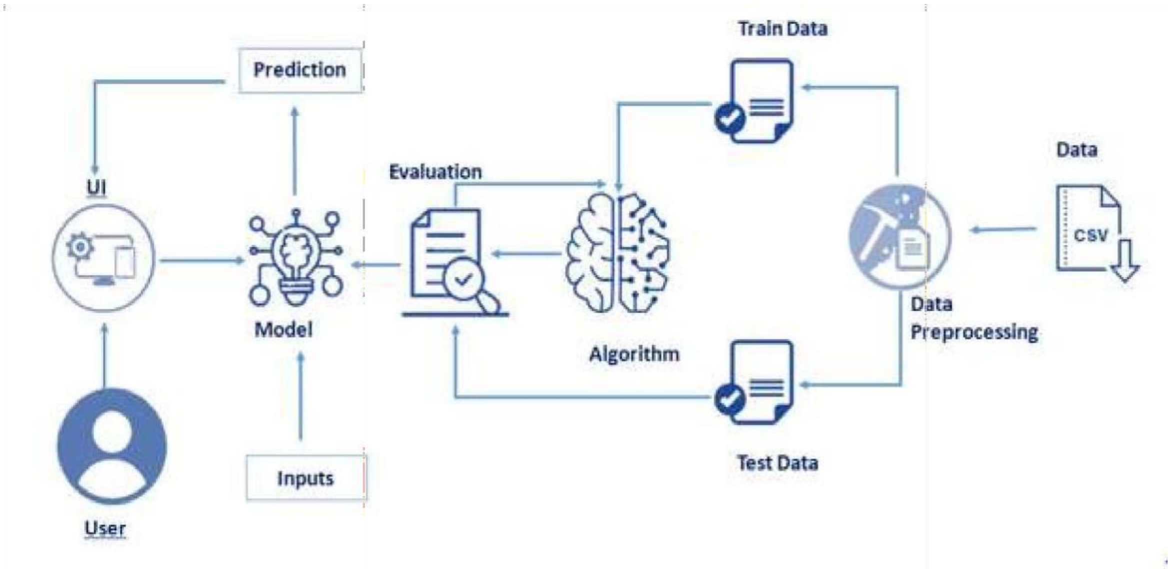
CHAPTER-5

PROJECT DESIGN

5.1 Data Flow :


UI of the web application
User (Student)
(User)Login page
Database
Unsuccessful or any error
Registration
Admission Eligibility Predictor
Successful
Taking GRE, TOEFL, SOP, LOR and CGPA of the students
Showing no chances
No
Predicting and checking eligibility
Upload relevant documents of the candidate.
Yes
Showing chances and available list of colleges

5.2 SOLUTION ARCHETECTURE



5.3 USER STORIES

Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority
Customer (Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	We can access my account / **User** dashboard	High
		USN-2	As a user, I will receive confirmation email once I have registered for the application	We can receive confirmation email & click confirm	High
		USN-3	As a user, I can register for the application through Facebook	We can register & access the dashboard with	Low

				Facebook Login	
		USN-4	As a user, I can register for the application through Gmail	We can register and access the dashboard	Medium
	Login	USN-5	As a user, I can log into the application by entering email & password	We can access various pages	High
	Dashboard	USN-6	As a user , I can search for various universities	We can access several pages	High
	Search	USN-7	As a user , I can search for Universities with different field	We can receive information related to universities on various locations	High
	View	USN-8	As a user , I can view the University details	We will get the information on seat availability, eligibility criteria.	High
	Receive notification	USN-9	As a user, I will receive notifications about the Suggested universities based on student marks	We will get frequent updates of the preferred universities	Low
	Chat with expert	USN-10	As a user, I can chat with the expert for clarifications	We can clear my doubts through chat with expert option	Medium
Admin	Analysis	USN-11	As an admin, I will analyze the given dataset	We can analyze the dataset	High
	Predict	USN-12	As an admin, I will predict the admission	We can predict eligibility for admission	High

CHAPTER-6

PROJECT PLANNING AND SCHEDULING

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 can register for the application by entering my email, password, and confirming my password.	2	High	2
Sprint -1		USN-2	As a user, I will receive confirmation email once I have registered for the application	1	High	1
Sprint -1		USN-3	As a user, I can register for the application through Gmail	1	High	1
Sprint-1	Login	USN-4	As a user, I can log into the application by entering email & password	3	High	3

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint -2	Update Profile	USN-5	As a user, after logging in, I will have to update my profile by providing	5	High	5

			all the required details			
Sprint-3	Choose university	USN-6	As a user, I will be able to view the list of Universities that the students are eligible toapply	5	Medium	5
Sprint-3	Choose course	USN-7	As a user, I will be able to view the details of Admissionprocess like date and venue ofcertification verification	2	Low	2
Sprint-4	Admission process	USN-8	As a user, I will be able to view the list of courses thatthe students are eligible to apply	3	High	3
Sprint-1	Authentication	USN-9	As a admin, the login credential of the user is authenticated my me	2	High	2

6.2 SPRINT DELIVARY SCHUDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (ason Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	2 0	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	2 0	50 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	2 0	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	2 0	19 Nov 2022

CHAPTER-7

CODING AND SOLUTIONING

7.1 FEATURE 1

FLASK CODE

```
#University Admit Eligibility predictor
from flask import Flask, render_template, request
from math import ceil
import pickle

app = Flask(__name__)
model = pickle.load(open("model1.pkl","rb"))

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

@app.route('/predict',methods=["GET","POST"])
def predict():
    gre=(eval(request.form["gre"])-290)/(340-290)
    toefl=(eval(request.form["toefl"])-92)/(120-92)
    rating=(eval(request.form["rating"])-1.0)/4.0
    sop=(eval(request.form["sop"])-1.0)/4.0
    lor=(eval(request.form["lor"])-1.0)/4.0
    cgpa=(eval(request.form["cgpa"])-6.7)/(10.0-6.7)
    research=request.form["research"]
    if research == 1:
        research = 1
    else:
        research = 0
    pred = [[gre,toefl,rating,sop,lor,cgpa,research]]
    result=model.predict(pred)
    if result > 0.5:
        return render_template("chance.html",msg="Congratulation, you are eligible. As the probability is "+str(ceil(result[0]*100))+"%")
    return render_template("nochance.html",msg="Sorry, Unfortunately you aren't eligible. As the probability is "+str(ceil(result[0]*100))+"%")


if __name__ == "__main__":
    app.run(port=8801)
```

7.2 FEATURE 2

INDEX

```
<!DOCTYPE html>


<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Universitty Admit Eligibility Predictor</title>
    <!-- <link rel="stylesheet" href="{{ url_for("static", filename="css/styles.css") }}"> -->
    <link rel="stylesheet" href="./static/css/styles.css">
    <!-- UIkit CSS -->
    <!-- <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/uikit@3.15.12/dist/css/uikit.min.css" /> -->
    <!-- UIkit JS -->
    <!-- <script src="https://cdn.jsdelivr.net/npm/uikit@3.15.12/dist/js/uikit.min.js"></script> -->
    <!-- <script src="https://cdn.jsdelivr.net/npm/uikit@3.15.12/dist/js/uikit-icons.min.js"></script> -->
</head>
<body>
    <header class="uk-width-1-1">
        <h1><center> <b> UNIVERSITY ADMIT ELIGIBILITY<br> PREDICTOR SYSTEM</b></center></h1>
        <h5>Enter your details and get probability of your admission</h5>
    </header>
    <section class="uk-flex uk-flex-center uk-padding-large uk-width-1-1">
        <div class="card uk-padding">

            <div class="uk-divider-icon"></div>
            <div class="form">
                <form action="/predict" method="post">
                    <div class="label">
                        <label for="gre"><b>GRE Score</b></label>
                    </div>
                    <div class="input">
                        <input type="text" name="gre" id="gre" placeholder="Enter your score between 0 - 360" required/>
                    </div>
                    <div class="label">
                        <label for="toefl"><b>TOEFL Score</b></label>
                    </div>
                    <div class="input">
                        <input type="text" name="toefl" id="toefl" placeholder="Enter your TOEFL score between 0 - 120" required/>
                    </div>
                    <div class="label">
                        <label for="rating"><b>University Rating</b></label>
                    </div>
                    <div class="input">
                        <select name="rating" id="rating" required>
                            <option disabled>select a rating</option>
                            <option value="1">1</option>
                            <option value="2">2</option>
                            <option value="3">3</option>
                            <option value="4">4</option>
                            <option value="5">5</option>
                        </select>
                    </div>
                    <div class="label">
                        <label for="sop"><b>SOP</b></label>
```

```
                </div>
                <div class="input">
                    <input type="text" name="sop" id="sop" placeholder="Enter your SOP score between 0 - 5" required/>
                </div>
                <div class="label">
                    <label for="lor"><b>LOR</b></label>
                </div>
                <div class="input">
                    <input type="text" name="lor" id="lor" placeholder="Enter your LOR score between 0 - 5" required/>
                </div>
                <div class="label">
                    <label for="cgpa"><b>CGPA</b></label>
                </div>
                <div class="input">
                    <input type="text" name="cgpa" id="cgpa" placeholder="Enter your CGPA score between 0 - 10" required/>
                </div>
                <div class="label">
                    <label for="research"><b>Research</b></label>
                </div>
                <div class="input">
                    <select name="research" id="research" required>
                        <option value="1">Yes</option>
                        <option value="0">No</option>
                        </select>
                    </div>
                    <div class="btn uk-flex uk-flex-center uk-margin-top">
                        <input type="submit" value="Predict" />
                    </div>
                </form>
            </div>

        </div>
    </section>
</body>
<script src={{url_for("static",filename="js/scripts.js")}}></script>
</html>
```

CHANCE

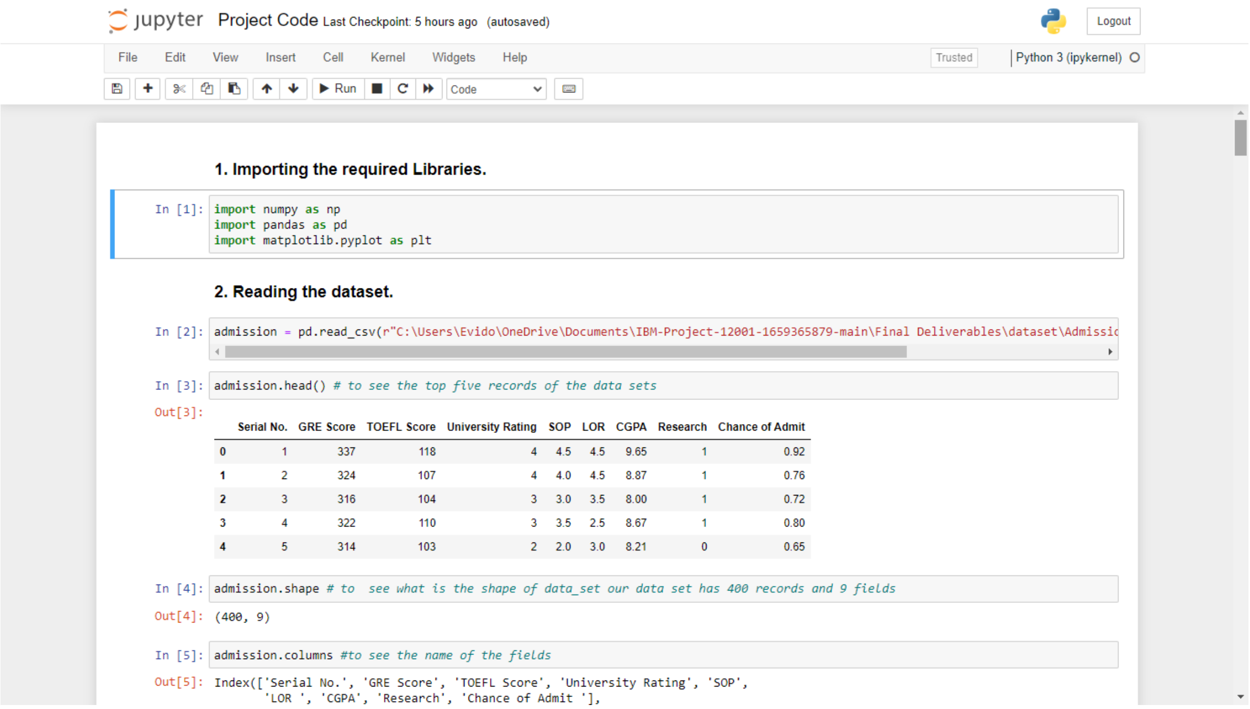
```
<!DOCTYPE html>

<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Universitty Admit Eligibility Predictor</title>
    <link rel="stylesheet" href="{{ url_for("static", filename="css/styles.css") }}">
    <!-- <link rel="stylesheet" href="../static/css/styles.css"> -->
    <!-- UIkit CSS -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/uikit@3.15.12/dist/css/uikit.min.css" />
    <!-- UIkit JS -->
    <script src="https://cdn.jsdelivr.net/npm/uikit@3.15.12/dist/js/uikit.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/uikit@3.15.12/dist/js/uikit-icons.min.js"></script>
</head>
<body>
    <header class="uk-width-1-1">
        <h1><center> <b> UNIVERSITY ADMIT ELIGIBILITY<br> PREDICTOR SYSTEM</b></center></h1>
    </header>
    <section class="uk-flex uk-flex-center uk-flex-column uk-padding-large uk-width-1-1">
        <div class="card uk-padding uk-box-shadow-xlarge">
            <div class="img">
                <!-- <img src="../static/images/chance.jpeg" alt="response">-->
                <img src="{{ url_for("static", filename="images/chance.jpg") }}" alt="Chance">
            </div>
            <div class="msg uk-margin-top uk-padding uk-width-1-1 uk-text-justify">
                <p>{{msg}}<p>
            </div>
            <div class="nav uk-flex uk-flex-center uk-margin-top uk-width-1-1 uk-link-toggle uk-text-large">
                <a class="uk-link-heading" href="/"><b>Back to Form</b></a>
            </div>
        </div>
    </section>
</body>
</html>
```

NO CHANCE

```
<!DOCTYPE html>


<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Universitty Admit Eligibility Predictor</title>
    <link rel="stylesheet" href="{{ url_for("static", filename="css/styles.css") }}">
    <!-- <link rel="stylesheet" href="../static/css/styles.css">  -->
    <!-- UIkit CSS -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/uikit@3.15.12/dist/css/uikit.min.css" />
    <!-- UIkit JS -->
    <script src="https://cdn.jsdelivr.net/npm/uikit@3.15.12/dist/js/uikit.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/uikit@3.15.12/dist/js/uikit-icons.min.js"></script>
</head>
<body>
    <header class="uk-width-1-1">
```

```
        <h1><center> <b> UNIVERSITY ADMIT ELIGIBILITY<br> PREDICTOR SYSTEM</b></center></h1>


    </header>
    <section class="uk-flex uk-flex-center uk-flex-column uk-padding-large uk-width-1-1">
        <div class="card uk-padding uk-box-shadow-xlarge">
            <div class="img">
                <img src="{{ url_for("static", filename="images/nochance.jpg") }}" alt="No chance">
            </div>
            <div class="msg uk-margin-top uk-padding uk-width-1-1 uk-text-justify">
                {{msg}}
            </div>
            <div class="nav uk-flex uk-flex-center uk-margin-top uk-width-1-1  uk-link-toggle uk-text-large">
                <a class="uk-link-heading" href="/"><b>Back to Form</b></a>
            </div>
        </div>
    </section>
</body>
    </html>
```

7.3 DATABASE SCHEMA



1. Importing the required Libraries.

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

2. Reading the dataset.

In [2]:
```
admission = pd.read_csv(r"C:\Users\Evido\OneDrive\Documents\IBM-Project-12001-1659365879-main\Final Deliverables\dataset\Admissic
```

In [3]:
```
admission.head() # to see the top five records of the data sets
```

Out[3]:

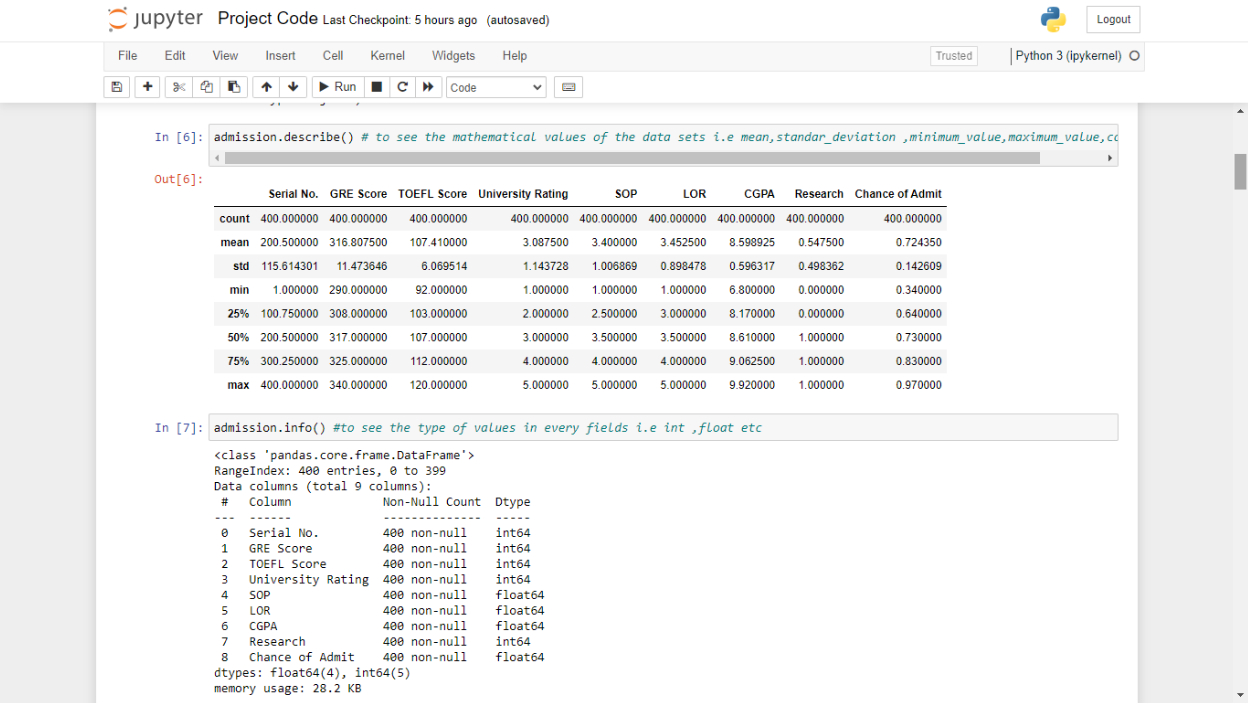
	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	1	337	118	4	4.5	4.5	9.65	1	0.92
1	2	324	107	4	4.0	4.5	8.87	1	0.76
2	3	316	104	3	3.0	3.5	8.00	1	0.72
3	4	322	110	3	3.5	2.5	8.67	1	0.80
4	5	314	103	2	2.0	3.0	8.21	0	0.65

In [4]:
```
admission.shape # to  see what is the shape of data_set our data set has 400 records and 9 fields
```

Out[4]:
```
(400, 9)
```

In [5]:
```
admission.columns #to see the name of the fields
```

Out[5]:
```
Index(['Serial No.', 'GRE Score', 'TOEFL Score', 'University Rating', 'SOP',
       'LOR ', 'CGPA', 'Research', 'Chance of Admit '],
```



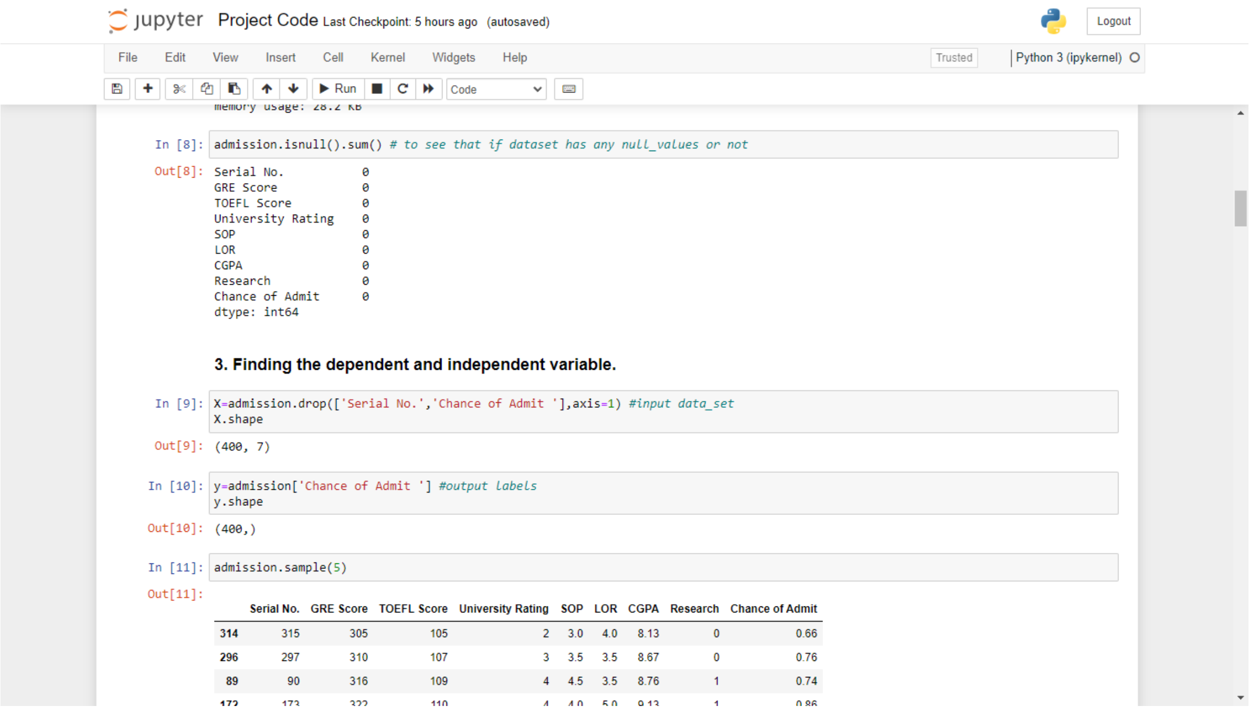
In [6]:
```
admission.describe() # to see the mathematical values of the data sets i.e mean,standar_deviation ,minimum_value,maximum_value,cc
```

Out[6]:

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
count	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000
mean	200.500000	316.807500	107.410000	3.087500	3.400000	3.452500	8.598925	0.547500	0.724350
std	115.614301	11.473646	6.069514	1.143728	1.006869	0.898478	0.596317	0.498362	0.142609
min	1.000000	290.000000	92.000000	1.000000	1.000000	1.000000	6.800000	0.000000	0.340000
25%	100.750000	308.000000	103.000000	2.000000	2.500000	3.000000	8.170000	0.000000	0.640000
50%	200.500000	317.000000	107.000000	3.000000	3.500000	3.500000	8.610000	1.000000	0.730000
75%	300.250000	325.000000	112.000000	4.000000	4.000000	4.000000	9.062500	1.000000	0.830000
max	400.000000	340.000000	120.000000	5.000000	5.000000	5.000000	9.920000	1.000000	0.970000

In [7]:
```
admission.info() #to see the type of values in every fields i.e int ,float etc
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 9 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Serial No.         400 non-null    int64  
 1   GRE Score          400 non-null    int64  
 2   TOEFL Score        400 non-null    int64  
 3   University Rating  400 non-null    int64  
 4   SOP                400 non-null    float64
 5   LOR                400 non-null    float64
 6   CGPA               400 non-null    float64
 7   Research           400 non-null    int64  
 8   Chance of Admit    400 non-null    float64
dtypes: float64(4), int64(5)
memory usage: 28.2 KB
```



```
memory usage: 28.2 KB
```

```
In [8]: admission.isnull().sum() # to see that if dataset has any null_values or not
```

```
Out[8]: Serial No.           0
        GRE Score            0
        TOEFL Score          0
        University Rating    0
        SOP                  0
        LOR                  0
        CGPA                 0
        Research             0
        Chance of Admit      0
        dtype: int64
```

3. Finding the dependent and independent variable.

```
In [9]: X=admission.drop(['Serial No.','Chance of Admit '],axis=1) #input data_set
        X.shape
```

```
Out[9]: (400, 7)
```

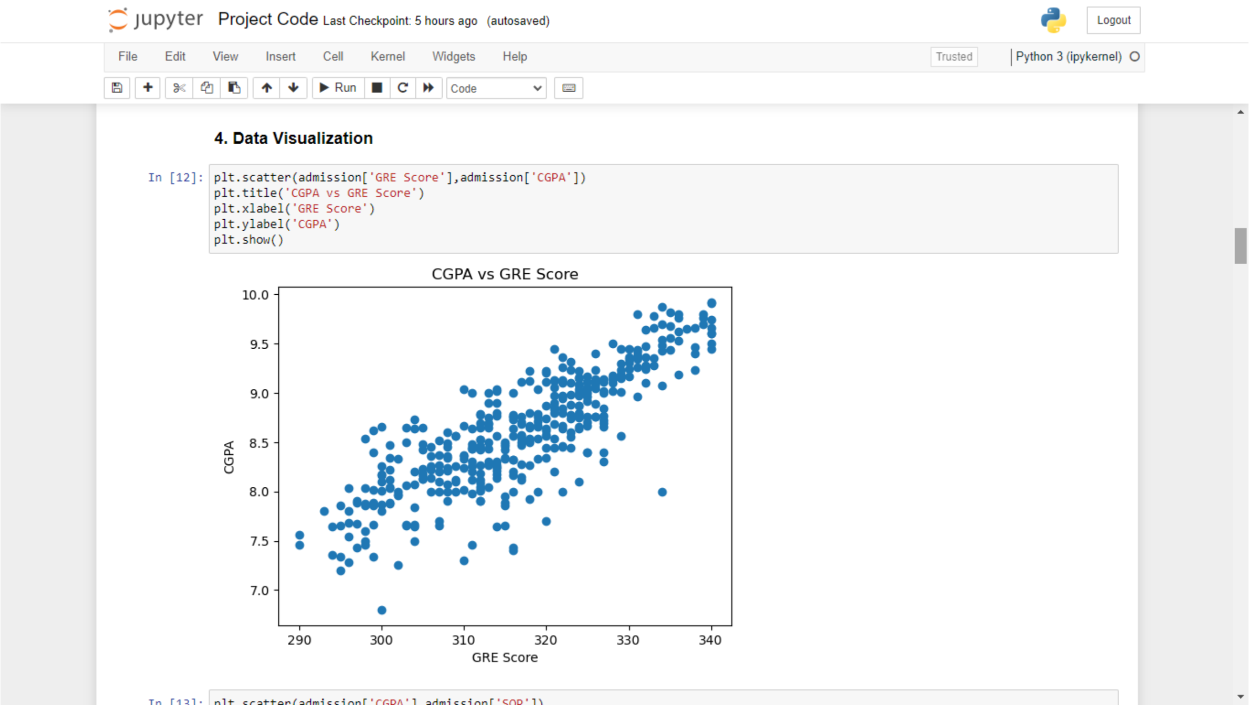
```
In [10]: y=admission['Chance of Admit '] #output labels
         y.shape
```

```
Out[10]: (400,)
```

```
In [11]: admission.sample(5)
```

Out[11]:

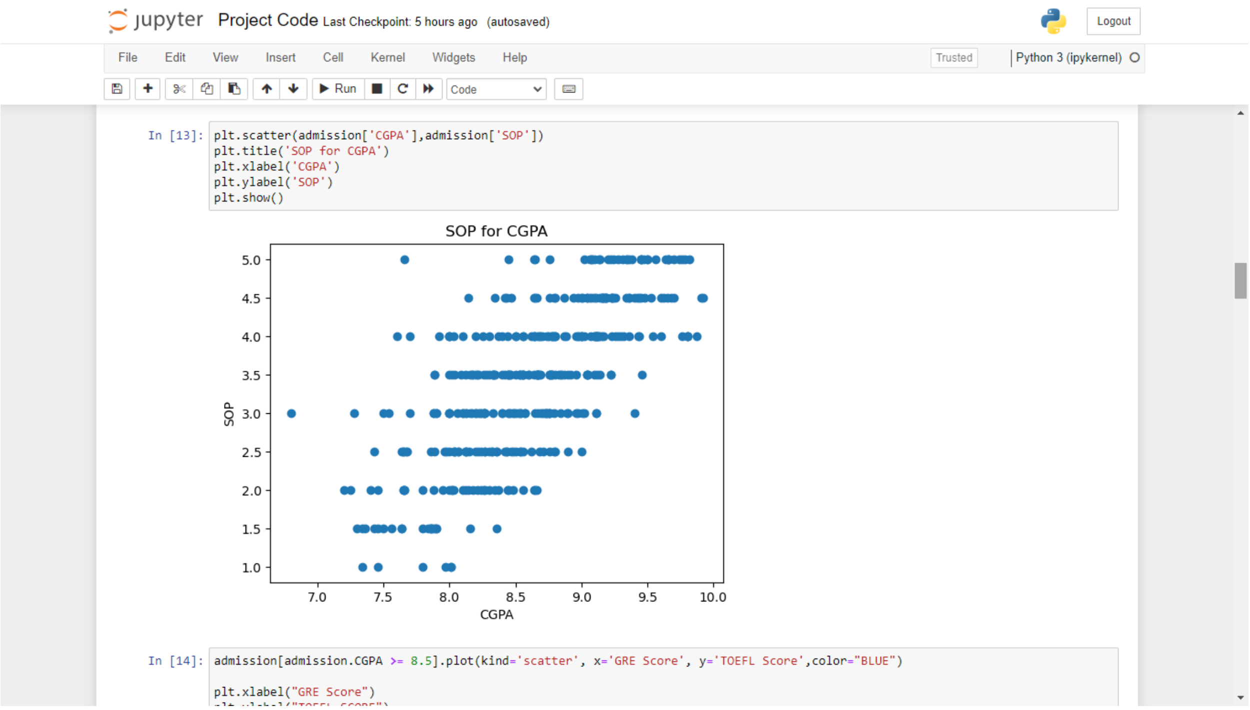
	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
314	315	305	105	2	3.0	4.0	8.13	0	0.66
296	297	310	107	3	3.5	3.5	8.67	0	0.76
89	90	316	109	4	4.5	3.5	8.76	1	0.74
172	173	322	110	4	4.0	5.0	9.13	1	0.86



4. Data Visualization

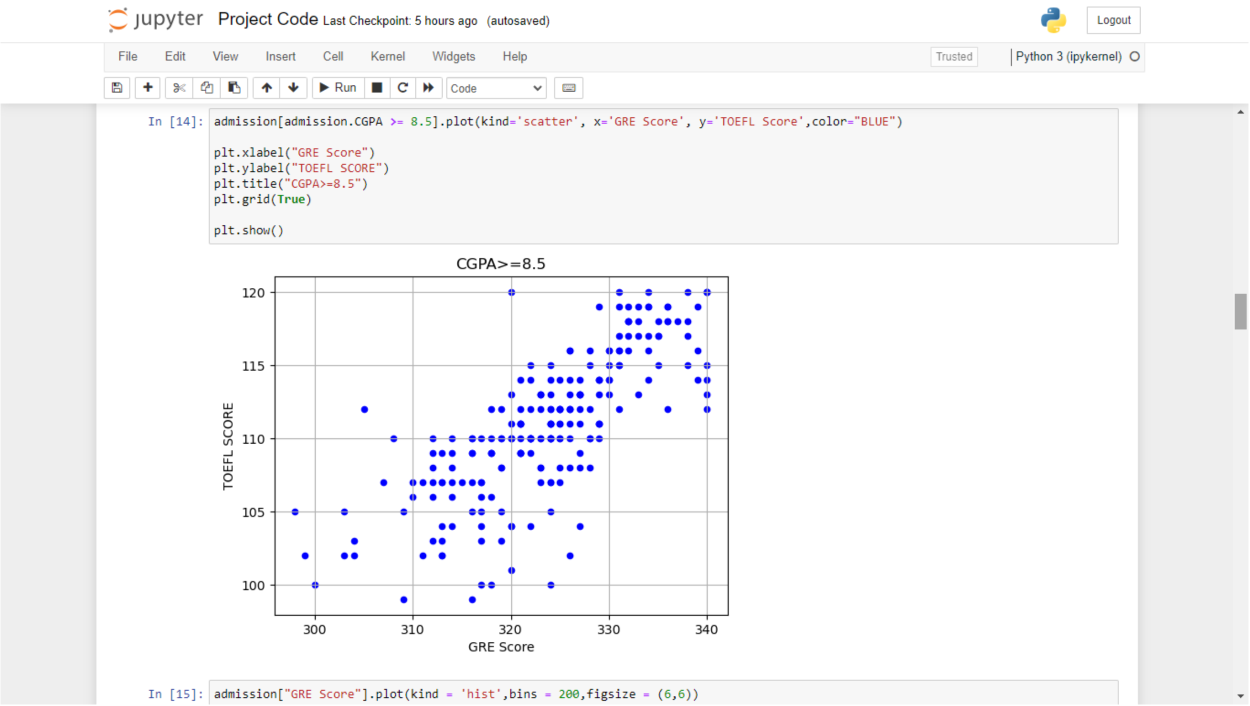
```
In [12]: plt.scatter(admission['GRE Score'],admission['CGPA'])
         plt.title('CGPA vs GRE Score')
         plt.xlabel('GRE Score')
         plt.ylabel('CGPA')
         plt.show()
```

```
In [13]: plt.scatter(admission['CGPA'],admission['SOP'])
```



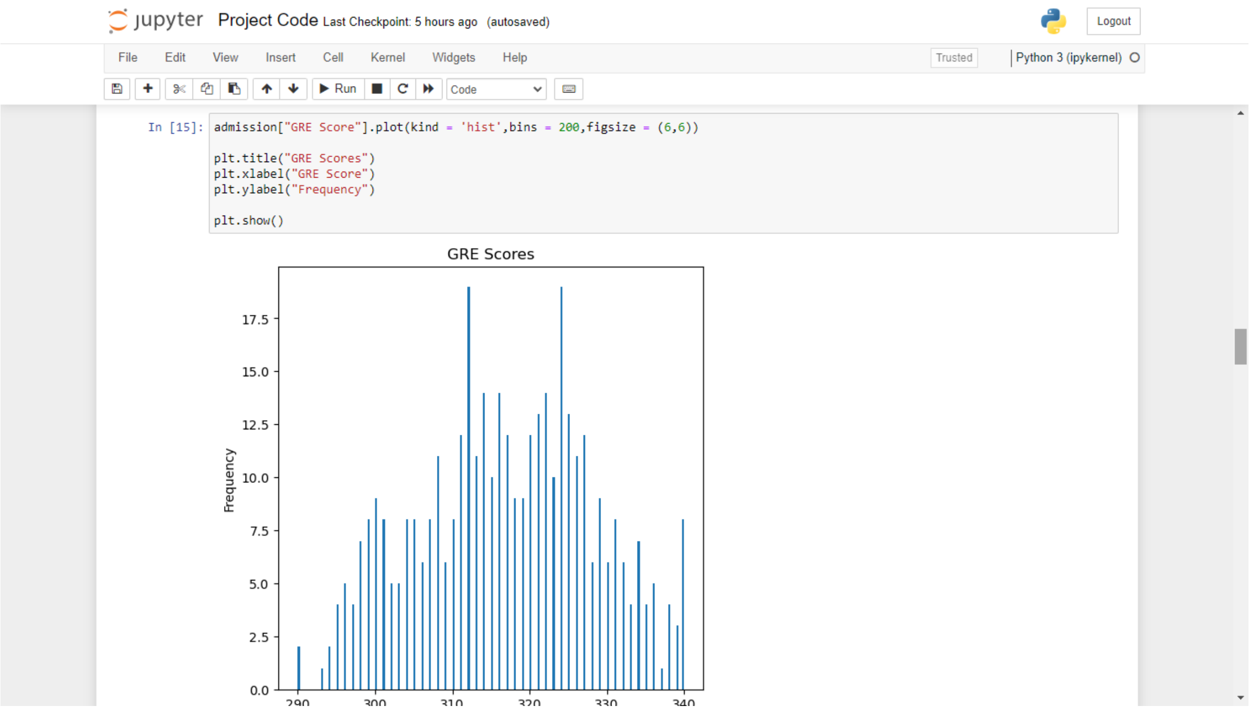
```
plt.scatter(admission['CGPA'],admission['SOP'])
plt.title('SOP for CGPA')
plt.xlabel('CGPA')
plt.ylabel('SOP')
plt.show()
```

```
admission[admission.CGPA >= 8.5].plot(kind='scatter', x='GRE Score', y='TOEFL Score',color="BLUE")

plt.xlabel("GRE Score")
```

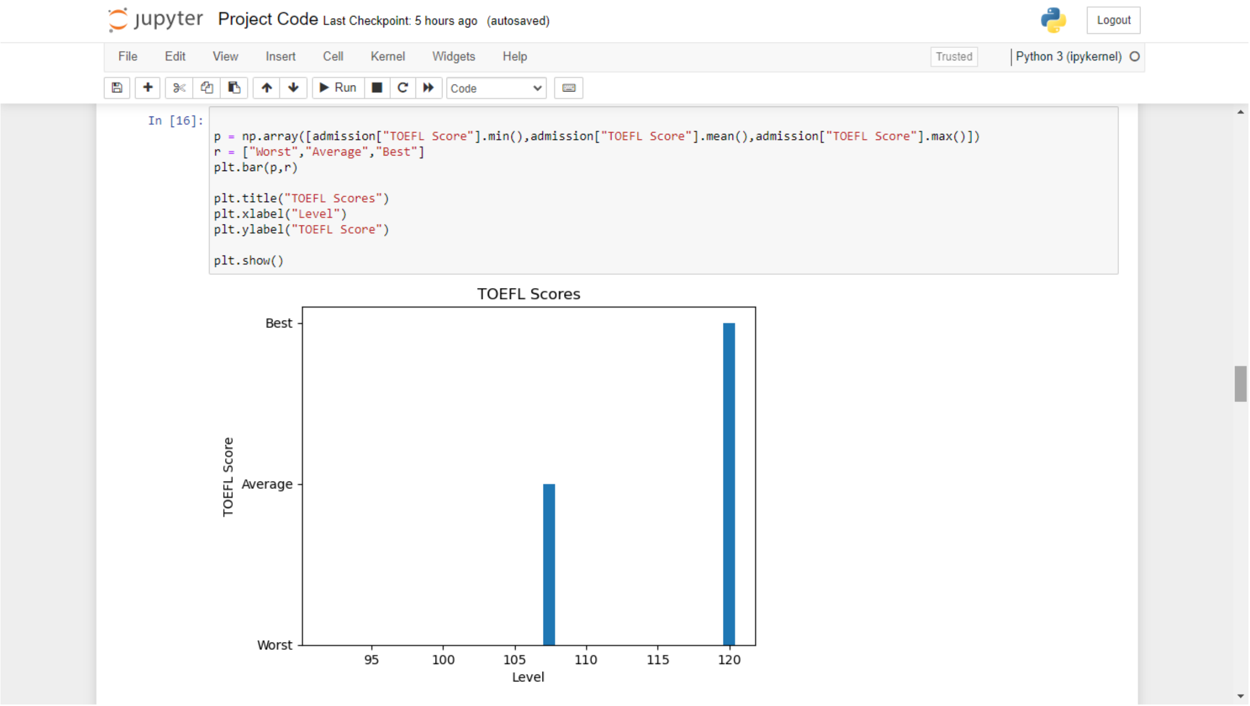


```
admission[admission.CGPA >= 8.5].plot(kind='scatter', x='GRE Score', y='TOEFL Score',color="BLUE")

plt.xlabel("GRE Score")
plt.ylabel("TOEFL SCORE")
plt.title("CGPA>=8.5")
plt.grid(True)

plt.show()
```

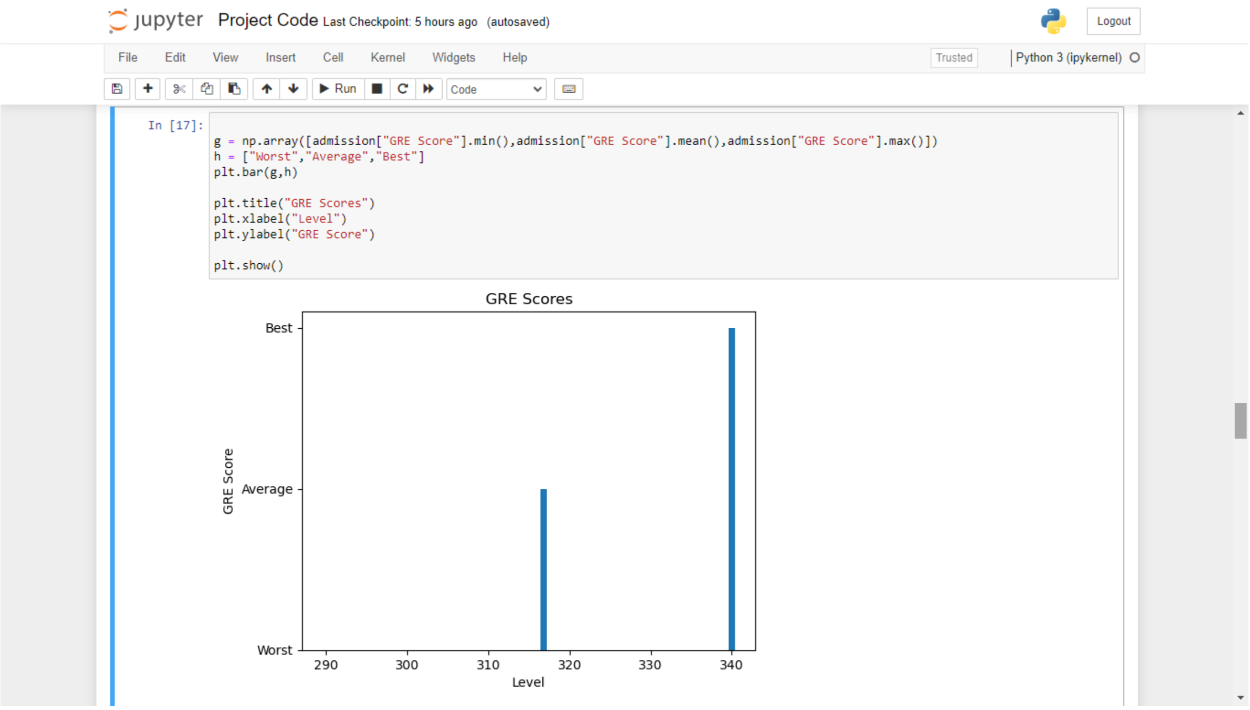
```
admission["GRE Score"].plot(kind = 'hist',bins = 200,figsize = (6,6))
```



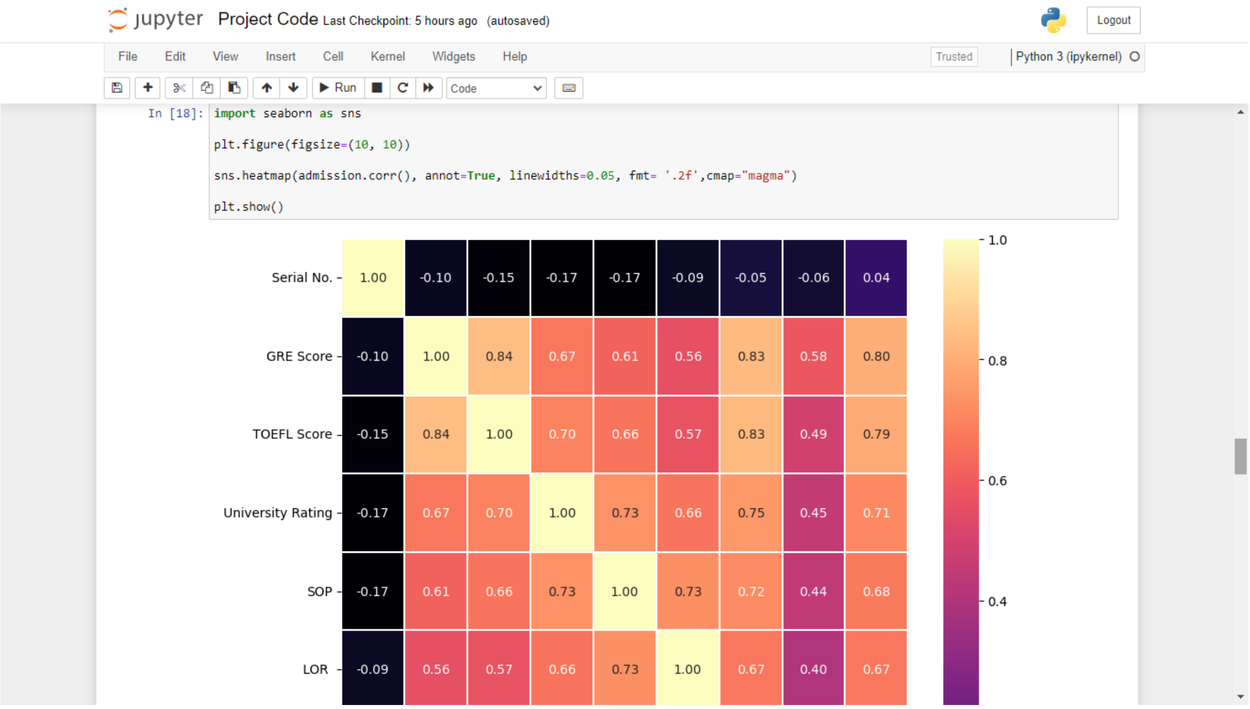
```
In [15]: admission["GRE Score"].plot(kind = 'hist',bins = 200,figsize = (6,6))

         plt.title("GRE Scores")
         plt.xlabel("GRE Score")
         plt.ylabel("Frequency")

         plt.show()
```



```
In [16]:
         p = np.array([admission["TOEFL Score"].min(),admission["TOEFL Score"].mean(),admission["TOEFL Score"].max()])
         r = ["Worst","Average","Best"]
         plt.bar(p,r)

         plt.title("TOEFL Scores")
         plt.xlabel("Level")
         plt.ylabel("TOEFL Score")

         plt.show()
```

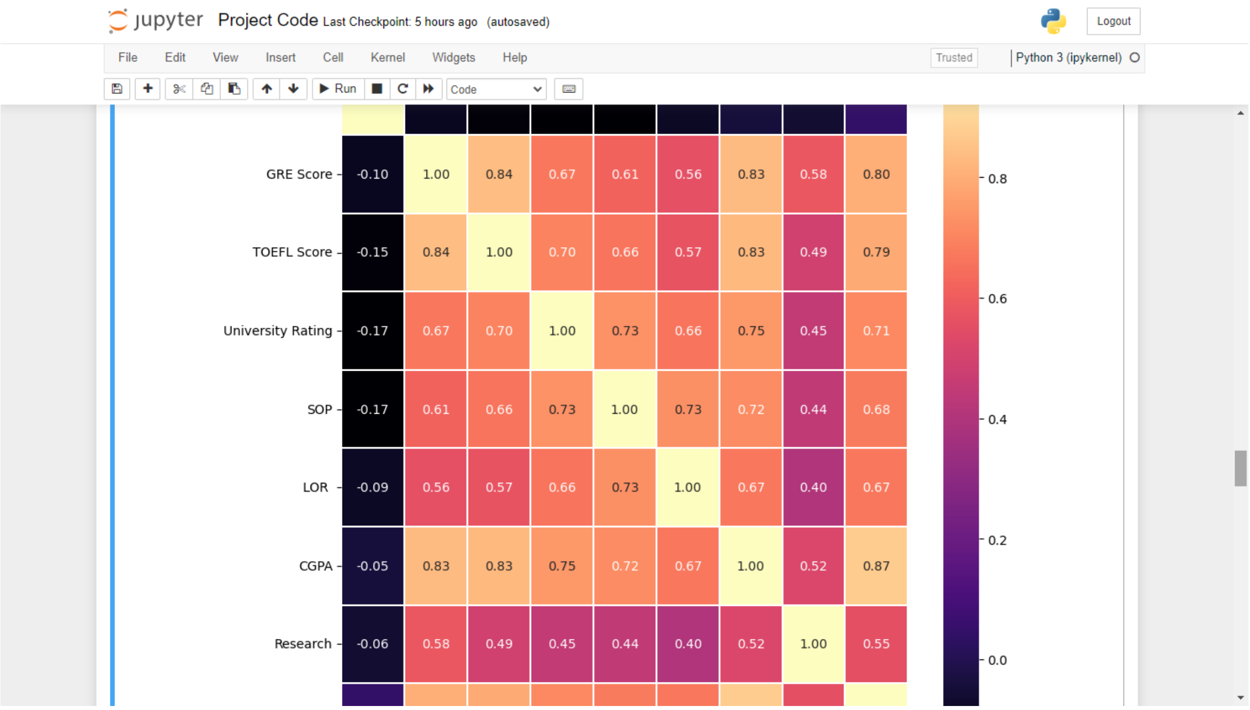


```
g = np.array([admission["GRE Score"].min(),admission["GRE Score"].mean(),admission["GRE Score"].max()])
h = ["Worst","Average","Best"]
plt.bar(g,h)

plt.title("GRE Scores")
plt.xlabel("Level")
plt.ylabel("GRE Score")

plt.show()
```

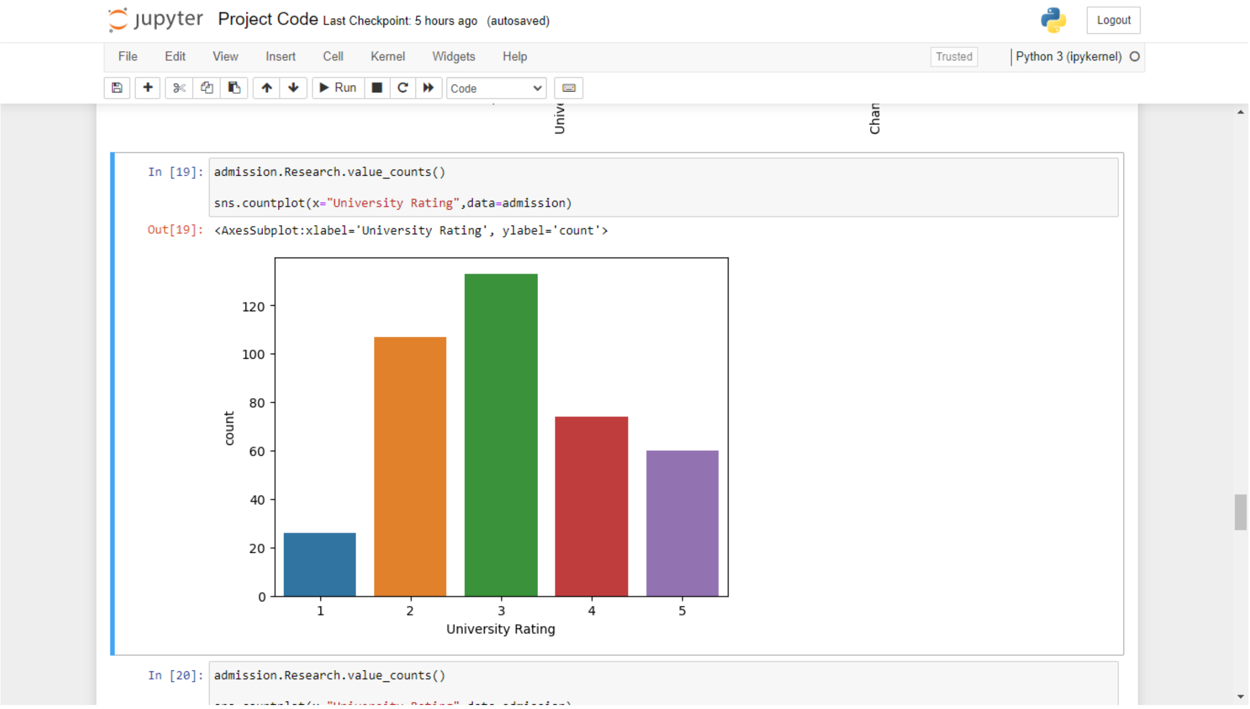


```
import seaborn as sns

plt.figure(figsize=(10, 10))

sns.heatmap(admission.corr(), annot=True, linewidths=0.05, fmt= '.2f',cmap="magma")

plt.show()
```

Serial No.	1.00	-0.10	-0.15	-0.17	-0.17	-0.09	-0.05	-0.06	0.04
GRE Score	-0.10	1.00	0.84	0.67	0.61	0.56	0.83	0.58	0.80
TOEFL Score	-0.15	0.84	1.00	0.70	0.66	0.57	0.83	0.49	0.79
University Rating	-0.17	0.67	0.70	1.00	0.73	0.66	0.75	0.45	0.71
SOP	-0.17	0.61	0.66	0.73	1.00	0.73	0.72	0.44	0.68
LOR	-0.09	0.56	0.57	0.66	0.73	1.00	0.67	0.40	0.67

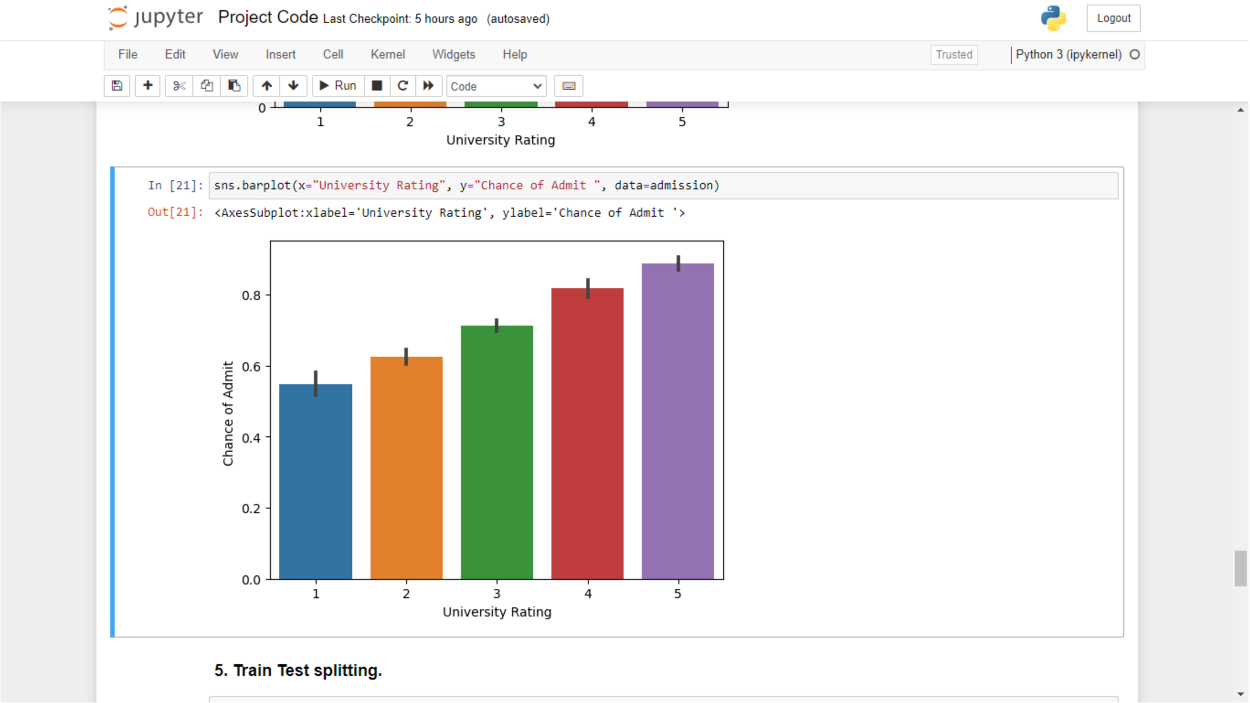


GRE Score	-0.10	1.00	0.84	0.67	0.61	0.56	0.83	0.58	0.80
TOEFL Score	-0.15	0.84	1.00	0.70	0.66	0.57	0.83	0.49	0.79
University Rating	-0.17	0.67	0.70	1.00	0.73	0.66	0.75	0.45	0.71
SOP	-0.17	0.61	0.66	0.73	1.00	0.73	0.72	0.44	0.68
LOR	-0.09	0.56	0.57	0.66	0.73	1.00	0.67	0.40	0.67
CGPA	-0.05	0.83	0.83	0.75	0.72	0.67	1.00	0.52	0.87
Research	-0.06	0.58	0.49	0.45	0.44	0.40	0.52	1.00	0.55



```
In [19]: admission.Research.value_counts()

         sns.countplot(x="University Rating",data=admission)

Out[19]: <AxesSubplot:xlabel='University Rating', ylabel='count'>
```

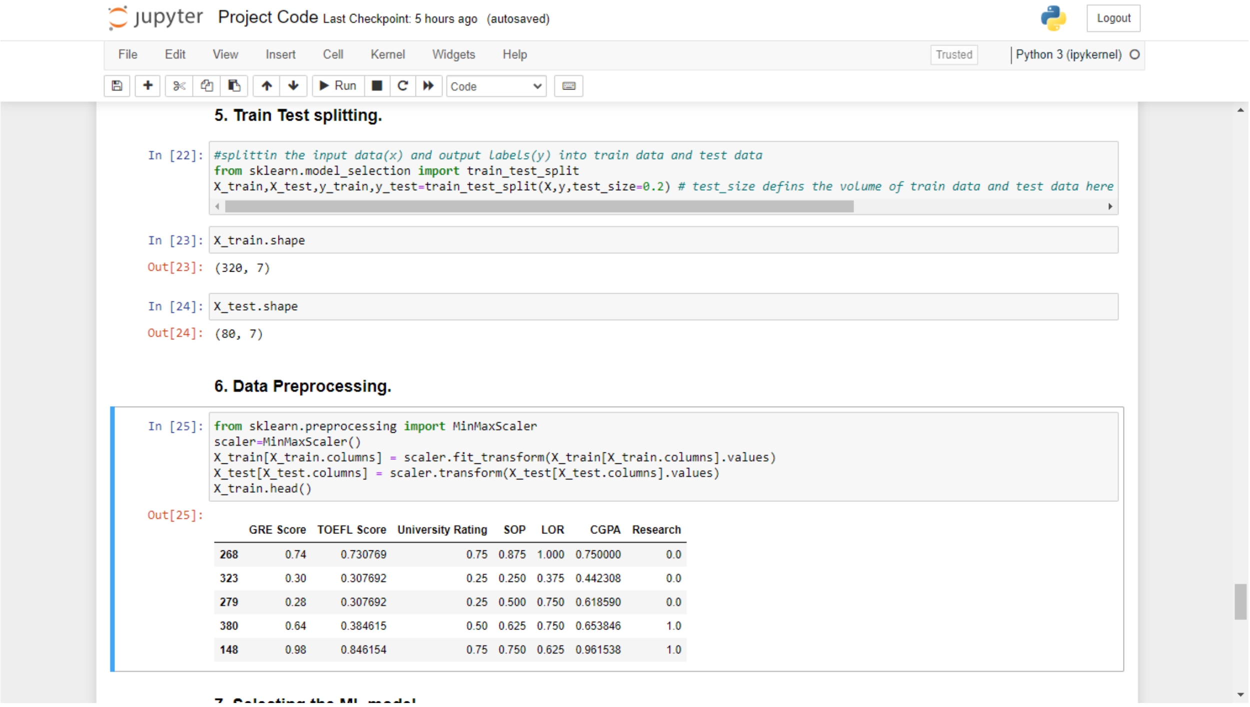
```
In [20]: admission.Research.value_counts()

         sns.countplot(x="University Rating",data=admission)
```



jupyter
Project Code Last Checkpoint: 5 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
0
1
2
3
4
5
University Rating

```
In [21]: sns.barplot(x="University Rating", y="Chance of Admit ", data=admission)

Out[21]: <AxesSubplot:xlabel='University Rating', ylabel='Chance of Admit '>
```

5. Train Test splitting.



jupyter
Project Code Last Checkpoint: 5 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

5. Train Test splitting.

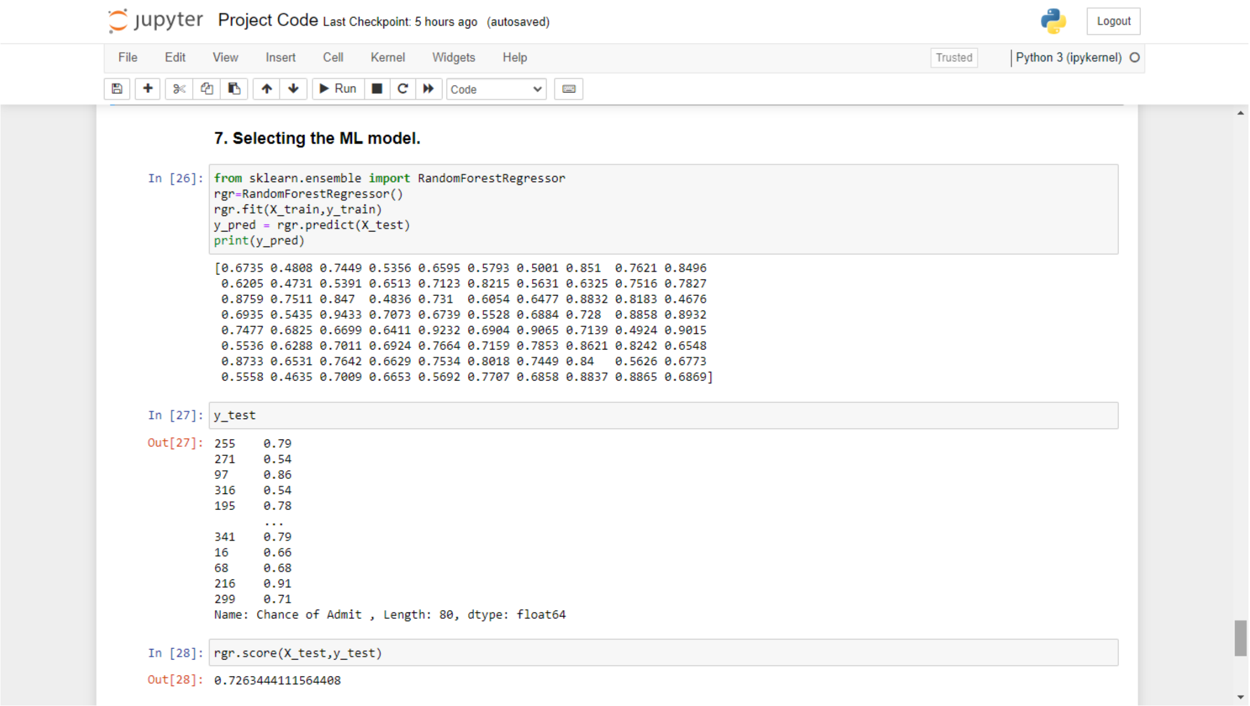
```
In [22]: #splittin the input data(x) and output labels(y) into train data and test data
         from sklearn.model_selection import train_test_split
         X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2) # test_size defins the volume of train data and test data here

In [23]: X_train.shape

Out[23]: (320, 7)

In [24]: X_test.shape

Out[24]: (80, 7)
```

6. Data Preprocessing.

```
In [25]: from sklearn.preprocessing import MinMaxScaler
         scaler=MinMaxScaler()
         X_train[X_train.columns] = scaler.fit_transform(X_train[X_train.columns].values)
         X_test[X_test.columns] = scaler.transform(X_test[X_test.columns].values)
         X_train.head()

Out[25]:
```

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research
268	0.74	0.730769	0.75	0.875	1.000	0.750000	0.0
323	0.30	0.307692	0.25	0.250	0.375	0.442308	0.0
279	0.28	0.307692	0.25	0.500	0.750	0.618590	0.0
380	0.64	0.384615	0.50	0.625	0.750	0.653846	1.0
148	0.98	0.846154	0.75	0.750	0.625	0.961538	1.0

7. Selecting the ML model

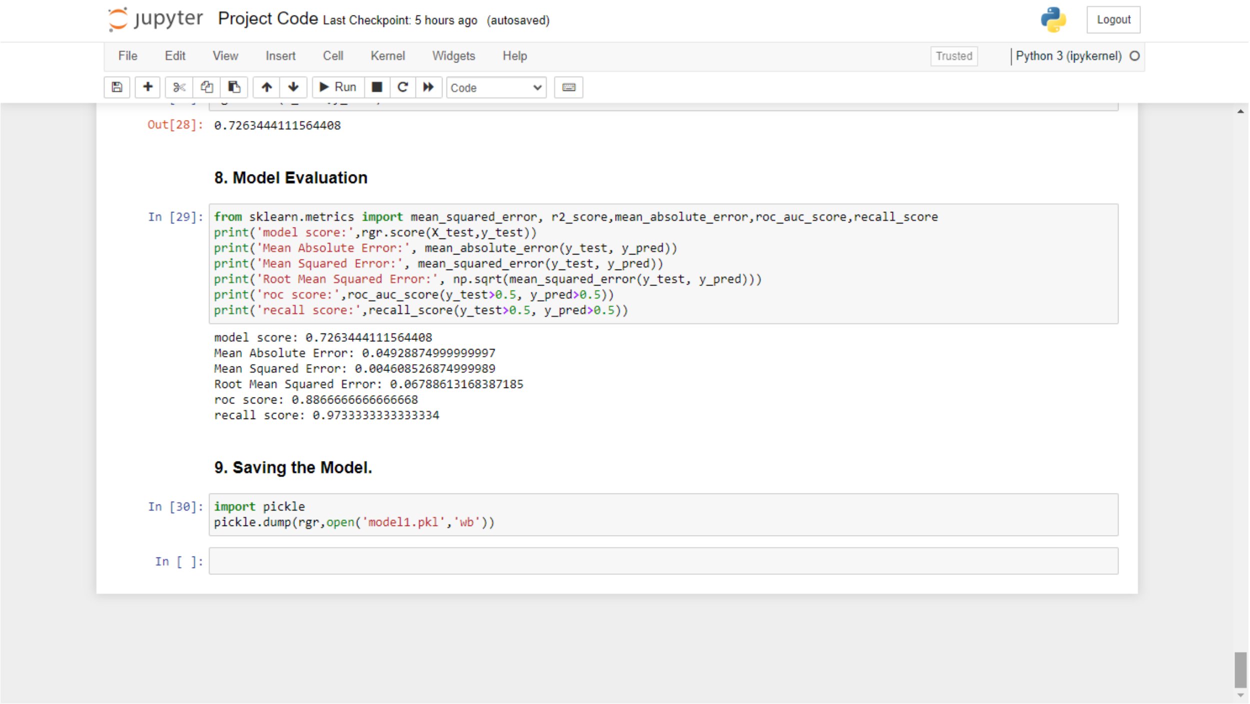

jupyter Project Code Last Checkpoint: 5 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

7. Selecting the ML model.

```
In [26]: from sklearn.ensemble import RandomForestRegressor
         rgr=RandomForestRegressor()
         rgr.fit(X_train,y_train)
         y_pred = rgr.predict(X_test)
         print(y_pred)

         [0.6735 0.4808 0.7449 0.5356 0.6595 0.5793 0.5001 0.851  0.7621 0.8496
          0.6205 0.4731 0.5391 0.6513 0.7123 0.8215 0.5631 0.6325 0.7516 0.7827
          0.8759 0.7511 0.847  0.4836 0.731  0.6054 0.6477 0.8832 0.8183 0.4676
          0.6935 0.5435 0.9433 0.7073 0.6739 0.5528 0.6884 0.728  0.8858 0.8932
          0.7477 0.6825 0.6699 0.6411 0.9232 0.6904 0.9065 0.7139 0.4924 0.9015
          0.5536 0.6288 0.7011 0.6924 0.7664 0.7159 0.7853 0.8621 0.8242 0.6548
          0.8733 0.6531 0.7642 0.6629 0.7534 0.8018 0.7449 0.84   0.5626 0.6773
          0.5558 0.4635 0.7009 0.6653 0.5692 0.7707 0.6858 0.8837 0.8865 0.6869]
```

```
In [27]: y_test

Out[27]: 255    0.79
         271    0.54
         97     0.86
         316    0.54
         195    0.78
                ...
         341    0.79
         16     0.66
         68     0.68
         216    0.91
         299    0.71
         Name: Chance of Admit , Length: 80, dtype: float64
```

```
In [28]: rgr.score(X_test,y_test)

Out[28]: 0.7263444111564408
```


jupyter Project Code Last Checkpoint: 5 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

```
Out[28]: 0.7263444111564408
```

8. Model Evaluation

```
In [29]: from sklearn.metrics import mean_squared_error, r2_score,mean_absolute_error,roc_auc_score,recall_score
         print('model score:',rgr.score(X_test,y_test))
         print('Mean Absolute Error:', mean_absolute_error(y_test, y_pred))
         print('Mean Squared Error:', mean_squared_error(y_test, y_pred))
         print('Root Mean Squared Error:', np.sqrt(mean_squared_error(y_test, y_pred)))
         print('roc score:',roc_auc_score(y_test>0.5, y_pred>0.5))
         print('recall score:',recall_score(y_test>0.5, y_pred>0.5))

         model score: 0.7263444111564408
         Mean Absolute Error: 0.04928874999999997
         Mean Squared Error: 0.004608526874999989
         Root Mean Squared Error: 0.06788613168387185
         roc score: 0.8866666666666668
         recall score: 0.9733333333333334
```

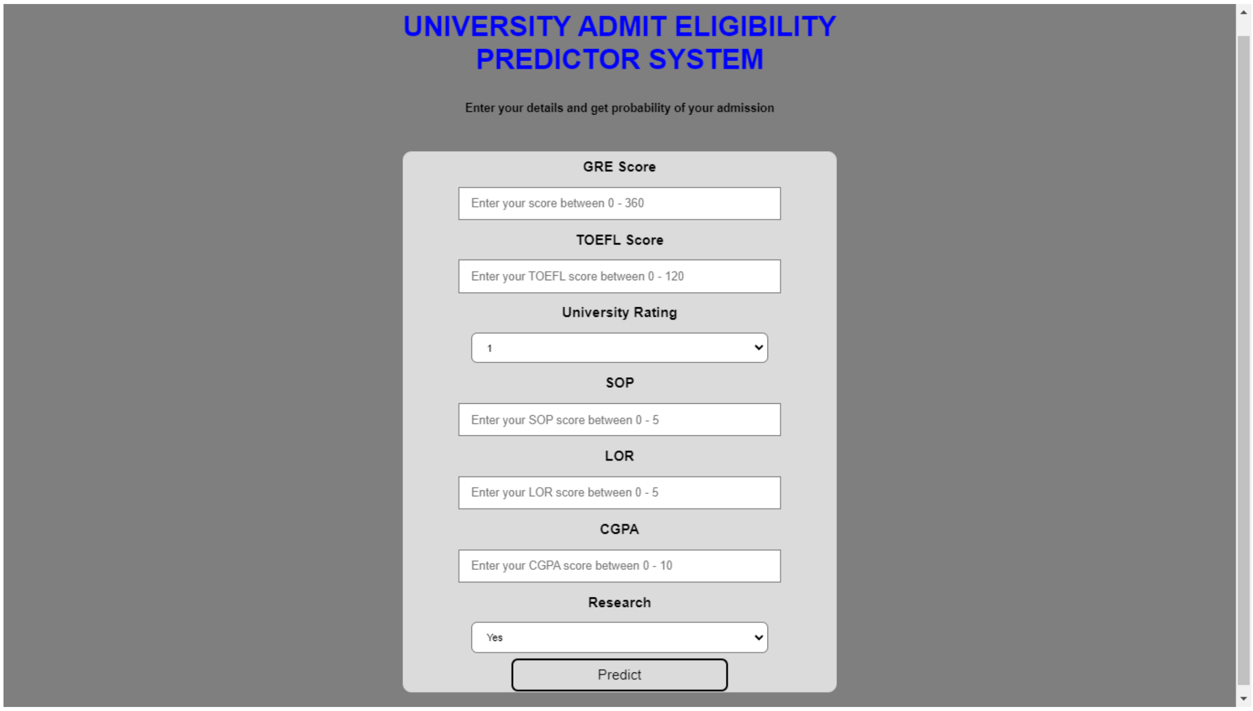
9. Saving the Model.

```
In [30]: import pickle
         pickle.dump(rgr,open('model1.pkl','wb'))

In [ ]:
```

CHAPTER-8

TESTING

8.1 TEST CASES



UNIVERSITY ADMIT ELIGIBILITY PREDICTOR SYSTEM

Enter your details and get probability of your admission

GRE Score

Enter your score between 0 - 360

TOEFL Score

Enter your TOEFL score between 0 - 120

University Rating

1

SOP

Enter your SOP score between 0 - 5

LOR

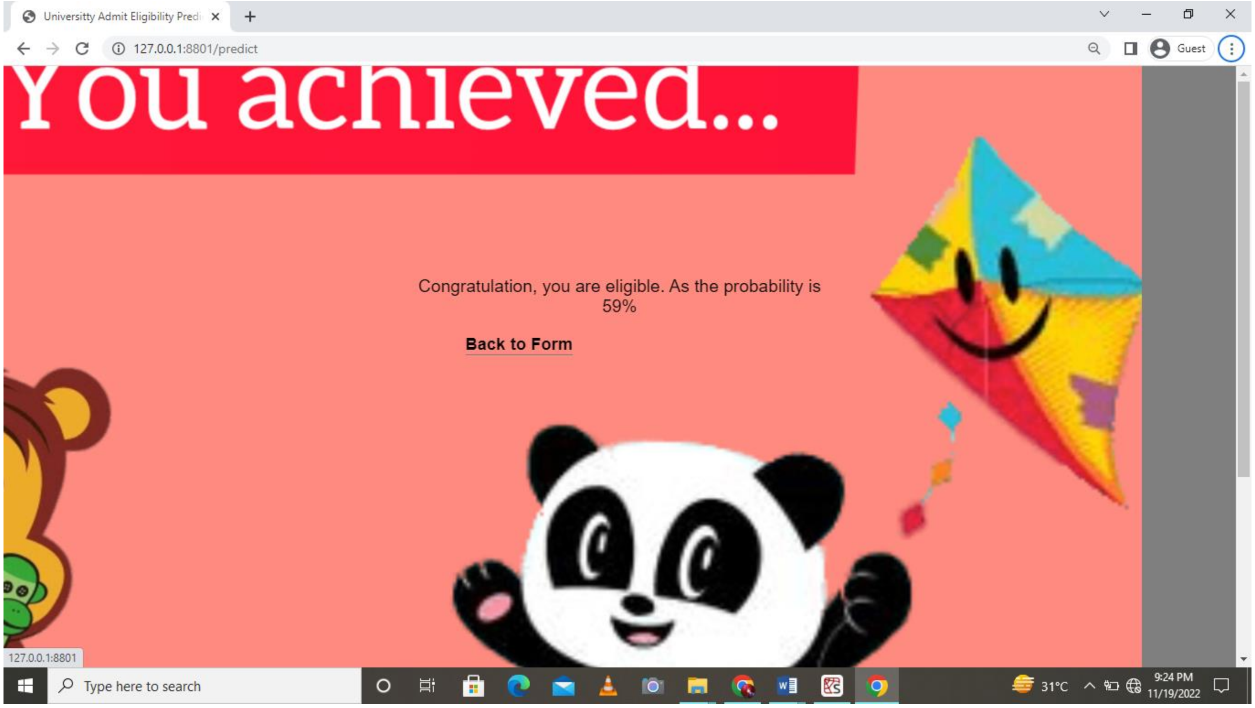
Enter your LOR score between 0 - 5

CGPA

Enter your CGPA score between 0 - 10

Research

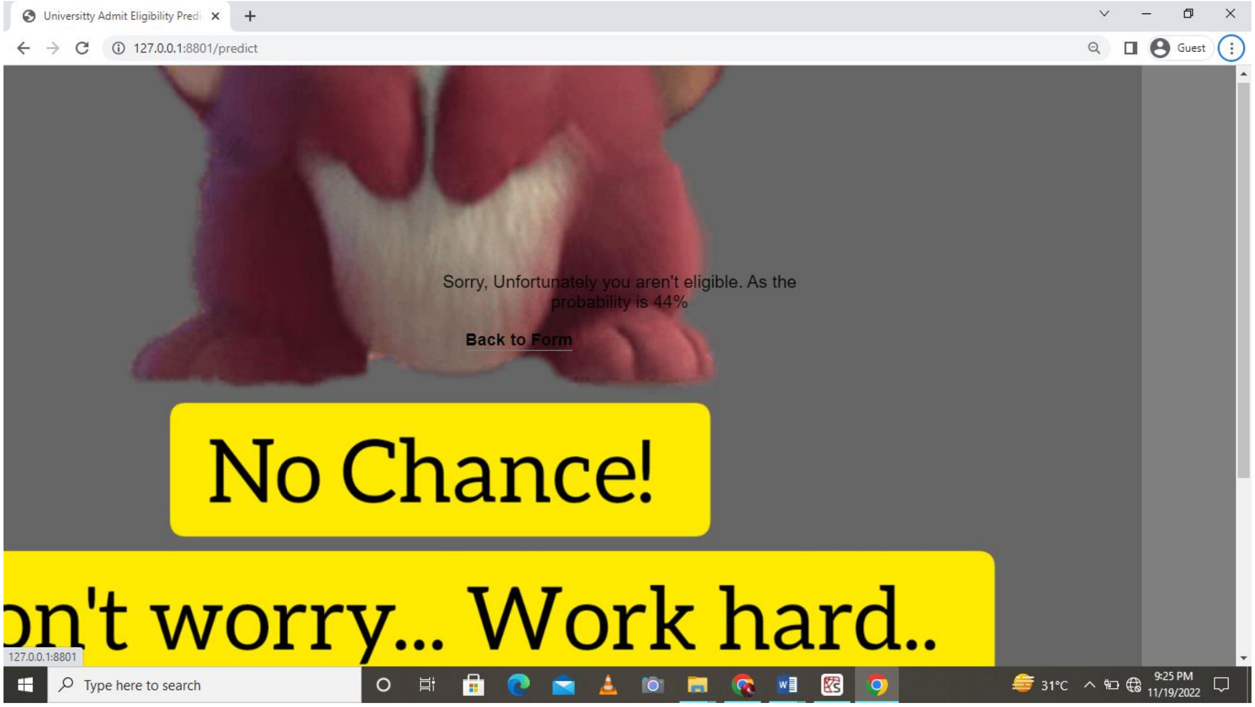
Yes



You achieved...

Congratulation, you are eligible. As the probability is 59%

Back to Form



Sorry, Unfortunately you aren't eligible. As the probability is 44%

Back to Form

No Chance!

on't worry... Work hard..

8.2 USER ACCEPTANCE

User Acceptance Testing (UAT) is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing is done.

Purpose of UAT

The main Purpose of UAT is to validate end to end business flow. It does not focus on cosmetic errors, spelling mistakes or system testing. User Acceptance Testing is carried out in a separate testing environment with production-like data setup. It is kind of black box testing where two or more end-users will be involved. UAT is performed by -

- Client
- End users

Need of User Acceptance Testing arises once software has undergone Unit, Integration and System testing because developers might have built software based on requirements document by their own understanding and further required changes during development may not be effectively communicated to them, so for testing whether the final product is accepted by client/end-user, user acceptance testing is needed.

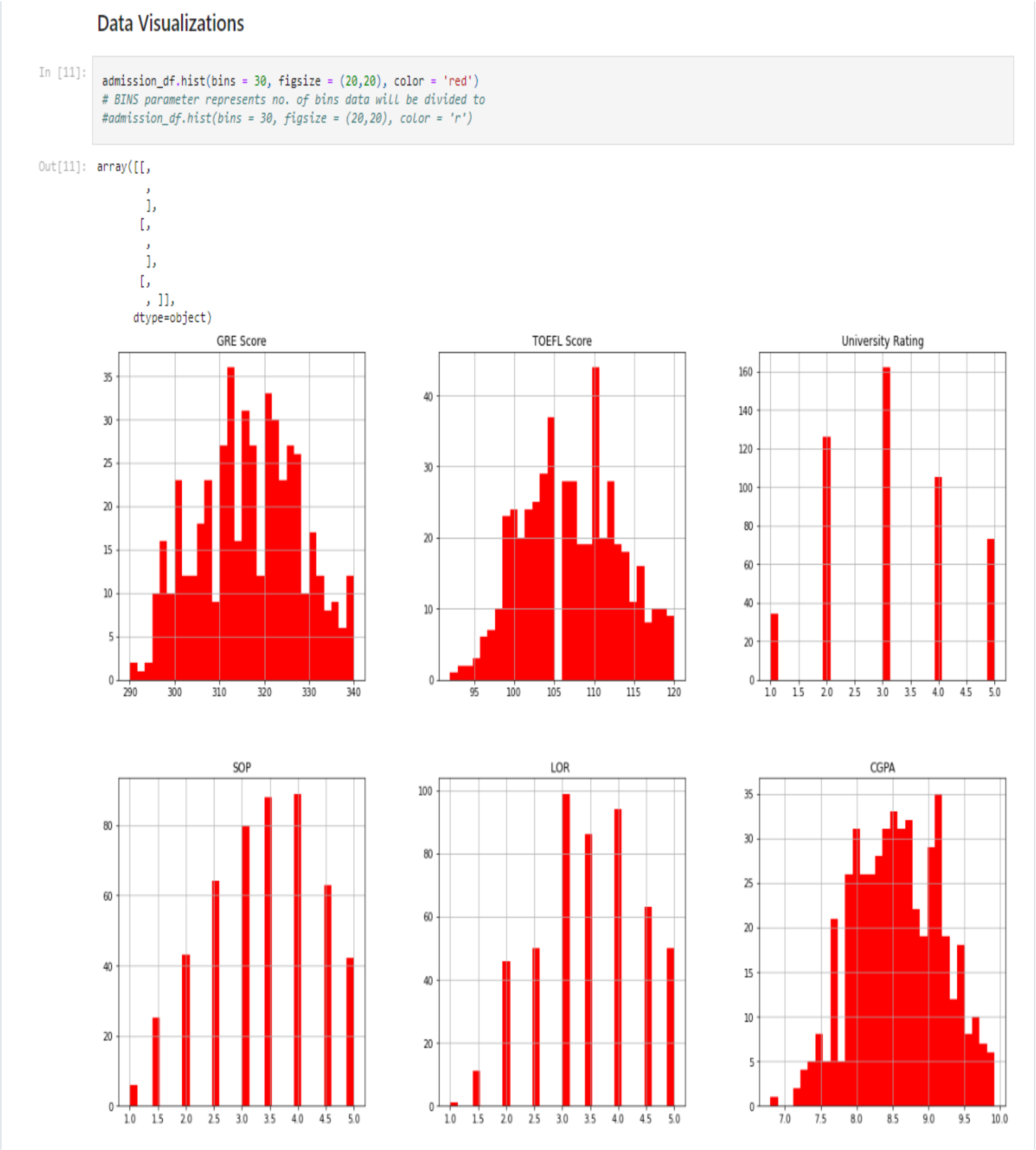
Developers code software based on requirements document which is their “own” understanding of the requirements and may not actually be what the client needs from the software.

Requirements changes during the course of the project may not be communicated effectively to the developers.

CHAPTER-9

RESULTS

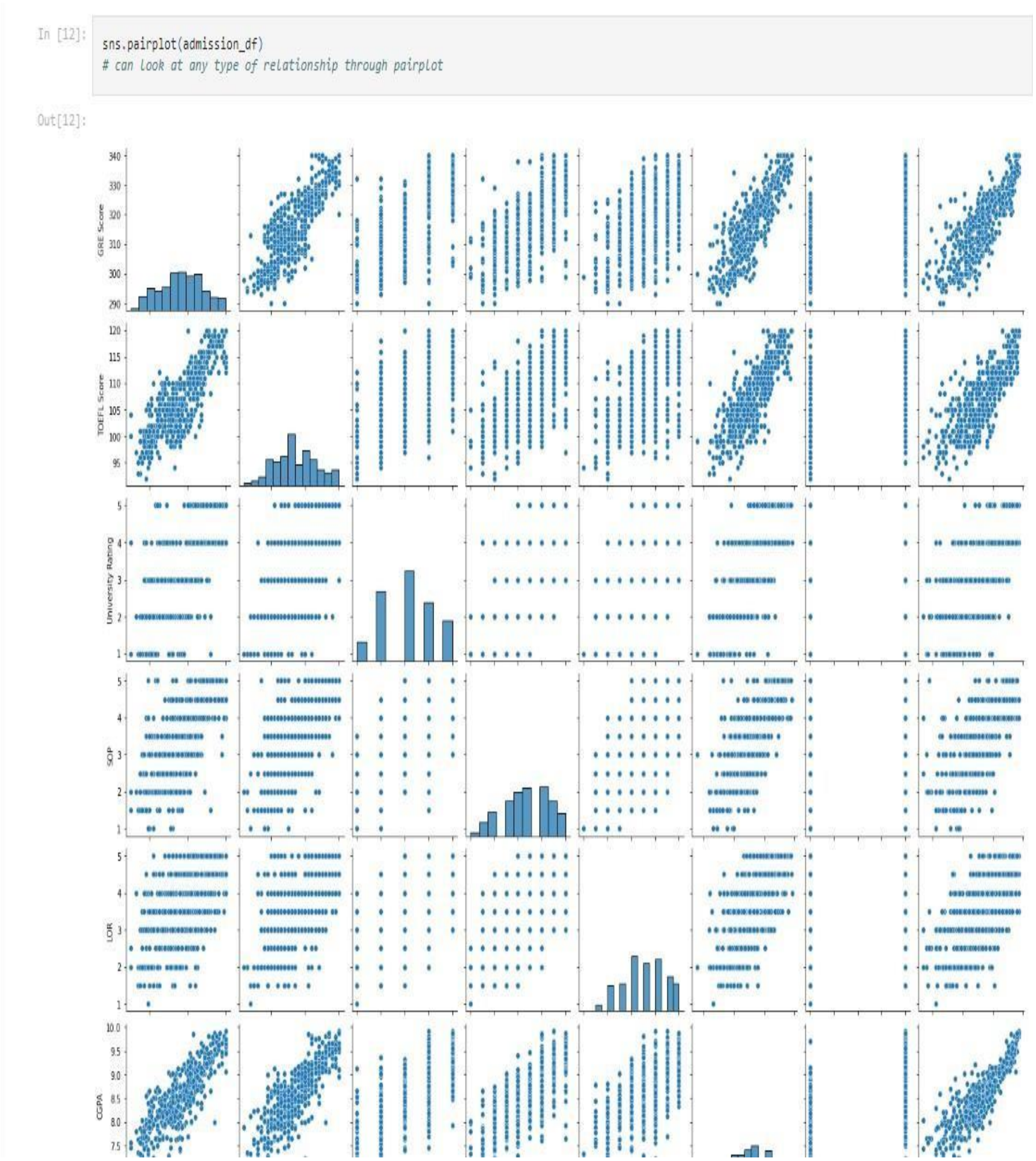
9.1 PERFORMANCE METRICS



Data Visualizations

```
In [11]:  admission_df.hist(bins = 30, figsize = (20,20), color = 'red')
          # BINS parameter represents no. of bins data will be divided to
          #admission_df.hist(bins = 30, figsize = (20,20), color = 'r')
```

```
Out[11]:  array([[,
                 ,
                 ],
                [,
                 ,
                 ],
                [,
                 , ]],
               dtype=object)
```



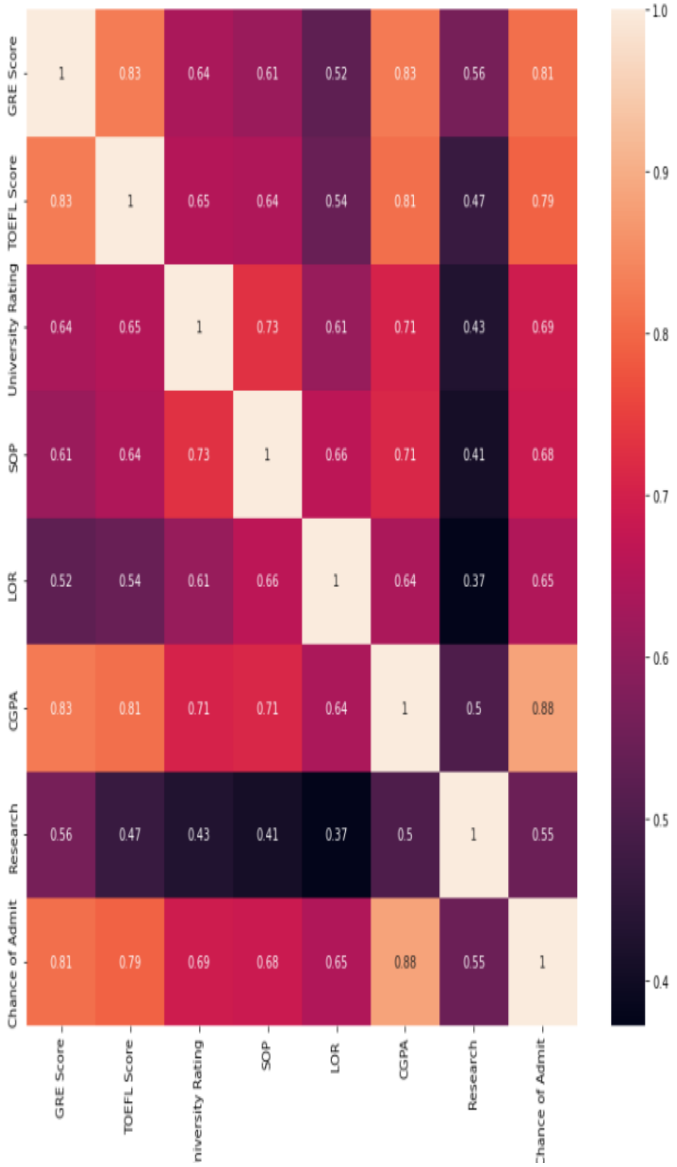
In [12]:

```
sns.pairplot(admission_df)
# can look at any type of relationship through pairplot
```

Out[12]:

In [13]:

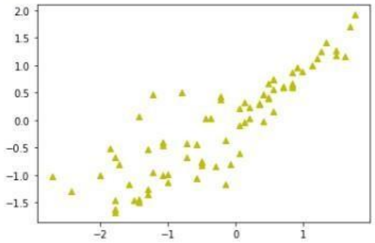
```
# correlation matrix, heatmap
corr_matrix = admission_df.corr()
plt.figure(figsize = (12,12))
sns.heatmap(corr_matrix, annot = True)   # annot: so that numbers are shown in the heatmap
plt.show()
```



Calculating Regression Model

```
In [41]:
y_predict = regressor.predict(X_test)
plt.plot(y_test, y_predict, '^', color = 'y') # '^' means to show data points in a triangular format
#plt.xlabel('y_test (Ground Truth)')
```

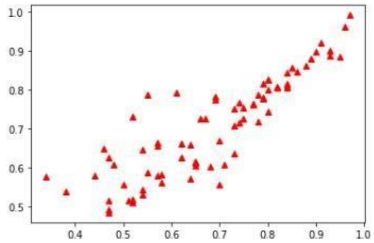
```
Out[41]: []
```



```
In [42]:
# applying inverse transform to represent the data in its original units
#(as earlier we did feature scaling, gonna reverse that so that data makes sense in its original units)
y_predict_orig = scaler_y.inverse_transform(y_predict)
y_test_orig = scaler_y.inverse_transform(y_test)
```

```
In [43]:
plt.plot(y_test_orig, y_predict_orig, '^', color = 'r')
```

```
Out[43]: []
```



```
In [32]:
result = ANN_model.evaluate(X_test, y_test)
accuracy_ANN = 1 - result
print("Accuracy : {}".format(accuracy_ANN))
```

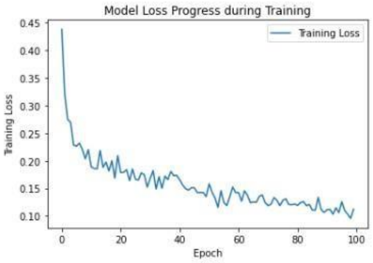
```
3/3 [==============================] - 0s 3ms/step - loss: 0.4360
Accuracy : 0.5640202462673187
```

```
In [33]:
# Looking at the progression of the network throughout the number of epochs
epochs_hist.history.keys()
```

```
Out[33]: dict_keys(['loss'])
```

```
In [34]:
# Loss => progression of network throughout the no. of epochs
plt.plot(epochs_hist.history['loss'])
plt.title('Model Loss Progress during Training')
plt.xlabel('Epoch')
plt.ylabel('Training Loss')
plt.legend(['Training Loss'])  # for label/box in the plot
```

```
Out[34]:
```



77.8% accuracy of ANN model

CHAPTER-10

ADVANTAGES & DISADVANTAGES

ADVANTAGES

- Give more accurate options for foreign universities.
- Train more quickly, especially with bigger datasets.
- GRADE has minimal software and hardware dependencies.
- It is implemented in Python with the Pandas and Skit-learn packages, which are open sourceand freely available.
- It uses many algorithms like random forest, K-nearest neighbour, multi layer preceptIon.
- The simplest way to determine whether a person is eligible for a university or college as wellas being completely objective and transparent.
- Easy to implement evaluation of the conditional probability is simple.
- Individuals would no longer need to rely on consultancies that may have contracts withschools and universities that are somewhat off the list.

DISADVANTAGES

- It is used to display the result as a dotted graph.
- It is to predict the approximate value.
- Contrarily, linear regression presumes that the relationship between the dependent and independent variables is linear. The implies that it considers their relationship to be linear. The independence of the qualities is assumed.
- It is not always true that the conditional independence assumption is true. The feature often exhibits some sort of dependence.

CHAPTER-11

CONCLUSION

CONCLUSION

University Admit Eligibility Prediction representation is build by using different types of algorithms. Same dataset is trained for the three algorithms and by training the representation we obtained the results. Finally, Random Forest is the best algorithm to provide accurate result for the dataset that is available. The accuracy level we obtained using Random Forest is94%, which is very encouraging. A user interface to interact with scholars to see the result is created using Node-RED.

CHAPTER-12

FUTURE WORKS

FUTURE WORKS

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 internetconnection.

The user had not need to travel a long distance for the admission and his/her time is also savedas a result of this automated system.

APPENDIX

SOURCE CODE

FLASK CODE

```
#University Admit Eligibility predictor
from flask import Flask, render_template, request
from math import ceil
import pickle

app = Flask(__name__)
model = pickle.load(open("model1.pkl","rb"))

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

@app.route('/predict',methods=["GET","POST"])
def predict():
    gre=(eval(request.form["gre"])-290)/(340-290)
    toefl=(eval(request.form["toefl"])-92)/(120-92)
    rating=(eval(request.form["rating"])-1.0)/4.0
    sop=(eval(request.form["sop"])-1.0)/4.0
    lor=(eval(request.form["lor"])-1.0)/4.0
    cgpa=(eval(request.form["cgpa"])-6.7)/(10.0-6.7)
    research=request.form["research"]
    if research == 1:
        research = 1
    else:
        research = 0
    pred = [[gre,toefl,rating,sop,lor,cgpa,research]]
    result=model.predict(pred)
    if result > 0.5:
        return render_template("chance.html",msg="Congratulation, you are eligible. As the probability is "+str(ceil(result[0]*100))+"%")
    return render_template("nochance.html",msg="Sorry, Unfortunately you aren't eligible. As the probability is "+str(ceil(result[0]*100))+"%")


if __name__ == "__main__":
    app.run(port=8801)
```

FEATURE 2

INDEX

```
<!DOCTYPE html>


<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Universitty Admit Eligibility Predictor</title>
    <!-- <link rel="stylesheet" href="{{ url_for("static", filename="css/styles.css") }}"> -->
    <link rel="stylesheet" href="./static/css/styles.css">
    <!-- UIkit CSS -->
    <!-- <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/uikit@3.15.12/dist/css/uikit.min.css" /> -->
    <!-- UIkit JS -->
    <!-- <script src="https://cdn.jsdelivr.net/npm/uikit@3.15.12/dist/js/uikit.min.js"></script> -->
    <!-- <script src="https://cdn.jsdelivr.net/npm/uikit@3.15.12/dist/js/uikit-icons.min.js"></script> -->
</head>
<body>
    <header class="uk-width-1-1">
        <h1><center> <b> UNIVERSITY ADMIT ELIGIBILITY<br> PREDICTOR SYSTEM</b></center></h1>
        <h5>Enter your details and get probability of your admission</h5>
    </header>
    <section class="uk-flex uk-flex-center uk-padding-large uk-width-1-1">
        <div class="card uk-padding">

            <div class="uk-divider-icon"></div>
            <div class="form">
                <form action="/predict" method="post">
                    <div class="label">
                        <label for="gre"><b>GRE Score</b></label>
                    </div>
                    <div class="input">
                        <input type="text" name="gre" id="gre" placeholder="Enter your score between 0 - 360" required/>
                    </div>
                    <div class="label">
                        <label for="toefl"><b>TOEFL Score</b></label>
                    </div>
                    <div class="input">
                        <input type="text" name="toefl" id="toefl" placeholder="Enter your TOEFL score between 0 - 120" required/>
                    </div>
                    <div class="label">
                        <label for="rating"><b>University Rating</b></label>
                    </div>
                    <div class="input">
                        <select name="rating" id="rating" required>
                            <option disabled>select a rating</option>
                            <option value="1">1</option>
                            <option value="2">2</option>
                            <option value="3">3</option>
                            <option value="4">4</option>
                            <option value="5">5</option>
                        </select>
                    </div>
                    <div class="label">
                        <label for="sop"><b>SOP</b></label>
                    </div>
                    <div class="input">
                        <input type="text" name="sop" id="sop" placeholder="Enter your SOP score between 0 - 5" required/>
                    </div>
                    <div class="label">
                        <label for="lor"><b>LOR</b></label>
                    </div>
                    <div class="input">
                        <input type="text" name="lor" id="lor" placeholder="Enter your LOR score between 0 - 5" required/>
                    </div>
                    <div class="label">
                        <label for="cgpa"><b>CGPA</b></label>
                    </div>
                    <div class="input">
                        <input type="text" name="cgpa" id="cgpa" placeholder="Enter your CGPA score between 0 - 10" required/>
                    </div>
                    <div class="label">
                        <label for="research"><b>Research</b></label>
                    </div>
                    <div class="input">
                        <select name="research" id="research" required>
                            <option value="1">Yes</option>
                            <option value="0">No</option>
                        </select>
                    </div>
                    <div class="btn uk-flex uk-flex-center uk-margin-top">
                        <input type="submit" value="Predict" />
                    </div>
                </form>
            </div>

        </div>
    </section>
</body>
<script src={{url_for("static",filename="js/scripts.js")}}></script>
</html>
```

CHANCE

```
<!DOCTYPE html>

<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Universitty Admit Eligibility Predictor</title>
    <link rel="stylesheet" href="{{ url_for("static", filename="css/styles.css") }}">
    <!-- <link rel="stylesheet" href="../static/css/styles.css"> -->
    <!-- UIkit CSS -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/uikit@3.15.12/dist/css/uikit.min.css" />

    <!-- UIkit JS -->

    <script src="https://cdn.jsdelivr.net/npm/uikit@3.15.12/dist/js/uikit.min.js"></script>

    <script src="https://cdn.jsdelivr.net/npm/uikit@3.15.12/dist/js/uikit-icons.min.js"></script>

</head>

<body>

    <header class="uk-width-1-1">

        <h1><center> <b> UNIVERSITY ADMIT ELIGIBILITY<br> PREDICTOR SYSTEM</b></center></h1>

    </header>

    <section class="uk-flex uk-flex-center uk-flex-column uk-padding-large uk-width-1-1">

        <div class="card uk-padding uk-box-shadow-xlarge">

            <div class="img">

                <!-- <img src="../static/images/chance.jpeg" alt="response">-->

                <img src="{{ url_for("static", filename="images/chance.jpg") }}" alt="Chance">

            </div>

            <div class="msg uk-margin-top uk-padding uk-width-1-1 uk-text-justify">

                <p>{{msg}}<p>

            </div>

            <div class="nav uk-flex uk-flex-center uk-margin-top uk-width-1-1 uk-link-toggle uk-text-large">

                <a class="uk-link-heading" href="/"><b>Back to Form</b></a>

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

NO CHANCE

```
<!DOCTYPE html>


<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Universitty Admit Eligibility Predictor</title>
    <link rel="stylesheet" href="{{ url_for("static", filename="css/styles.css") }}">
    <!-- <link rel="stylesheet" href="../static/css/styles.css">  -->
    <!-- UIkit CSS -->
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/uikit@3.15.12/dist/css/uikit.min.css" />
    <!-- UIkit JS -->
    <script src="https://cdn.jsdelivr.net/npm/uikit@3.15.12/dist/js/uikit.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/uikit@3.15.12/dist/js/uikit-icons.min.js"></script>
</head>
<body>
    <header class="uk-width-1-1">
```

```
        <h1><center> <b> UNIVERSITY ADMIT ELIGIBILITY<br> PREDICTOR SYSTEM</b></center></h1>


    </header>
    <section class="uk-flex uk-flex-center uk-flex-column uk-padding-large uk-width-1-1">
        <div class="card uk-padding uk-box-shadow-xlarge">
            <div class="img">
                <img src="{{ url_for("static", filename="images/nochance.jpg") }}" alt="No chance">
            </div>
            <div class="msg uk-margin-top uk-padding uk-width-1-1 uk-text-justify">
                {{msg}}
            </div>
            <div class="nav uk-flex uk-flex-center uk-margin-top uk-width-1-1  uk-link-toggle uk-text-large">
                <a class="uk-link-heading" href="/"><b>Back to Form</b></a>
            </div>
        </div>
    </section>
</body>
    </html>
```

STYLE SHEETS

```
*{
    padding: 0;
    margin: 0;
    font-family: sans-serif;
}

body{
    width:100%;
    min-height:100vh;
    height:fit-content;
    max-height:fit-content;
    background: url("./images/university.jpg") no-repeat fixed;
    background-color:rgba(0,0,0,0.5);
    background-size:cover;
}

.bor{
    border:1px solid brown;
}

.heading{
    width:100%;
    background: #252525;
    font-size: 1.2rem;
}
.uk-width-1-1{

    text-align: center;
}
```

```
h1{
    color: blue;
}
h5{
    margin-top: 2%;
}
.uk-width-1-1{
    margin-top: 2%;
    font-size: larger;
    font-weight: 400px;
}

.heading a, .heading a:hover{
    color: #fff;
    text-decoration: none;
}

.uk-flex{
    display: flex;
    align-items: center;
    justify-content: center;
    text-align: center;
}
```

```
.card{
    margin: 1% 0%;
    width:35%;
    height:fit-content;
    backdrop-filter: blur(0px) saturate(200%);
    -webkit-backdrop-filter: blur(0px) saturate(200%);
    background-color: rgba(255, 255, 255, 0.72);
    border-radius: 12px;
    border: 1px solid rgba(209, 213, 219, 0.3);
    align-items: center;
    justify-content: center;
}

.info{
    font-size: 1.3rem;
}

.label{
    width:inherit;
    padding:0.5rem 0;
    /* border:1px solid green; */
}

.input{
    width:inherit;
    padding:0.5rem 0;
    /* border:1px solid blue; */
    margin : 0px 5px;
}

.input input{
    width:70%;
    padding:0.75rem 1rem;
    outline: none;
    border:1.3px solid #7a7a7a;
    border-radius: 0.50%;
    font-size: 1rem;
}

.label label{
    width: 70%;
}

.input select{
    width:70%;
    padding:0.75rem 1rem;
    outline: none;
    border:1.3px solid #7a7a7a;
    border-radius: 0.50rem;
}
```

```
.input input:hover, .input select:hover{
    border:1.3px solid white;
}
.input select:focus, .input input:focus{
    border:1.3px solid white;
}

.btn{
    width:100%;
}

.btn input{
    width:50%;
    padding:0.5rem 1rem;
    outline: none;
    border:3px solid #000;
    border-radius: 0.55rem;
    background:transparent;
    color:black;
    font-size: 1.2rem;
}
.btn uk-flex uk-flex-center uk-margin-top{
    color: black;
    background: transparent;
```

```
}
.btn input:hover{
   background-color:gray;
}
.img
{
   width: 100px;
   height: 100px;
   display: flex;
   align-items: center;
   justify-content: center;
}

.msg{
   font-size: 1.5rem;
}

.nav{
   width:50%;
   margin-left:10px ;
   align-items: center;
   margin-top:3%;
}
.nav a{
    margin-top: 2%;
    text-decoration: none;
    color: #000;
}
.nav a:hover{
    border-bottom: 2px solid grey;

}
/* small mobile devices.. */
@media screen and (min-width: 320px) and (max-width:480px){

}

/* ipads, tablets and large mobile devices.. */
@media screen and (min-width: 481px) and (max-width:768px){

}

/* small screen, Laptops.. */
@media  screen and (min-width: 769px) and (max-width:1024px){

}

/* Desktops and Large screens.. */
@media screen and (min-width:1025px) and (max-width:1200px)
```

```
}

@media screen and (min-width:1201px){
}
```

GITHUB: https://github.com/IBM-EPBL/IBM-Project-20254-1659715845

DEMONSTRATION VIDEO LINK: https://drive.google.com/drive/folders/1BB576E1iB8GP5-O1qkxUj6JiX97H9mTX?usp=sharing