1. INTRODUCTION

1.1 Project Overview

Students are often worried about their chances of admission to university. The aim of this project is to help students in shortlisting universities with their profiles. The predicted output gives them a fair idea about their admission chances in a particular university. This analysis should also help students who are currently preparing or will be preparing to get a better idea. This project University Admission Predictor System is web-based application in which students can register with their personal as well as marks details for prediction the admission in colleges and the administrator can allot the seats for the students. Administrator can add the college details and he batch details. Using this software, the entrance seat allotment became easier and can be implemented using system. The main advantage of the project is the computerization of the entrance seat allotment process. Administrator has the power for the allotment. He can add the allotted seats into a file and the details are saved into the system. The total time for the entrance allotment became lesser and the allotment process became faster.

1.2 Purpose

The primary purpose is to discuss the prediction of student admission to university based on numerous factors and using logistic regression. Many prospective students apply for Master’s programs. The admission decision depends on criteria within the particular college or degree program. The independent variables in this study will be measured statistically to predict graduate school admission. Exploration and data analysis, if successful, would allow predictive models to allow better prioritization of the applicants screening process to Master’s degree programe which in turn provides the admission to the right candidates.

2. LITERATURE SURVEY

2.1 Existing problem

Today in college's student details are entered manually .The student details in separate records are tedious task .Referring to all these records updating is needed .There is a chance for more manual errors.

- When the student comes in college.
- First of all, he/she takes admission form from reception
- Fills it and submits it into office.
- Filled form is first checked with documents like merit list an details came from university and verified by an official person ,if there is any mistake then it is corrected.
- At the time of submission of it the fee side posited by the candidate.
- At the time of submission of admission form admission number is assigned to the candidate by the institute.
- Candidate gets the receipt of fees deposition.

DISADVANTAGES OF EXISTING SYSTEM

- Require much man power i.e., much efforts, much cost and hard to operate and maintain.
- Since, all the work is done in papers so it is very hard to locate a particular student record when it is required.

2.2 References

Abdul Fatah S; M, A. H. (2012). Hybrid Recommender System for Predicting College Admission, pp. 107–113.

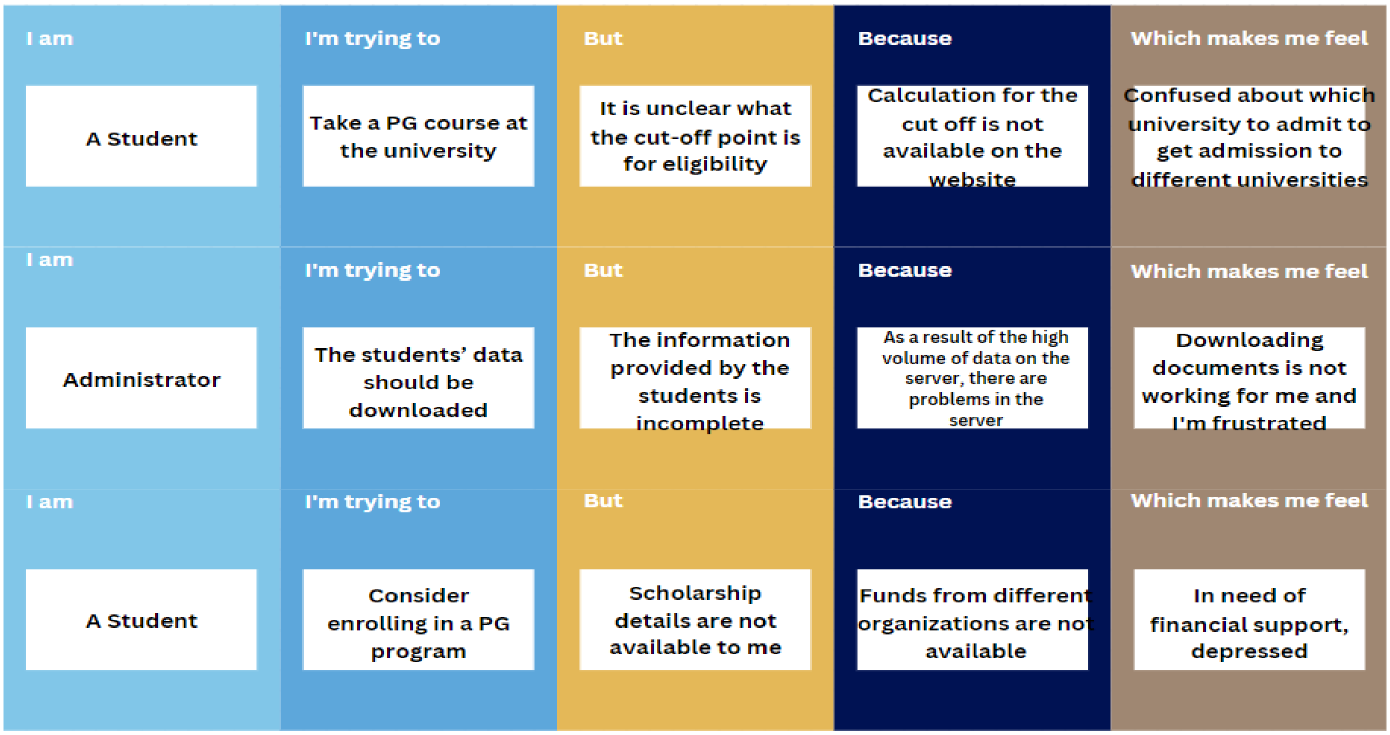
Bibodi, J., Vadodara, A., Rawat, A. and Patel, J. (n.d.). Admission Prediction System Using Machine Learning.

Eberle, W., Simpson, E., Talbert, D., Roberts, L. and Pope, A. (n.d.). Using Machine Learning and Predictive Modeling to Assess Admission Policies and Standards.

Jamison, J. (2017). Applying Machine Learning to Predict Davidson College

2.3 Problem Statement Definition

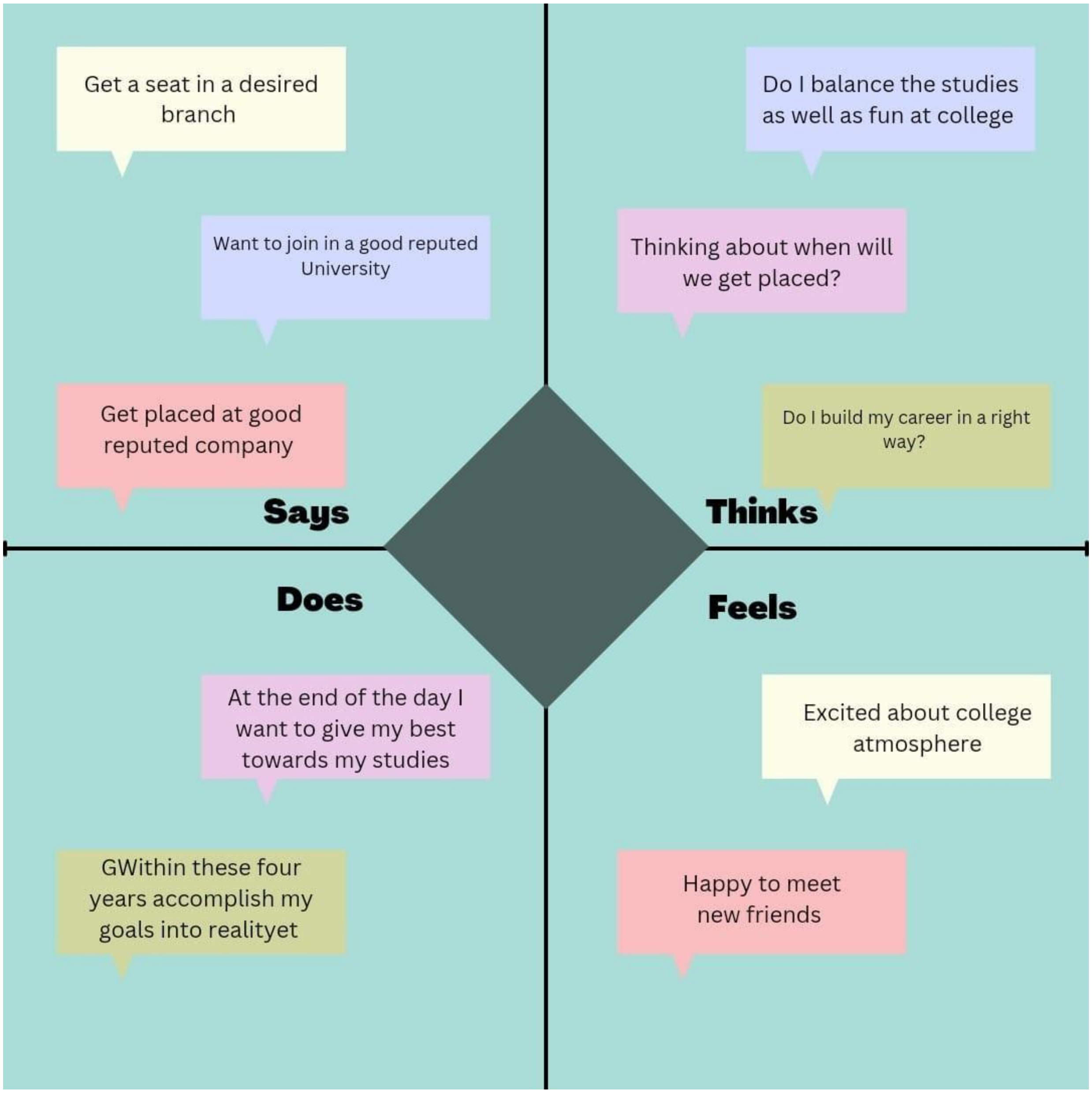
Problem Statement (PS)	I am (Customer)	I’m trying to	But	Because	Which Makes me feel
PS-1	A student	Take a PG course at the university	Itis unclear what the cut-off point is for eligibility	The calculation for the cut off is not available onthe website	Confused about which university to admit to get admission to different universities
PS-2	Administrator tor	Thestudent s’ data should be downloaded	Theinformation provided by thestudents is incomplete	Asa result of the highvolumeof data on the server, thereare	Downloading documents is not working for me and I'mfrustrated



I am	I'm trying to	But	Because	Which makes me feel
A Student	Take a PG course at the university	It is unclear what the cut-off point is for eligibility	Calculation for the cut off is not available on the website	Confused about which university to admit to get admission to different universities
I am	I'm trying to	But	Because	Which makes me feel
Administrator	The students’ data should be downloaded	The information provided by the students is incomplete	As a result of the high volume of data on the server, there are problems in the server	Downloading documents is not working for me and I'm frustrated
I am	I'm trying to	But	Because	Which makes me feel
A Student	Consider enrolling in a PG program	Scholarship details are not available to me	Funds from different organizations are not available	In need of financial support, depressed

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



Get a seat in a desired branch

Want to join in a good reputed University

Get placed at good reputed company

Says

Do I balance the studies as well as fun at college

Thinking about when will we get placed?

Do I build my career in a right way?

Thinks

Does

At the end of the day I want to give my best towards my studies

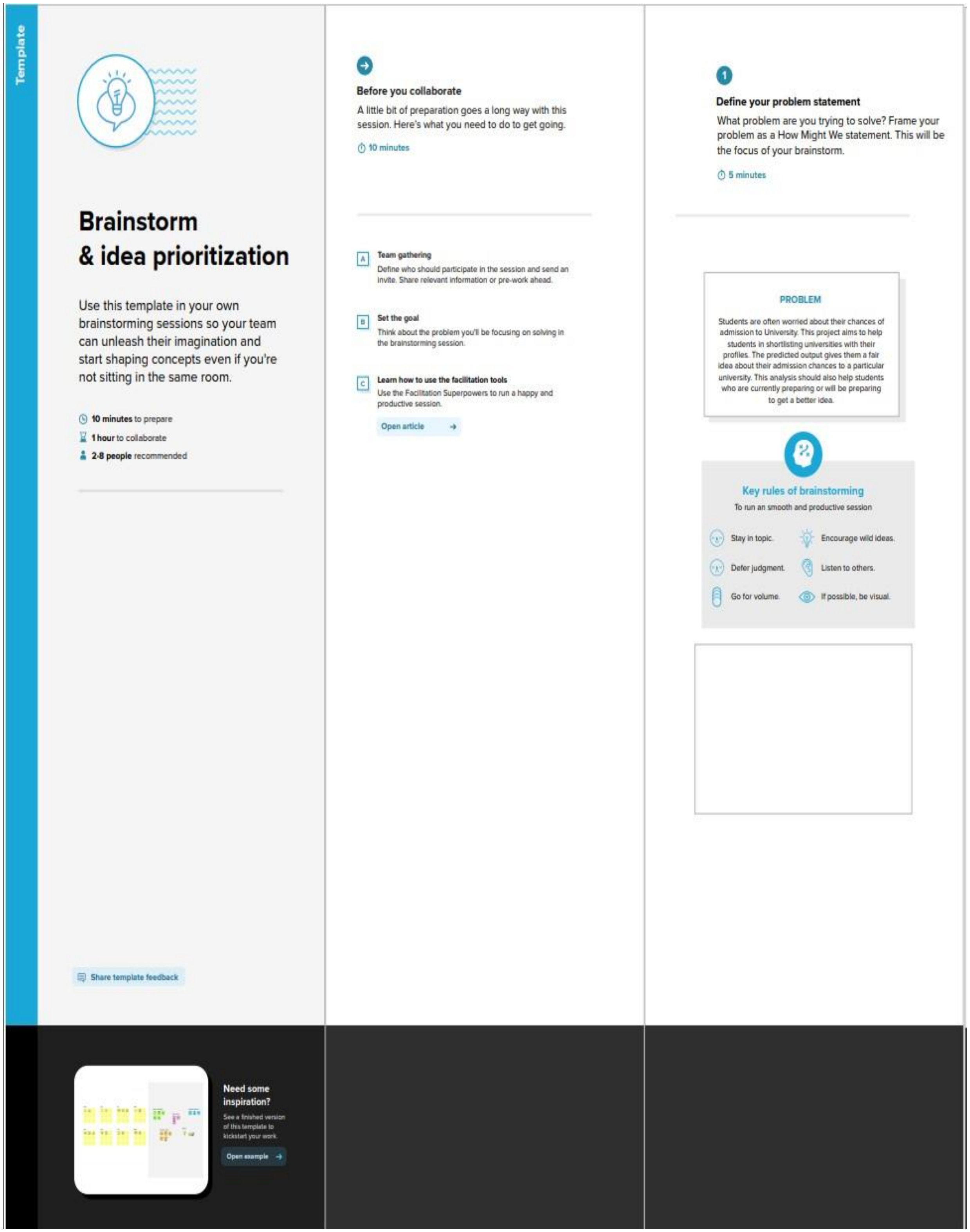
GWithin these four years accomplish my goals into realityet

Feels

Excited about college atmosphere

Happy to meet new friends

3.2 Ideation & Brainstorming



Brainstorm & idea prioritization

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

- **10 minutes** to prepare
- **1 hour** to collaborate
- **2-8 people** recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

1 **Define your problem statement**

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

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

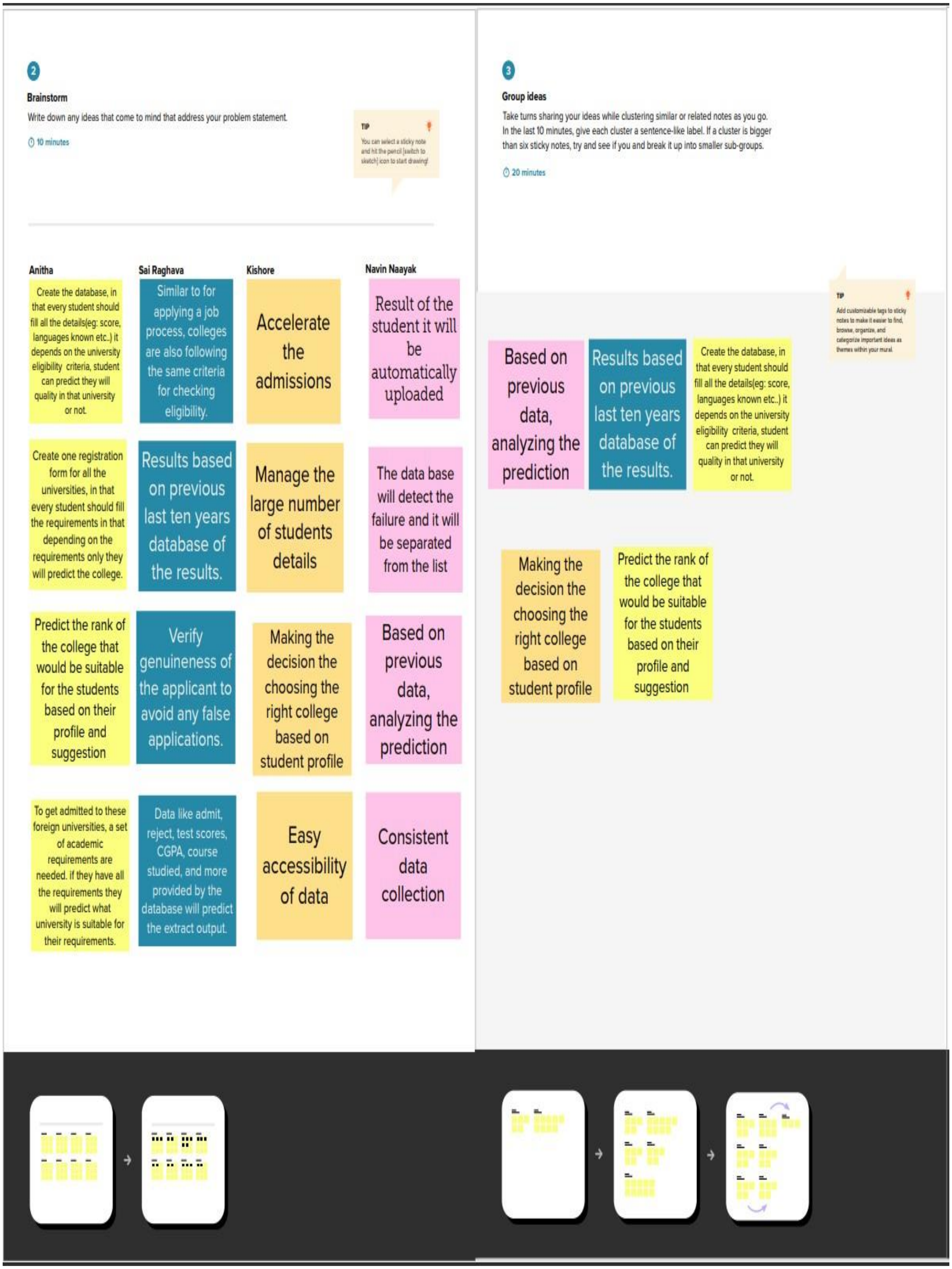
Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

Need some inspiration?

See a finished version of this template to kickstart your work.



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Anitha

Create the database, in that every student should fill all the details(eg: score, languages known etc.) it depends on the university eligibility criteria, student can predict they will quality in that university or not.

Create one registration form for all the universities, in that every student should fill the requirements in that depending on the requirements only they will predict the college.

Predict the rank of the college that would be suitable for the students based on their profile and suggestion

To get admitted to these foreign universities, a set of academic requirements are needed. if they have all the requirements they will predict what university is suitable for their requirements.

Sai Raghava

Similar to for applying a job process, colleges are also following the same criteria for checking eligibility.

Results based on previous last ten years database of the results.

Verify genuineness of the applicant to avoid any false applications.

Data like admit, reject, test scores, CGPA, course studied, and more provided by the database will predict the extract output.

Kishore

Accelerate the admissions

Manage the large number of students details

Making the decision the choosing the right college based on student profile

Easy accessibility of data

Navin Naayak

Result of the student it will be automatically uploaded

The data base will detect the failure and it will be separated from the list

Based on previous data, analyzing the prediction

Consistent data collection

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

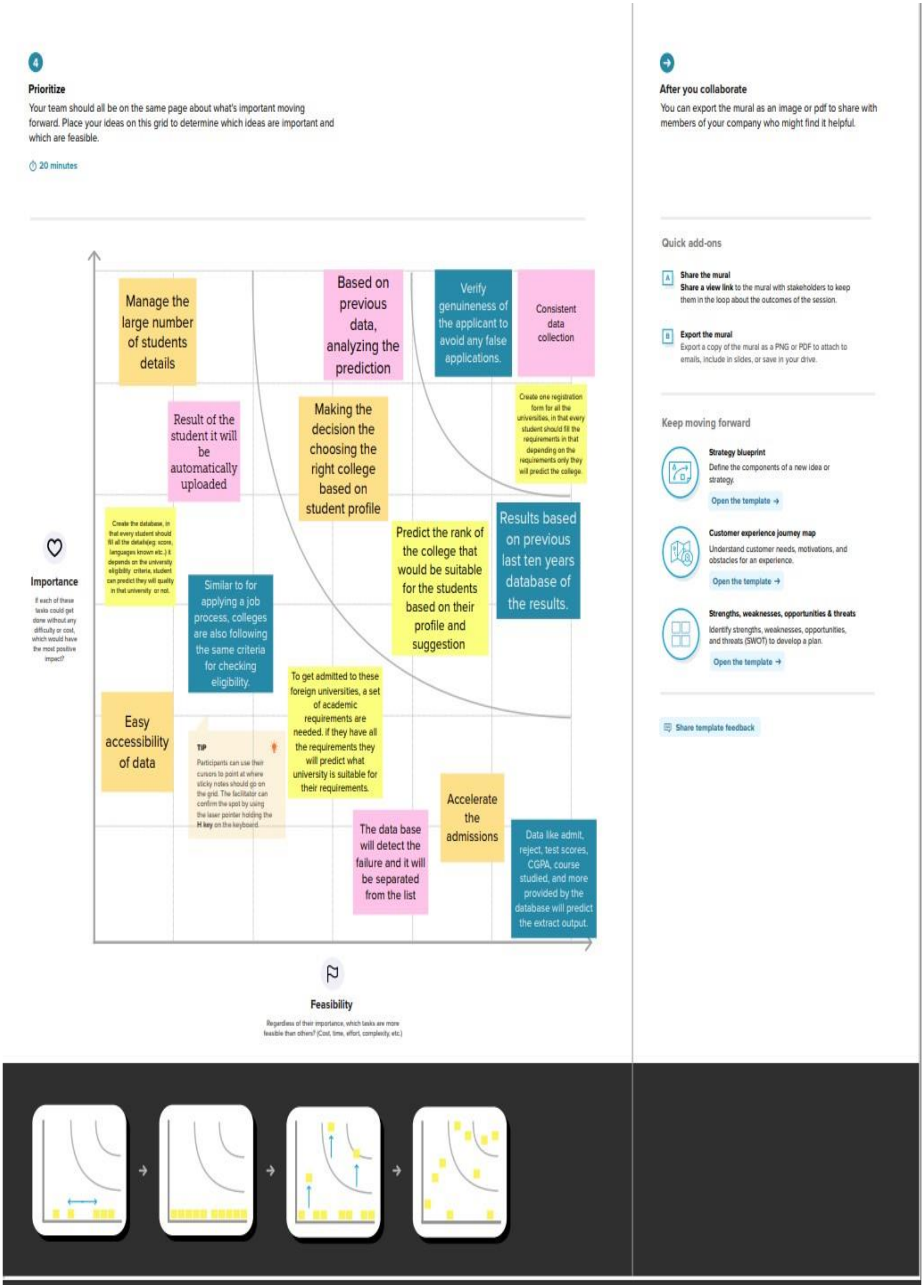
Based on previous data, analyzing the prediction

Results based on previous last ten years database of the results.

Create the database, in that every student should fill all the details(eg: score, languages known etc.) it depends on the university eligibility criteria, student can predict they will quality in that university or not.

Making the decision the choosing the right college based on student profile

Predict the rank of the college that would be suitable for the students based on their profile and suggestion



4

Prioritize

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

⏱ 20 minutes

Manage the large number of students details

Based on previous data, analyzing the prediction

Verify genuineness of the applicant to avoid any false applications.

Consistent data collection

Result of the student it will be automatically uploaded

Making the decision the choosing the right college based on student profile

Create one registration form for all the universities, in that every student should fill the requirements in that depending on the requirements only they will predict the college.

Create the database, in that every student should fill all the details(eg: score, languages known etc.) it depends on the university eligibility criteria, student can predict they will quality in that university or not.

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Results based on previous last ten years database of the results.

Predict the rank of the college that would be suitable for the students based on their profile and suggestion

Similar to for applying a job process, colleges are also following the same criteria for checking eligibility.

To get admitted to these foreign universities, a set of academic requirements are needed. if they have all the requirements they will predict what university is suitable for their requirements.

Easy accessibility of data

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

Accelerate the admissions

The data base will detect the failure and it will be separated from the list

Data like admit, reject, test scores, CGPA, course studied, and more provided by the database will predict the extract output.

Feasibility

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

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A **Share the mural**
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B **Export the mural**
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

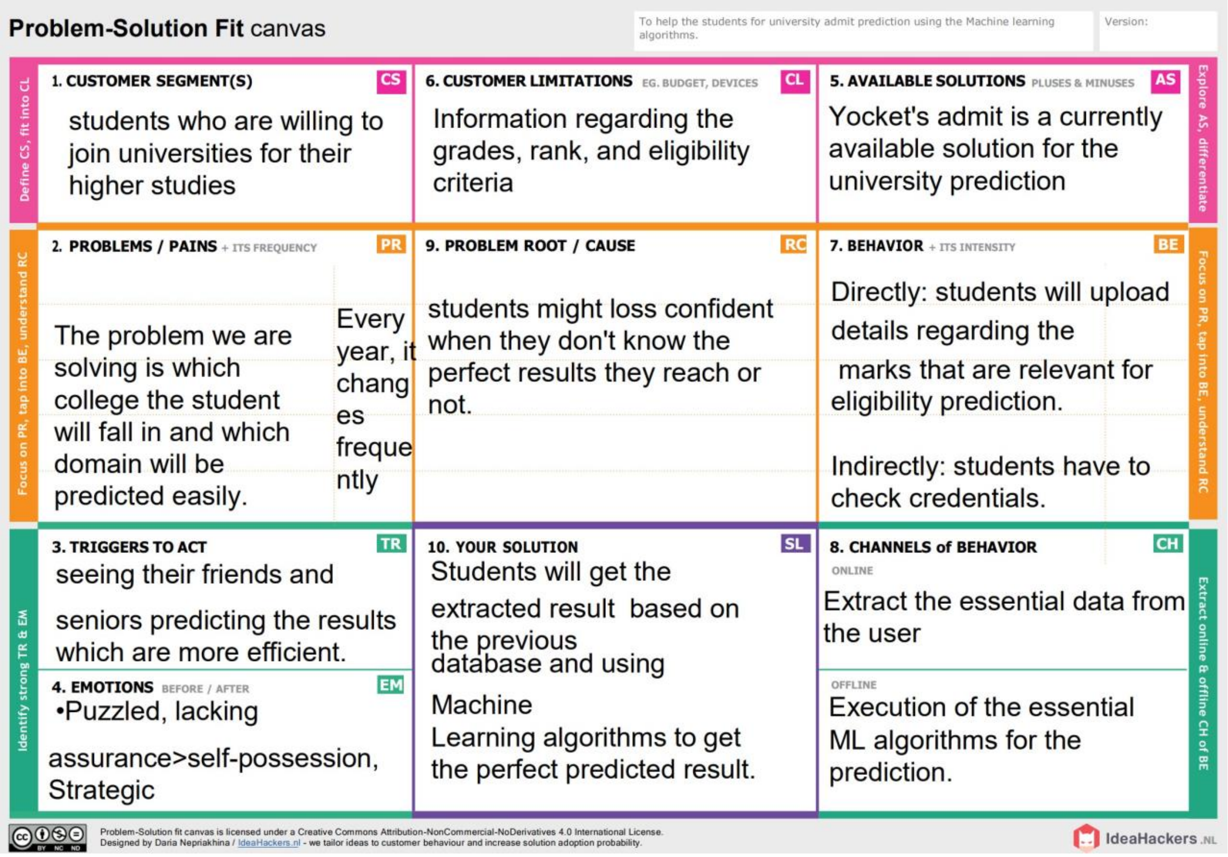
Share template feedback

Proposed solution template:

S.NO	parameter	Description
1.	Problem statement (problem to be solved)	Students are often worried about their chances ofadmission to University. The aim of this project is to help students in shortlistinguniversities with their profiles. The predicted output gives them a fair idea about their admission chances to a particular university. This analysis should also help studentswho are currently preparingor will be preparing to get a better idea.
2.	Idea/ solution description	● College admission prediction based on their performance. Inputs like class 10th marks, and 12th marks. GRE rank has taken then predicted the best-suited college for them. ● Based on the previous based data, dependent on that. we can predict which college is suitable. ● Use a random forest to predict the probability

3.	Novelty/uniqueness	● This application is useful for students those who are applying to universities. ● They can predict, which university they can get a seat.
4.	Social impact/ customer satisfaction	● It helps the students to save the time and money that they have to spend at education consultancy firms. ● Building an efficient university research site for the students who have been planning to apply for master programs in various disciplines.
5.	Business Model (Revenue Model)	● Many higher education institutions are investing in new business models with an emphasis on attracting and engaging lifelong learners. And innovation is more important than tradition, new models

3.4 Problem Solution fit



Problem-Solution Fit canvas

To help the students for university admit prediction using the Machine learning algorithms.

Version:

Define CS, fit into CL

1. CUSTOMER SEGMENT(S) CS

students who are willing to join universities for their higher studies

6. CUSTOMER LIMITATIONS EG. BUDGET, DEVICES CL

Information regarding the grades, rank, and eligibility criteria

5. AVAILABLE SOLUTIONS PLUSES & MINUSES AS

Yocket's admit is a currently available solution for the university prediction

Explore AS, differentiate

Focus on PR, tap into BE, understand RC

2. PROBLEMS / PAINS + ITS FREQUENCY PR

The problem we are solving is which college the student will fall in and which domain will be predicted easily.

Every year, it chang es freque ntly

9. PROBLEM ROOT / CAUSE RC

students might loss confident when they don't know the perfect results they reach or not.

7. BEHAVIOR + ITS INTENSITY BE

Directly: students will upload details regarding the marks that are relevant for eligibility prediction.

Indirectly: students have to check credentials.

Focus on PR, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TO ACT TR

seeing their friends and seniors predicting the results which are more efficient.

4. EMOTIONS BEFORE / AFTER EM

•Puzzled, lacking assurance>self-possession, Strategic

10. YOUR SOLUTION SL

Students will get the extracted result based on the previous database and using Machine Learning algorithms to get the perfect predicted result.

8. CHANNELS of BEHAVIOR CH

ONLINE

Extract the essential data from the user

OFFLINE

Execution of the essential ML algorithms for the prediction.

Extract online & offline CH of BE

IdeaHackers.NL

4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through GmailRegistration through LinkedIN
FR-2	User Confirmation	Confirmation via EmailConfirmation via OTP
FR-3	User DataCollection	The following details of Students' Score are collected: HSC scores, SSLCgrades, and CGPAif they are applying to PG programs.
FR-4	Evaluation	Using ML algorithms to analyse thedata provided by the students and testing the developed MLmodel withthe supplied data.
FR-5	Prediction	Prediction is done based on the result of evaluation, the universities wherestudents areeligible to apply will be displayed.
FR-6	Output	Based on their eligibility, students move forwardwiththe admissions procedure totheir predicted university and course.

4.2 Non-Functional requirements

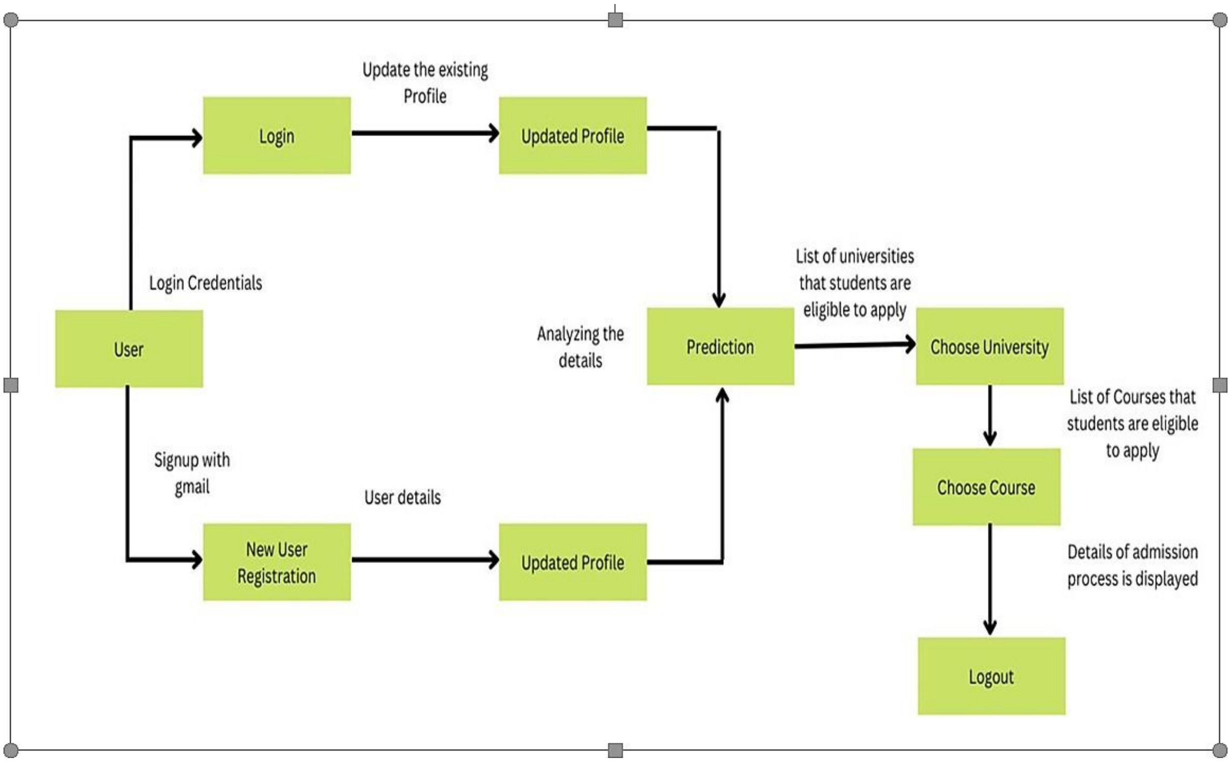
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	User Interface Design for Interactive and Effective Visualization of Progress Customer Satisfaction and Learning Ease
NFR-2	**Security**	Frequent Updates using the Customers feedback. Automatic Logout when the app isnot in use to prevent unauthorized access to useraccounts.
NFR-3	**Reliability**	It is essential thatthe predictor systemwill be consistent in order for the system to produce trustworthy and accurate outcomes
NFR-4	**Performance**	Effectivebecause of usingLogistic regression.
NFR-5	**Availability**	System predictors willbe accessible at anytime,anywhere as needed.

5. PROJECT DESIGN

5.1 Data Flow Diagrams

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


Update the existing Profile
Login
Updated Profile
Login Credentials
List of universities that students are eligible to apply
User
Analyzing the details
Prediction
Choose University
List of Courses that students are eligible to apply
Signup with gmail
Choose Course
User details
New User Registration
Updated Profile
Details of admission process is displayed
Logout

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
	Update Profile	USN-5	Once logged in, I Will be required to update my profile by providing the necessary information	I can complete the profile in order to continue the prediction process.	High	Sprint-2
	Choose University	USN-6	User can view the list of universities to which students are eligible to apply.	I can choose the University from the List of University provided in the dropdown window.	High	Sprint-3
	Choose Course	USN-7	The students will be able to view the courses that they are eligible to participate in as auser.	I can choose the University from the List of University provided in the dropdown window.	Medium	Sprint-3

	Admission Process	USN-8	Using this program, I will be able to view information regarding the Admissions process, including the Date and location Of the certification verification.	Upon completion of prediction, I can view the details of Admission process being displayed.	Low	Sprint-4
Administer	Authentication	USN-9	Authenticating a user's credentials is my responsibilityas an administrator.	There is no problem retrieving and using All the user informationn	High	Sprint-1
	Update Profile	USN-10	In my capacity as an admin, I am able to verify the information Entered by the user.	Upon Confirmation I can access the user's details.	High	Sprint-2
	Prediction	USN-11	As an administrator, I am able to test trained machine learning models	I can test the data provided by the users	High	Sprint-3

5.2 Solution & Technical ArchitectureSolution Architecture

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

- Machine learning algorithms to be used depending on the data we are going to process such as images, sound, text, and numerical values. The algorithms that we can choose according to the objective that you might have it may be Classification algorithms are Regression algorithms. As it is a kind of classification problem you can apply any of the following Algorithms
  - 1. Logistic Regression
  - 2.Decision Tree Classifier
  - 3.Random Forest Classifier
  - 4. KNN

Logistic Regression:

Logistic Regression is used when the dependent variable (target) is categorical. For example,

- To predict whether an email is spam (1) or (0)
- Whether the tumor is malignant (1) or not (0)
- Out of all the algorithms Logistic Regresson got the highest accuracy

There are many numbers of model evaluation techniques for the classification type of machine learning models. the following are widely used as **Accuracy score**, **Confusion matrix,** and **Roc- Auc Curve**

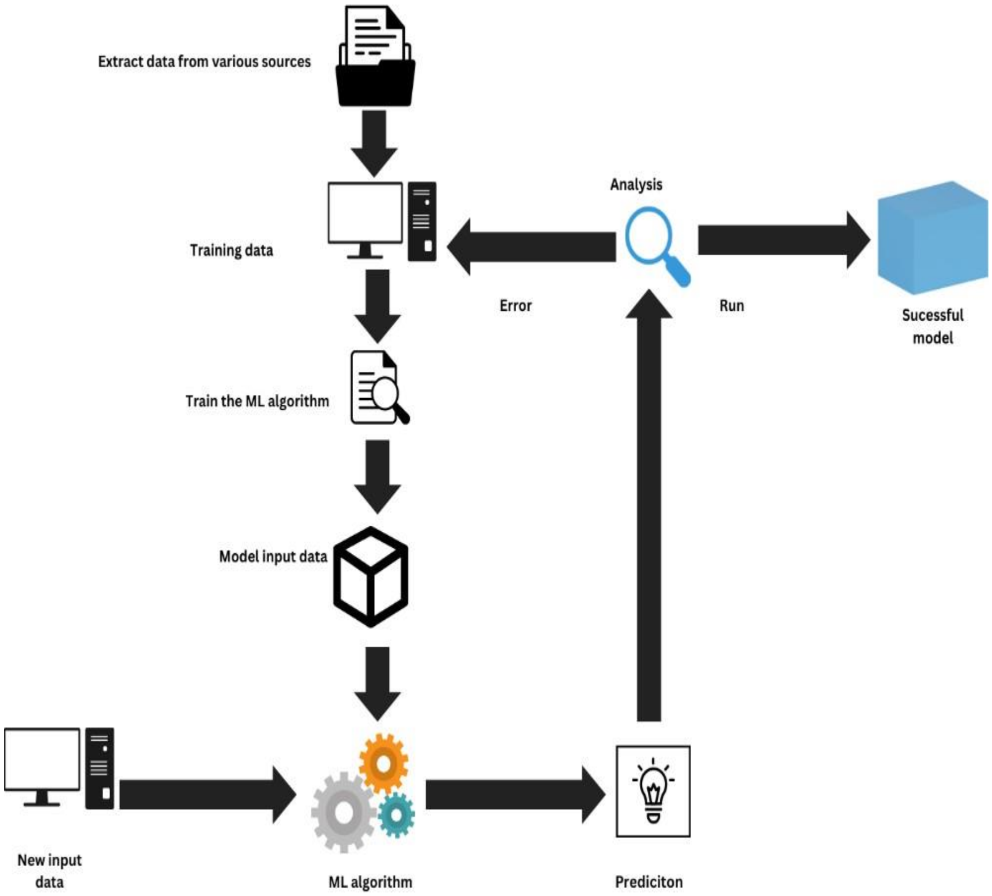
Next, we will be building a web application that is integrated tothe model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

This section has the following tasks

- **Building HTML Pages**
- **Building server-side script**

Next, we will integrate the flask and IBM cloud for deployment. Then finally flask is integrated with the scoring endpoint.

Solution Architecture Diagram:



Technical Architecture:

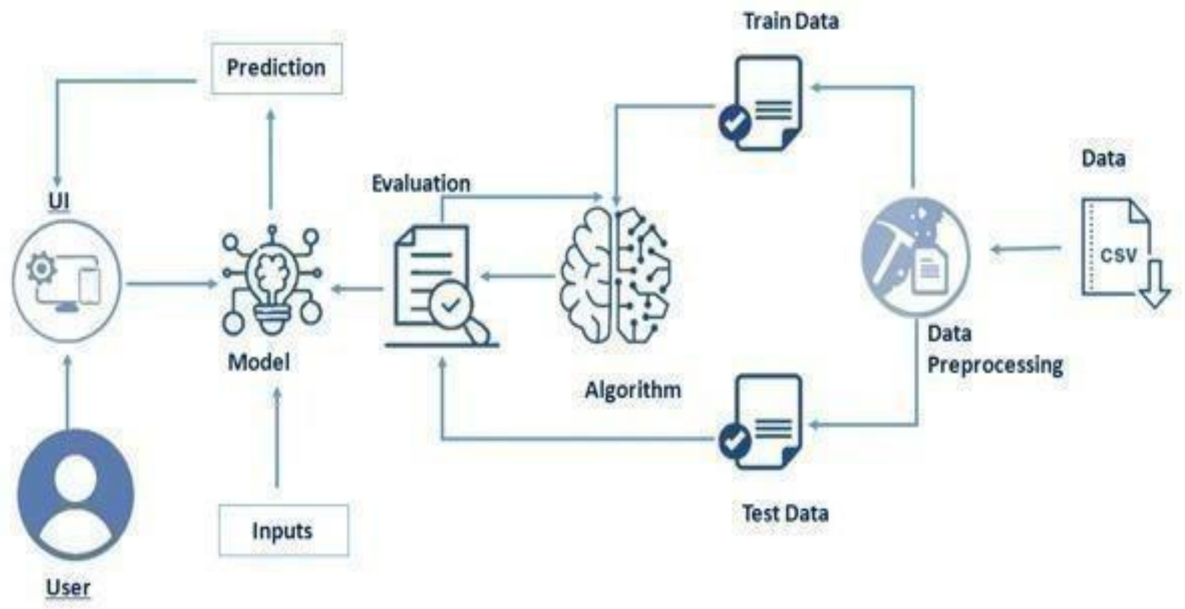


Table – 1 : Components & Technologies

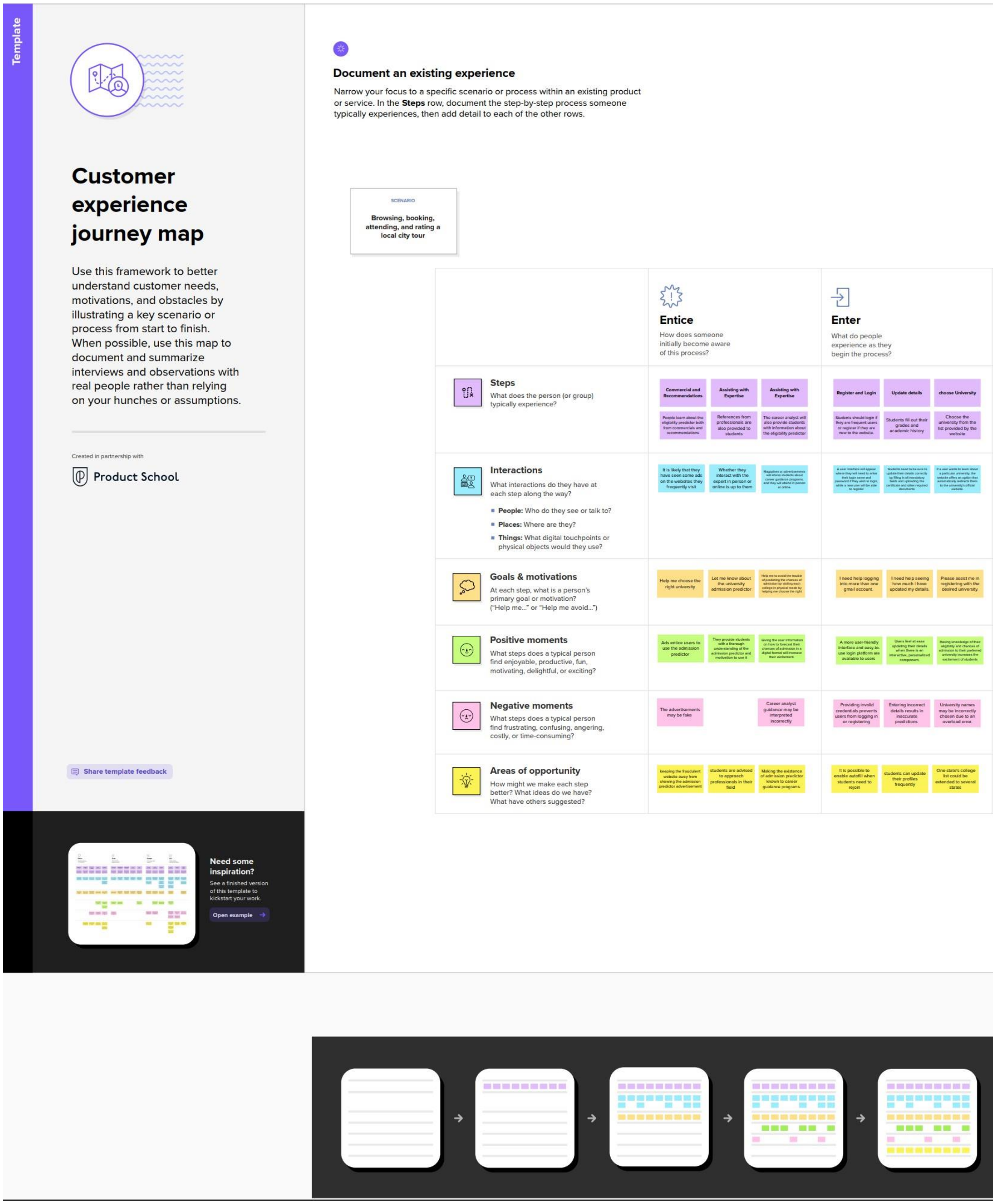
S.No	Component	Description	Technology
1	User Interface	The Front-end part of the application	HTML, CSS
2	Application Logic-1	Logic for a process in the application	Python
3	Application Logic-2	Logic for a process in the application	IBM Watson
4	Application Logic-3	Logic for a process in the application	IBM Watson
5	Database	Data type, Configuration	MySQL
6	Cloud Database	Data base services on cloud	IBM DB2, IBM Cloudant, etc.
7	Libraries	Import Libraries into data	NumPy, Pandas, Seaborn, Matplotlib

8	File Storage	File storage requirements	Local File System
9	Machine Learning Model	Purpose of Machine Learning Model	Admission Prediction Model
10	Training and Testing data	Purpose of training and testing data	Logistic Regression algorithm
11	Accuracy	Accuracy of the tested andtrained data	Root Mean Squared Logarithmic Error (RMSLE), Mean Squared Error (MSE)
12	Infrastructure	Cloud Local Server Configuration	Local

Table - 2: Application Characteristics

Characteristics	Descripton	Technologies Used
-Source Frameworks	List the open-sourceframe works used	Flask Framework
Implementations	The user profile has been stored ina secured way	Encryptions
Architecture	Many computations can be done in a time saving and effectiveway	Logistic Regression
Availability	Our web application is available at any time and at anyplace	IBM Load Balancer
Performance	As logistic regression is applied to develop the performance will be more	Logistic Regression

5.3 User Stories



Template

Customer experience journey map

Use this framework to better understand customer needs, motivations, and obstacles by illustrating a key scenario or process from start to finish. When possible, use this map to document and summarize interviews and observations with real people rather than relying on your hunches or assumptions.

Created in partnership with

Product School

Share template feedback

Need some inspiration?

See a finished version of this template to kickstart your work.

Open example →

Document an existing experience

Narrow your focus to a specific scenario or process within an existing product or service. In the **Steps** row, document the step-by-step process someone typically experiences, then add detail to each of the other rows.

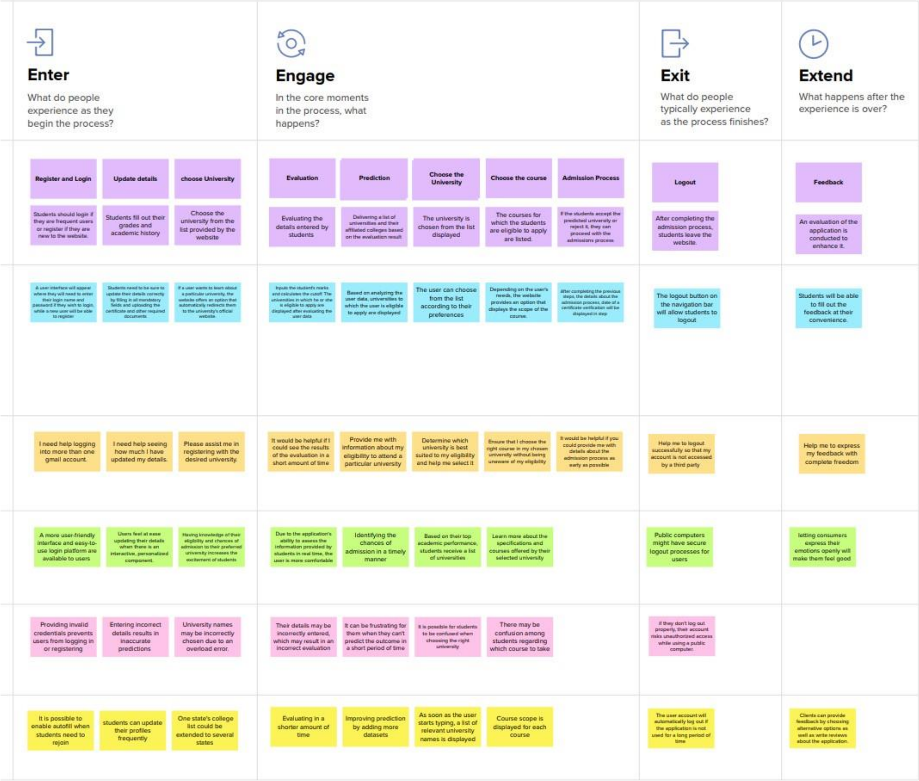
SCENARIO

Browsing, booking, attending, and rating a local city tour

	Entice How does someone initially become aware of this process?			**Enter** What do people experience as they begin the process?		
Steps What does the person (or group) typically experience?	**Commercial and Recommendations**	**Assisting with Expertise**	**Assisting with Expertise**	**Register and Login**	**Update details**	**choose University**
	People learn about the eligibility predictor both from commercials and recommendations	References from professionals are also provided to students	The career analyst will also provide students with information about the eligibility predictor	Students should login if they are frequent users or register if they are new to the website.	Students fill out their grades and academic history	Choose the university from the list provided by the website
Interactions What interactions do they have at each step along the way? ▪ **People:** Who do they see or talk to? ▪ **Places:** Where are they? ▪ **Things:** What digital touchpoints or physical objects would they use?	It is likely that they have seen some ads on the websites they frequently visit	Whether they interact with the expert in person or online is up to them	Magazines or advertisements will inform students about career guidance programs, and they will attend in person or online.	A user interface will appear where they will need to enter their login name and password if they wish to login, while a new user will be able to register	Students need to be sure to update their details correctly by filling in all mandatory fields and uploading the certificate and other required documents	If a user wants to learn about a particular university, the website offers an option that automatically redirects them to the university's official website.
Goals & motivations At each step, what is a person's primary goal or motivation? ("Help me…" or "Help me avoid…")	Help me choose the right university	Let me know about the university admission predictor	Help me to avoid the trouble of predicting the chances of admission by visiting each college in physical mode by helping me choose the right	I need help logging into more than one gmail account.	I need help seeing how much I have updated my details.	Please assist me in registering with the desired university.
Positive moments What steps does a typical person find enjoyable, productive, fun, motivating, delightful, or exciting?	Ads entice users to use the admission predictor	They provide students with a thorough understanding of the admission predictor and motivation to use it	Giving the user information on how to forecast their chances of admission in a digital format will increase their excitement.	A more user-friendly interface and easy-to-use login platform are available to users	Users feel at ease updating their details when there is an interactive, personalized component.	Having knowledge of their eligibility and chances of admission to their preferred university increases the excitement of students
Negative moments What steps does a typical person find frustrating, confusing, angering, costly, or time-consuming?	The advertisements may be fake		Career analyst guidance may be interpreted incorrectly	Providing invalid credentials prevents users from logging in or registering	Entering incorrect details results in inaccurate predictions	University names may be incorrectly chosen due to an overload error.
Areas of opportunity How might we make each step better? What ideas do we have? What have others suggested?	keeping the fraudulent website away from showing the admission predictor advertisement	students are advised to approach professionals in their field	Making the existence of admission predictor known to career guidance programs.	It is possible to enable autofill when students need to rejoin	students can update their profiles frequently	One state's college list could be extended to several states

TIP

As you add steps to the experience, move each these "Five Es" the left or right depending on the scenario you are documenting.



Enter What do people experience as they begin the process?	Engage In the core moments in the process, what happens?	Exit What do people typically experience as the process finishes?	Extend What happens after the experience is over?
Register and Login **Update details** **choose University**	**Evaluation** **Prediction** **Choose the University** **Choose the course** **Admission Process**	**Logout**	**Feedback**
Students should login if they are frequent users or register if they are new to the website. Students fill out their grades and academic history Choose the university from the list provided by the website	Evaluating the details entered by students Delivering a list of universities and their affiliated colleges based on the evaluation result The university is chosen from the list displayed The courses for which the students are eligible to apply are listed. If the students accept the predicted university or reject it, they can proceed with the admissions process	After completing the admission process, students leave the website.	An evaluation of the application is conducted to enhance it.
A user interface will appear where they will need to enter their login name and password if they wish to login, while a new user will be able to register Students need to be sure to update their details correctly by filling in all mandatory fields and uploading the certificate and other required documents If a user wants to learn about a particular university, the website offers an option that automatically redirects them to the university's official website	Inputs the student's marks and calculates the cutoff. The universities in which he or she is eligible to apply are displayed after evaluating the user data Based on analyzing the user data, universities to which the user is eligible to apply are displayed The user can choose from the list according to their preferences Depending on the user's needs, the website provides an option that displays the scope of the course. After completing the previous steps, the details about the admission process, date of a certificate verification will be displayed in step	The logout button on the navigation bar will allow students to logout	Students will be able to fill out the feedback at their convenience.
I need help logging into more than one gmail account. I need help seeing how much I have updated my details. Please assist me in registering with the desired university.	It would be helpful if I could see the results of the evaluation in a short amount of time Provide me with information about my eligibility to attend a particular university Determine which university is best suited to my eligibility and help me select it Ensure that I choose the right course in my chosen university without being unaware of my eligibility It would be helpful if you could provide me with details about the admission process as early as possible	Help me to logout successfully so that my account is not accessed by a third party	Help me to express my feedback with complete freedom
A more user-friendly interface and easy-to-use login platform are available to users Users feel at ease updating their details when there is an interactive, personalized component. Having knowledge of their eligibility and chances of admission to their preferred university increases the excitement of students	Due to the application's ability to assess the information provided by students in real time, the user is more comfortable Identifying the chances of admission in a timely manner Based on their top academic performance, students receive a list of universities Learn more about the specifications and courses offered by their selected university	Public computers might have secure logout processes for users	letting consumers express their emotions openly will make them feel good
Providing invalid credentials prevents users from logging in or registering Entering incorrect details results in inaccurate predictions University names may be incorrectly chosen due to an overload error.	Their details may be incorrectly entered, which may result in an incorrect evaluation It can be frustrating for them when they can't predict the outcome in a short period of time It is possible for students to be confused when choosing the right university There may be confusion among students regarding which course to take	If they don't log out properly, their account risks unauthorized access while using a public computer.	
It is possible to enable autofill when students need to rejoin students can update their profiles frequently One state's college list could be extended to several states	Evaluating in a shorter amount of time Improving prediction by adding more datasets As soon as the user starts typing, a list of relevant university names is displayed Course scope is displayed for each course	The user account will automatically log out if the application is not used for a long period of time	Clients can provide feedback by choosing alternative options as well as write reviews about the application.

6. PROJECT PLANNING & SCHEDULING

Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story/ Task	Story Points	Priority	Team Members
Sprint-1	University Registration	USN-1	As a student, I can registerforth application by entering my email, password by confirming my password.	2	High	2
Sprint-1		USN-2	As a student, I will receive confirmationemail once I have registered for the application from the university.	1	High	1
Sprint-2		USN-3	As a student, I can registerfor the application throughuniversity by uploading my mark statements. Upload original copy of the Marksheets.	2	Low	2
Sprint-3		USN-4	As a Students, I can registerforthe application through Gmailwithall eligibility. Students can upload extra course completion certificates.	2	Medium	2
Sprint-4	Login by user name	USN-5	As a Student, I can loginto the application byentering email & password	1	High	2
	Dashboard		Check dashboard and upload the details according to university criteria.			4

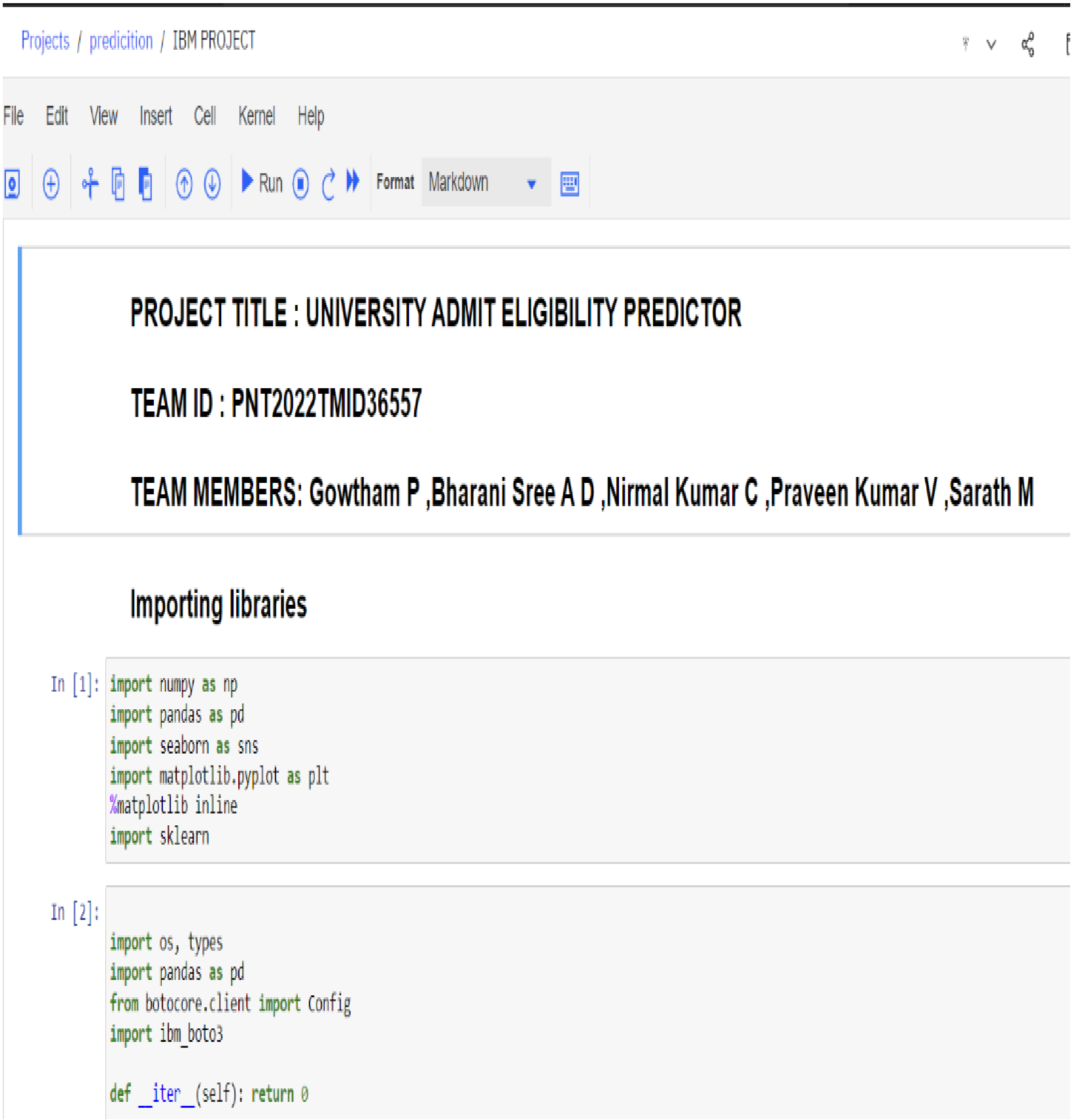
6.1 Sprint Delivery Schedule

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	8	6 Days	24th Oct 2022	29th Oct 2022	8	29th Oct 2022
Sprint-2	5	6 Days	30th Oct 2022	05th Nov 2022	5	05th Nov 2022
Sprint-3	15	6 Days	7th Nov 2022	12th Nov 2022	15	12th Nov 2022
Sprint-4	5	6 Days	14th Nov 2022	15th Nov 2022	5	15th Nov 2022

7. CODING & SOLUTIONING

7.1 Feature 1

Firstly developed a Jupiter notebook file then deployed in IBM Cloud by using services like Watson studio, machine learning and cloud object.


Projects / predicition / IBM PROJECT
File Edit View Insert Cell Kernel Help
Run
Format
Markdown

PROJECT TITLE : UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

TEAM ID : PNT2022TMID36557

TEAM MEMBERS: Gowtham P ,Bharani Sree A D ,Nirmal Kumar C ,Praveen Kumar V ,Sarath M

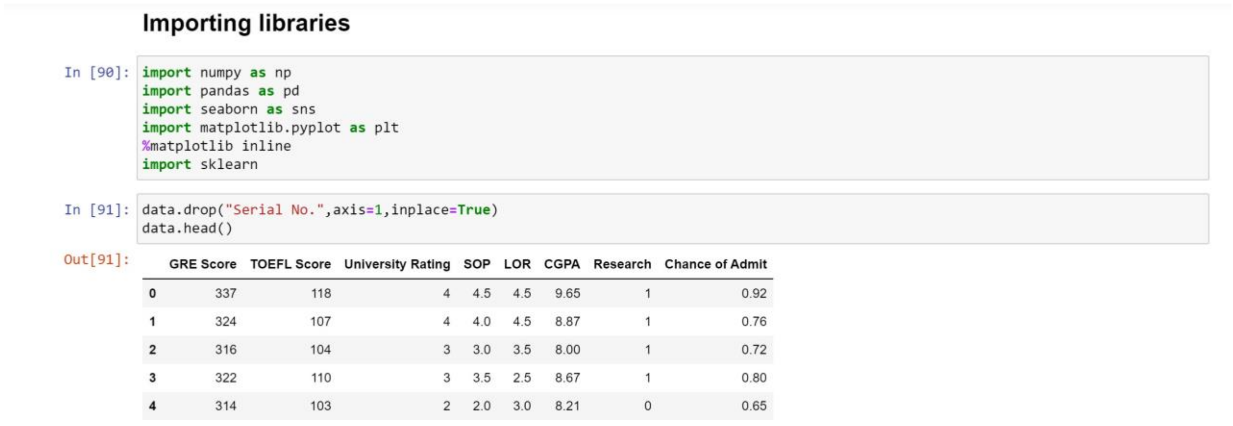
Importing libraries

In [1]:

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

In [2]:

```

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

def __iter__(self): return 0
```



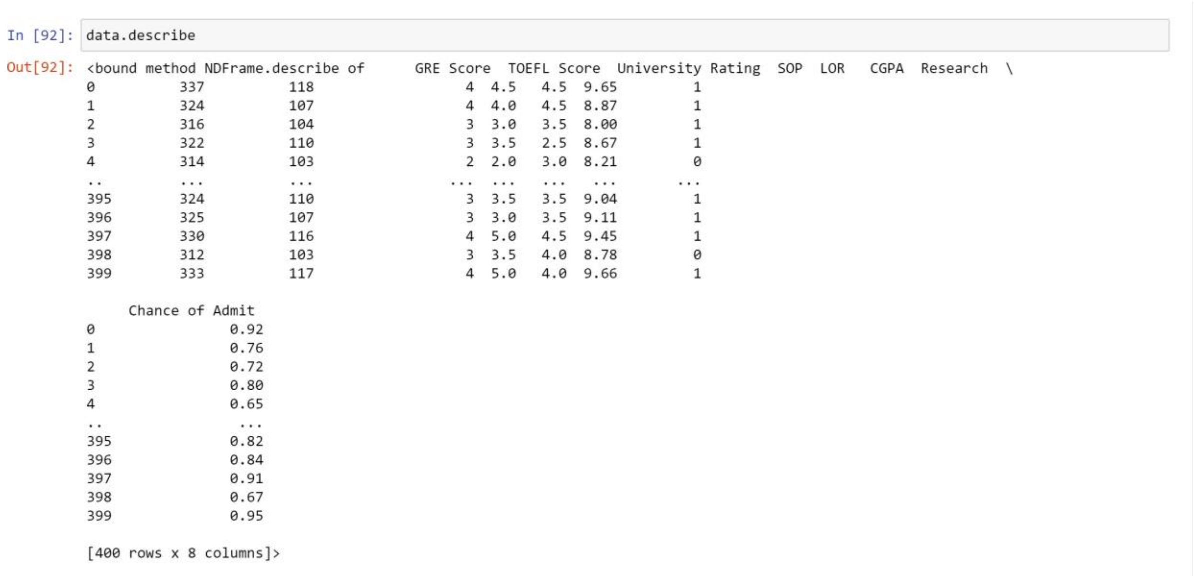
Importing libraries

In [90]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import sklearn
```

In [91]:
```
data.drop("Serial No.",axis=1,inplace=True)
data.head()
```

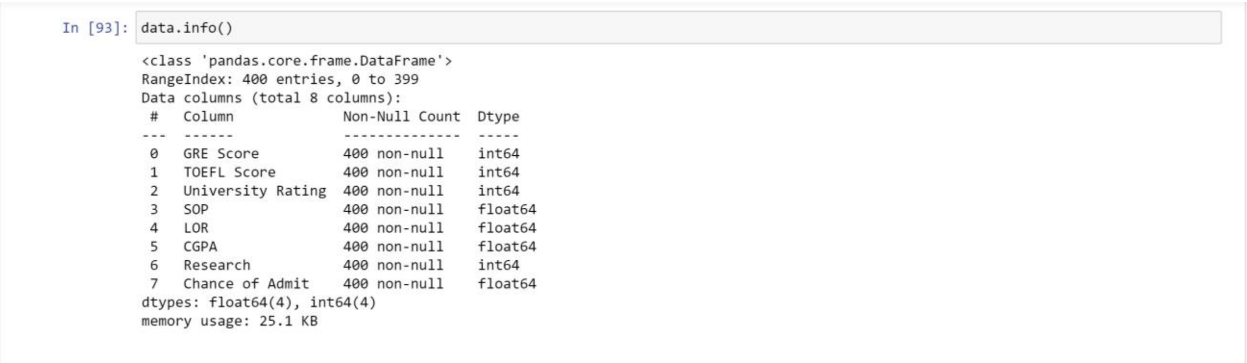
Out[91]:

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



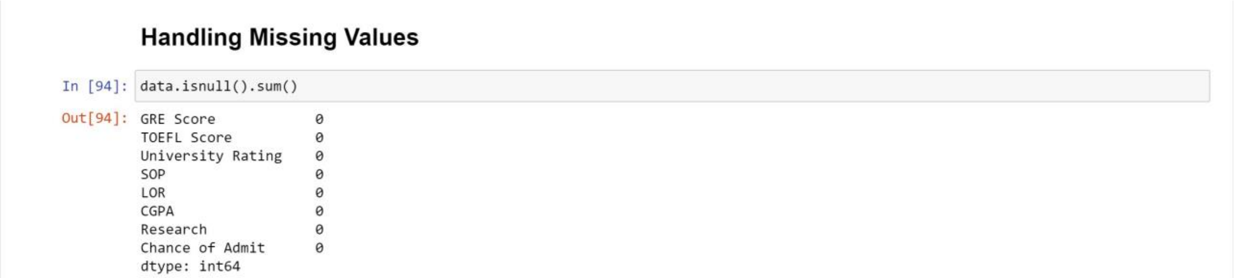
In [92]:
```
data.describe
```

Out[92]:
```
<bound method NDFrame.describe of      GRE Score  TOEFL Score  University Rating  SOP  LOR   CGPA  Research  \
0          337          118                  4  4.5  4.5  9.65         1   
1          324          107                  4  4.0  4.5  8.87         1   
2          316          104                  3  3.0  3.5  8.00         1   
3          322          110                  3  3.5  2.5  8.67         1   
4          314          103                  2  2.0  3.0  8.21         0   
..         ...          ...                ...  ...  ...   ...       ...   
395        324          110                  3  3.5  3.5  9.04         1   
396        325          107                  3  3.0  3.5  9.11         1   
397        330          116                  4  5.0  4.5  9.45         1   
398        312          103                  3  3.5  4.0  8.78         0   
399        333          117                  4  5.0  4.0  9.66         1   

     Chance of Admit  
0               0.92  
1               0.76  
2               0.72  
3               0.80  
4               0.65  
..               ...  
395             0.82  
396             0.84  
397             0.91  
398             0.67  
399             0.95  

[400 rows x 8 columns]>
```



In [93]:
```
data.info()
```

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



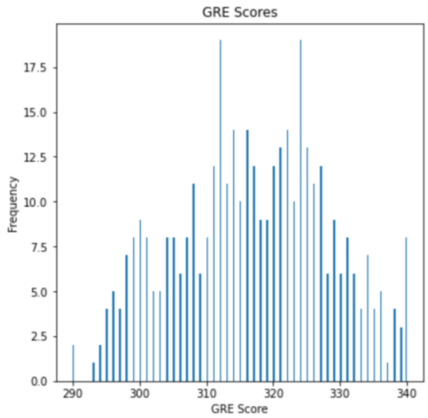
Handling Missing Values

In [94]:
```
data.isnull().sum()
```

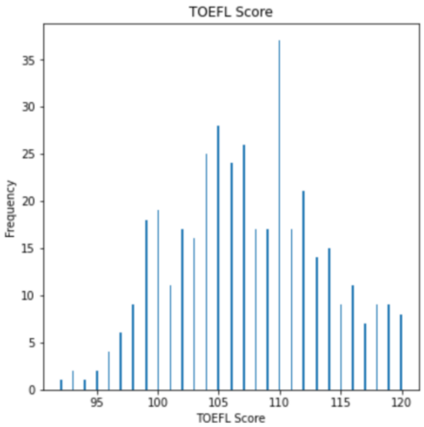
Out[94]:
```
GRE Score            0
TOEFL Score          0
University Rating    0
SOP                  0
LOR                  0
CGPA                 0
Research             0
Chance of Admit      0
dtype: int64
```

Data Visualization

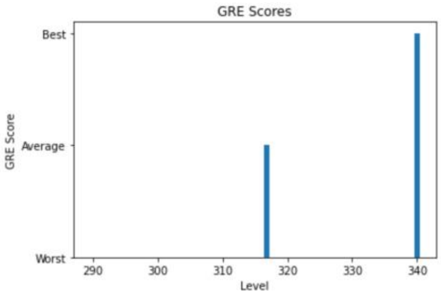
```
In [95]: data["GRE Score"].plot(kind = 'hist',bins = 200,figsize = (6,6))
         plt.title("GRE Scores")
         plt.xlabel("GRE Score")
         plt.ylabel("Frequency")
         plt.show()
```



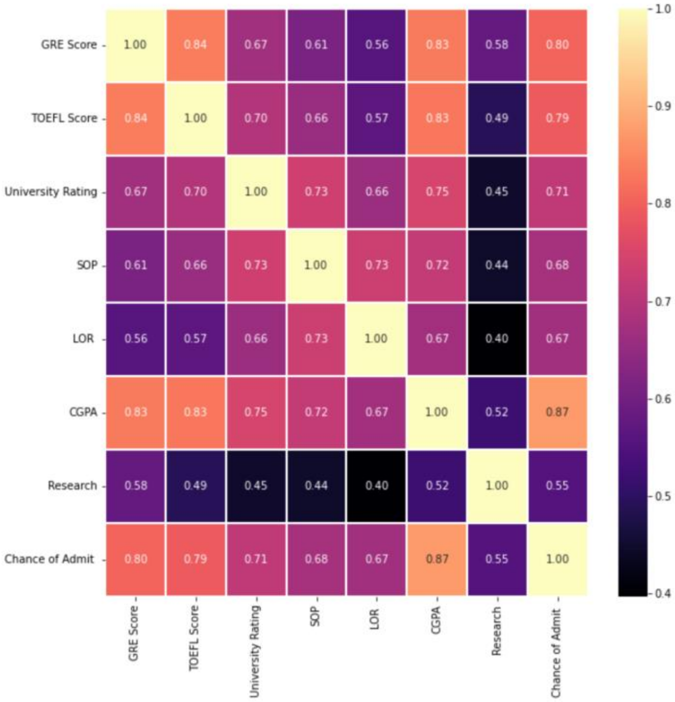
```
In [96]: data["TOEFL Score"].plot(kind = 'hist',bins = 200,figsize = (6,6))
         plt.title("TOEFL Score")
         plt.xlabel("TOEFL Score")
         plt.ylabel("Frequency")
         plt.show()
```



```
In [97]: g = np.array([data["GRE Score"].min(),data["GRE Score"].mean(),data["GRE Score"].max()])
         h = ["Worst","Average","Best"]
         plt.bar(g,h)
         plt.title("GRE Scores")
         plt.xlabel("Level")
         plt.ylabel("GRE Score")
         plt.show()
```

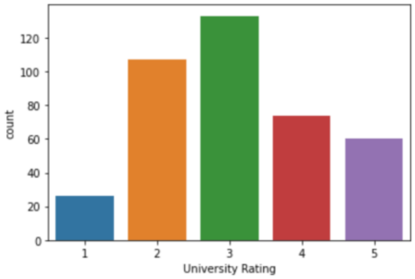


```
In [98]: plt.figure(figsize=(10, 10))
         sns.heatmap(data.corr(), annot=True, linewidths=0.05, fmt= '.2f',cmap="magma")
         plt.show()
```



```
In [99]: data.Research.value_counts()
         sns.countplot(x="University Rating",data=data)
```

```
Out[99]: <AxesSubplot:xlabel='University Rating', ylabel='count'>
```



Training and Testing Split

In [100]:
```
X=data.drop(['Chance of Admit '],axis=1)
y=data['Chance of Admit ']
```

In [101]:
```
X
```

Out[101]:

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research
0	337	118	4	4.5	4.5	9.65	1
1	324	107	4	4.0	4.5	8.87	1
2	316	104	3	3.0	3.5	8.00	1
3	322	110	3	3.5	2.5	8.67	1
4	314	103	2	2.0	3.0	8.21	0
...	...	...	...	...	...	...	...
395	324	110	3	3.5	3.5	9.04	1
396	325	107	3	3.0	3.5	9.11	1
397	330	116	4	5.0	4.5	9.45	1
398	312	103	3	3.5	4.0	8.78	0
399	333	117	4	5.0	4.0	9.66	1

400 rows × 7 columns

In [102]:
```
y
```

Out[102]:
```
0      0.92
1      0.76
2      0.72
3      0.80
4      0.65
       ...
395    0.82
396    0.84
397    0.91
398    0.67
399    0.95
Name: Chance of Admit , Length: 400, dtype: float64
```

In [103]:
```
from sklearn.model_selection import train_test_split
X_Train, X_Test, y_Train, y_Test = train_test_split(X, y, test_size=0.15)
```

In [104]:
```
X_Train.shape
```

Out[104]:
```
(340, 7)
```

In [105]:
```
y_Train.shape
```

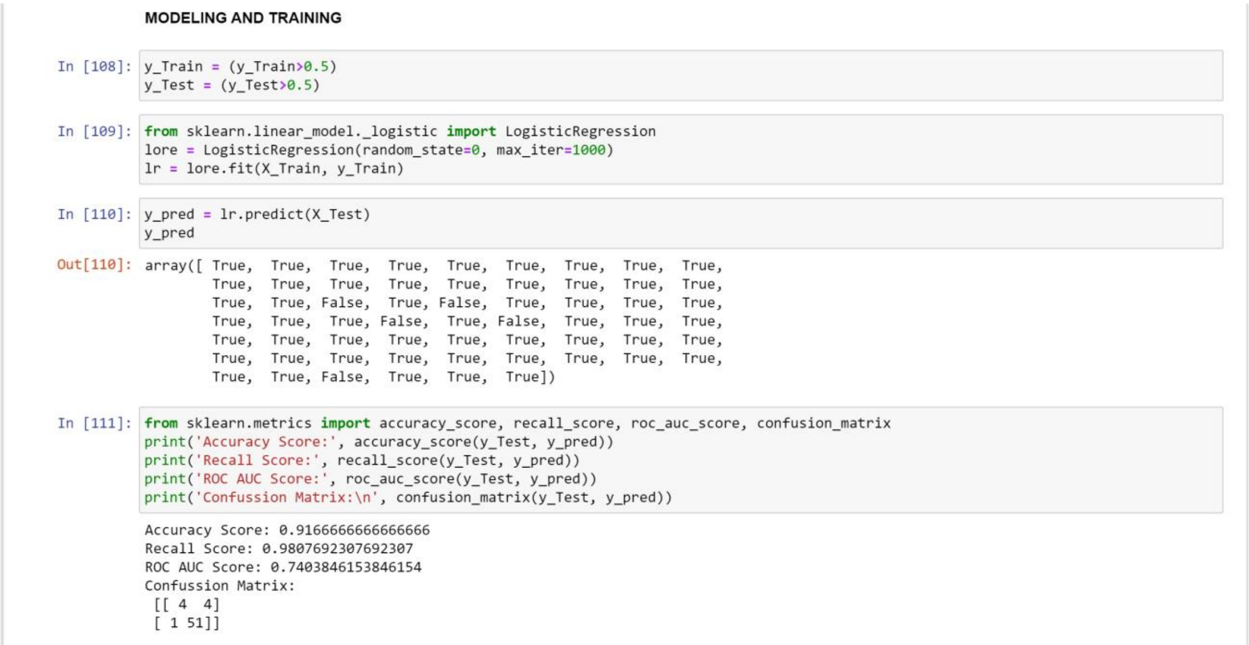
Out[105]:
```
(340,)
```

In [106]:
```
y_Test.shape
```

Out[106]:
```
(60,)
```

In [107]:
```
X_Test.shape
```

Out[107]:
```
(60, 7)
```



MODELING AND TRAINING

```
In [108]: y_Train = (y_Train>0.5)
          y_Test = (y_Test>0.5)
```

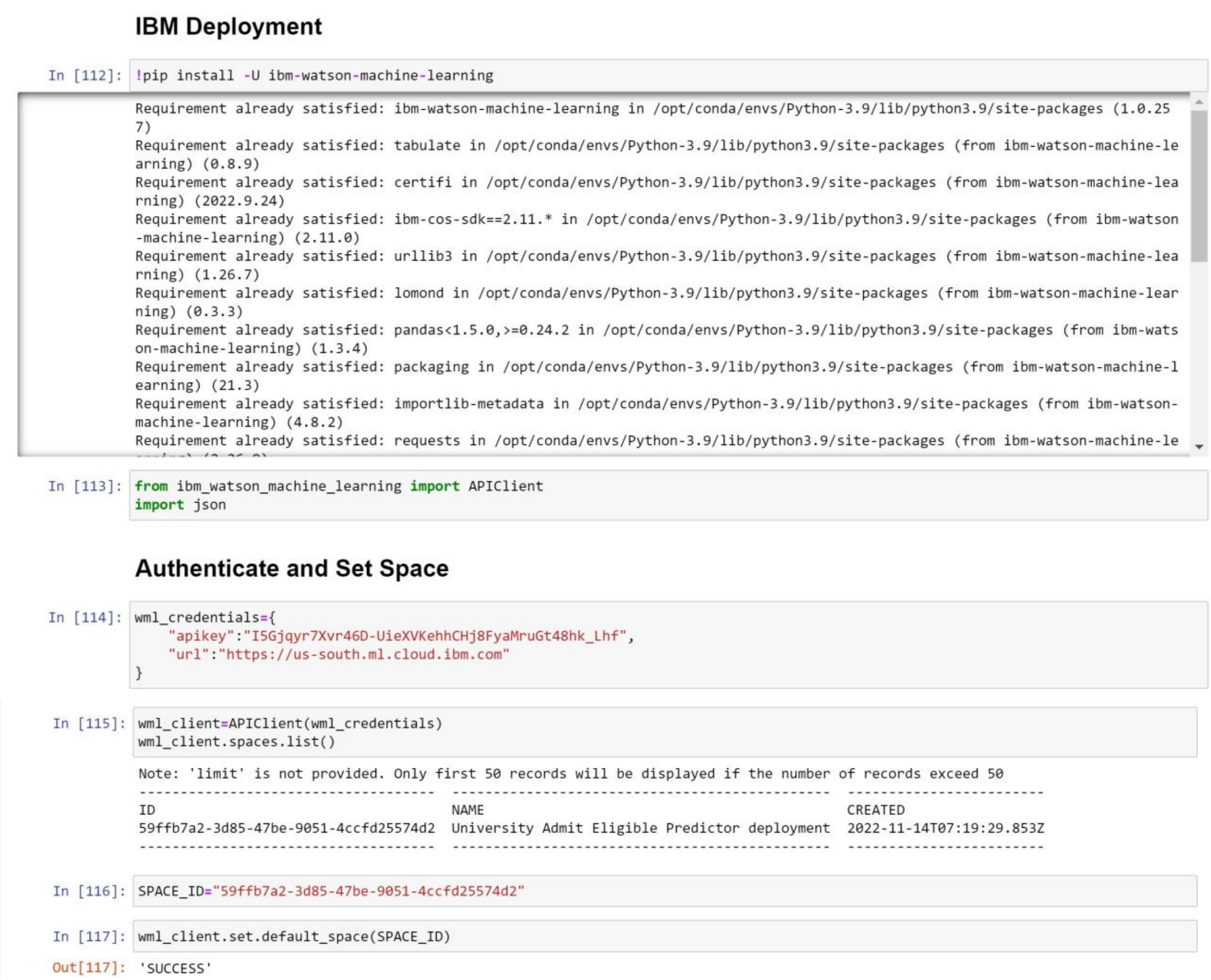
```
In [109]: from sklearn.linear_model._logistic import LogisticRegression
          lore = LogisticRegression(random_state=0, max_iter=1000)
          lr = lore.fit(X_Train, y_Train)
```

```
In [110]: y_pred = lr.predict(X_Test)
          y_pred
```

```
Out[110]: array([ True,  True,  True,  True,  True,  True,  True,  True,  True,
                  True,  True,  True,  True,  True,  True,  True,  True,  True,
                  True,  True, False,  True, False,  True,  True,  True,  True,
                  True,  True,  True, False,  True, False,  True,  True,  True,
                  True,  True,  True,  True,  True,  True,  True,  True,  True,
                  True,  True,  True,  True,  True,  True,  True,  True,  True,
                  True,  True, False,  True,  True,  True])
```

```
In [111]: from sklearn.metrics import accuracy_score, recall_score, roc_auc_score, confusion_matrix
          print('Accuracy Score:', accuracy_score(y_Test, y_pred))
          print('Recall Score:', recall_score(y_Test, y_pred))
          print('ROC AUC Score:', roc_auc_score(y_Test, y_pred))
          print('Confussion Matrix:\n', confusion_matrix(y_Test, y_pred))
```

```
Accuracy Score: 0.9166666666666666
Recall Score: 0.9807692307692307
ROC AUC Score: 0.7403846153846154
Confussion Matrix:
 [[ 4  4]
 [ 1 51]]
```



IBM Deployment

```
In [112]: !pip install -U ibm-watson-machine-learning
```

```
Requirement already satisfied: ibm-watson-machine-learning in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.25
7)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-le
arning) (0.8.9)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-lea
rning) (2022.9.24)
Requirement already satisfied: ibm-cos-sdk==2.11.* in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson
-machine-learning) (2.11.0)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-lea
rning) (1.26.7)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-lear
ning) (0.3.3)
Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-wats
on-machine-learning) (1.3.4)
Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-l
earning) (21.3)
Requirement already satisfied: importlib-metadata in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-
machine-learning) (4.8.2)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-le
```

```
In [113]: from ibm_watson_machine_learning import APIClient
          import json
```

Authenticate and Set Space

```
In [114]: wml_credentials={
              "apikey":"I5Gjqyr7Xvr46D-UieXVKehhCHj8FyaMruGt48hk_Lhf",
              "url":"https://us-south.ml.cloud.ibm.com"
          }
```

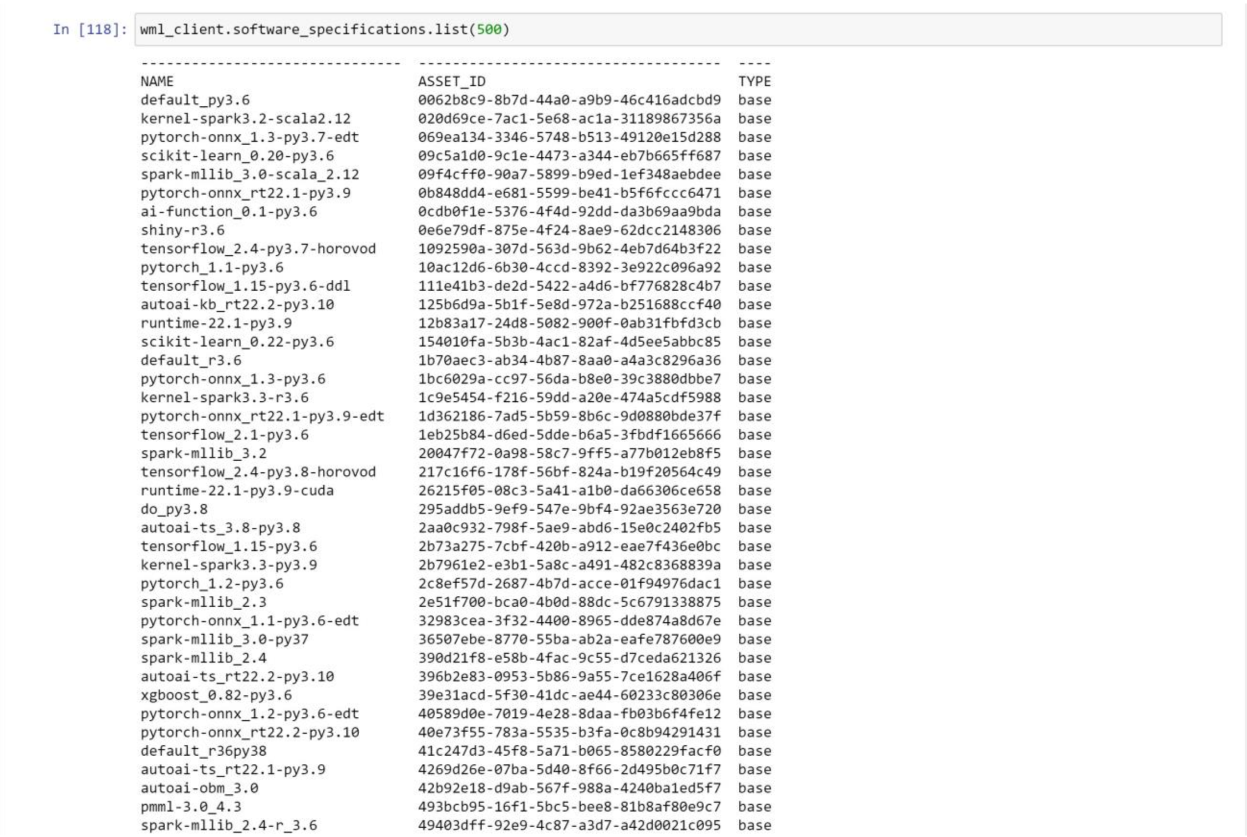
```
In [115]: wml_client=APIClient(wml_credentials)
          wml_client.spaces.list()
```

```
Note: 'limit' is not provided. Only first 50 records will be displayed if the number of records exceed 50
------------------------------------  ----------------------------------------------  ------------------------
ID                                    NAME                                            CREATED
59ffb7a2-3d85-47be-9051-4ccfd25574d2  University Admit Eligible Predictor deployment  2022-11-14T07:19:29.853Z
------------------------------------  ----------------------------------------------  ------------------------
```

```
In [116]: SPACE_ID="59ffb7a2-3d85-47be-9051-4ccfd25574d2"
```

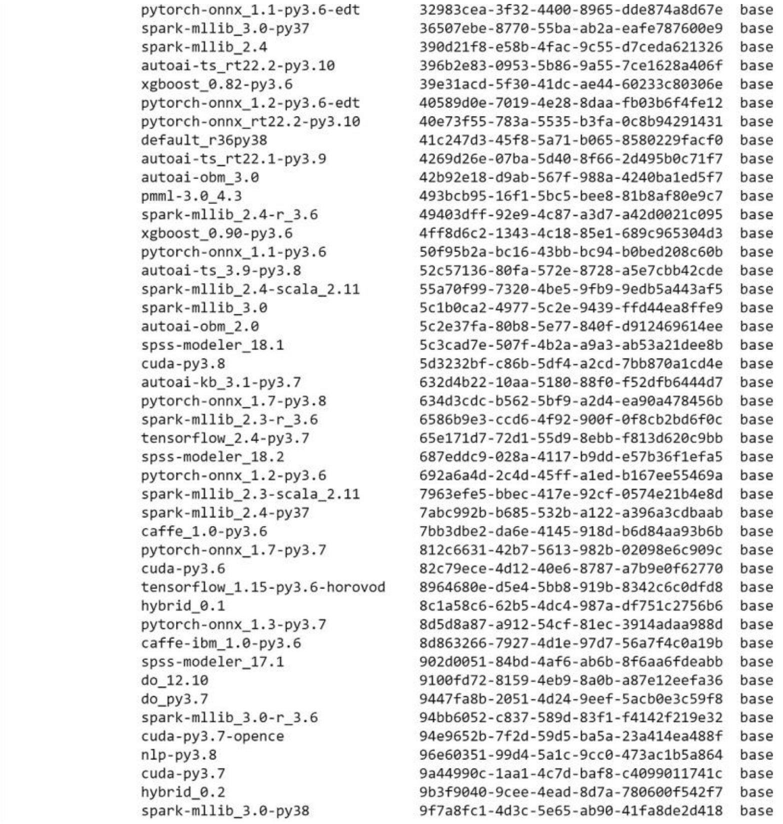
```
In [117]: wml_client.set.default_space(SPACE_ID)
```

```
Out[117]: 'SUCCESS'
```



```
In [118]: wml_client.software_specifications.list(500)
```

```
------------------------------  ------------------------------------  ----
NAME                            ASSET_ID                              TYPE
default_py3.6                   0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
kernel-spark3.2-scala2.12       020d69ce-7ac1-5e68-ac1a-31189867356a  base
pytorch-onnx_1.3-py3.7-edt      069ea134-3346-5748-b513-49120e15d288  base
scikit-learn_0.20-py3.6         09c5a1d0-9c1e-4473-a344-eb7b665ff687  base
spark-mllib_3.0-scala_2.12      09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
pytorch-onnx_rt22.1-py3.9       0b848dd4-e681-5599-be41-b5f6fccc6471  base
ai-function_0.1-py3.6           0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  base
shiny-r3.6                      0e6e79df-875e-4f24-8ae9-62dcc2148306  base
tensorflow_2.4-py3.7-horovod    1092590a-307d-563d-9b62-4eb7d64b3f22  base
pytorch_1.1-py3.6               10ac12d6-6b30-4ccd-8392-3e922c096a92  base
tensorflow_1.15-py3.6-ddl       111e41b3-de2d-5422-a4d6-bf776828c4b7  base
autoai-kb_rt22.2-py3.10         125b6d9a-5b1f-5e8d-972a-b251688ccf40  base
runtime-22.1-py3.9              12b83a17-24d8-5082-900f-0ab31fbfd3cb  base
scikit-learn_0.22-py3.6         154010fa-5b3b-4ac1-82af-4d5ee5abbc85  base
default_r3.6                    1b70aec3-ab34-4b87-8aa0-a4a3c8296a36  base
pytorch-onnx_1.3-py3.6          1bc6029a-cc97-56da-b8e0-39c3880dbbe7  base
kernel-spark3.3-r3.6            1c9e5454-f216-59dd-a20e-474a5cdf5988  base
pytorch-onnx_rt22.1-py3.9-edt   1d362186-7ad5-5b59-8b6c-9d0880bde37f  base
tensorflow_2.1-py3.6            1eb25b84-d6ed-5dde-b6a5-3fbdf1665666  base
spark-mllib_3.2                 20047f72-0a98-58c7-9ff5-a77b012eb8f5  base
tensorflow_2.4-py3.8-horovod    217c16f6-178f-56bf-824a-b19f20564c49  base
runtime-22.1-py3.9-cuda         26215f05-08c3-5a41-a1b0-da66306ce658  base
do_py3.8                        295addb5-9ef9-547e-9bf4-92ae3563e720  base
autoai-ts_3.8-py3.8             2aa0c932-798f-5ae9-abd6-15e0c2402fb5  base
tensorflow_1.15-py3.6           2b73a275-7cbf-420b-a912-eae7f436e0bc  base
kernel-spark3.3-py3.9           2b7961e2-e3b1-5a8c-a491-482c8368839a  base
pytorch_1.2-py3.6               2c8ef57d-2687-4b7d-acce-01f94976dac1  base
spark-mllib_2.3                 2e51f700-bca0-4b0d-88dc-5c6791338875  base
pytorch-onnx_1.1-py3.6-edt      32983cea-3f32-4400-8965-dde874a8d67e  base
spark-mllib_3.0-py37            36507ebe-8770-55ba-ab2a-eafe787600e9  base
spark-mllib_2.4                 390d21f8-e58b-4fac-9c55-d7ceda621326  base
autoai-ts_rt22.2-py3.10         396b2e83-0953-5b86-9a55-7ce1628a406f  base
xgboost_0.82-py3.6              39e31acd-5f30-41dc-ae44-60233c80306e  base
pytorch-onnx_1.2-py3.6-edt      40589d0e-7019-4e28-8daa-fb03b6f4fe12  base
pytorch-onnx_rt22.2-py3.10      40e73f55-783a-5535-b3fa-0c8b94291431  base
default_r36py38                 41c247d3-45f8-5a71-b065-8580229facf0  base
autoai-ts_rt22.1-py3.9          4269d26e-07ba-5d40-8f66-2d495b0c71f7  base
autoai-obm_3.0                  42b92e18-d9ab-567f-988a-4240ba1ed5f7  base
pmml-3.0_4.3                    493bcb95-16f1-5bc5-bee8-81b8af80e9c7  base
spark-mllib_2.4-r_3.6           49403dff-92e9-4c87-a3d7-a42d0021c095  base
```



```
pytorch-onnx_1.1-py3.6-edt      32983cea-3f32-4400-8965-dde874a8d67e  base
spark-mllib_3.0-py37            36507ebe-8770-55ba-ab2a-eafe787600e9  base
spark-mllib_2.4                 390d21f8-e58b-4fac-9c55-d7ceda621326  base
autoai-ts_rt22.2-py3.10         396b2e83-0953-5b86-9a55-7ce1628a406f  base
xgboost_0.82-py3.6              39e31acd-5f30-41dc-ae44-60233c80306e  base
pytorch-onnx_1.2-py3.6-edt      40589d0e-7019-4e28-8daa-fb03b6f4fe12  base
pytorch-onnx_rt22.2-py3.10      40e73f55-783a-5535-b3fa-0c8b94291431  base
default_r36py38                 41c247d3-45f8-5a71-b065-8580229facf0  base
autoai-ts_rt22.1-py3.9          4269d26e-07ba-5d40-8f66-2d495b0c71f7  base
autoai-obm_3.0                  42b92e18-d9ab-567f-988a-4240ba1ed5f7  base
pmml-3.0_4.3                    493bcb95-16f1-5bc5-bee8-81b8af80e9c7  base
spark-mllib_2.4-r_3.6           49403dff-92e9-4c87-a3d7-a42d0021c095  base
```

```
xgboost_0.90-py3.6              4ff8d6c2-1343-4c18-85e1-689c965304d3  base
pytorch-onnx_1.1-py3.6          50f95b2a-bc16-43bb-bc94-b0bed208c60b  base
autoai-ts_3.9-py3.8             52c57136-80fa-572e-8728-a5e7cbb42cde  base
spark-mllib_2.4-scala_2.11      55a70f99-7320-4be5-9fb9-9edb5a443af5  base
spark-mllib_3.0                 5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9  base
autoai-obm_2.0                  5c2e37fa-80b8-5e77-840f-d912469614ee  base
spss-modeler_18.1               5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b  base
cuda-py3.8                      5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e  base
autoai-kb_3.1-py3.7             632d4b22-10aa-5180-88f0-f52dfb6444d7  base
pytorch-onnx_1.7-py3.8          634d3cdc-b562-5bf9-a2d4-ea90a478456b  base
spark-mllib_2.3-r_3.6           6586b9e3-ccd6-4f92-900f-0f8cb2bd6f0c  base
tensorflow_2.4-py3.7            65e171d7-72d1-55d9-8ebb-f813d620c9bb  base
spss-modeler_18.2               687eddc9-028a-4117-b9dd-e57b36f1efa5  base
pytorch-onnx_1.2-py3.6          692a6a4d-2c4d-45ff-a1ed-b167ee55469a  base
spark-mllib_2.3-scala_2.11      7963efe5-bbec-417e-92cf-0574e21b4e8d  base
spark-mllib_2.4-py37            7abc992b-b685-532b-a122-a396a3cdbaab  base
caffe_1.0-py3.6                 7bb3dbe2-da6e-4145-918d-b6d84aa93b6b  base
pytorch-onnx_1.7-py3.7          812c6631-42b7-5613-982b-02098e6c909c  base
cuda-py3.6                      82c79ece-4d12-40e6-8787-a7b9e0f62770  base
tensorflow_1.15-py3.6-horovod   8964680e-d5e4-5bb8-919b-8342c6c0dfd8  base
hybrid_0.1                      8c1a58c6-62b5-4dc4-987a-df751c2756b6  base
pytorch-onnx_1.3-py3.7          8d5d8a87-a912-54cf-81ec-3914adaa988d  base
caffe-ibm_1.0-py3.6             8d863266-7927-4d1e-97d7-56a7f4c0a19b  base
spss-modeler_17.1               902d0051-84bd-4af6-ab6b-8f6aa6fdeabb  base
do_12.10                        9100fd72-8159-4eb9-8a0b-a87e12eefa36  base
do_py3.7                        9447fa8b-2051-4d24-9eef-5acb0e3c59f8  base
spark-mllib_3.0-r_3.6           94bb6052-c837-589d-83f1-f4142f219e32  base
cuda-py3.7-opence               94e9652b-7f2d-59d5-ba5a-23a414ea488f  base
nlp-py3.8                       96e60351-99d4-5a1c-9cc0-473ac1b5a864  base
cuda-py3.7                      9a44990c-1aa1-4c7d-baf8-c4099011741c  base
hybrid_0.2                      9b3f9040-9cee-4ead-8d7a-780600f542f7  base
spark-mllib_3.0-py38            9f7a8fc1-4d3c-5e65-ab90-41fa8de2d418  base
```

```
autoai-kb_3.0-py3.6              d139f196-e04b-5d8b-9140-9a10ca1fa91a  base
spark-mllib_3.0-py36             d82546d5-dd78-5fbb-9131-2ec309bc56ed  base
autoai-kb_3.4-py3.8              da9b39c3-758c-5a4f-9cfd-457dd4d8c395  base
kernel-spark3.2-r3.6             db2fe4d6-d641-5d05-9972-73c654c60e0a  base
autoai-kb_rt22.1-py3.9           db6afe93-665f-5910-b117-d879897404d9  base
tensorflow_rt22.1-py3.9-horovod  dda170cc-ca67-5da7-9b7a-cf84c6987fae  base
autoai-ts_1.0-py3.7              deef04f0-0c42-5147-9711-89f9904299db  base
tensorflow_2.1-py3.7-horovod     e384fce5-fdd1-53f8-bc71-11326c9c635f  base
default_py3.7                    e4429883-c883-42b6-87a8-f419d64088cd  base
do_22.1                          e51999ba-6452-5f1f-8287-17228b88b652  base
autoai-obm_3.2                   eae86aab-da30-5229-a6a6-1d0d4e368983  base
tensorflow_rt22.2-py3.10         f65bd165-f057-55de-b5cb-f97cf2c0f393  base
do_20.1                          f686cdd9-7904-5f9d-a732-01b0d6b10dc5  base
pytorch-onnx_rt22.2-py3.10-edt   f8a05d07-e7cd-57bb-a10b-23f1d4b837ac  base
scikit-learn_0.19-py3.6          f963fa9d-4bb7-5652-9c5d-8d9289ef6ad9  base
tensorflow_2.4-py3.8             fe185c44-9a99-5425-986b-59bd1d2eda46  base
-------------------------------  ------------------------------------  ----
```

Save and Deploy the model

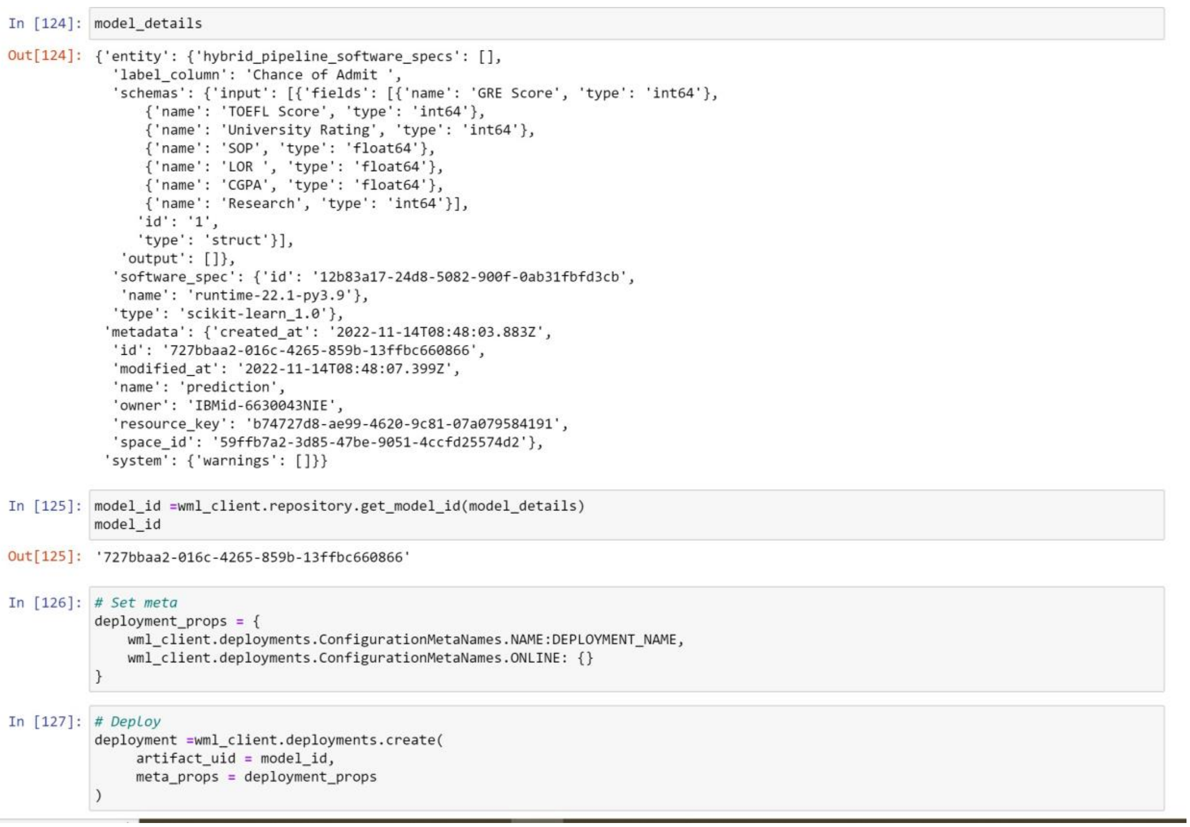
```
In [119]: import sklearn
          sklearn.__version__

Out[119]: '1.0.2'
```

```
In [120]: MODEL_NAME='prediction'
          DEPLOYMENT_NAME='University Admit Eligible Predictor deployment'
          DEMO_MODEL=lr
```

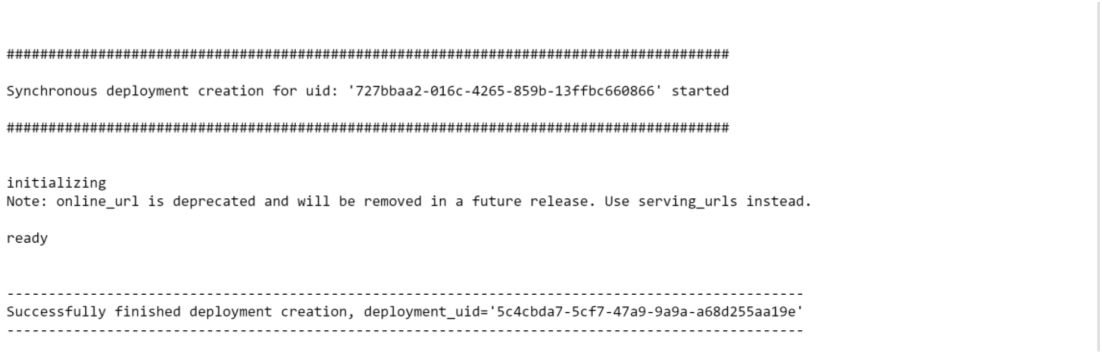
```
In [121]: # Set Python Version
          software_spec_uid=wml_client.software_specifications.get_id_by_name('runtime-22.1-py3.9')
```

```
In [122]: #Setup model meta
          model_props = {
              wml_client.repository.ModelMetaNames.NAME: MODEL_NAME,
              wml_client.repository.ModelMetaNames.TYPE: 'scikit-learn_1.0 ',
              wml_client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid
          }
```

```
In [123]: #Save model
          model_details =wml_client.repository.store_model(
              model = DEMO_MODEL,
              meta_props = model_props,
              training_data = X_Train,
              training_target = y_Train
          )
```



```
In [124]: model_details

Out[124]: {'entity': {'hybrid_pipeline_software_specs': [],
            'label_column': 'Chance of Admit ',
            'schemas': {'input': [{'fields': [{'name': 'GRE Score', 'type': 'int64'},
                {'name': 'TOEFL Score', 'type': 'int64'},
                {'name': 'University Rating', 'type': 'int64'},
                {'name': 'SOP', 'type': 'float64'},
                {'name': 'LOR ', 'type': 'float64'},
                {'name': 'CGPA', 'type': 'float64'},
                {'name': 'Research', 'type': 'int64'}],
               'id': '1',
               'type': 'struct'}],
             'output': []},
            'software_spec': {'id': '12b83a17-24d8-5082-900f-0ab31fbfd3cb',
             'name': 'runtime-22.1-py3.9'},
            'type': 'scikit-learn_1.0'},
           'metadata': {'created_at': '2022-11-14T08:48:03.883Z',
            'id': '727bbaa2-016c-4265-859b-13ffbc660866',
            'modified_at': '2022-11-14T08:48:07.399Z',
            'name': 'prediction',
            'owner': 'IBMid-6630043NIE',
            'resource_key': 'b74727d8-ae99-4620-9c81-07a079584191',
            'space_id': '59ffb7a2-3d85-47be-9051-4ccfd25574d2'},
           'system': {'warnings': []}}

In [125]: model_id =wml_client.repository.get_model_id(model_details)
          model_id

Out[125]: '727bbaa2-016c-4265-859b-13ffbc660866'

In [126]: # Set meta
          deployment_props = {
              wml_client.deployments.ConfigurationMetaNames.NAME:DEPLOYMENT_NAME,
              wml_client.deployments.ConfigurationMetaNames.ONLINE: {}
          }

In [127]: # Deploy
          deployment =wml_client.deployments.create(
               artifact_uid = model_id,
               meta_props = deployment_props
          )
```

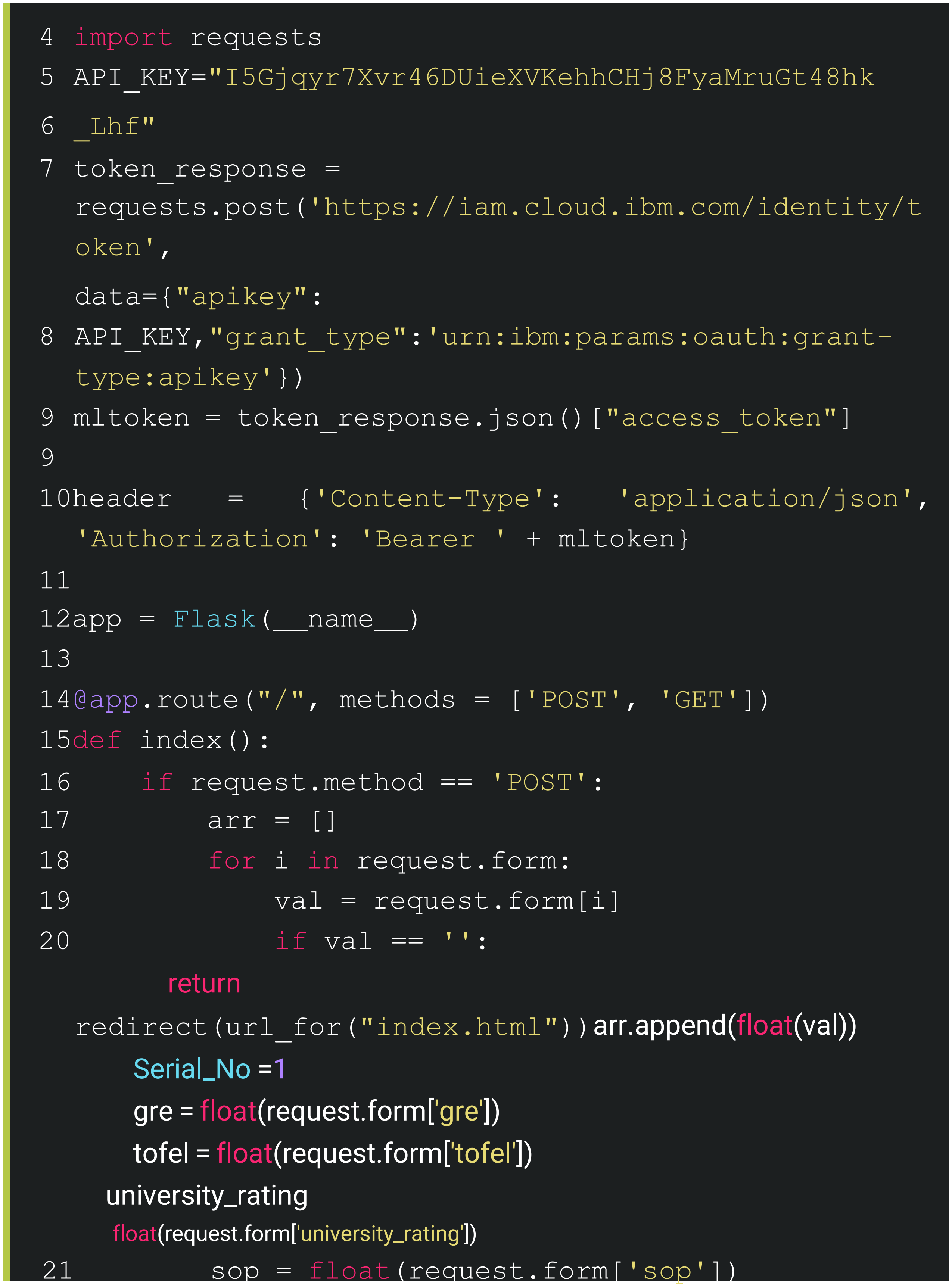


```
#######################################################################################

Synchronous deployment creation for uid: '727bbaa2-016c-4265-859b-13ffbc660866' started

#######################################################################################


initializing
Note: online_url is deprecated and will be removed in a future release. Use serving_urls instead.

ready


------------------------------------------------------------------------------------------------
Successfully finished deployment creation, deployment_uid='5c4cbda7-5cf7-47a9-9a9a-a68d255aa19e'
------------------------------------------------------------------------------------------------
```

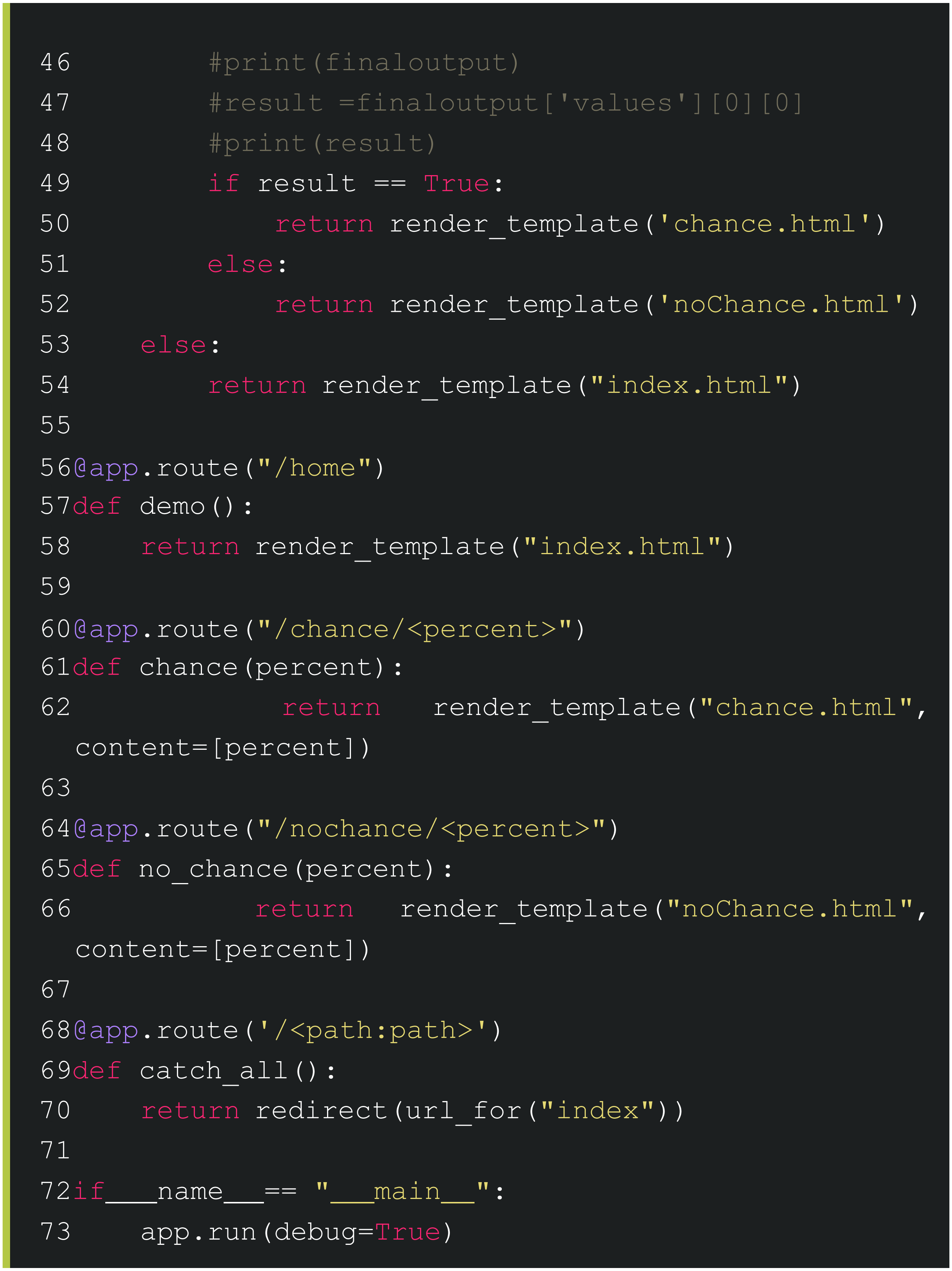
7.2 Feature 2

Developed a IBM_app.py file with integrated deployment and scoring points of IBM cloud.

```
from flask import Flask, render_template, redirect, url_for, request
import pickle
#import sklearn
```



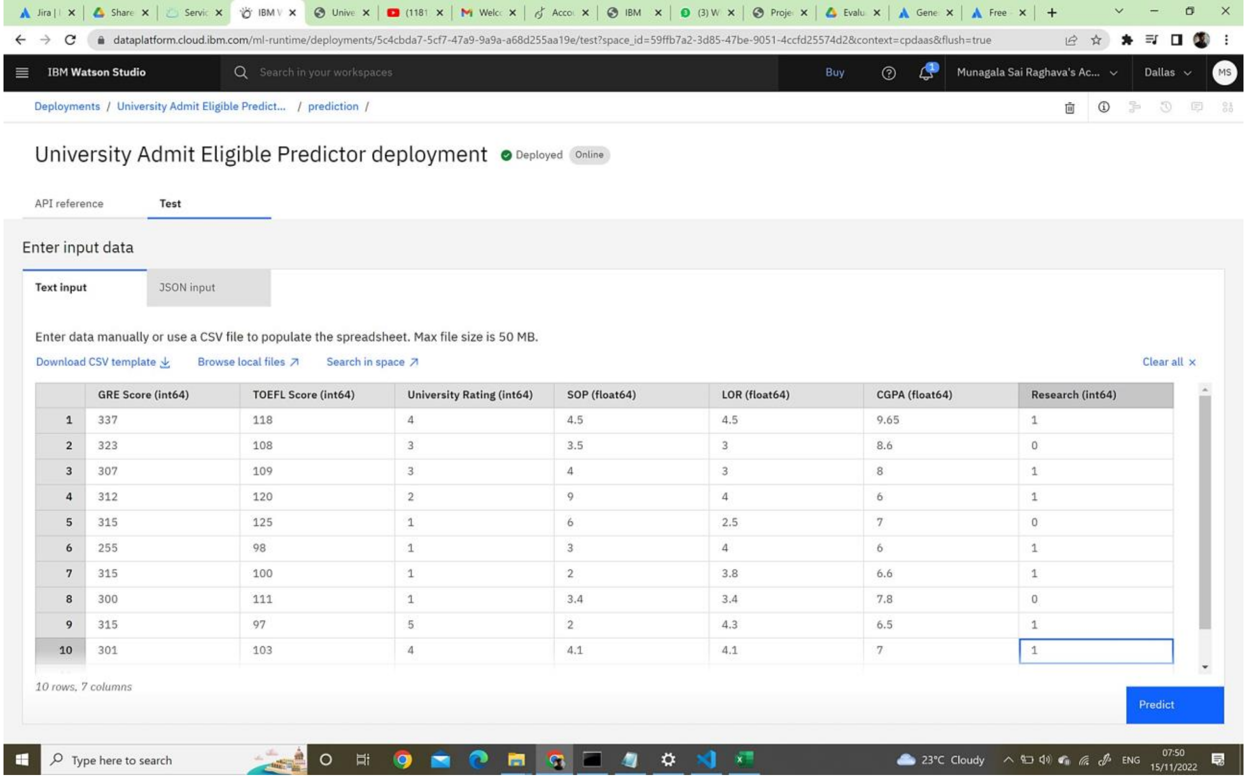
```
import requests
API_KEY="I5Gjqyr7Xvr46DUieXVKehhCHj8FyaMruGt48hk
_Lhf"
token_response =
 requests.post('https://iam.cloud.ibm.com/identity/t
 oken',
 data={"apikey":
API_KEY,"grant_type":'urn:ibm:params:oauth:grant-
 type:apikey'})
mltoken = token_response.json()["access_token"]

header   =   {'Content-Type':   'application/json',
 'Authorization': 'Bearer ' + mltoken}

app = Flask(__name__)

@app.route("/", methods = ['POST', 'GET'])
def index():
    if request.method == 'POST':
        arr = []
        for i in request.form:
            val = request.form[i]
            if val == '':
       return
 redirect(url_for("index.html"))arr.append(float(val))
     Serial_No =1
     gre = float(request.form['gre'])
     tofel = float(request.form['tofel'])
   university_rating
   float(request.form['university_rating'])
        sop = float(request.form['sop'])
        lor = float(request.form['lor'])
        cgpa = float(request.form['cgpa'])
        yes_no_radio                          =
  float(request.form['yes_no_radio'])
X=[[gre,tofel,university_rating,sop,lor,cgpa,yes_no_r
payload_scoring = {"input_data": [{"field":[ "GRE
  Score","TOEFL Score","University
  Rating","SOP","LOR","CGPA","Research"],
  "values": X}]}
  response_scoring                           =
  requests.post('https://us-
  south.ml.cloud.ibm.com/ml/v4/deployments/5c4cbda7-
  5cf7-47a9-9a9a-
  a68d255aa19e/predictions?version=2022-11-14',
  json=payload_scoring,
headers={'Authorization':  'Bearer  '  +mltoken})
        print(response_scoring)
        finaloutput=response_scoring.json()

 result=finaloutput['predictions'][0]['values'][0][O]
```



```
        #print(finaloutput)
        #result =finaloutput['values'][0][0]
        #print(result)
        if result == True:
            return render_template('chance.html')
        else:
            return render_template('noChance.html')
    else:
        return render_template("index.html")

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

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

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

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

if___name___== "___main__":
    app.run(debug=True)
```

8. TESTING

8.1 Test Cases


IBM Watson Studio
Search in your workspaces
Buy
Munagala Sai Raghava's Ac...
Dallas
MS
Deployments / University Admit Eligible Predict... / prediction /

University Admit Eligible Predictor deployment Deployed Online

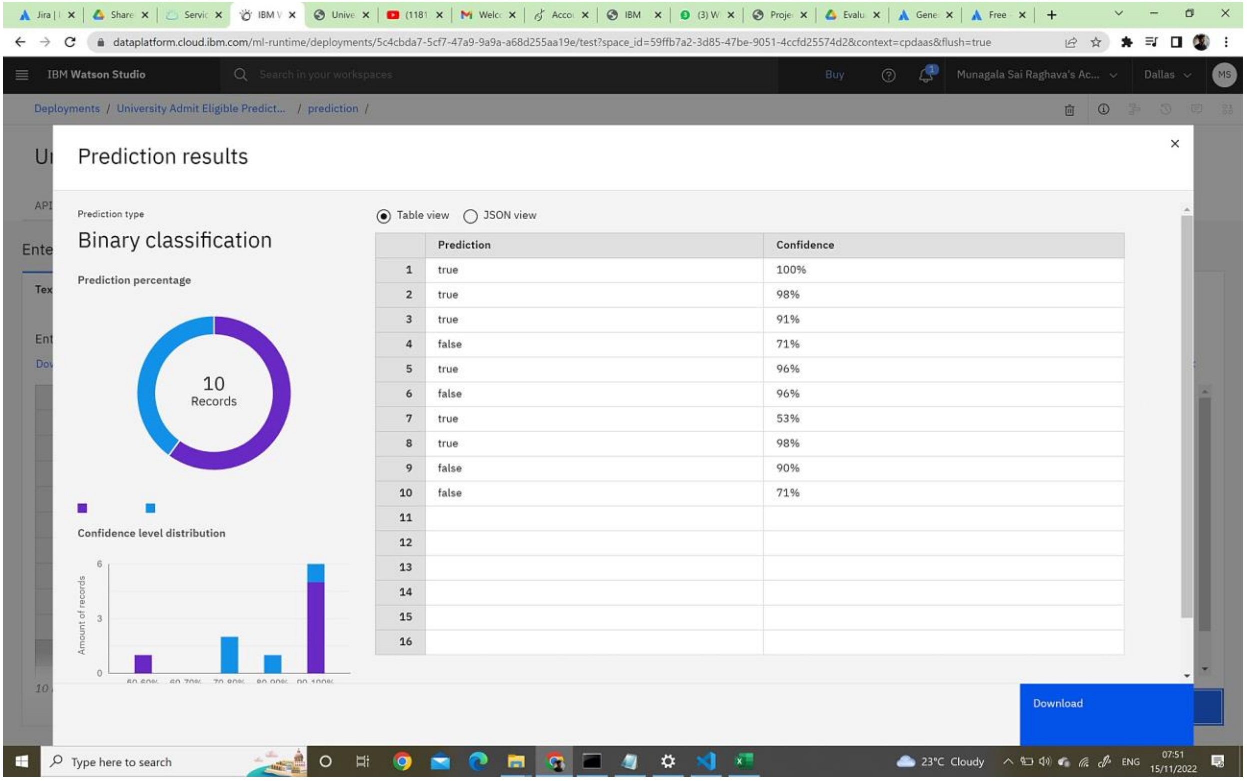
Enter input data

Enter data manually or use a CSV file to populate the spreadsheet. Max file size is 50 MB.

	GRE Score (int64)	TOEFL Score (int64)	University Rating (int64)	SOP (float64)	LOR (float64)	CGPA (float64)	Research (int64)
1	337	118	4	4.5	4.5	9.65	1
2	323	108	3	3.5	3	8.6	0
3	307	109	3	4	3	8	1
4	312	120	2	9	4	6	1
5	315	125	1	6	2.5	7	0
6	255	98	1	3	4	6	1
7	315	100	1	2	3.8	6.6	1
8	300	111	1	3.4	3.4	7.8	0
9	315	97	5	2	4.3	6.5	1
10	301	103	4	4.1	4.1	7	1

10 rows, 7 columns

If the student is eligible for the university, it will give output as True. Otherwise, the output will be False.



dataplatform.cloud.ibm.com/ml-runtime/deployments/5c4cbda7-5cf7-47a9-9a9a-a68d255aa19e/test?space_id=59ffb7a2-3d85-47be-9051-4ccfd25574d2&context=cpdaas&flush=true

Deployments / University Admit Eligible Predict... / prediction /

Prediction results

Prediction type

Binary classification

Prediction percentage

10 Records

Confidence level distribution

● Table view ○ JSON view

	Prediction	Confidence
1	true	100%
2	true	98%
3	true	91%
4	false	71%
5	true	96%
6	false	96%
7	true	53%
8	true	98%
9	false	90%
10	false	71%
11		
12		
13		
14		
15		
16		

8.2 User Acceptance Testing

acceptance TESTING

UAT Execution & Report and Submission

Purpose of Documentation:

The purpose of this documentation proved information, to giveinstructions, to persude the reader, and to enact something.

1. Defect Analysis

Pre solution	Severty1	Severty2	Severty3	Severty4	Severty5
By Design	10	4	2	2	20
Duplicates	1	0	3	0	4
External	2	3	0	1	6

Fixed	11	2	4	20	37
Not reported	0	0	1	0	1
Skipped	0	0	1	1	2
Wont FLs	0	5	2	1	7
total	24	14	13	26	77

2. Testcase Analysis

section	testcases	Not tables	fall	pass
Print engine	7	0	0	7
Client application	31	0	0	81
security	2	0	0	2
Customer shipping	3	0	0	3
Exception Handling	9	0	0	9
Final report output	4	0	0	4
Version control	2	0	0	2

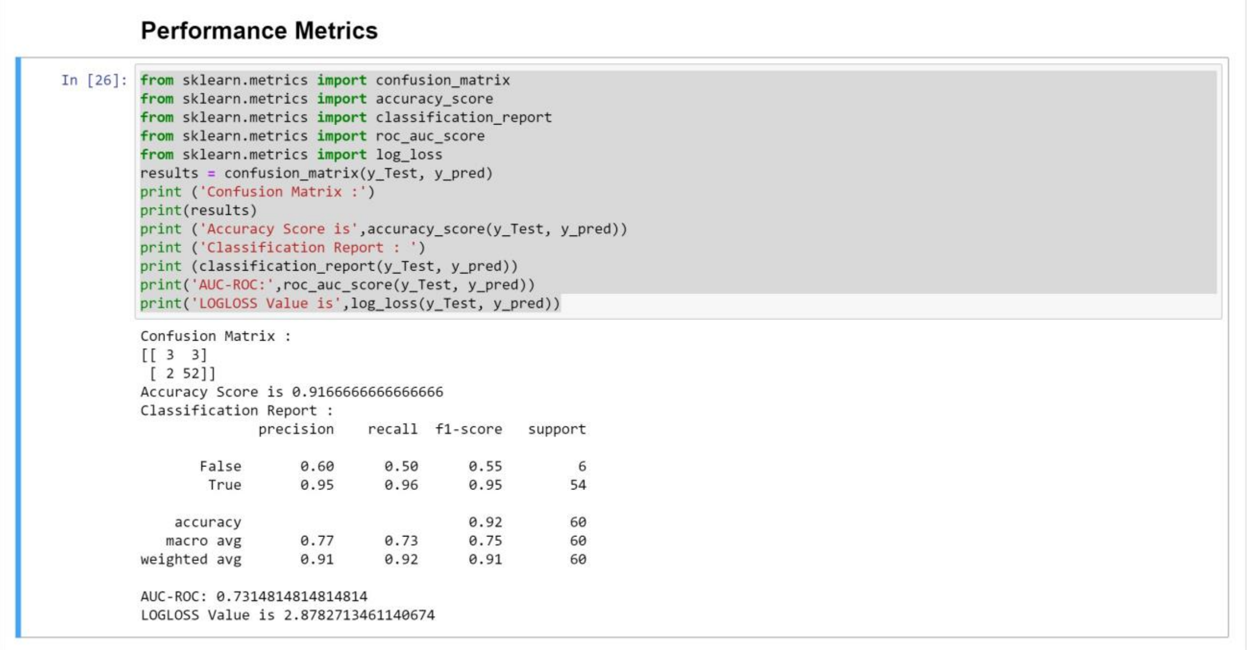
9. RESULTS

9.1 Performance Metrics

There are various metrics which we can use to evaluate the performance of ML algorithms, classification as well as regression algorithms. We must carefully choose the metrics for evaluating ML performance because –

- How the performance of ML algorithms is measured and compared will be dependent entirely on the metric you choose.
- How you weight the importance of various characteristics in the result will be influenced completely by the metric you choose.

```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report
from sklearn.metrics import roc_auc_score
from sklearn.metrics import log_loss
results = confusion_matrix(y_Test, y_pred)
print ('Confusion Matrix :')
print(results)
print  ('Accuracy  Score  is',accuracy_score(y_Test, y_pred))
print ('Classification Report : ')
print (classification_report(y_Test, y_pred))
print('AUC-ROC:',roc_auc_score(y_Test, y_pred))
print('LOGLOSS Value is',log_loss(y_Test, y_pred))
```



Performance Metrics

```
In [26]: from sklearn.metrics import confusion_matrix
         from sklearn.metrics import accuracy_score
         from sklearn.metrics import classification_report
         from sklearn.metrics import roc_auc_score
         from sklearn.metrics import log_loss
         results = confusion_matrix(y_Test, y_pred)
         print ('Confusion Matrix :')
         print(results)
         print ('Accuracy Score is',accuracy_score(y_Test, y_pred))
         print ('Classification Report : ')
         print (classification_report(y_Test, y_pred))
         print('AUC-ROC:',roc_auc_score(y_Test, y_pred))
         print('LOGLOSS Value is',log_loss(y_Test, y_pred))
```

```
Confusion Matrix :
[[ 3  3]
 [ 2 52]]
Accuracy Score is 0.9166666666666666
Classification Report :
              precision    recall  f1-score   support

       False       0.60      0.50      0.55         6
        True       0.95      0.96      0.95        54

    accuracy                           0.92        60
   macro avg       0.77      0.73      0.75        60
weighted avg       0.91      0.92      0.91        60

AUC-ROC: 0.7314814814814814
LOGLOSS Value is 2.8782713461140674
```

10. ADVANTAGES & DISADVANTAGESADVANTAGES

- Here the chance of occurrence of error is less when compared withthe existing system.
- It is fast, efficient and reliable.
- Avoids data redundancy and inconsistency.
- Very user-friendly.
- Easy accessibility of data.

DISADVANTAGES

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

11. CONCLUSION

The subject of this examination was to determine if the below variables contribute to the admission of student to Master's degree program.

GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA

The results of this examination appear to indicate that it greatly contributes to the response variable 'Chance of Admit'. Higher the GRE, TOEFL score then higher the admit chances. The model predicts 91.5% accuracy and can be used for predicting the admit chances based on the above factors. This model will be helpful for the universities to predict the admission and ease their process of selection and timelines. As part of the hypothesis, the model proved that admission to Master's degree program is dependent on GRE, TOEFL and other scores. This model would likely be greatly improved by the gathering of additional data of students from different universities which has similar selection criteria to choose the candidates for Master's program.

FUTURE SCOPE

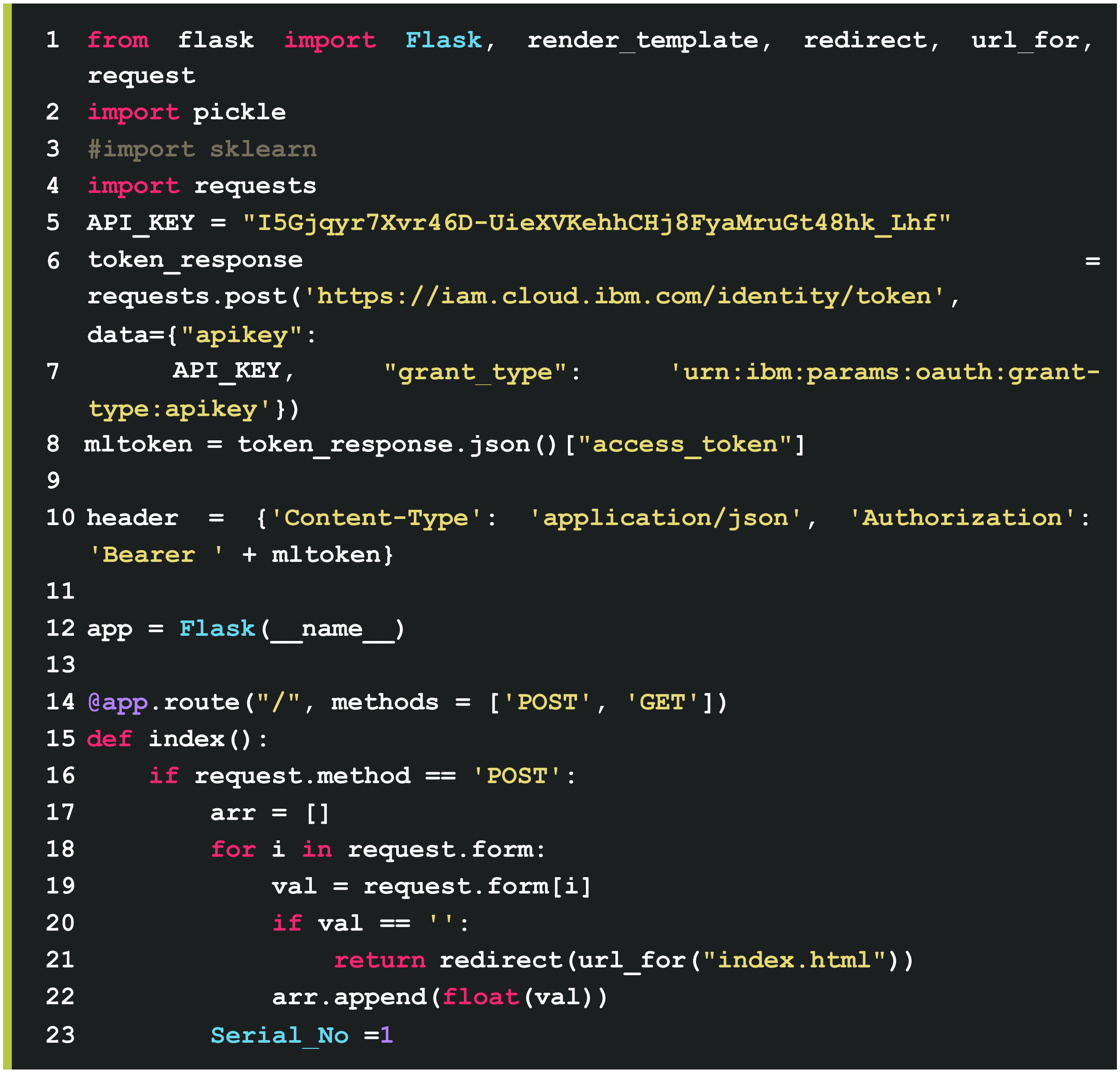
The future scope of this project is very broad.Few
of them are:

- 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.

12. APPENDIX

Source Code GitHub & Project Demo Link

Source Code for Flask Application



```
from flask import Flask, render_template, redirect, url_for, request
import pickle
#import sklearn
import requests
API_KEY = "I5Gjqyr7Xvr46D-UieXVKehhCHj8FyaMruGt48hk_Lhf"
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("/", methods = ['POST', 'GET'])
def index():
    if request.method == 'POST':
        arr = []
        for i in request.form:
            val = request.form[i]
            if val == '':
                return redirect(url_for("index.html"))
            arr.append(float(val))
        Serial_No =1
```

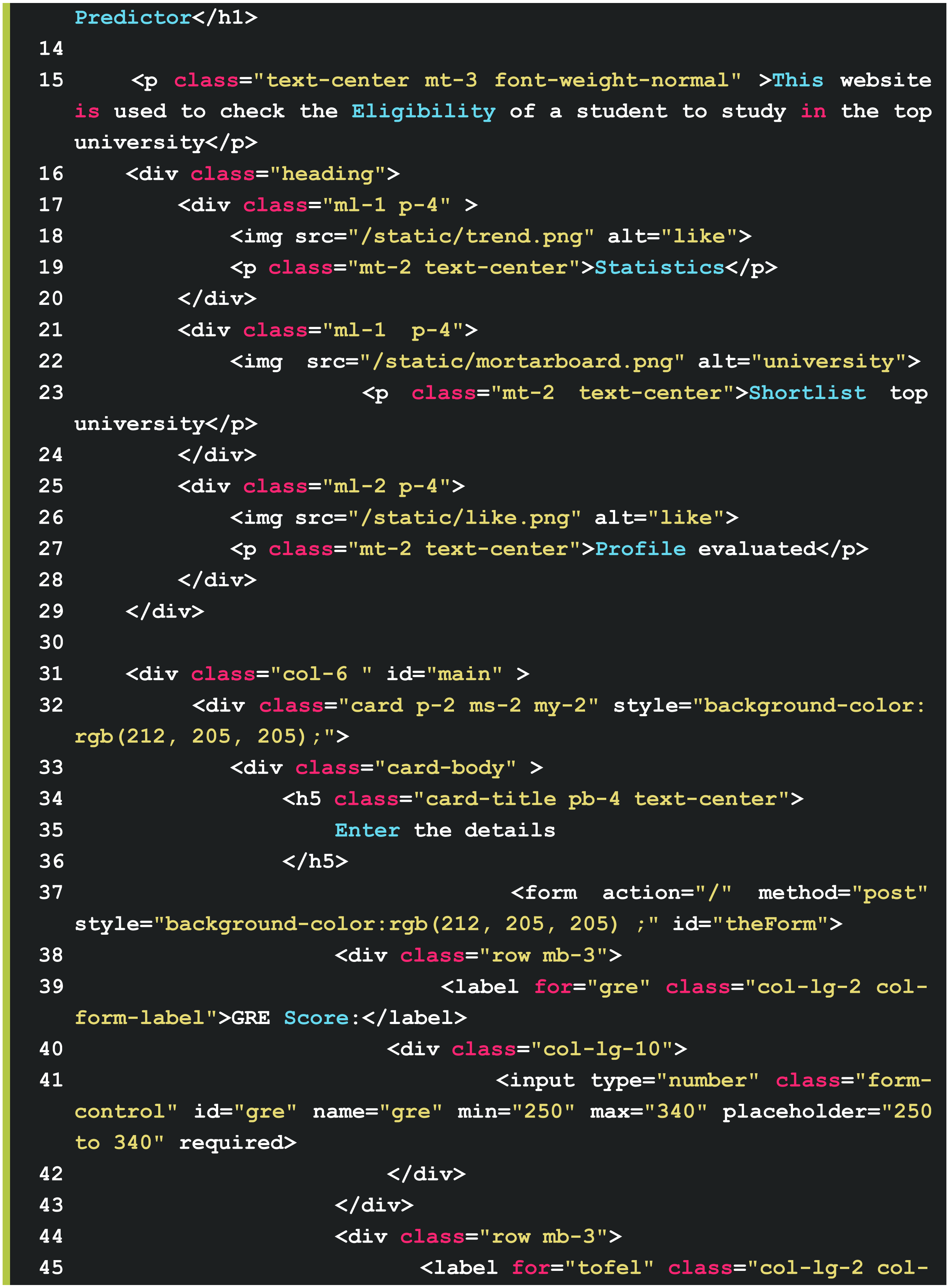
```python
        gre = float(request.form['gre'])
        tofel = float(request.form['tofel'])
        university_rating = float(request.form['university_rating'])
        sop = float(request.form['sop'])
        lor = float(request.form['lor'])
        cgpa = float(request.form['cgpa'])
        yes_no_radio = float(request.form['yes_no_radio'])

        X =[[gre,tofel,university_rating,sop,lor,cgpa,yes_no_radio]]

        payload_scoring = {"input_data": [{"field": [ "GRE Score",
                                "TOEFL Score",
                                "University Rating",
                                "SOP",
                                "LOR ",
                                "CGPA",
                                "Research"], "values": X}]}
        response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/5c4cbda7-5cf7-47a9-9a9a-a68d255aa19e/predictions?version=2022-11-14', json=payload_scoring,
        headers={'Authorization': 'Bearer ' + mltoken})
        print(response_scoring)
        finaloutput=response_scoring.json()
        result=finaloutput['predictions'][0]['values'][0][0]
        if result == True:
            return render_template('chance.html')
        else:
            return render_template('noChance.html')
    else:
        return render_template("index.html")

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

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

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

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

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

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

```

Front End Code HTML

Files1.Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
     <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../static/style.css">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" crossorigin="anonymous">

    <title>University Eligibility Predictor</title>
</head>
<body>
          <h1 class="text-center mt-3">University Eligibility
```



```
Predictor</h1>

  <p class="text-center mt-3 font-weight-normal" >This website is used to check the Eligibility of a student to study in the top university</p>
  <div class="heading">
    <div class="ml-1 p-4" >
      <img src="/static/trend.png" alt="like">
      <p class="mt-2 text-center">Statistics</p>
    </div>
    <div class="ml-1  p-4">
      <img  src="/static/mortarboard.png" alt="university">
          <p class="mt-2 text-center">Shortlist top university</p>
    </div>
    <div class="ml-2 p-4">
      <img src="/static/like.png" alt="like">
      <p class="mt-2 text-center">Profile evaluated</p>
    </div>
  </div>

  <div class="col-6 " id="main" >
     <div class="card p-2 ms-2 my-2" style="background-color: rgb(212, 205, 205);">
      <div class="card-body" >
        <h5 class="card-title pb-4 text-center">
          Enter the details
        </h5>
          <form action="/" method="post" style="background-color:rgb(212, 205, 205) ;" id="theForm">
          <div class="row mb-3">
              <label for="gre" class="col-lg-2 col-form-label">GRE Score:</label>
            <div class="col-lg-10">
                <input type="number" class="form-control" id="gre" name="gre" min="250" max="340" placeholder="250 to 340" required>
            </div>
          </div>
          <div class="row mb-3">
             <label for="tofel" class="col-lg-2 col-
```

```
form-label">TOFEL Score:</label>
                        <div class="col-lg-10">
                            <input type="number" class="form-control" id="tofel" name="tofel" min="50" max="120" placeholder="50 to 120" required>
                        </div>
                    </div>
                    <div class="row mb-3">
                        <label for="university_rating" class="col-lg-2 col-form-label">University Rating:</label>
                        <div class="col-lg-10">
                            <input type="number" class="form- control" id="university_rating" step="0.01" name="university_rating" min="1" max="5" placeholder="1 to 5" required>
                        </div>
                    </div>
                    <div class="row mb-3">
                        <label for="sop" class="col-lg-2 col-form-label">SOP:</label>
                        <div class="col-lg-10">
                            <input type="number" class="form-control" id="sop" name="sop" step="0.01" min="1" max="5" placeholder="1 to 5" required>
                        </div>
                    </div>
                    <div class="row mb-3">
                        <label for="lor" class="col-lg-2 col-form-label">LOR:</label>
                        <div class="col-lg-10">
                            <input type="number" class="form-control" id="lor" name="lor" step="0.01" min="1" max="5" placeholder="1 to 5" required>
                        </div>
                    </div>
                    <div class="row mb-3">
                        <label for="cgpa" class="col-lg-2 col-form-label">CGPA:</label>
                        <div class="col-lg-10">
                            <input type="number" class="form-control" id="cgpa" name="cgpa" step="0.01" min="5" max="10"
```

```html
placeholder="5 to 10" required>
                      </div>
                  </div>
                  <fieldset class="row mb-3">
                         <legend class="col-form-label col-sm-2 pt-0">Research:</legend>
                      <div class="col-sm-10">
                          <div class="form-check">
                                <input class="form-check-input" type="radio" name="yes_no_radio" id="gridRadios1" value="1">
                                <label class="form-check-label" for="yes_no_radio">
                                  Yes
                                  </label>
                          </div>
                          <div class="form-check">
                                <input class="form-check-input" type="radio" name="yes_no_radio" id="gridRadios2" value="0" checked>
                                <label class="form-check-label" for="yes_no_radio">
                                  No
                                  </label>
                          </div>
                      </div>
                  </fieldset>

                  <div class="row lg-3 justify-content-center">
                      <div class="col-lg-2 mb-2 me-3">
                 <button type="submit" class="btn btn- primary m-auto" style="padding:8px 25px; " id="button">Predict</button>
                      </div>

              </form>
              </div>
          </div>
        </div>

</body>
```

```
</html>
```

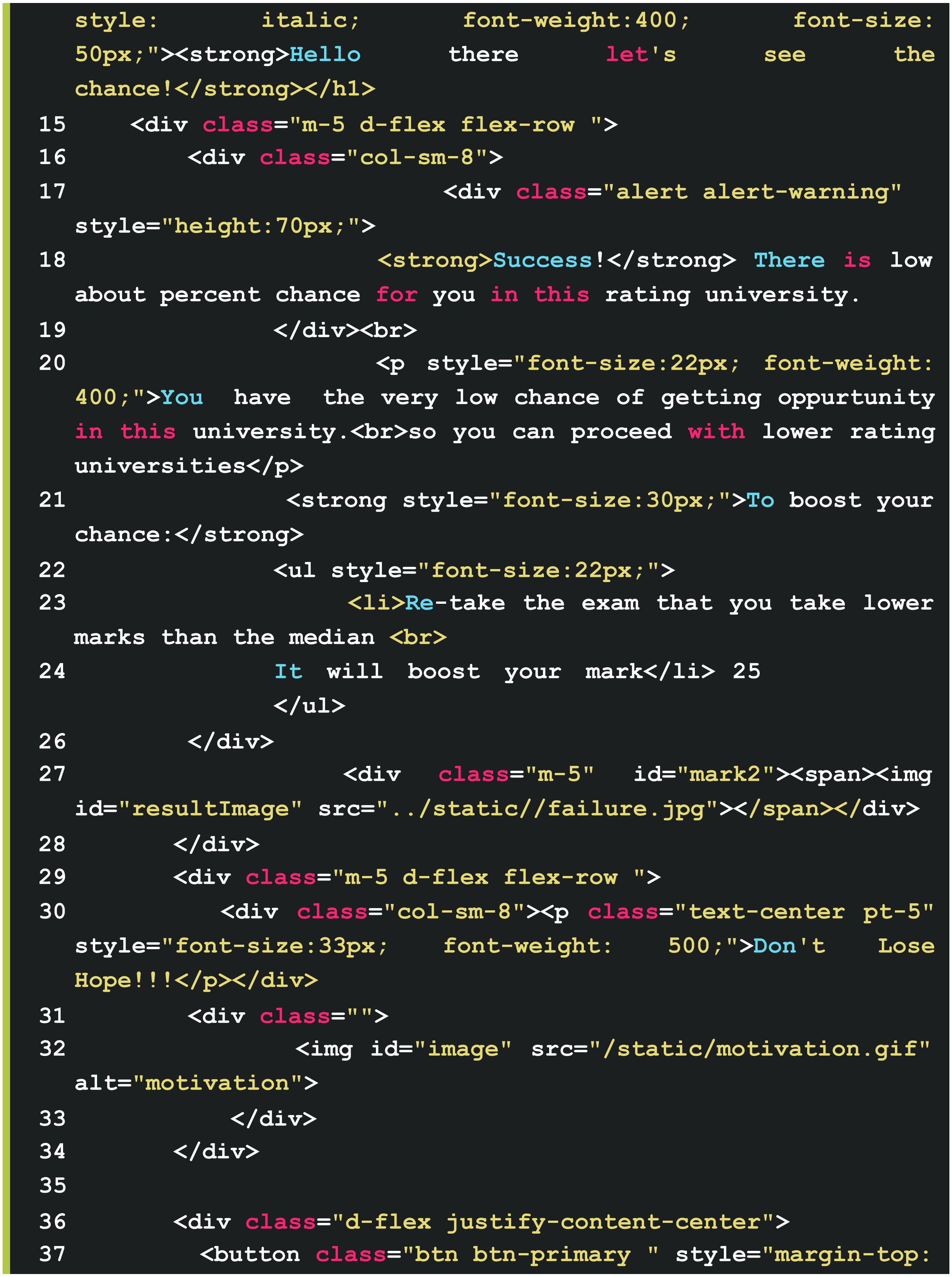
2.chance.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../static/style.css">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" crossorigin="anonymous">

    <title>Document</title>
</head>
<body>

     <h1 class="text-center" id="title" ><strong>Hello there let's see the chance!</strong></h1>
    <div class="m-5 d-flex flex-row ">
        <div class="col-sm-8">
                <div class="alert alert-success" style="height:70px;">
                    <strong>Success!</strong> There is high about percent chance for you in this rating university.
               </div>
                 <p style="font-size:22px; font-weight: 500;">You have the high chance of getting oppurtunity in this university.<ul style="font-size:22px;"><li> You can apply for this rating universities</li><li>OR You can try higher rating universities</li></ul></p>
        </div>
          <div class="m-5" id="mark1"><span><img id="resultImage" src="../static/green thumb.jpg"></span></div>
       </div>
```

```
        <div class="m-5 d-flex flex-row ">
                <div class="col-sm-8"><p class="text-center pt-5" style="font-size:33px; font-weight: 500;">Good Luck!!!</p></div>
         <div class="">
                        <img id="image"src="/static/success1.gif" alt="success">
            </div>
        </div>

        <div class="d-flex justify-content-center">
            <button class="btn btn-primary " style="margin-top: 0;"><a href="/" style="color:white; text-decoration : none" >Go Back</a></button>
        </div>
</body>
</html>
```

3.NoChance.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../static/style.css">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" crossorigin="anonymous">

    <title>Document</title>
</head>
<body>

      <h1 class="text-center" style="margin-top:25px; font-
```



```
style: italic; font-weight:400; font-size: 50px;"><strong>Hello there let's see the chance!</strong></h1>
    <div class="m-5 d-flex flex-row ">
        <div class="col-sm-8">
                                <div class="alert alert-warning" style="height:70px;">
                          <strong>Success!</strong> There is low about percent chance for you in this rating university.
                 </div><br>
                          <p style="font-size:22px; font-weight: 400;">You have the very low chance of getting oppurtunity in this university.<br>so you can proceed with lower rating universities</p>
                  <strong style="font-size:30px;">To boost your chance:</strong>
                 <ul style="font-size:22px;">
                      <li>Re-take the exam that you take lower marks than the median <br>
                 It will boost your mark</li>
                 </ul>
        </div>
                      <div class="m-5" id="mark2"><span><img id="resultImage" src="../static//failure.jpg"></span></div>
       </div>
       <div class="m-5 d-flex flex-row ">
           <div class="col-sm-8"><p class="text-center pt-5" style="font-size:33px; font-weight: 500;">Don't Lose Hope!!!</p></div>
        <div class="">
                <img id="image" src="/static/motivation.gif" alt="motivation">
           </div>
       </div>

       <div class="d-flex justify-content-center">
         <button class="btn btn-primary " style="margin-top:
```

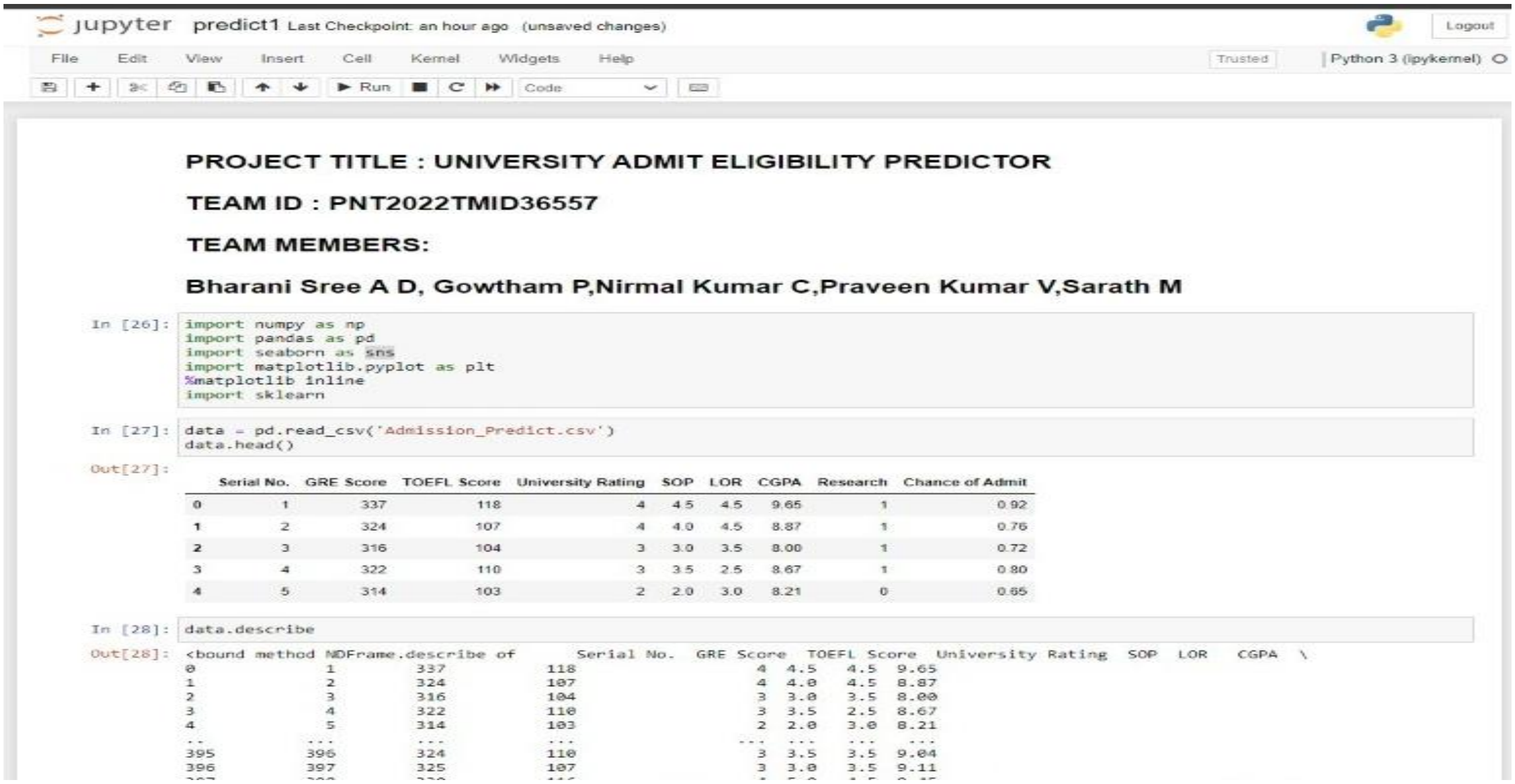
```
0;"><a href="/" style="color:white; text-decoration : none"
>Go Back</a></button>
      </div>
</body>
</html>
```

4.style.css

```
.heading{
    display:flex;
     justify-content: space-around;
      margin-top: 45px;
}
#resultImage{
    height: 280px;
    width: 280px;
}
#title{
    margin-top:50px;
    font-style: italic;
    font-weight:400;
    font-size: 50px;
}
#image{
    height: 335px;
    width: 335px;
}
img{
    width:100px;
    height: 100px;
}

```

```
#main{
    margin:auto;
}

#mark1{
    font-size:110px;
    color: rgb(62, 180, 62)
}
#mark2{
    font-size:110px;
    color: red;
}
```

5. Jupiter Note Book



PROJECT TITLE : UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

TEAM ID : PNT2022TMID36557

TEAM MEMBERS:

Bharani Sree A D, Gowtham P,Nirmal Kumar C,Praveen Kumar V,Sarath M

```
In [26]: import numpy as np
         import pandas as pd
         import seaborn as sns
         import matplotlib.pyplot as plt
         %matplotlib inline
         import sklearn
```

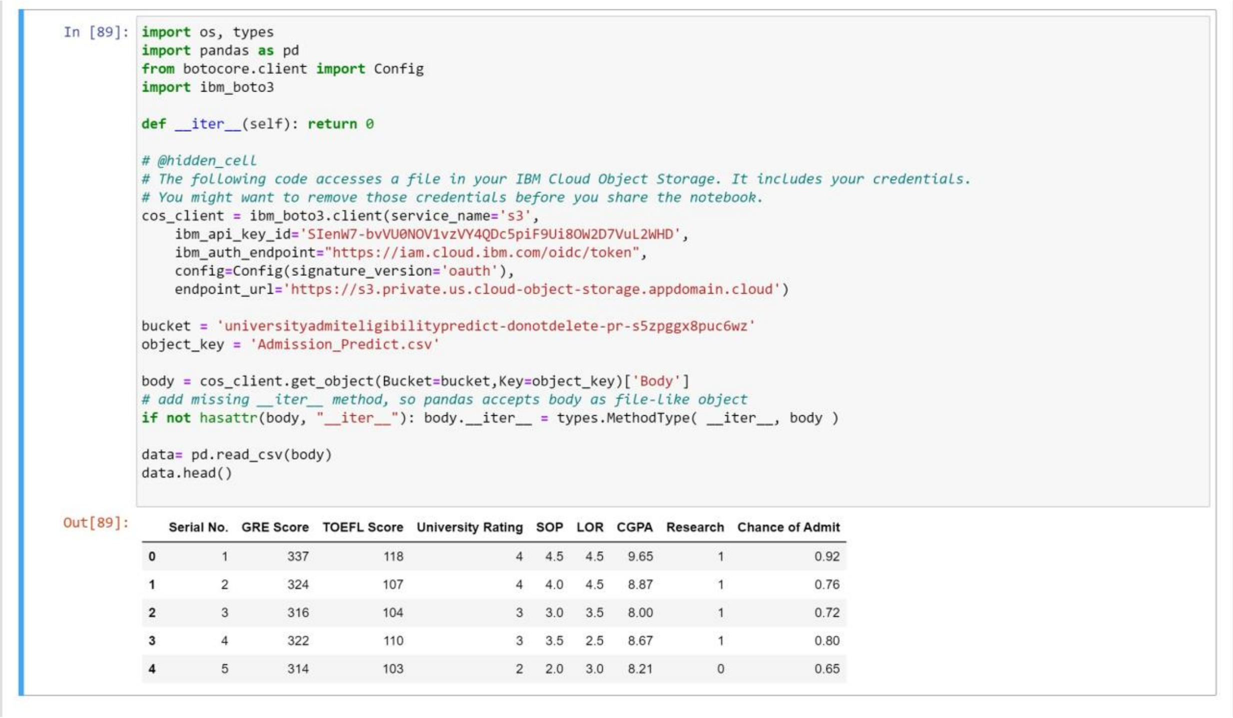
```
In [27]: data = pd.read_csv('Admission_Predict.csv')
         data.head()
```

Out[27]:

	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 [28]: data.describe
```

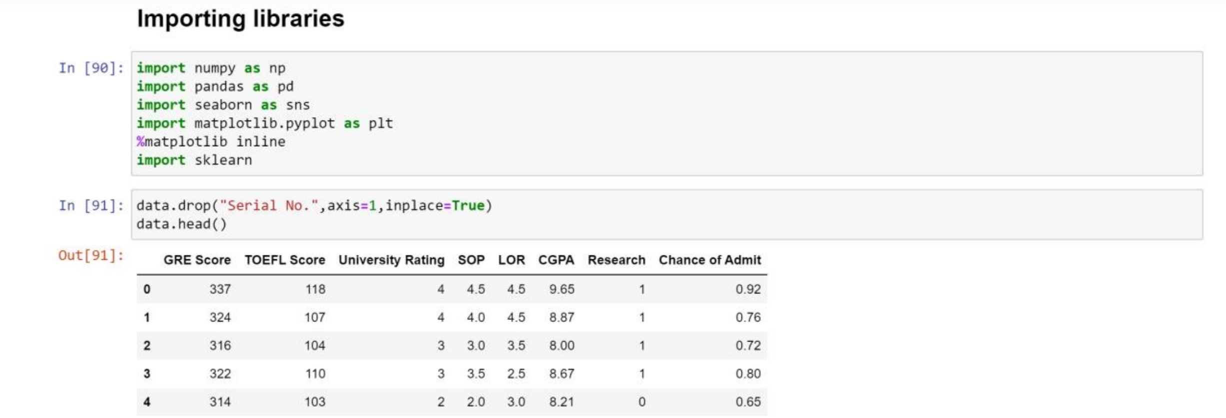
```
Out[28]: <bound method NDFrame.describe of      Serial No.  GRE Score  TOEFL Score  University Rating  SOP  LOR   CGPA  \
         0             1        337          118                  4  4.5   4.5  9.65
         1             2        324          107                  4  4.0   4.5  8.87
         2             3        316          104                  3  3.0   3.5  8.00
         3             4        322          110                  3  3.5   2.5  8.67
         4             5        314          103                  2  2.0   3.0  8.21
         ..          ...        ...          ...                ...  ...   ...   ...
         395         396        324          110                  3  3.5   3.5  9.04
         396         397        325          107                  3  3.0   3.5  9.11
```



In [89]:
```
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='SIenW7-bvVU0NOV1vzVY4QDc5piF9Ui8OW2D7VuL2WHD',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'universityadmiteligibilitypredict-donotdelete-pr-s5zpggx8puc6wz'
object_key = 'Admission_Predict.csv'

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

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

Out[89]:

	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



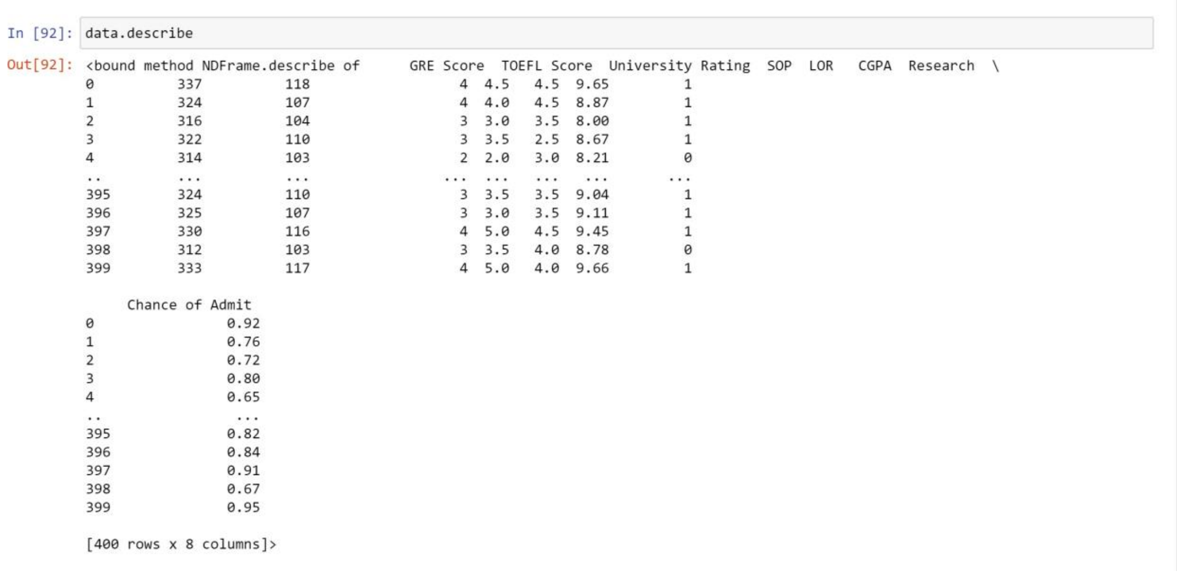
Importing libraries

In [90]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import sklearn
```

In [91]:
```
data.drop("Serial No.",axis=1,inplace=True)
data.head()
```

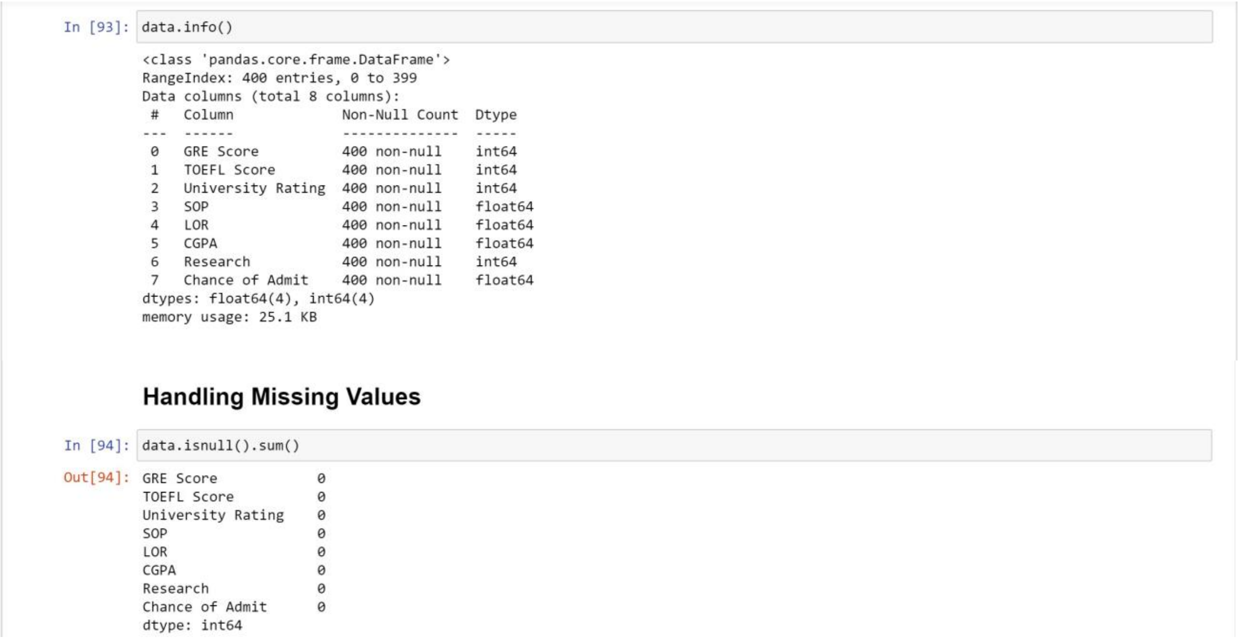
Out[91]:

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



```
In [92]: data.describe
```

```
Out[92]: <bound method NDFrame.describe of      GRE Score  TOEFL Score  University Rating  SOP  LOR   CGPA  Research  \
         0          337          118                  4  4.5   4.5  9.65         1   
         1          324          107                  4  4.0   4.5  8.87         1   
         2          316          104                  3  3.0   3.5  8.00         1   
         3          322          110                  3  3.5   2.5  8.67         1   
         4          314          103                  2  2.0   3.0  8.21         0   
         ..         ...          ...                ...  ...   ...   ...       ...   
         395        324          110                  3  3.5   3.5  9.04         1   
         396        325          107                  3  3.0   3.5  9.11         1   
         397        330          116                  4  5.0   4.5  9.45         1   
         398        312          103                  3  3.5   4.0  8.78         0   
         399        333          117                  4  5.0   4.0  9.66         1   

              Chance of Admit   
         0               0.92  
         1               0.76  
         2               0.72  
         3               0.80  
         4               0.65  
         ..               ...  
         395             0.82  
         396             0.84  
         397             0.91  
         398             0.67  
         399             0.95  

         [400 rows x 8 columns]>
```



```
In [93]: data.info()
```

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

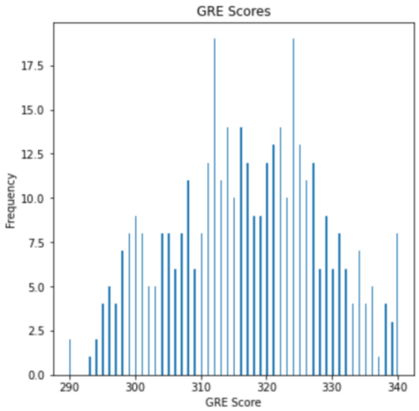
Handling Missing Values

```
In [94]: data.isnull().sum()
```

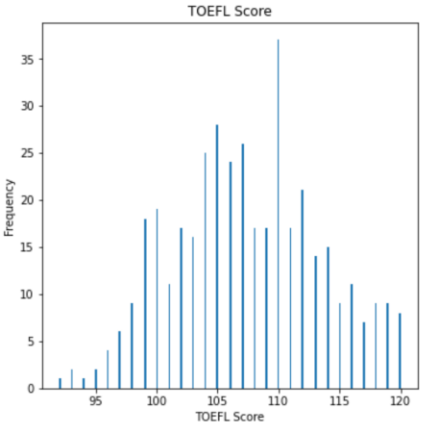
```
Out[94]: GRE Score            0
         TOEFL Score          0
         University Rating    0
         SOP                  0
         LOR                  0
         CGPA                 0
         Research             0
         Chance of Admit      0
         dtype: int64
```

Data Visualization

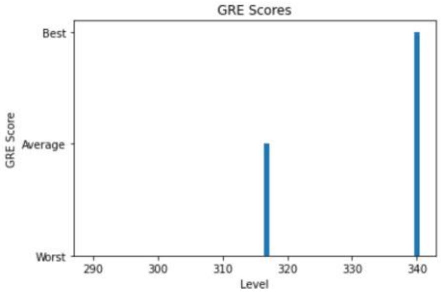
```
In [95]: data["GRE Score"].plot(kind = 'hist',bins = 200,figsize = (6,6))
         plt.title("GRE Scores")
         plt.xlabel("GRE Score")
         plt.ylabel("Frequency")
         plt.show()
```



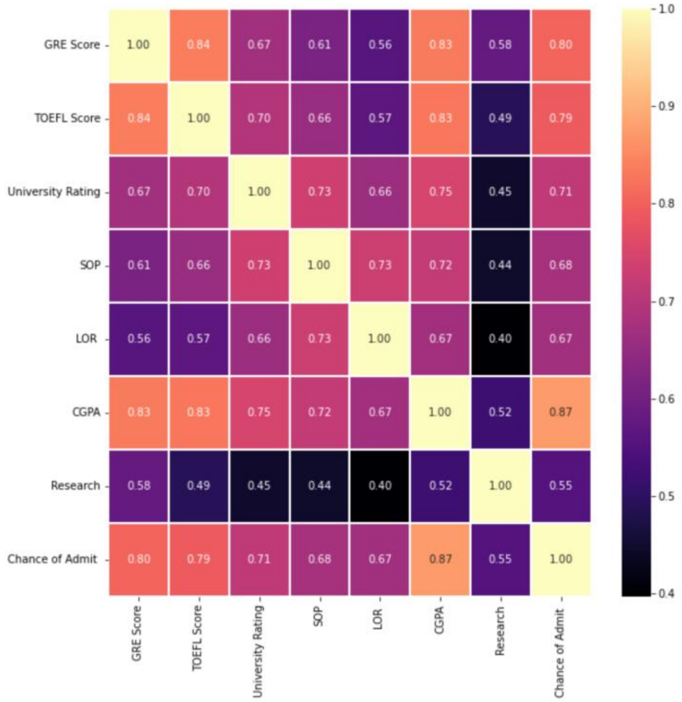
```
In [96]: data["TOEFL Score"].plot(kind = 'hist',bins = 200,figsize = (6,6))
         plt.title("TOEFL Score")
         plt.xlabel("TOEFL Score")
         plt.ylabel("Frequency")
         plt.show()
```



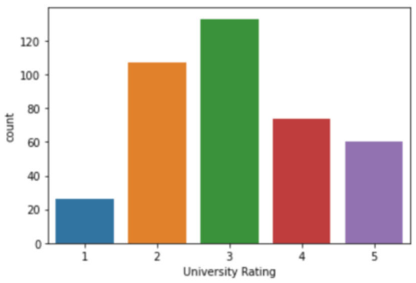
```
In [97]: g = np.array([data["GRE Score"].min(),data["GRE Score"].mean(),data["GRE Score"].max()])
         h = ["Worst","Average","Best"]
         plt.bar(g,h)
         plt.title("GRE Scores")
         plt.xlabel("Level")
         plt.ylabel("GRE Score")
         plt.show()
```



```
In [98]: plt.figure(figsize=(10, 10))
         sns.heatmap(data.corr(), annot=True, linewidths=0.05, fmt= '.2f',cmap="magma")
         plt.show()
```



```
In [99]: data.Research.value_counts()
         sns.countplot(x="University Rating",data=data)

Out[99]: <AxesSubplot:xlabel='University Rating', ylabel='count'>
```



Training and Testing Split

```
In [100]: X=data.drop(['Chance of Admit '],axis=1)
          y=data['Chance of Admit ']
```

```
In [101]: X
```

Out[101]:

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research
0	337	118	4	4.5	4.5	9.65	1
1	324	107	4	4.0	4.5	8.87	1
2	316	104	3	3.0	3.5	8.00	1
3	322	110	3	3.5	2.5	8.67	1
4	314	103	2	2.0	3.0	8.21	0
...	...	...	...	...	...	...	...
395	324	110	3	3.5	3.5	9.04	1
396	325	107	3	3.0	3.5	9.11	1
397	330	116	4	5.0	4.5	9.45	1
398	312	103	3	3.5	4.0	8.78	0
399	333	117	4	5.0	4.0	9.66	1

400 rows × 7 columns

```
In [102]: y
```

```
Out[102]: 0      0.92
          1      0.76
          2      0.72
          3      0.80
          4      0.65
                 ...
          395    0.82
          396    0.84
          397    0.91
          398    0.67
          399    0.95
          Name: Chance of Admit , Length: 400, dtype: float64
```

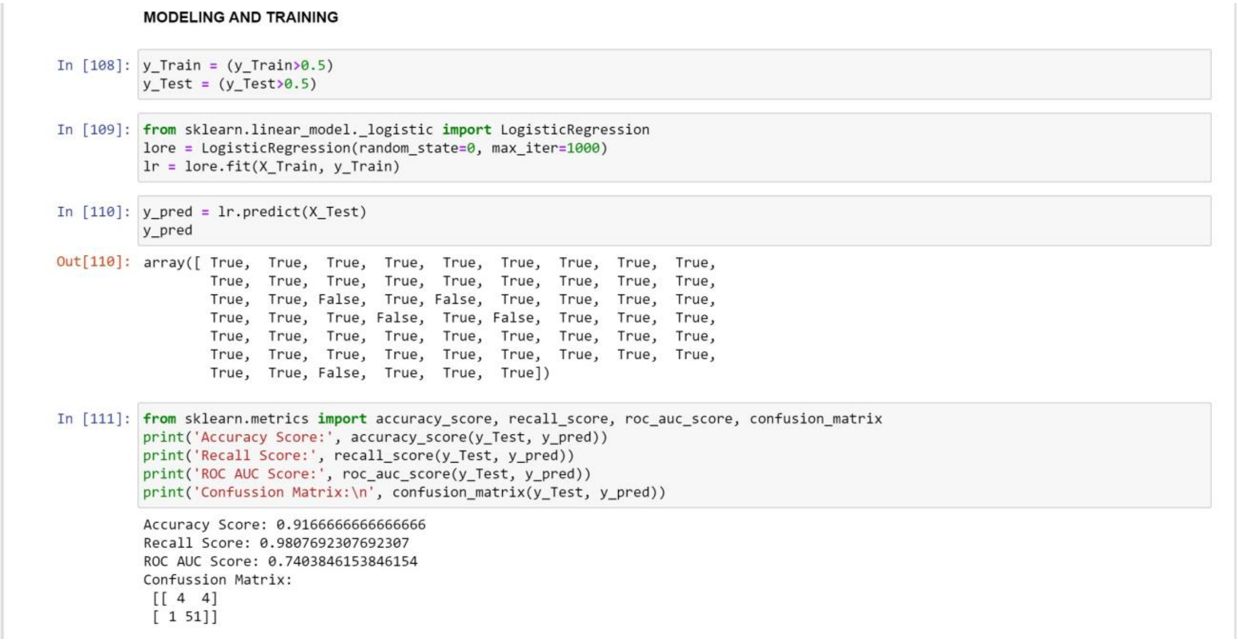
```
In [103]: from sklearn.model_selection import train_test_split
          X_Train, X_Test, y_Train, y_Test = train_test_split(X, y, test_size=0.15)
```

```
In [104]: X_Train.shape

Out[104]: (340, 7)
```

```
In [105]: y_Train.shape

Out[105]: (340,)
```

```
In [106]: y_Test.shape

Out[106]: (60,)
```

```
In [107]: X_Test.shape

Out[107]: (60, 7)
```



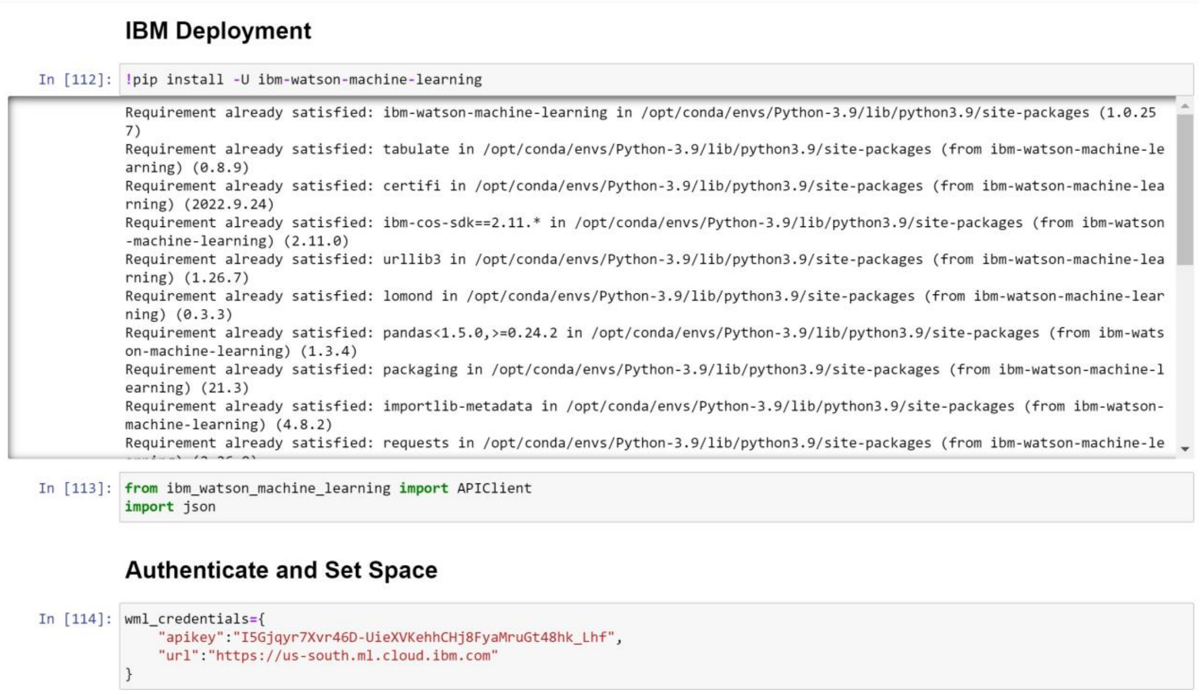
MODELING AND TRAINING

```
In [108]: y_Train = (y_Train>0.5)
          y_Test = (y_Test>0.5)
```

```
In [109]: from sklearn.linear_model._logistic import LogisticRegression
          lore = LogisticRegression(random_state=0, max_iter=1000)
          lr = lore.fit(X_Train, y_Train)
```

```
In [110]: y_pred = lr.predict(X_Test)
          y_pred

Out[110]: array([ True,  True,  True,  True,  True,  True,  True,  True,  True,
                  True,  True,  True,  True,  True,  True,  True,  True,  True,
                  True,  True, False,  True, False,  True,  True,  True,  True,
                  True,  True,  True, False,  True, False,  True,  True,  True,
                  True,  True,  True,  True,  True,  True,  True,  True,  True,
                  True,  True,  True,  True,  True,  True,  True,  True,  True,
                  True,  True, False,  True,  True,  True])
```

```
In [111]: from sklearn.metrics import accuracy_score, recall_score, roc_auc_score, confusion_matrix
          print('Accuracy Score:', accuracy_score(y_Test, y_pred))
          print('Recall Score:', recall_score(y_Test, y_pred))
          print('ROC AUC Score:', roc_auc_score(y_Test, y_pred))
          print('Confussion Matrix:\n', confusion_matrix(y_Test, y_pred))

          Accuracy Score: 0.9166666666666666
          Recall Score: 0.9807692307692307
          ROC AUC Score: 0.7403846153846154
          Confussion Matrix:
           [[ 4  4]
           [ 1 51]]
```



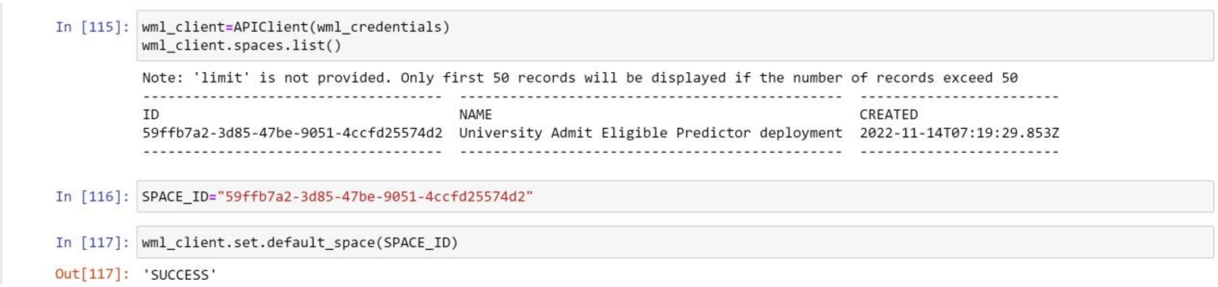
IBM Deployment

```
In [112]: !pip install -U ibm-watson-machine-learning

          Requirement already satisfied: ibm-watson-machine-learning in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.25
          7)
          Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-le
          arning) (0.8.9)
          Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-lea
          rning) (2022.9.24)
          Requirement already satisfied: ibm-cos-sdk==2.11.* in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson
          -machine-learning) (2.11.0)
          Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-lea
          rning) (1.26.7)
          Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-lear
          ning) (0.3.3)
          Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-wats
          on-machine-learning) (1.3.4)
          Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-l
          earning) (21.3)
          Requirement already satisfied: importlib-metadata in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-
          machine-learning) (4.8.2)
          Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-le
```

```
In [113]: from ibm_watson_machine_learning import APIClient
          import json
```

Authenticate and Set Space

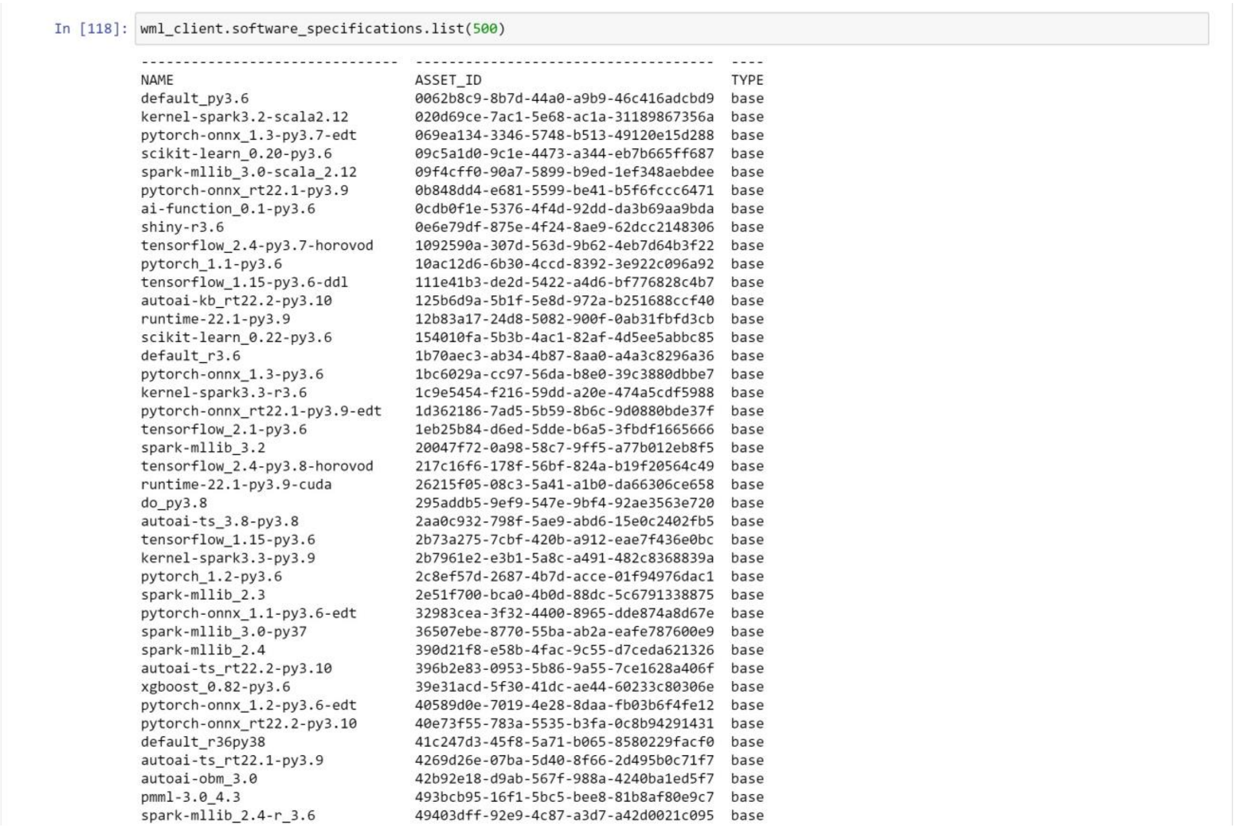
```
In [114]: wml_credentials={
              "apikey":"I5Gjqyr7Xvr46D-UieXVKehhCHj8FyaMruGt48hk_Lhf",
              "url":"https://us-south.ml.cloud.ibm.com"
          }
```



```
In [115]: wml_client=APIClient(wml_credentials)
          wml_client.spaces.list()

          Note: 'limit' is not provided. Only first 50 records will be displayed if the number of records exceed 50
          ------------------------------------  --------------------------------------------  ------------------------
          ID                                    NAME                                          CREATED
          59ffb7a2-3d85-47be-9051-4ccfd25574d2  University Admit Eligible Predictor deployment  2022-11-14T07:19:29.853Z
          ------------------------------------  --------------------------------------------  ------------------------
```

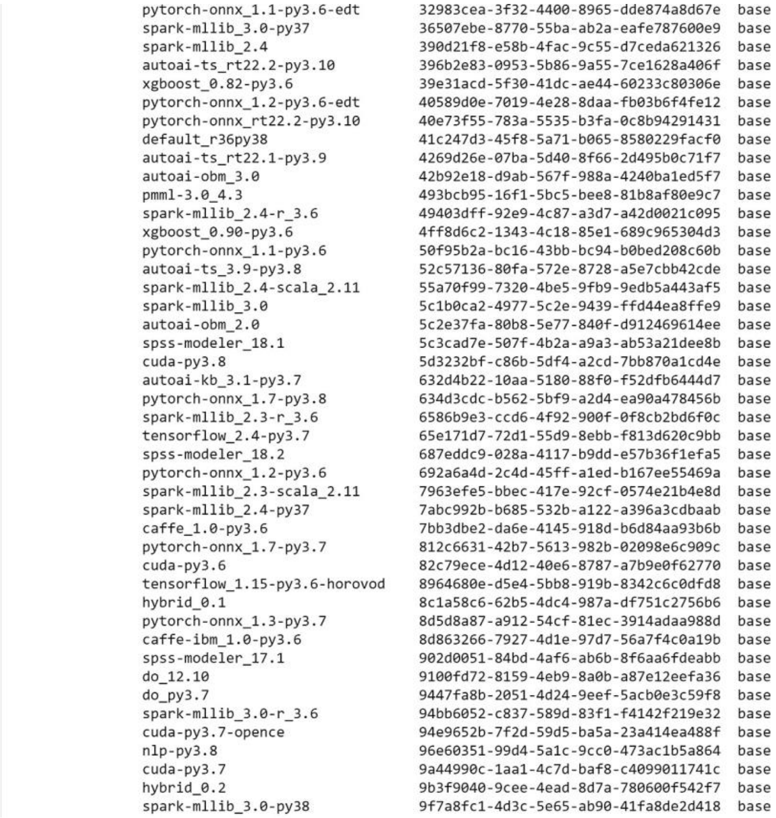
```
In [116]: SPACE_ID="59ffb7a2-3d85-47be-9051-4ccfd25574d2"
```

```
In [117]: wml_client.set.default_space(SPACE_ID)

Out[117]: 'SUCCESS'
```



```
In [118]: wml_client.software_specifications.list(500)
```

```
-------------------------------  ------------------------------------  ----
NAME                             ASSET_ID                              TYPE
default_py3.6                    0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
kernel-spark3.2-scala2.12        020d69ce-7ac1-5e68-ac1a-31189867356a  base
pytorch-onnx_1.3-py3.7-edt       069ea134-3346-5748-b513-49120e15d288  base
scikit-learn_0.20-py3.6          09c5a1d0-9c1e-4473-a344-eb7b665ff687  base
spark-mllib_3.0-scala_2.12       09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
pytorch-onnx_rt22.1-py3.9        0b848dd4-e681-5599-be41-b5f6fccc6471  base
ai-function_0.1-py3.6            0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  base
shiny-r3.6                       0e6e79df-875e-4f24-8ae9-62dcc2148306  base
tensorflow_2.4-py3.7-horovod     1092590a-307d-563d-9b62-4eb7d64b3f22  base
pytorch_1.1-py3.6                10ac12d6-6b30-4ccd-8392-3e922c096a92  base
tensorflow_1.15-py3.6-ddl        111e41b3-de2d-5422-a4d6-bf776828c4b7  base
autoai-kb_rt22.2-py3.10          125b6d9a-5b1f-5e8d-972a-b251688ccf40  base
runtime-22.1-py3.9               12b83a17-24d8-5082-900f-0ab31fbfd3cb  base
scikit-learn_0.22-py3.6          154010fa-5b3b-4ac1-82af-4d5ee5abbc85  base
default_r3.6                     1b70aec3-ab34-4b87-8aa0-a4a3c8296a36  base
pytorch-onnx_1.3-py3.6           1bc6029a-cc97-56da-b8e0-39c3880dbbe7  base
kernel-spark3.3-r3.6             1c9e5454-f216-59dd-a20e-474a5cdf5988  base
pytorch-onnx_rt22.1-py3.9-edt    1d362186-7ad5-5b59-8b6c-9d0880bde37f  base
tensorflow_2.1-py3.6             1eb25b84-d6ed-5dde-b6a5-3fbdf1665666  base
spark-mllib_3.2                  20047f72-0a98-58c7-9ff5-a77b012eb8f5  base
tensorflow_2.4-py3.8-horovod     217c16f6-178f-56bf-824a-b19f20564c49  base
runtime-22.1-py3.9-cuda          26215f05-08c3-5a41-a1b0-da66306ce658  base
do_py3.8                         295addb5-9ef9-547e-9bf4-92ae3563e720  base
autoai-ts_3.8-py3.8              2aa0c932-798f-5ae9-abd6-15e0c2402fb5  base
tensorflow_1.15-py3.6            2b73a275-7cbf-420b-a912-eae7f436e0bc  base
kernel-spark3.3-py3.9            2b7961e2-e3b1-5a8c-a491-482c8368839a  base
pytorch_1.2-py3.6                2c8ef57d-2687-4b7d-acce-01f94976dac1  base
spark-mllib_2.3                  2e51f700-bca0-4b0d-88dc-5c6791338875  base
pytorch-onnx_1.1-py3.6-edt       32983cea-3f32-4400-8965-dde874a8d67e  base
spark-mllib_3.0-py37             36507ebe-8770-55ba-ab2a-eafe787600e9  base
spark-mllib_2.4                  390d21f8-e58b-4fac-9c55-d7ceda621326  base
autoai-ts_rt22.2-py3.10          396b2e83-0953-5b86-9a55-7ce1628a406f  base
xgboost_0.82-py3.6               39e31acd-5f30-41dc-ae44-60233c80306e  base
pytorch-onnx_1.2-py3.6-edt       40589d0e-7019-4e28-8daa-fb03b6f4fe12  base
pytorch-onnx_rt22.2-py3.10       40e73f55-783a-5535-b3fa-0c8b94291431  base
default_r36py38                  41c247d3-45f8-5a71-b065-8580229facf0  base
autoai-ts_rt22.1-py3.9           4269d26e-07ba-5d40-8f66-2d495b0c71f7  base
autoai-obm_3.0                   42b92e18-d9ab-567f-988a-4240ba1ed5f7  base
pmml-3.0_4.3                     493bcb95-16f1-5bc5-bee8-81b8af80e9c7  base
spark-mllib_2.4-r_3.6            49403dff-92e9-4c87-a3d7-a42d0021c095  base
```



```
pytorch-onnx_1.1-py3.6-edt       32983cea-3f32-4400-8965-dde874a8d67e  base
spark-mllib_3.0-py37             36507ebe-8770-55ba-ab2a-eafe787600e9  base
spark-mllib_2.4                  390d21f8-e58b-4fac-9c55-d7ceda621326  base
autoai-ts_rt22.2-py3.10          396b2e83-0953-5b86-9a55-7ce1628a406f  base
xgboost_0.82-py3.6               39e31acd-5f30-41dc-ae44-60233c80306e  base
pytorch-onnx_1.2-py3.6-edt       40589d0e-7019-4e28-8daa-fb03b6f4fe12  base
pytorch-onnx_rt22.2-py3.10       40e73f55-783a-5535-b3fa-0c8b94291431  base
default_r36py38                  41c247d3-45f8-5a71-b065-8580229facf0  base
autoai-ts_rt22.1-py3.9           4269d26e-07ba-5d40-8f66-2d495b0c71f7  base
autoai-obm_3.0                   42b92e18-d9ab-567f-988a-4240ba1ed5f7  base
pmml-3.0_4.3                     493bcb95-16f1-5bc5-bee8-81b8af80e9c7  base
spark-mllib_2.4-r_3.6            49403dff-92e9-4c87-a3d7-a42d0021c095  base
```

```
xgboost_0.90-py3.6               4ff8d6c2-1343-4c18-85e1-689c965304d3  base
pytorch-onnx_1.1-py3.6           50f95b2a-bc16-43bb-bc94-b0bed208c60b  base
autoai-ts_3.9-py3.8              52c57136-80fa-572e-8728-a5e7cbb42cde  base
spark-mllib_2.4-scala_2.11       55a70f99-7320-4be5-9fb9-9edb5a443af5  base
spark-mllib_3.0                  5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9  base
autoai-obm_2.0                   5c2e37fa-80b8-5e77-840f-d912469614ee  base
spss-modeler_18.1                5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b  base
cuda-py3.8                       5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e  base
autoai-kb_3.1-py3.7              632d4b22-10aa-5180-88f0-f52dfb6444d7  base
pytorch-onnx_1.7-py3.8           634d3cdc-b562-5bf9-a2d4-ea90a478456b  base
spark-mllib_2.3-r_3.6            6586b9e3-ccd6-4f92-900f-0f8cb2bd6f0c  base
tensorflow_2.4-py3.7             65e171d7-72d1-55d9-8ebb-f813d620c9bb  base
spss-modeler_18.2                687eddc9-028a-4117-b9dd-e57b36f1efa5  base
pytorch-onnx_1.2-py3.6           692a6a4d-2c4d-45ff-a1ed-b167ee55469a  base
spark-mllib_2.3-scala_2.11       7963efe5-bbec-417e-92cf-0574e21b4e8d  base
spark-mllib_2.4-py37             7abc992b-b685-532b-a122-a396a3cdbaab  base
caffe_1.0-py3.6                  7bb3dbe2-da6e-4145-918d-b6d84aa93b6b  base
pytorch-onnx_1.7-py3.7           812c6631-42b7-5613-982b-02098e6c909c  base
cuda-py3.6                       82c79ece-4d12-40e6-8787-a7b9e0f62770  base
tensorflow_1.15-py3.6-horovod    8964680e-d5e4-5bb8-919b-8342c6c0dfd8  base
hybrid_0.1                       8c1a58c6-62b5-4dc4-987a-df751c2756b6  base
pytorch-onnx_1.3-py3.7           8d5d8a87-a912-54cf-81ec-3914adaa988d  base
caffe-ibm_1.0-py3.6              8d863266-7927-4d1e-97d7-56a7f4c0a19b  base
spss-modeler_17.1                902d0051-84bd-4af6-ab6b-8f6aa6fdeabb  base
do_12.10                         9100fd72-8159-4eb9-8a0b-a87e12eefa36  base
do_py3.7                         9447fa8b-2051-4d24-9eef-5acb0e3c59f8  base
spark-mllib_3.0-r_3.6            94bb6052-c837-589d-83f1-f4142f219e32  base
cuda-py3.7-opence                94e9652b-7f2d-59d5-ba5a-23a414ea488f  base
nlp-py3.8                        96e60351-99d4-5a1c-9cc0-473ac1b5a864  base
cuda-py3.7                       9a44990c-1aa1-4c7d-baf8-c4099011741c  base
hybrid_0.2                       9b3f9040-9cee-4ead-8d7a-780600f542f7  base
spark-mllib_3.0-py38             9f7a8fc1-4d3c-5e65-ab90-41fa8de2d418  base
```

```
autoai-kb_3.0-py3.6              d139f196-e04b-5d8b-9140-9a10ca1fa91a  base
spark-mllib_3.0-py36             d82546d5-dd78-5fbb-9131-2ec309bc56ed  base
autoai-kb_3.4-py3.8              da9b39c3-758c-5a4f-9cfd-457dd4d8c395  base
kernel-spark3.2-r3.6             db2fe4d6-d641-5d05-9972-73c654c60e0a  base
autoai-kb_rt22.1-py3.9           db6afe93-665f-5910-b117-d879897404d9  base
tensorflow_rt22.1-py3.9-horovod  dda170cc-ca67-5da7-9b7a-cf84c6987fae  base
autoai-ts_1.0-py3.7              deef04f0-0c42-5147-9711-89f9904299db  base
tensorflow_2.1-py3.7-horovod     e384fce5-fdd1-53f8-bc71-11326c9c635f  base
default_py3.7                    e4429883-c883-42b6-87a8-f419d64088cd  base
do_22.1                          e51999ba-6452-5f1f-8287-17228b88b652  base
autoai-obm_3.2                   eae86aab-da30-5229-a6a6-1d0d4e368983  base
tensorflow_rt22.2-py3.10         f65bd165-f057-55de-b5cb-f97cf2c0f393  base
do_20.1                          f686cdd9-7904-5f9d-a732-01b0d6b10dc5  base
pytorch-onnx_rt22.2-py3.10-edt   f8a05d07-e7cd-57bb-a10b-23f1d4b837ac  base
scikit-learn_0.19-py3.6          f963fa9d-4bb7-5652-9c5d-8d9289ef6ad9  base
tensorflow_2.4-py3.8             fe185c44-9a99-5425-986b-59bd1d2eda46  base
-------------------------------  ------------------------------------  ----
```

Save and Deploy the model

In [119]:
```python
import sklearn
sklearn.__version__
```

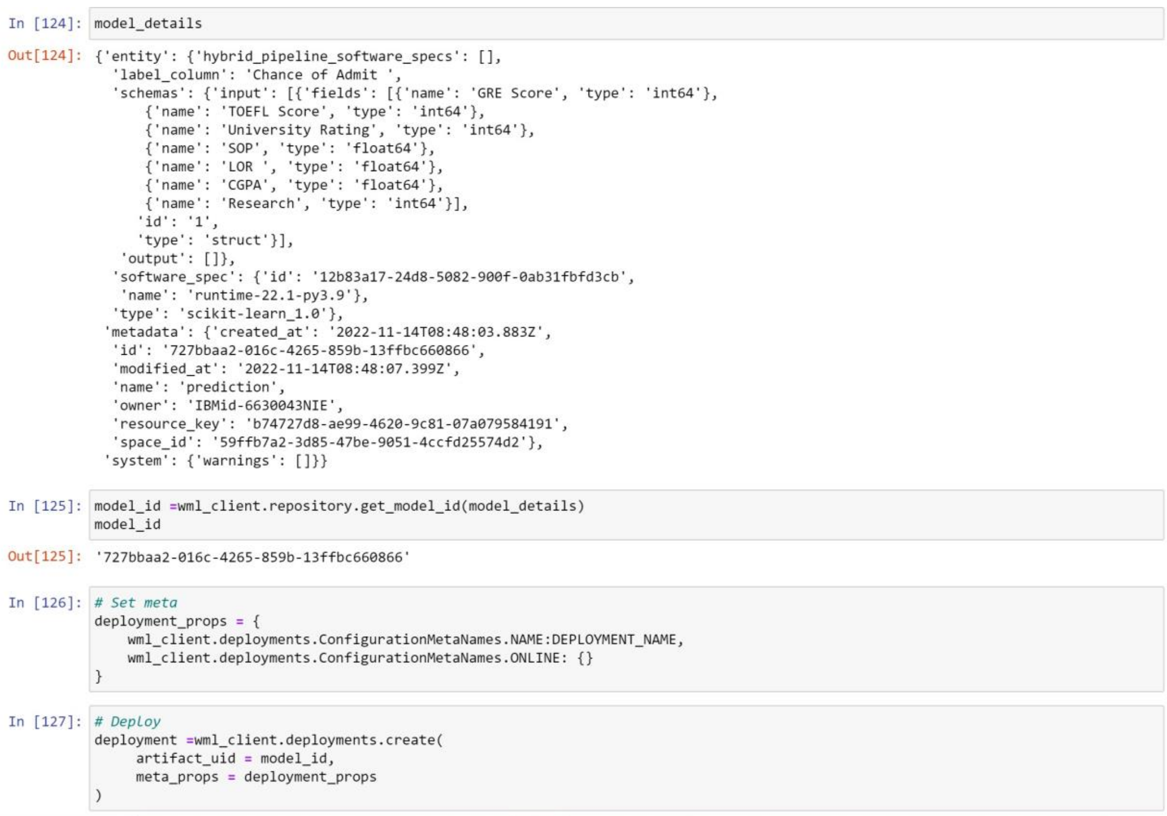
Out[119]:
```
'1.0.2'
```

In [120]:
```python
MODEL_NAME='prediction'
DEPLOYMENT_NAME='University Admit Eligible Predictor deployment'
DEMO_MODEL=lr
```

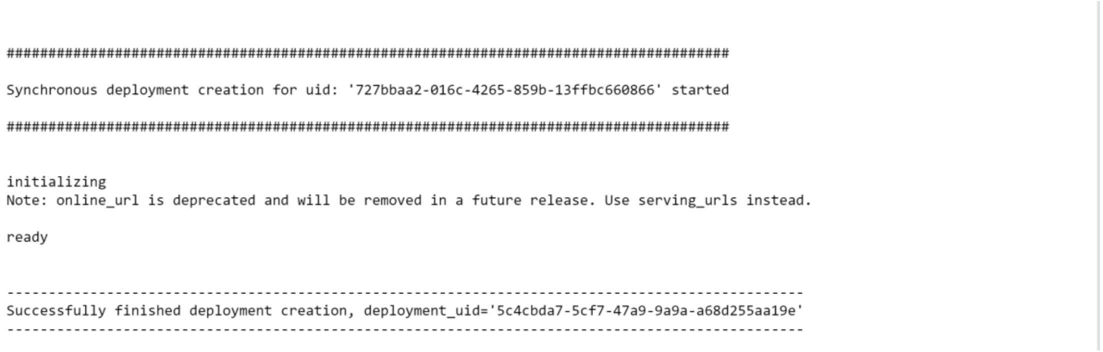
In [121]:
```python
# Set Python Version
software_spec_uid=wml_client.software_specifications.get_id_by_name('runtime-22.1-py3.9')
```

In [122]:
```python
#Setup model meta
model_props = {
    wml_client.repository.ModelMetaNames.NAME: MODEL_NAME,
    wml_client.repository.ModelMetaNames.TYPE: 'scikit-learn_1.0 ',
    wml_client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid
}
```

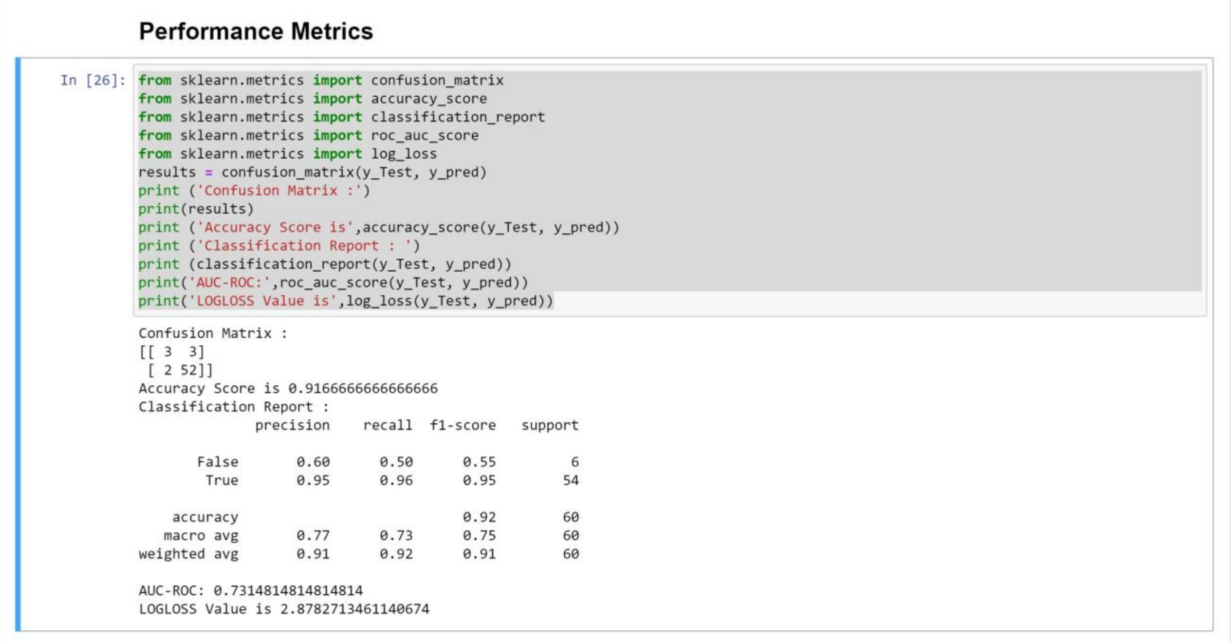
In [123]:
```python
#Save model
model_details =wml_client.repository.store_model(
    model = DEMO_MODEL,
    meta_props = model_props,
    training_data = X_Train,
    training_target = y_Train
)
```



```
In [124]: model_details

Out[124]: {'entity': {'hybrid_pipeline_software_specs': [],
            'label_column': 'Chance of Admit ',
            'schemas': {'input': [{'fields': [{'name': 'GRE Score', 'type': 'int64'},
                {'name': 'TOEFL Score', 'type': 'int64'},
                {'name': 'University Rating', 'type': 'int64'},
                {'name': 'SOP', 'type': 'float64'},
                {'name': 'LOR ', 'type': 'float64'},
                {'name': 'CGPA', 'type': 'float64'},
                {'name': 'Research', 'type': 'int64'}],
               'id': '1',
               'type': 'struct'}],
             'output': []},
            'software_spec': {'id': '12b83a17-24d8-5082-900f-0ab31fbfd3cb',
             'name': 'runtime-22.1-py3.9'},
            'type': 'scikit-learn_1.0'},
           'metadata': {'created_at': '2022-11-14T08:48:03.883Z',
            'id': '727bbaa2-016c-4265-859b-13ffbc660866',
            'modified_at': '2022-11-14T08:48:07.399Z',
            'name': 'prediction',
            'owner': 'IBMid-6630043NIE',
            'resource_key': 'b74727d8-ae99-4620-9c81-07a079584191',
            'space_id': '59ffb7a2-3d85-47be-9051-4ccfd25574d2'},
           'system': {'warnings': []}}

In [125]: model_id =wml_client.repository.get_model_id(model_details)
          model_id

Out[125]: '727bbaa2-016c-4265-859b-13ffbc660866'

In [126]: # Set meta
          deployment_props = {
              wml_client.deployments.ConfigurationMetaNames.NAME:DEPLOYMENT_NAME,
              wml_client.deployments.ConfigurationMetaNames.ONLINE: {}
          }

In [127]: # Deploy
          deployment =wml_client.deployments.create(
               artifact_uid = model_id,
               meta_props = deployment_props
          )
```



```
#######################################################################################

Synchronous deployment creation for uid: '727bbaa2-016c-4265-859b-13ffbc660866' started

#######################################################################################


initializing
Note: online_url is deprecated and will be removed in a future release. Use serving_urls instead.

ready


------------------------------------------------------------------------------------------------
Successfully finished deployment creation, deployment_uid='5c4cbda7-5cf7-47a9-9a9a-a68d255aa19e'
------------------------------------------------------------------------------------------------
```



Performance Metrics

```
In [26]: from sklearn.metrics import confusion_matrix
         from sklearn.metrics import accuracy_score
         from sklearn.metrics import classification_report
         from sklearn.metrics import roc_auc_score
         from sklearn.metrics import log_loss
         results = confusion_matrix(y_Test, y_pred)
         print ('Confusion Matrix :')
         print(results)
         print ('Accuracy Score is',accuracy_score(y_Test, y_pred))
         print ('Classification Report : ')
         print (classification_report(y_Test, y_pred))
         print('AUC-ROC:',roc_auc_score(y_Test, y_pred))
         print('LOGLOSS Value is',log_loss(y_Test, y_pred))
```

```
Confusion Matrix :
[[ 3  3]
 [ 2 52]]
Accuracy Score is 0.9166666666666666
Classification Report :
              precision    recall  f1-score   support

       False       0.60      0.50      0.55         6
        True       0.95      0.96      0.95        54

    accuracy                           0.92        60
   macro avg       0.77      0.73      0.75        60
weighted avg       0.91      0.92      0.91        60

AUC-ROC: 0.7314814814814814
LOGLOSS Value is 2.8782713461140674
```

GitHub Link: GitHub - IBM-EPBL/IBM-Project-43388-1660716500: University Admit Eligibility Predictor