IBM-Project-21048-1659771001

Project Name: University Admit Eligibility Predictor

Team Id : PNT2022TMID20129

Domain : Applied Data Science

Team Members:

Dhanarajan G (Team Leader)

Gokul Karthick P

Sanjay Kumar S

Ajay Rajendran S

CONTENTS

1. Introduction

1.1 Project Overview

1.2 Purpose
2. Literature Survey
2.1 Existing Problem
2.2 References
2.3 Problem Statement Definitions
3. Ideation & Proposed Solution
3.1 Empathy Map Canvas
3.2 Ideation & Brainstorming
3.3 Proposed Solution
3.4 Problem Solution Fit
4. Requirement Analysis
4.1 Functional Requirement
4.2 Non-Functional Requirement
5. Project Design
5.1 Data Flow Diagrams
5.2 Solution & Technical Architecture
5.3 User Stories
6. Project Planning & Scheduling
6.1 Sprint Planning & Estimation
6.2 Sprint Delivery Schedule

7. Coding & Solutioning
7.1 Feature 1
7.2 Feature 2
8. Testing
8.1 Test Cases
8.2 User Acceptance Testing
9. Results
9.1 Performance Metrics

10. Advantages & Disadvantages
11. Conclusion
12. Future Scope
13. Appendix
13.1 Source Code
13.2 GitHub
13.3 Project Demo Link

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 better idea.

1.2 Purpose

This university admit eligibility predictor is a web based application in which students can register with their personal as well as mark details for pediction the 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 in the system. The main advantage of the project is the computerization of the entrance seat allotment process. Administrator has the power of the allotment. He can add the alloted seats into a file and the details are saved 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. It helps student for making decision for choosing a right college and money that they have to spend at education consultancy firms and also it will help them to limit their application to small number by proving them the suggestion of the universities where they have best chance of securing admission thus saving more money on the application fees.

LITERATURE SURVEY

2.1 Exsisting problem

In this web based application, the scope of this project is a web application that allows user to enter their academic data and get predicitons of their chances of admissions in the university tier of their choosing. It also provides them to answer to most comman FAQ's that araise when thinking of admissions for for post graduate studies. It also provides analysis based on the dataset shows how different parameters affect the chances of admissions. A Database will also be implemented for the system so that students can save their profile. Issuses of web security other than password protection within the website are part of this project. Another issuse is to avoid time consuming high and make the task completed the task in a quickly manner.

2.2 References

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

[2] Bibodi, J., Vadodaria, A., Rawat, A. and Patel, J. (n.d.). Admission Prediction System Using Machine Learning.

[3] Jamison, Applying Machine learning to predict Davidson college's admissions Yeild , pp 765-766(2017)

[4] 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.

[5] Lapovsky, L The Changing Business for Colleges and Universities. Forbes(2018)

[6] Mane, R. V. (2016). Predicting Student Admission decisions byAssociation Rule Mining with Pattern Growth Approach, pp. 202–207

[7] J. Bodailla et al."Knowledge -Based System" Elseiver B.V

[8] B. Ghai , "Analysis and prediction of american graduate admissions Process"2018

[9] B.K. Bardwaj and s. pal "mining education data to analyse student's performance",2012

[10] c.c. Aggarwal, Data mining: The textbook ,2015

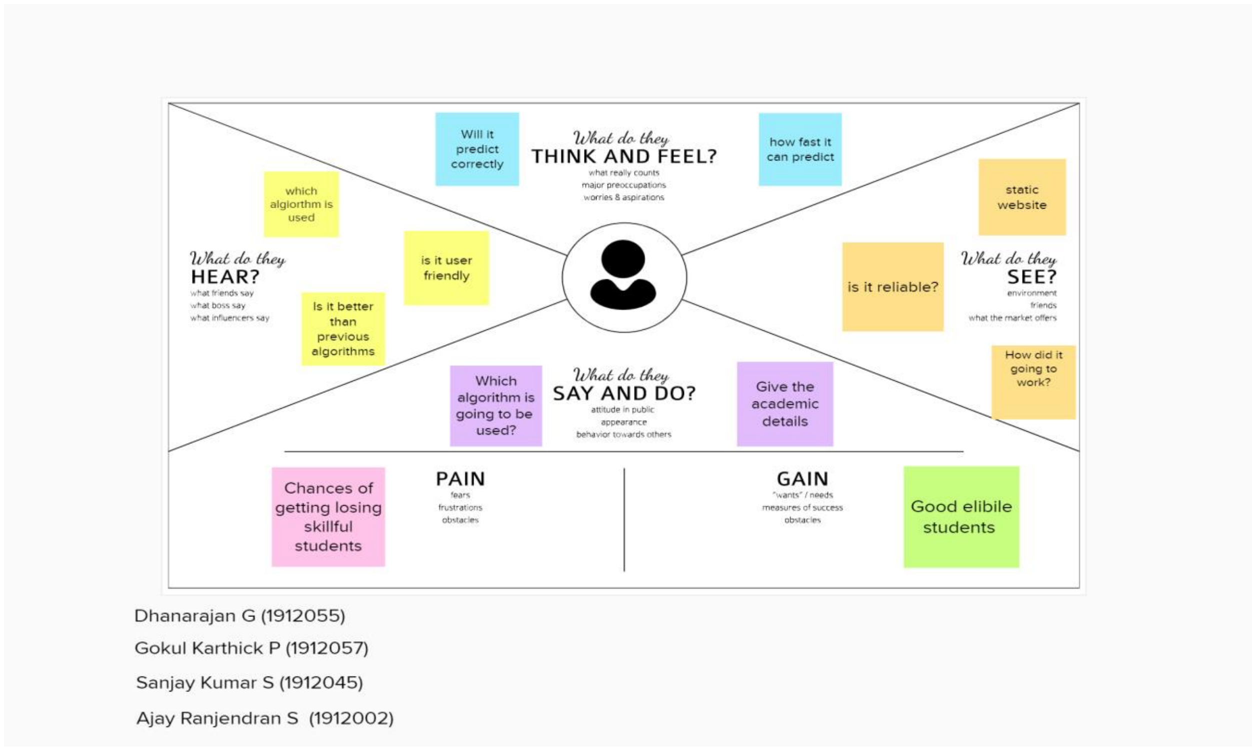
2.3 Problem Statement Definition

The problem in this process it is far away from the place so it takes time to for apply college and process is very tedious. It is also very difficult when it comes to offline and it takes long time for the result of the application. Because of time consumption and delay in result makes student's life questionable and career path unclear. This will be pioneer to make students to choose correct career and

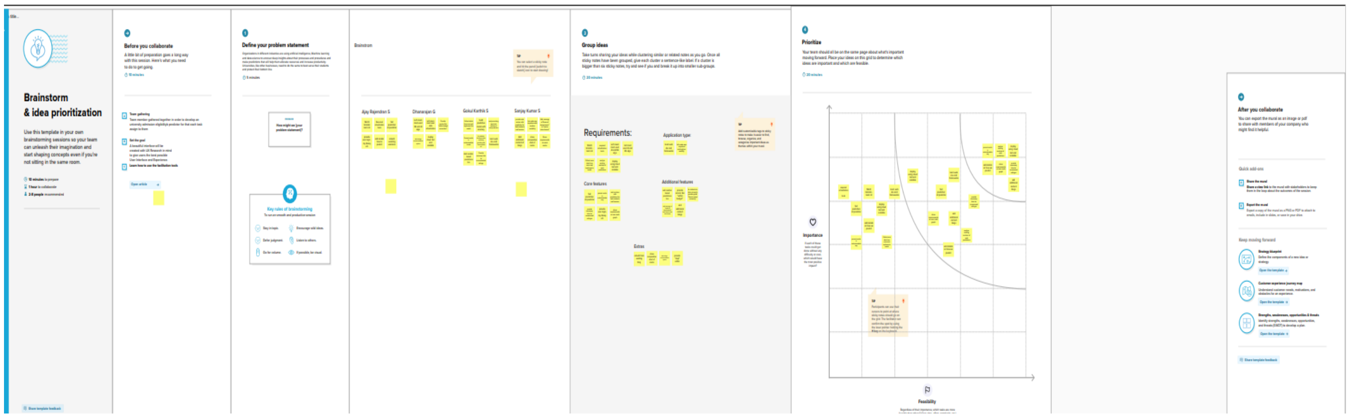
IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community.



3.2 Ideation & Brainstorming



Ideation is often closely related to the practice of brainstorming, a specific technique that is utilized to generate new ideas. A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity.As you can see, ideation is not just a one time idea generation or a brainstorming session. In fact, we can divide ideation in these three stages: generation, selection, and development.Brainstorming is one of the primary methods employed during the Ideation stage of a typical Design Thinking process.

Brainstorming is a method of generating ideas and sharing knowledge to solve a particular commercial or technical problem, in which participants are encouraged to think without interruption. Brainstorming is a group activity where each participant shares their ideas as soon as they come to mind.Group brainstorming stimulates creativity and invites participation from everyone, making it a great tool for generating a wide variety of ideas in a short amount of time. It's especially helpful when trying to solve a problem that you are really close to. Sometimes getting outside perspectives can breathe new life into a project and drive momentum towards a solution.

3.3 Problem Solution

Problem solving is a basic task for the project management. It is a process for developing and applying a solution for the occurred problems. The probability of the success rises, if a particular method is implemented to the project work.

3.4 Problem Solution fit

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

1.Customer Segment

The possible student who have completed their schooling and UG searching for university to study PG

2.Jobs to be done

The major task is to design a university admission prediction system and to provide probabilistic insight into the university rating, cuttoffs, intake count

3.Triggers

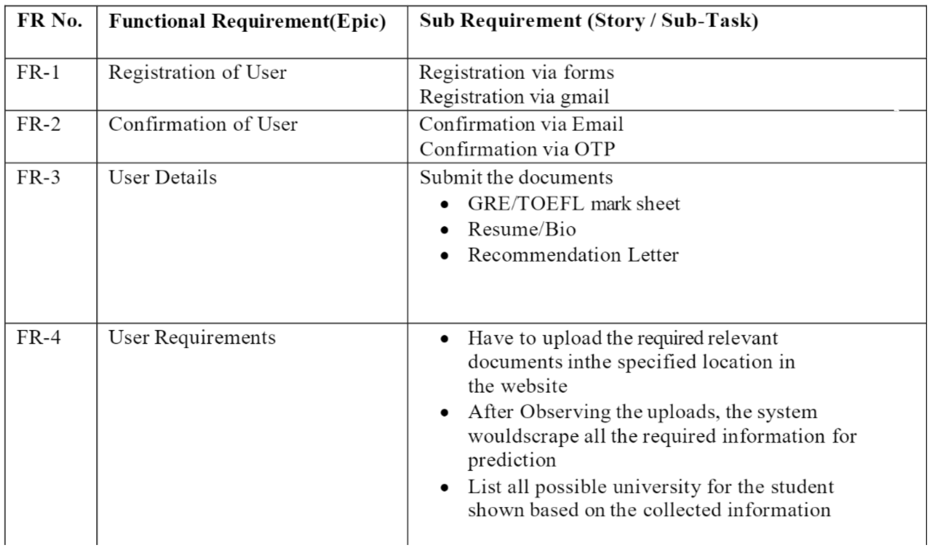
Students often get tensed and anxious about their admission chances of their desired universities
4.Emotions
Before : Insecure and unaware of the process, suffering to select the best suited-university
After : Secure, user friendly and aware of process
5.Available Solutions
Lack dynamic nature and scalability.
Incomplete training information.

REQUIREMENT ANALYSIS

4.1 Functional requirements

A functional requirement defines a function of a system or its component, where a function is described as a specification of behavior between inputs and outputs. Functional requirements may involve calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish. Behavioral requirements describe all the cases where the system uses the functional requirements, these are captured in use cases.

Following are the functional requirements of the proposed solution.



FR No.	Functional Requirement(Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Registration of User	Registration via forms Registration via gmail
FR-2	Confirmation of User	Confirmation via Email Confirmation via OTP
FR-3	User Details	Submit the documents • GRE/TOEFL mark sheet • Resume/Bio • Recommendation Letter
FR-4	User Requirements	• Have to upload the required relevant documents inthe specified location in the website • After Observing the uploads, the system wouldscrape all the required information for prediction • List all possible university for the student shown based on the collected information

4.2 Non Functional requirements

Nonfunctional Requirements (NFRs) define system attributes such as security, reliability, performance, maintainability, scalability, and usability. They serve as constraints or restrictions on the design of the system across the different backlogs.

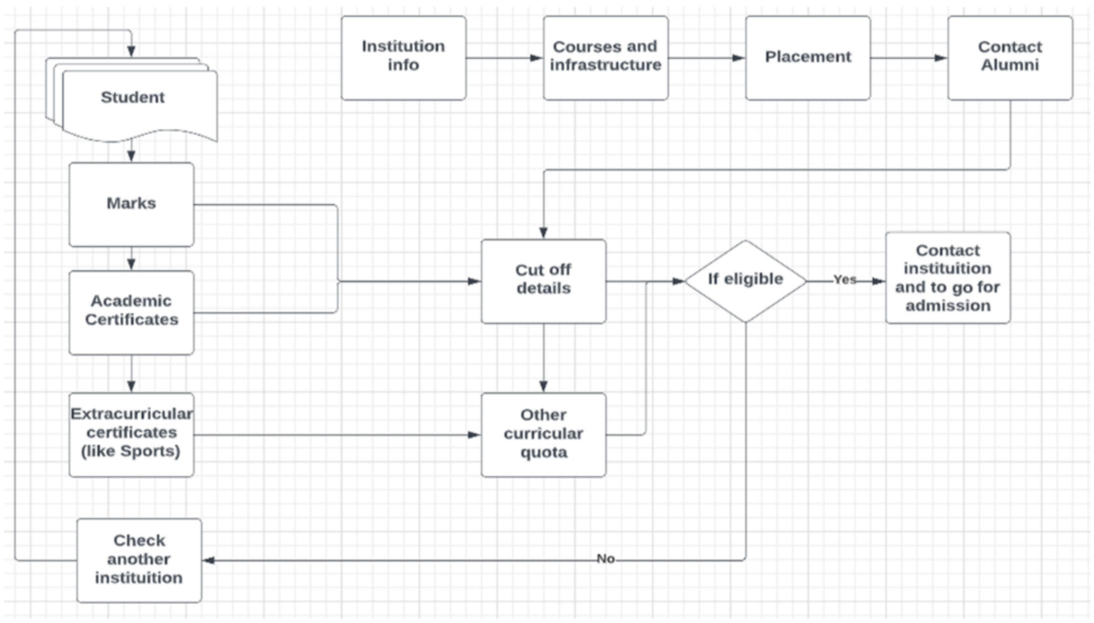
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	• This system doesn't expect any technical pre-requisite/knowledge from the user • It is very user friendly • Reduced focus on Short Term memoryload Focus on Internal Locus of Control • It load the content and display them (< 30 seconds) i.e user doesn't wait for a while • The fields in the website is quite easy to understand and self explanatory
NFR-2	**Security**	• The authenticated user could be able to access the services of the site. • User Data is saved and updated every hour • In case of failure, the system should be able to come back to normal operation with in 1 hour
NFR-3	**Reliability**	• The website would always try to get back normal with maximum reliability due to the damages that could be caused because of incomplete and incorrect data
NFR-4	**Performance**	• It can balance the traffic efficiently by service the request quickly as much as possible
NFR-5	**Availability**	• Little bit of data redundancy • There may be some prone to error • Quick,Compatible and robust • It works 24x7
NFR-6	**Scalability**	• The number of students is getting increasing now days so it is crucial that a more number of users/students should be able to access the system at the same time without any delay • The admission season is probably when the system will be under the moststrain. • It must therefore be able to manage numerous concurrent users.

PROJECT DESIGN

5.1 Data Flow Diagrams

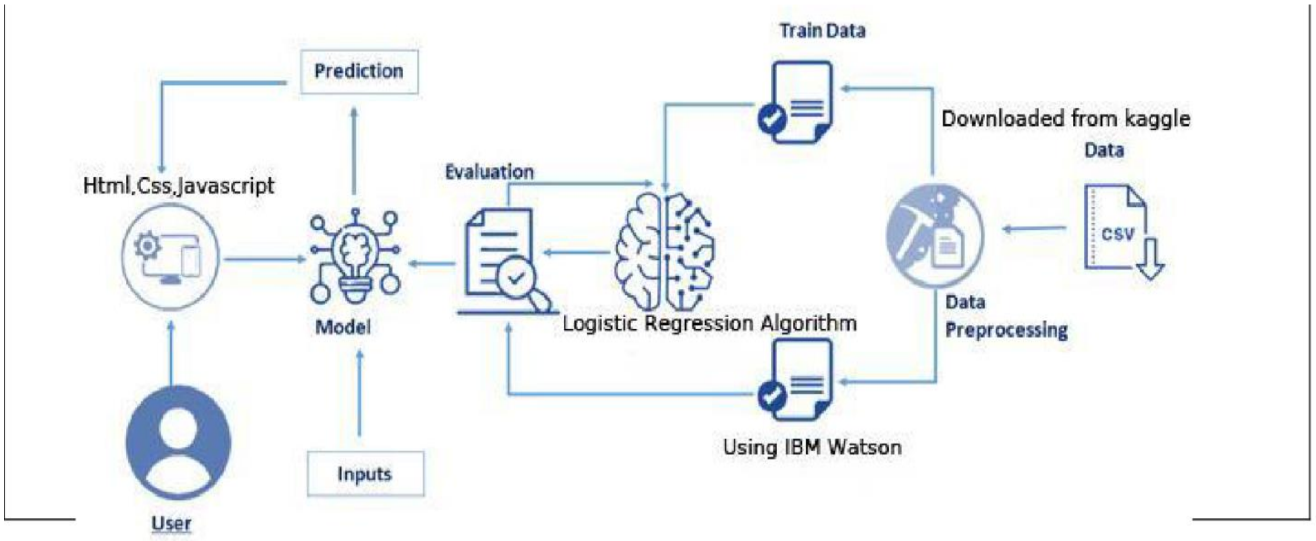
Data flow diagram is a graphical or visual representation using a standardized set of symbols and notations to describe the business operations through data movement.



Institution info
Courses and infrastructure
Placement
Contact Alumni
Student
Marks
Academic Certificates
Extracurricular certificates (like Sports)
Cut off details
Other curricular quota
If eligible
Yes
Contact instituition and to go for admission
No
Check another instituition

5.2 Solution and Technical Architecture

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



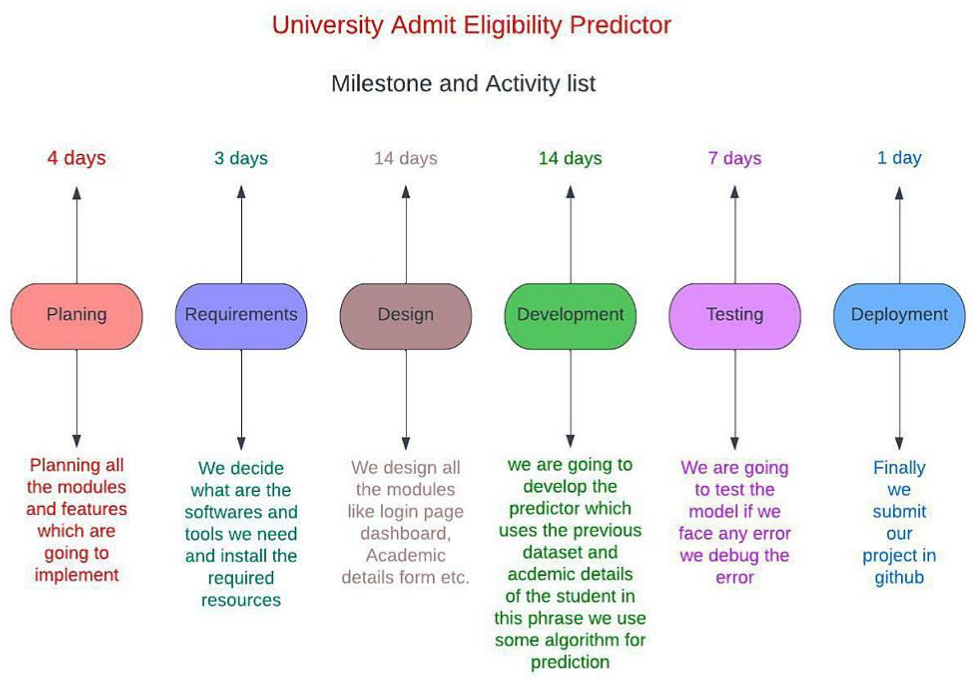
5.3 User Stories

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

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer	Landing page	USN-1	As a user, I can view the details about the institution.	I can access the university landing page	Medium	Sprint-1
		USN-2	As a user, I can view courses, infrastructure of university.	I can get the details about the university.	Medium	Sprint-1
		USN-3	As a user, I can view the placement and training details.	I can ensure the placement confidence.	Medium	Sprint-1
		USN-4	As a user, I can fill the contact form for queries.	I can fill and submit the contact form	Low	Sprint-2
		USN-5	As a user, I can see the social media (instagram, linkedin, facebook) profiles of the university.	I can reach out to them via social media	Medium	Sprint-1
		USN-6	As a user, I can see testimonials of students who graduated from the university	I can access the testimonials	Medium	Sprint-1
	Admissions	USN-7	As a user, I can see the previous and present year cut-off marks and any extracurricular quota (sports).	I can ensure the marks and quota details. I can also download as a document.	High	Sprint-2
		USN-8	As a user, I can read about proud alumni of the university. If possible, I can contact him/her about this institution.	I can access the details of alumni of the university	Medium	Sprint-2

PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning & Estimation


University Admit Eligibility Predictor
Milestone and Activity list
4 days
3 days
14 days
14 days
7 days
1 day
Planing
Requirements
Design
Development
Testing
Deployment
Planning all the modules and features which are going to implement
We decide what are the softwares and tools we need and install the required resources
We design all the modules like login page dashboard, Academic details form etc.
we are going to develop the predictor which uses the previous dataset and acdemic details of the student in this phrase we use some algorithm for prediction
We are going to test the model if we face any error we debug the error
Finally we submit our project in github

6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, you can register in the application by entering your email address, password, and confirming the password	2	High	Gokul Karthick P
Sprint-1		USN-2	As a user, you will receive a confirmation email after registering in the application	1	High	Dhanarajan G
Sprint-2		USN-3	As a user, you can register in the application via Facebook	2	Low	Ajay Rajendran S
Sprint-1	Login	USN-4	As a user, you can login to the application by entering your email and password	1	High	Sanjay Kumar S

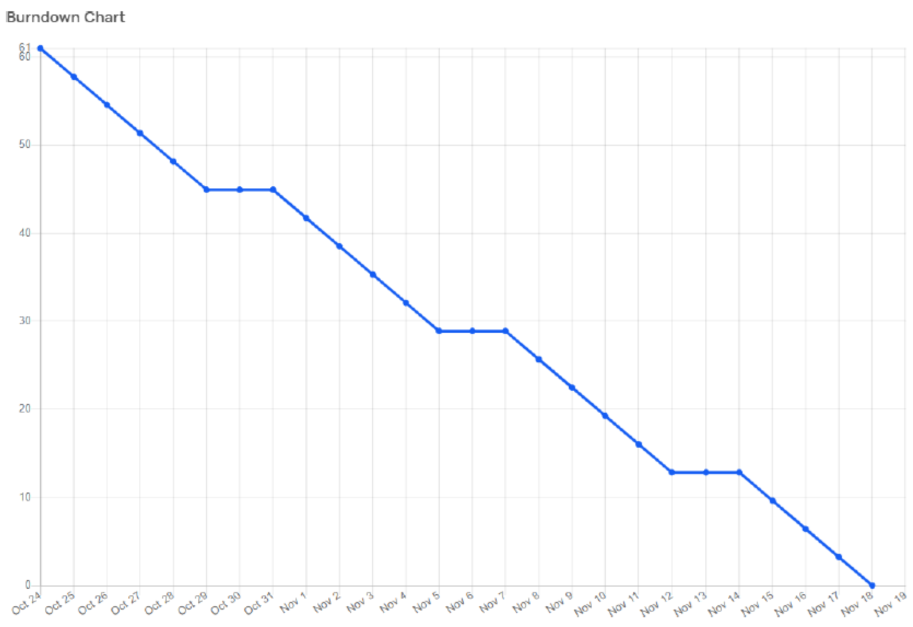
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	5 Days	29 Oct 2022	04 Nov 2022	20	03 Nov 2022
Sprint-2	20	4 Days	04 Oct 2022	08 Nov 2022	20	07 Nov 2022
Sprint-3	20	4 Days	08 Nov 2022	11 Nov 2022	20	10 Nov 2022
Sprint-4	20	4 Days	11 Nov 2022	14 Nov 2022	20	13 Nov 2022

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

6.3 Reports from JIRA

Burndown Chart:

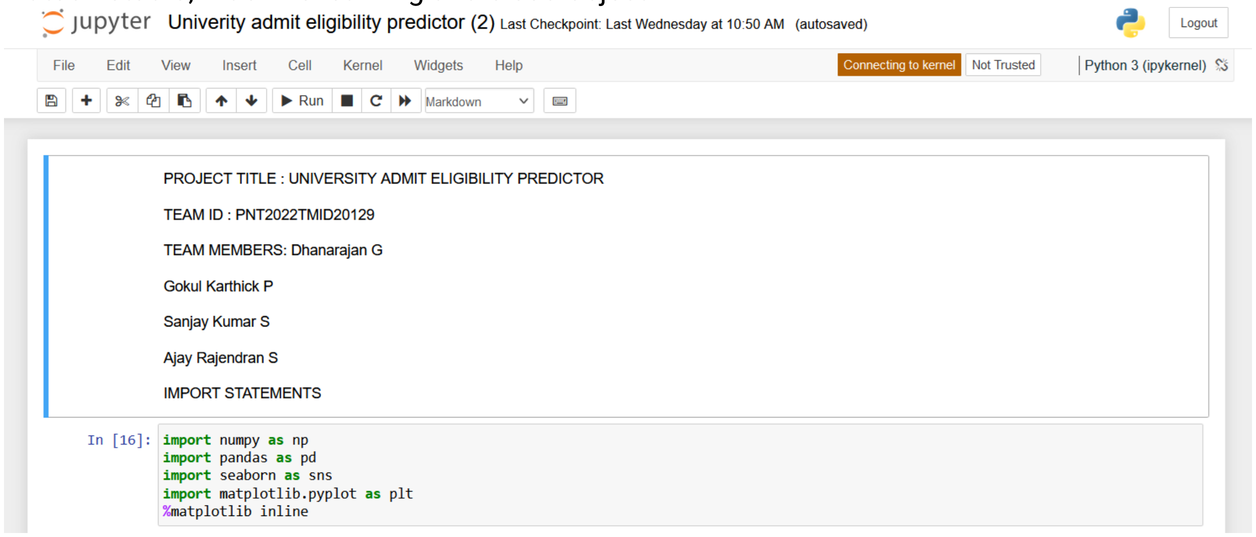
A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



CODING AND SOLUTIONING

7.1 Feature 1

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


jupyter Univerity admit eligibility predictor (2) Last Checkpoint: Last Wednesday at 10:50 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Connecting to kernel
Not Trusted
Python 3 (ipykernel)
Run
Markdown

PROJECT TITLE : UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

TEAM ID : PNT2022TMID20129

TEAM MEMBERS: Dhanarajan G

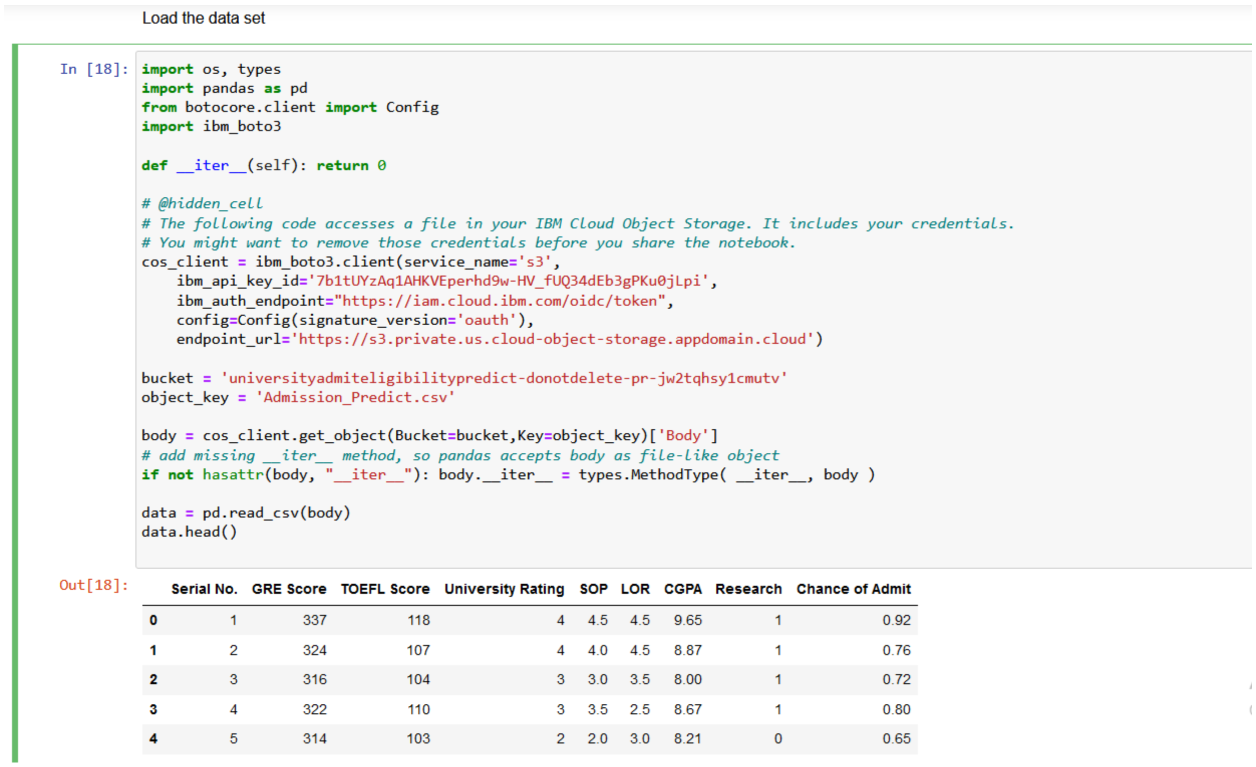
Gokul Karthick P

Sanjay Kumar S

Ajay Rajendran S

IMPORT STATEMENTS

```
In [16]: import numpy as np
         import pandas as pd
         import seaborn as sns
         import matplotlib.pyplot as plt
         %matplotlib inline
```



Load the data set

```
In [18]: 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='7b1tUYzAq1AHKVEperhd9w-HV_fUQ34dEb3gPKu0jLpi',
             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-jw2tqhsy1cmutv'
         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[18]:

	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 [19]:
```python
data.drop(["Serial No."], axis=1, inplace=True)
```

In [20]:
```python
data.describe()
```

Out[20]:

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
count	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000
mean	316.807500	107.410000	3.087500	3.400000	3.452500	8.598925	0.547500	0.724350
std	11.473646	6.069514	1.143728	1.006869	0.898478	0.596317	0.498362	0.142609
min	290.000000	92.000000	1.000000	1.000000	1.000000	6.800000	0.000000	0.340000
25%	308.000000	103.000000	2.000000	2.500000	3.000000	8.170000	0.000000	0.640000
50%	317.000000	107.000000	3.000000	3.500000	3.500000	8.610000	1.000000	0.730000
75%	325.000000	112.000000	4.000000	4.000000	4.000000	9.062500	1.000000	0.830000
max	340.000000	120.000000	5.000000	5.000000	5.000000	9.920000	1.000000	0.970000

In [21]:
```python
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
```

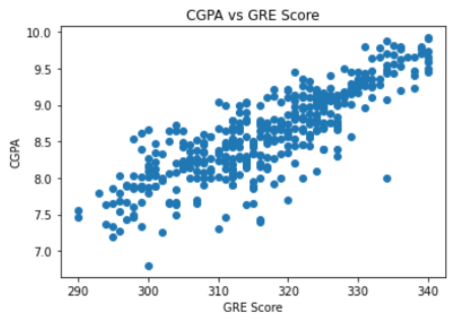
Handling Missing Values

In [22]:
```python
data.isnull().sum()
```

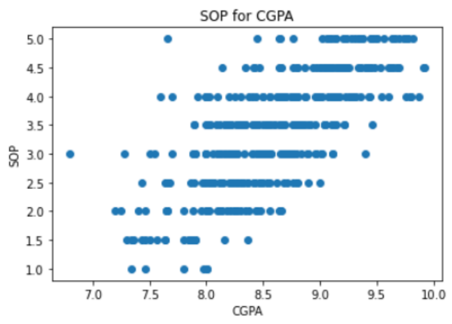
Out[22]:
```
GRE Score            0
TOEFL Score          0
University Rating    0
SOP                  0
LOR                  0
CGPA                 0
Research             0
Chance of Admit      0
dtype: int64
```

Visualization

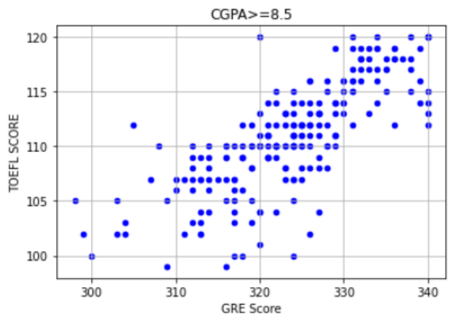
In [23]:
```python
plt.scatter(data['GRE Score'],data['CGPA'])
plt.title('CGPA vs GRE Score')
plt.xlabel('GRE Score')
plt.ylabel('CGPA')
plt.show()
```



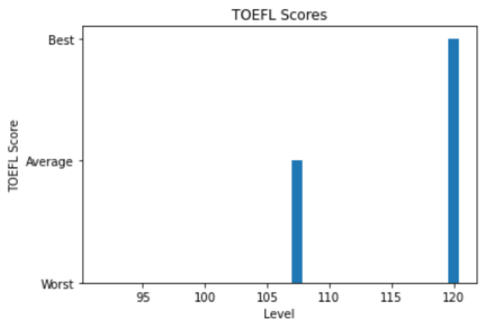
```
In [24]: plt.scatter(data['CGPA'],data['SOP'])
         plt.title('SOP for CGPA')
         plt.xlabel('CGPA')
         plt.ylabel('SOP')
         plt.show()
```



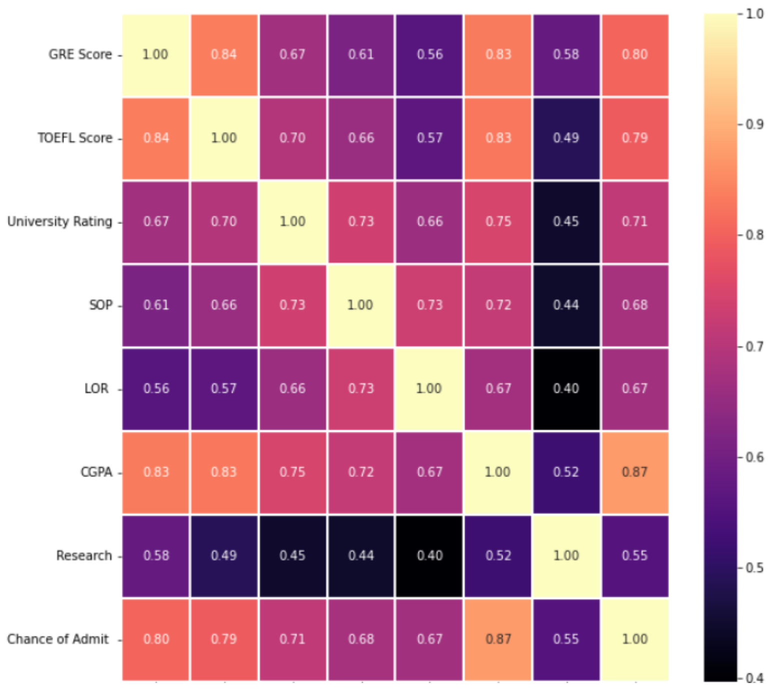
```
In [25]: data[data.CGPA >= 8.5].plot(kind='scatter', x='GRE Score', y='TOEFL Score',color="BLUE")

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

         plt.show()
```



```
In [26]: p = np.array([data["TOEFL Score"].min(),data["TOEFL Score"].mean(),data["TOEFL Score"].max()])
         r = ["Worst","Average","Best"]
         plt.bar(p,r)

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

         plt.show()
```



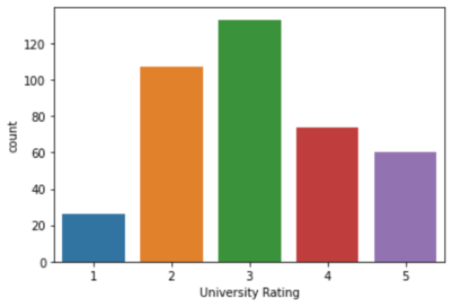
In [12]:
```
plt.figure(figsize=(10, 10))

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

plt.show()
```



GRE Score	1.00	0.84	0.67	0.61	0.56	0.83	0.58	0.80
TOEFL Score	0.84	1.00	0.70	0.66	0.57	0.83	0.49	0.79
University Rating	0.67	0.70	1.00	0.73	0.66	0.75	0.45	0.71
SOP	0.61	0.66	0.73	1.00	0.73	0.72	0.44	0.68
LOR	0.56	0.57	0.66	0.73	1.00	0.67	0.40	0.67
CGPA	0.83	0.83	0.75	0.72	0.67	1.00	0.52	0.87
Research	0.58	0.49	0.45	0.44	0.40	0.52	1.00	0.55
Chance of Admit	0.80	0.79	0.71	0.68	0.67	0.87	0.55	1.00

In [28]:
```
data.Research.value_counts()

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

Out[28]:
```
<AxesSubplot:xlabel='University Rating', ylabel='count'>
```



Trainning and test split

In [29]:
```
X=data.drop(['Chance of Admit '],axis=1) #input data_set
y=data['Chance of Admit '] #output labels
```

In [30]:
```
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)
```

MODELING AND TRAINING

```
In [31]: from sklearn.ensemble import GradientBoostingRegressor
         rgr = GradientBoostingRegressor()
         rgr.fit(X_train,y_train)

Out[31]: GradientBoostingRegressor()
```

```
In [32]: rgr.score(X_test,y_test)

Out[32]: 0.7345575572947072
```

```
In [33]: y_predict=rgr.predict(X_test)
```

```
In [34]: from sklearn.metrics import mean_squared_error, r2_score,mean_absolute_error
         import numpy as np
         print('Mean Absolute Error:', mean_absolute_error(y_test, y_predict))
         print('Mean Squared Error:', mean_squared_error(y_test, y_predict))
         print('Root Mean Squared Error:', np.sqrt(mean_squared_error(y_test, y_predict)))

         Mean Absolute Error: 0.048898416300819064
         Mean Squared Error: 0.00443403167036226
         Root Mean Squared Error: 0.06658852506522622
```

```
In [35]: y_train = (y_train>0.5)
         y_test = (y_test>0.5)
```

```
In [36]: from sklearn.linear_model._logistic import LogisticRegression

         lore = LogisticRegression(random_state=0, max_iter=1000)

         lr = lore.fit(X_train, y_train)
```

```
In [38]: 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: 0.9333333333333333
         Recall Score: 1.0
         ROC AUC Score: 0.6666666666666667
         Confussion Matrix:
          [[ 2  4]
          [ 0 54]]
```

SAVING THE MODEL

```
In [39]: import pickle
```

```
In [40]: pickle.dump(lr, open("university.pkl", "wb")) #logistic regression model
```

IBM Deployment

```
In [1]: 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
(1.0.257)
Requirement already satisfied: ibm-cos-sdk==2.11.* in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ib
m-watson-machine-learning) (2.11.0)
Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from
ibm-watson-machine-learning) (1.3.4)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-mac
hine-learning) (2022.9.24)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-ma
chine-learning) (2.26.0)
Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-m
achine-learning) (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: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-mac
hine-learning) (1.26.7)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-ma
chine-learning) (0.8.9)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-mach
ine-learning) (0.3.3)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (fr
om ibm-cos-sdk==2.11.*->ibm-watson-machine-learning) (2.11.0)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from
ibm-cos-sdk==2.11.*->ibm-watson-machine-learning) (0.10.0)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packag
es (from ibm-cos-sdk==2.11.*->ibm-watson-machine-learning) (2.11.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages
(from ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.*->ibm-watson-machine-learning) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.
5.0,>=0.24.2->ibm-watson-machine-learning) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas
<1.5.0,>=0.24.2->ibm-watson-machine-learning) (1.20.3)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateut
il<3.0.0,>=2.1->ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.*->ibm-watson-machine-learning) (1.15.0)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (f
rom requests->ibm-watson-machine-learning) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests-
>ibm-watson-machine-learning) (3.3)
Requirement already satisfied: zipp>=0.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from importlib-me
tadata->ibm-watson-machine-learning) (3.6.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (fr
om packaging->ibm-watson-machine-learning) (3.0.4)
Note: you may need to restart the kernel to use updated packages.
```

Authenticate and set space

```
In [2]: from ibm_watson_machine_learning import APIClient
        import json
```

```
In [49]: uml_credentials = {
             "url": "https://us-south.ml.cloud.ibm.com",
             "apikey": "PsONFf0TD_cZHNECesojE_TSP7JdrzmnKS2IgxvaVTE1"
         }

         client = APIClient(uml_credentials)
```

```
In [50]: def guid_from_space_name(client, space_name):
             space = client.spaces.get_details()
             idr = []
             for i in space['resources']:
                 idr.append(i['metadata']['id'])
             return idr
```

```
In [68]: space_uid = guid_from_space_name(client, "university")
         print("Space Id:",space_uid[0])
```

```
Space Id: dbdb029e-0578-4249-a7ef-a9afb6207be9
```

```
In [70]: client.software_specifications.list()
```

```
-----------------------------  ------------------------------------  ----
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
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
-----------------------------  ------------------------------------  ----
```

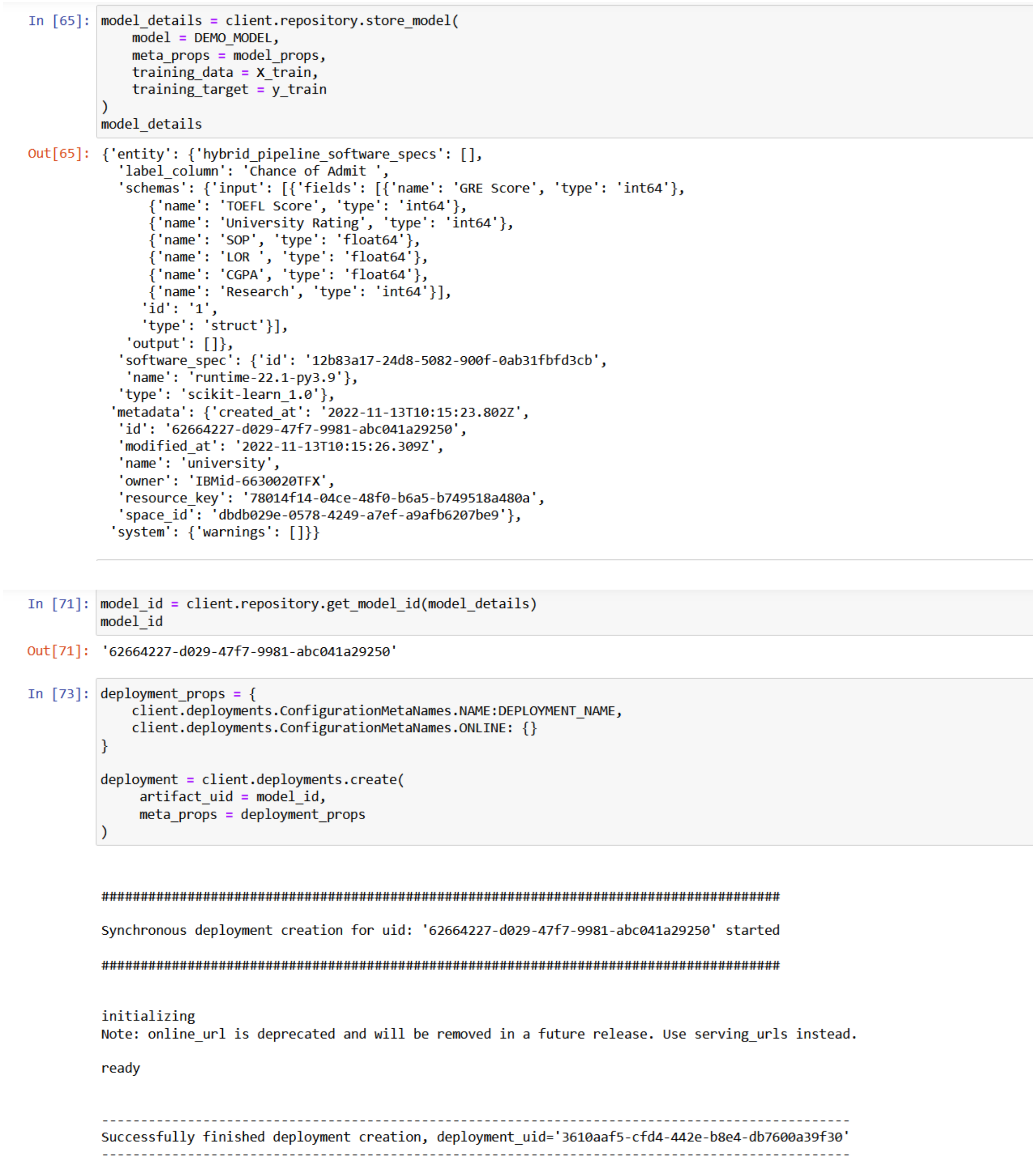
Save and Deploy the model

```
In [55]: import sklearn
         sklearn.__version__

Out[55]: '1.0.2'
```

```
In [57]: MODEL_NAME = 'university'
         DEPLOYMENT_NAME = 'uni'
         DEMO_MODEL = lr
```

```
In [61]: software_spec_uid = client.software_specifications.get_id_by_name('runtime-22.1-py3.9')
         software_spec_uid

Out[61]: '12b83a17-24d8-5082-900f-0ab31fbfd3cb'
```

```
In [59]: model_props = {
             client.repository.ModelMetaNames.NAME: MODEL_NAME,
             client.repository.ModelMetaNames.TYPE: 'scikit-learn_1.0 ',
             client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid
         }
```



```
In [65]: model_details = client.repository.store_model(
             model = DEMO_MODEL,
             meta_props = model_props,
             training_data = X_train,
             training_target = y_train
         )
         model_details
```

```
Out[65]: {'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': '2022-11-13T10:15:23.802Z',
           'id': '62664227-d029-47f7-9981-abc041a29250',
           'modified_at': '2022-11-13T10:15:26.309Z',
           'name': 'university',
           'owner': 'IBMid-6630020TFX',
           'resource_key': '78014f14-04ce-48f0-b6a5-b749518a480a',
           'space_id': 'dbdb029e-0578-4249-a7ef-a9afb6207be9'},
          'system': {'warnings': []}}
```

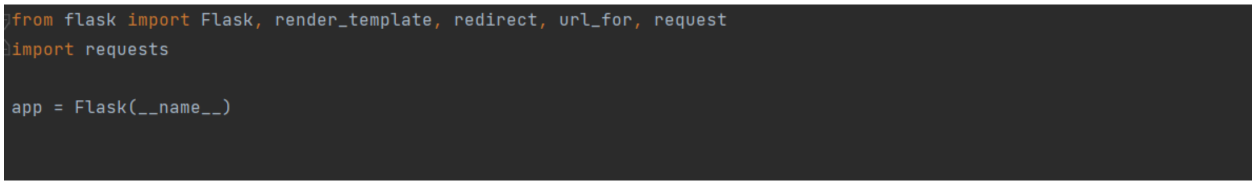
```
In [71]: model_id = client.repository.get_model_id(model_details)
         model_id

Out[71]: '62664227-d029-47f7-9981-abc041a29250'
```

```
In [73]: deployment_props = {
             client.deployments.ConfigurationMetaNames.NAME:DEPLOYMENT_NAME,
             client.deployments.ConfigurationMetaNames.ONLINE: {}
         }

         deployment = client.deployments.create(
              artifact_uid = model_id,
              meta_props = deployment_props
         )
```

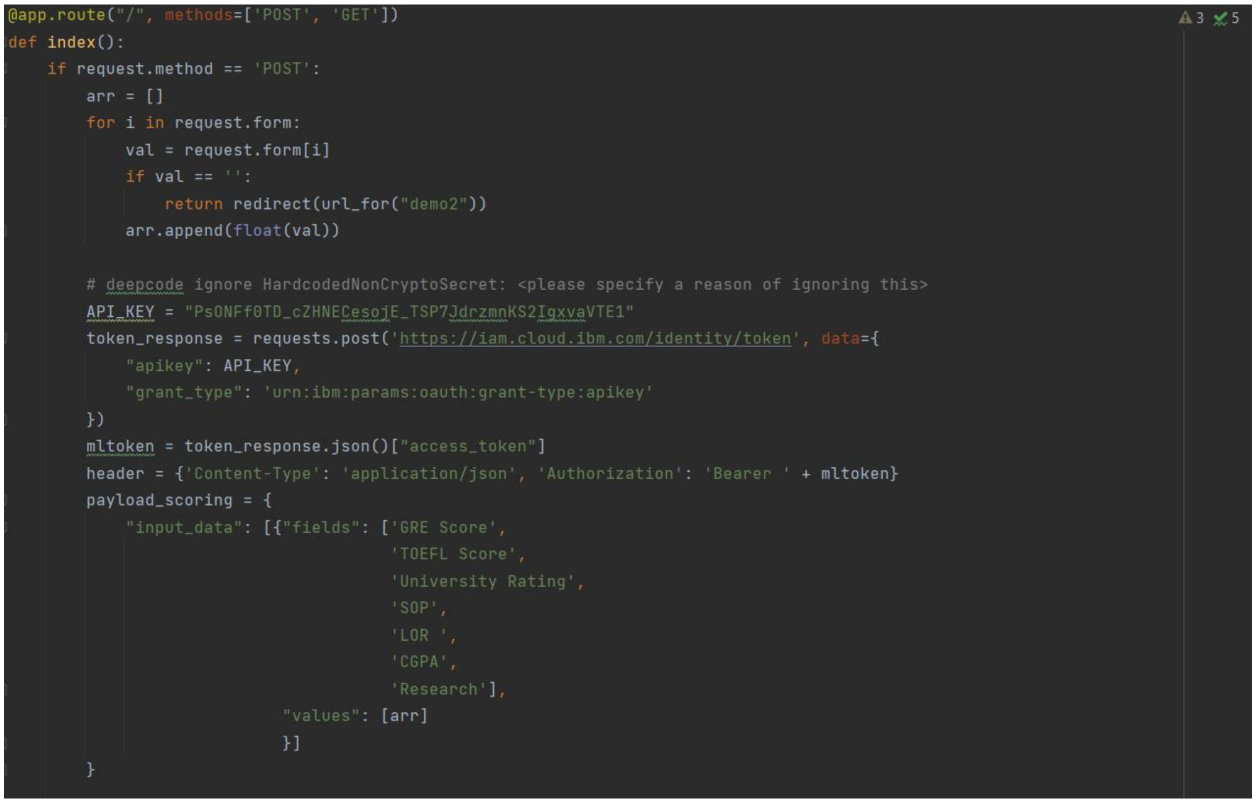
```
#######################################################################################

Synchronous deployment creation for uid: '62664227-d029-47f7-9981-abc041a29250' 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='3610aaf5-cfd4-442e-b8e4-db7600a39f30'
------------------------------------------------------------------------------------------------
```

7.2 Feature 2

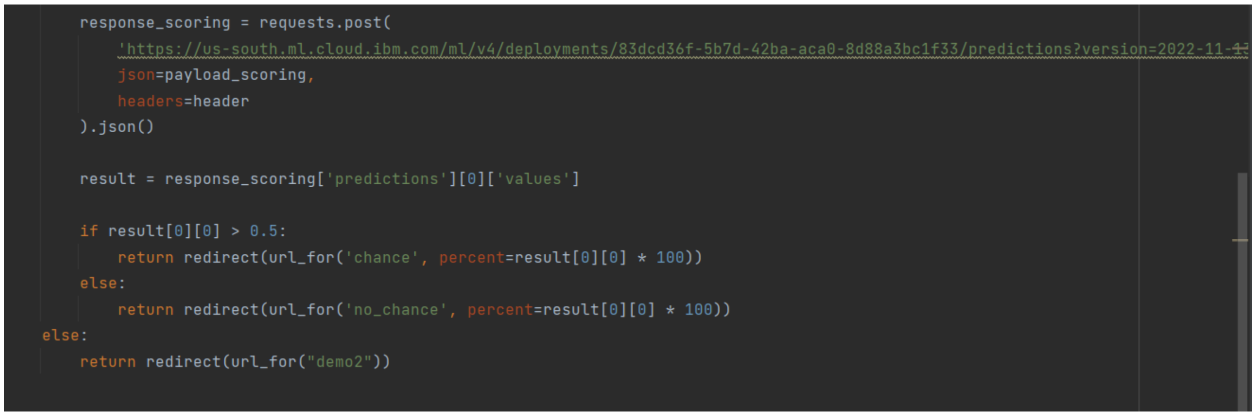
Developed an app.py file with integrated deployment and scoring points of IBM cloud.



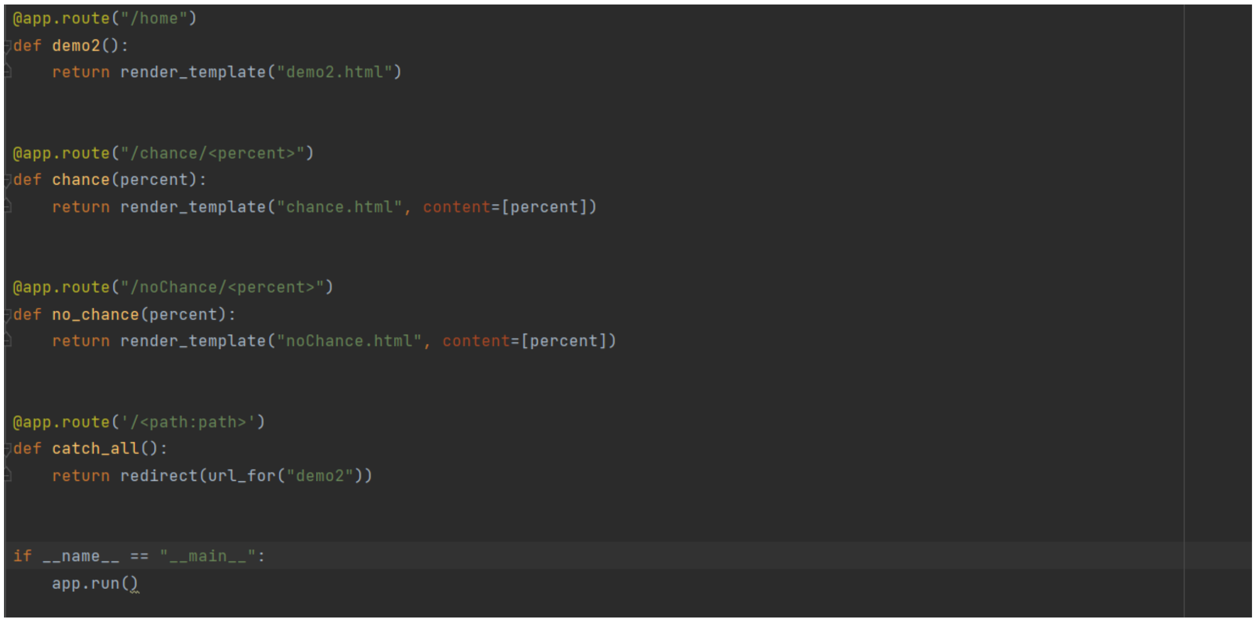
```python
from flask import Flask, render_template, redirect, url_for, request
import requests

app = Flask(__name__)
```



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

        # deepcode ignore HardcodedNonCryptoSecret: <please specify a reason of ignoring this>
        API_KEY = "PsONFf0TD_cZHNECesojE_TSP7JdrzmnKS2IgxvaVTE1"
        token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={
            "apikey": API_KEY,
            "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'
        })
        mltoken = token_response.json()["access_token"]
        header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
        payload_scoring = {
            "input_data": [{"fields": ['GRE Score',
                                       'TOEFL Score',
                                       'University Rating',
                                       'SOP',
                                       'LOR ',
                                       'CGPA',
                                       'Research'],
                            "values": [arr]
                            }]
        }
```



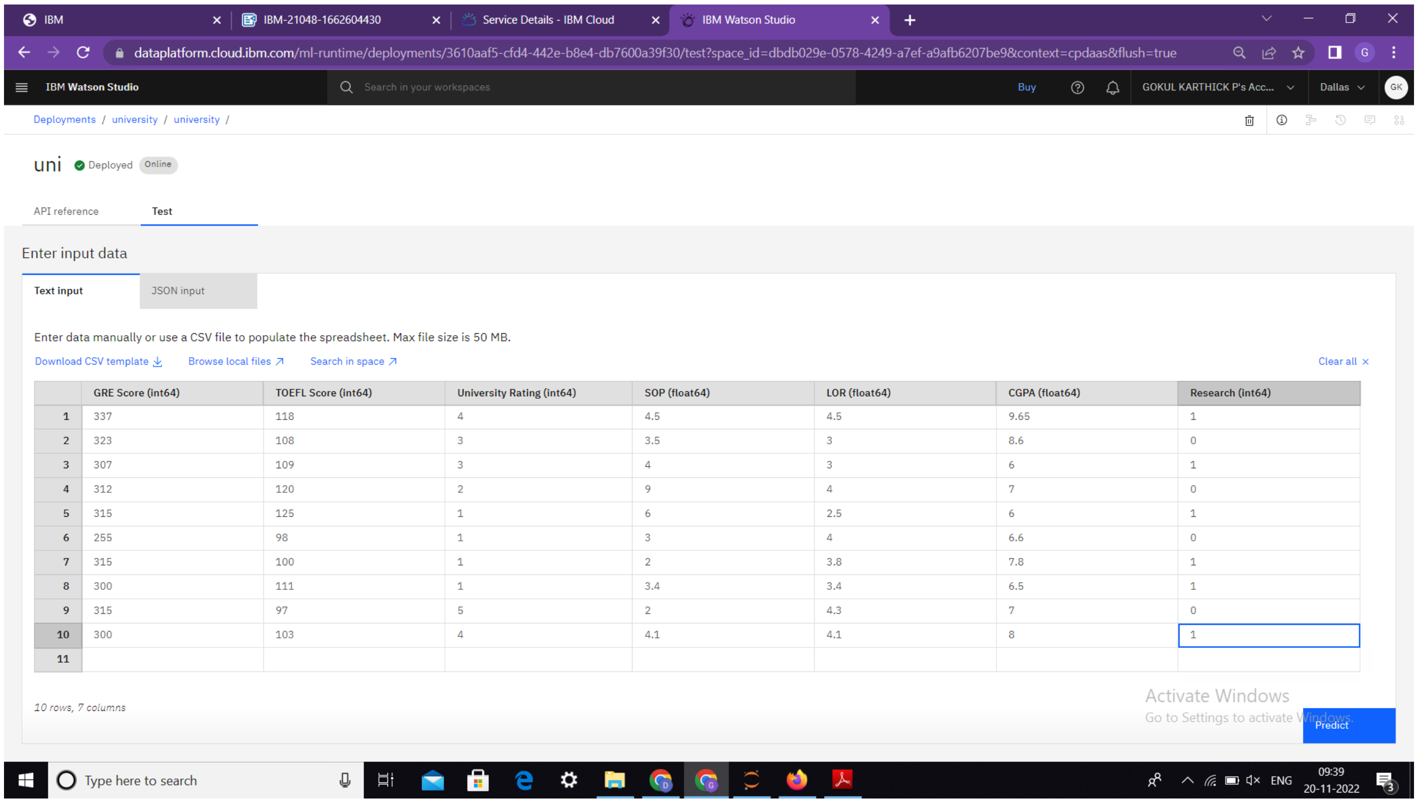
```python
        response_scoring = requests.post(
            'https://us-south.ml.cloud.ibm.com/ml/v4/deployments/83dcd36f-5b7d-42ba-aca0-8d88a3bc1f33/predictions?version=2022-11-1
            json=payload_scoring,
            headers=header
        ).json()

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

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



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


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


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


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


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

Testing

8.1 TestCases



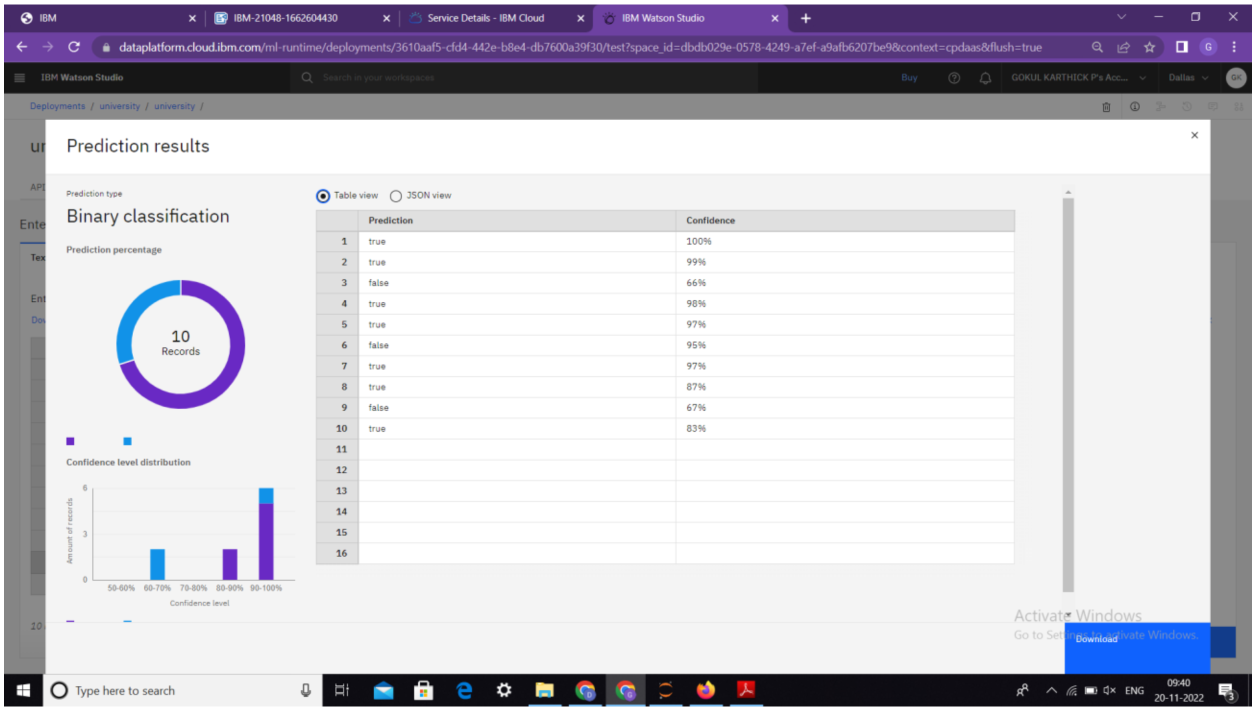
IBM
IBM-21048-1662604430
Service Details - IBM Cloud
IBM Watson Studio
dataplatform.cloud.ibm.com/ml-runtime/deployments/3610aaf5-cfd4-442e-b8e4-db7600a39f30/test?space_id=dbdb029e-0578-4249-a7ef-a9afb6207be9&context=cpdaas&flush=true
IBM Watson Studio
Search in your workspaces
Buy
GOKUL KARTHICK P's Acc...
Dallas
Deployments / university / university /
uni
Deployed
Online
API reference
Test
Enter input data
Text input
JSON input
Download CSV template
Browse local files
Search in space
Clear all
Activate Windows
Go to Settings to activate Windows.
Predict
Type here to search
ENG
09:39
20-11-2022

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	6	1
4	312	120	2	9	4	7	0
5	315	125	1	6	2.5	6	1
6	255	98	1	3	4	6.6	0
7	315	100	1	2	3.8	7.8	1
8	300	111	1	3.4	3.4	6.5	1
9	315	97	5	2	4.3	7	0
10	300	103	4	4.1	4.1	8	1
11							

10 rows, 7 columns

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


IBM Watson Studio
Deployments / university / university /
dataplatform.cloud.ibm.com/ml-runtime/deployments/3610aaf5-cfd4-442e-b8e4-db7600a39f30/test?space_id=dbdb029e-0578-4249-a7ef-a9afb6207be9&context=cpdaas&flush=true
Prediction results
Prediction type
Binary classification
Prediction percentage
10
Records
Confidence level distribution
Amount of records
50-60%
60-70%
70-80%
80-90%
90-100%
Confidence level
Table view
JSON view
Download

	Prediction	Confidence
1	true	100%
2	true	99%
3	false	66%
4	true	98%
5	true	97%
6	false	95%
7	true	97%
8	true	87%
9	false	67%
10	true	83%
11		
12		
13		
14		
15		
16		

8.2 User Acceptance Testing

Purpose of Documentation :

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

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

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

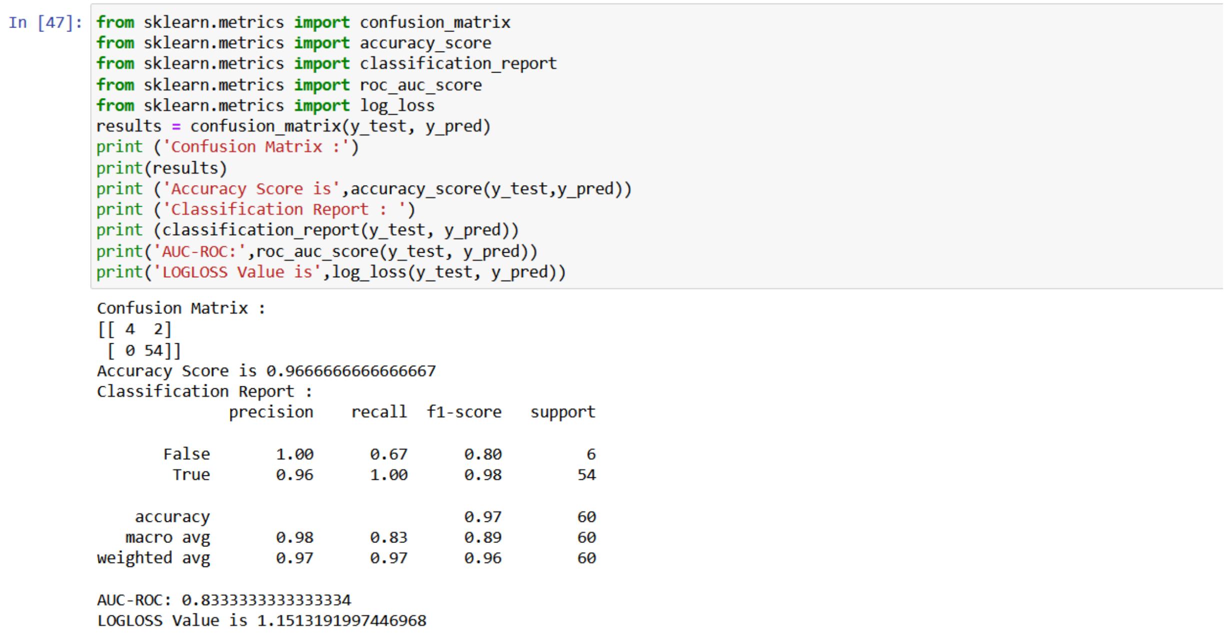
RESULTS

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.



```
In [47]: 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 :
[[ 4  2]
 [ 0 54]]
Accuracy Score is 0.9666666666666667
Classification Report :
              precision    recall  f1-score   support

       False       1.00      0.67      0.80         6
        True       0.96      1.00      0.98        54

    accuracy                           0.97        60
   macro avg       0.98      0.83      0.89        60
weighted avg       0.97      0.97      0.96        60

AUC-ROC: 0.8333333333333334
LOGLOSS Value is 1.1513191997446968
```

ADVANTAGES AND DISADVANTAGES

10.1 Advantages

● It helps students for making decision for choosing a right college.

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

10.2 Disadvantages
●Required active internet connection
●System will provide inaccurate results if data entered incorrectly.

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.

12. FUTURE SCOPE
The future scope of this project is very broad.
Few of them are:
●This can be implemented in less time for proper admission process.
●This can be accessed anytime anywhere, since it is a web application
provided only an internet connection.
●The user had not need to travel a long distance for the admission
and his/her time is also saved as a result of this automated system.

13. APPENDIX
Source Code GitHub & Project Demo Link

Source Code for Flask Application

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

app = Flask(__name__)


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

        # deepcode ignore HardcodedNonCryptoSecret: <please specify a reason of ignoring this>
        API_KEY = "PsONFf0TD_cZHNECesojE_TSP7JdrzmnKS2IgxvaVTE1"
        token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={
            "apikey": API_KEY,
            "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'
        })
        mltoken = token_response.json()["access_token"]
        header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
        payload_scoring = {
            "input_data": [{"fields": ['GRE Score',
                                      'TOEFL Score',
                                      'University Rating',
                                      'SOP',
                                      'LOR ',
                                      'CGPA',
                                      'Research'],
                            "values": [arr]
                            }]
        }

        response_scoring = requests.post(
            'https://us-south.ml.cloud.ibm.com/ml/v4/deployments/83dcd36f-5b7d-42ba-aca0-
8d88a3bc1f33/predictions?version=2022-11-13',
            json=payload_scoring,
            headers=header
        ).json()

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

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



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


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


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


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


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

Front End Code HTML Files
1.Index.html

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

{% block body %}

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

        <div class="card" style="width: 34rem;">
            <img src="..\static\img\yes.jpg" class="card-img-top" alt="...">
            <div class="card-body">
                <h5 class="card-title">Congratulations!! You Have Chance to get admission</h5>
                <a href="/home" class="btn btn-primary">Go Back</a>
            </div>
        </div>
    </div>
</div>

{% endblock %}
```

2.Demo2.html

```
{% extends 'index.html' %}
{% block body %}
    <div class="p-4">
        <div class="row mb-3">
            <div class="col-4">
                <h1 class="text-responsive-h">
                    Enter your Score details and get the chance of your Admission
                </h1>
                <p class="text-responsive">
```

```
                    This Admission Prediction System will help the Students to check their Chance of Admission in Universities
for their respective marks as they got in their final exams
                </p>
                <div class="d-flex justify-content-right">
                    <img src="../static/img/anime.png" class="card-img-top" alt="..." />
                </div>
                </div>
            <div class="col-8">
                <div class="card p-2 ms-2 my-2">
                    <div class="card-body">
                        <h5 class="card-title pb-4">
                            <p style="text-align:center">Enter the Mark details</p>
                        </h5>
                        <form action="/" method="post" id="theForm">
                            <div class="row mb-3">
                                <label for="gre" class="col-lg-2 col-form-label">GRE Score:</label>
                                <div class="col-lg-10">
                                    <input type="number" class="form-control" id="gre" name="gre" min="250" max="340" 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"
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" 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"
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"
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" required>
                                </div>
                            </div>
                            <fieldset class="row mb-3">
                                <legend class="col-form-label col-sm-2 pt-0">Research:</legend>
                                <div class="col-sm-10">
                                    <div class="form-check">
```

```
                                        <input class="form-check-input" type="radio" name="yes_no_radio" id="gridRadios1"
value="1">
                                        <label class="form-check-label" for="yes_no_radio">
                                        Yes
                                        </label>
                                    </div>
                                    <div class="form-check">
                                        <input class="form-check-input" type="radio" name="yes_no_radio" id="gridRadios2"
value="0" checked>
                                        <label class="form-check-label" for="yes_no_radio">
                                        No
                                        </label>
                                    </div>
                                </div>
                            </fieldset>
                            <div class="row lg-3">
                                <div class="col-lg-2 mb-2 me-3">
                                    <button type="submit" class="btn btn-primary" id="button">Predict</button>
                                </div>
                                <div class="col-lg-2" id="spinner">
                                    <div class="spinner-border text-primary m-1" role="status">
                                        <span class="visually-hidden">Loading...</span>
                                    </div>
                                    <div class="spinner-grow text-primary m-1" role="status">
                                        <span class="visually-hidden">Loading...</span>
                                    </div>
                                </div>
                        </form>
                    </div>
                </div>
            </div>
        </div>
    </div>
{% endblock %}

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
    <link rel="stylesheet" type="text/css" rel="noopener" target="_blank" href="../static/css/styles.css">
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" crossorigin="anonymous">
    <script type="text/javascript" src="../static/js/script.js" async></script>
    <title>University Admit Eligibility Predictor</title>
</head>
<body>
    <nav class="navbar navbar-expand-lg bg-light" >
        <div class="container-fluid">
            <a class="navbar-brand text-responsive-h" href="/">
                <img src="..\static\img\hat1.png" alt="Logo" width="30" height="24" class="d-inline-block align-text-top ">
                <marquee>University Admission Eligibility Prediction System</marquee>
            </a>
        </div>
    </nav>
    {% block body %}
    <h1> Index Page </h1>
    {% endblock %}
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-
OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3" crossorigin="anonymous"></script>
</body>
</html>
```

3.chance.html

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

<div class="container text-center p-4">
    <div class="d-flex justify-content-center">
        <div class="card" style="width: 34rem;">
            <img src="..\static\img\no.jpg" class="card-img-top" alt="...">
            <div class="card-body">
                <h5 class="card-title">OOPS!! You have a LOW / NO chance to get admission </h5>
                <p class="card-text">The model has predicted that you have no chance</p>
                <a href="/home" class="btn btn-primary">Go Back</a>
            </div>
        </div>
    </div>
</div>

{% endblock %}
```

4.nochance.html

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

<div class="container text-center p-4">
    <div class="d-flex justify-content-center">
        <div class="card" style="width: 34rem;">
            <img src="..\static\img\no.jpg" class="card-img-top" alt="...">
            <div class="card-body">
                <h5 class="card-title">OOPS!! You have a LOW / NO chance to get admission </h5>
                <p class="card-text">The model has predicted that you have no chance</p>
                <a href="/home" class="btn btn-primary">Go Back</a>
            </div>
        </div>
    </div>
</div>

{% endblock %}
```

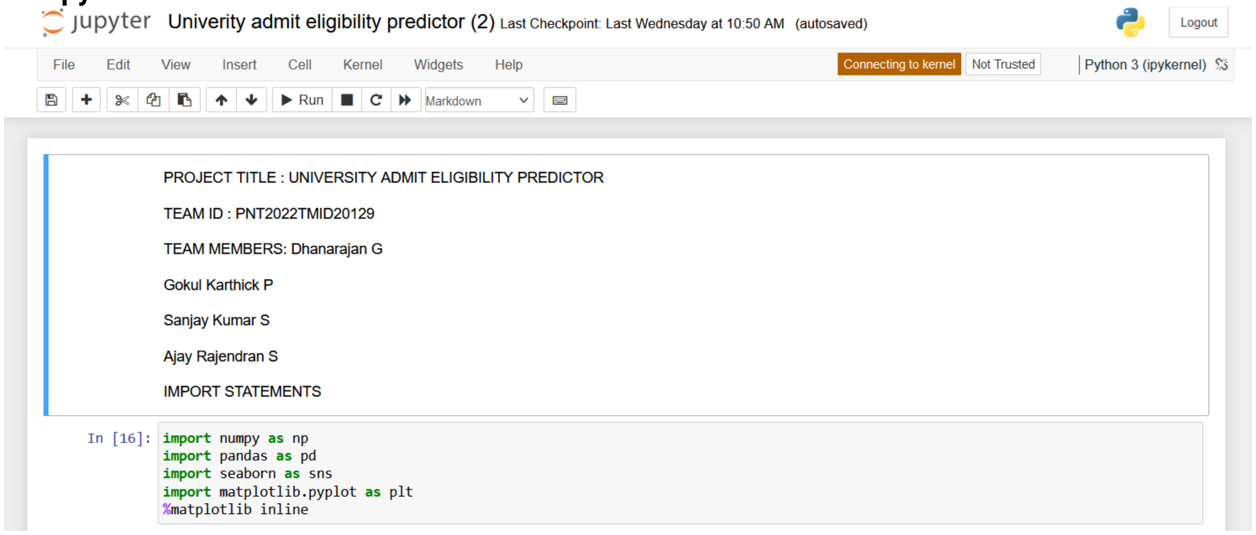
5.Styles.css

```
* {
    margin: 0;
    padding: 0;
    border: 0;
}
body {
    font: 62.5%/1.5  "Lucida Grande", "Lucida Sans", Tahoma, Verdana, sans-serif;
    background: #e0eafc;
    background: -webkit-linear-gradient(to right, #e0eafc, #cfdef3);
    background: linear-gradient(to right, #e0eafc, #cfdef3);
    color: #000000;
    text-align:center;
}

h1 {
    font-size: 2.2em;
}

h2 {
    font-size: 2.0em;
}

h4 {
    font-size: 1.6em;
}

p {
    font-size: 1.2em;
}



input.text
  {
  padding: 3px;
  border: 1px solid #999999;
}
.p-4{
background-color:#ffa07a;
}
img {
    max-width: auto;
    height: auto;
}

.text-responsive {
    font-size: calc(50% + 0.6vw + 0.6vh);
}

.text-responsive-h {
    font-size: calc(80% + 0.6vw + 0.6vh);
}
Footer
```

6.script.js

```
const button = document.getElementById('button');
const theForm = document.getElementById('theForm');
const loading = document.getElementById('spinner');

const disableButton = () => {
    console.log('Submitting form...');
    button.disabled = true;
    button.className = "btn btn-outline-primary";
    button.innerHTML = "Checking..."
    loading.style.display = "block"
};

const enableButton = () => {
    console.log('Loading window...');
    button.disabled = false;
    button.className = "btn btn-primary"
    button.innerHTML = "Check your chance"
    loading.style.display = "none"
}

theForm.onsubmit = disableButton;

window.onload = enableButton;
```

6 Jupyter Notebook



jupyter
Univerity admit eligibility predictor (2) Last Checkpoint: Last Wednesday at 10:50 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Connecting to kernel
Not Trusted
Python 3 (ipykernel)
Run
Markdown

PROJECT TITLE : UNIVERSITY ADMIT ELIGIBILITY PREDICTOR

TEAM ID : PNT2022TMID20129

TEAM MEMBERS: Dhanarajan G

Gokul Karthick P

Sanjay Kumar S

Ajay Rajendran S

IMPORT STATEMENTS

```
In [16]: import numpy as np
         import pandas as pd
         import seaborn as sns
         import matplotlib.pyplot as plt
         %matplotlib inline
```

Load the data set

In [18]:
```
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='7b1tUYzAq1AHKVEperhd9w-HV_fUQ34dEb3gPKu0jLpi',
    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-jw2tqhsy1cmutv'
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[18]:

	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 [19]:
```
data.drop(["Serial No."], axis=1, inplace=True)
```

In [20]:
```
data.describe()
```

Out[20]:

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
count	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000	400.000000
mean	316.807500	107.410000	3.087500	3.400000	3.452500	8.598925	0.547500	0.724350
std	11.473646	6.069514	1.143728	1.006869	0.898478	0.596317	0.498362	0.142609
min	290.000000	92.000000	1.000000	1.000000	1.000000	6.800000	0.000000	0.340000
25%	308.000000	103.000000	2.000000	2.500000	3.000000	8.170000	0.000000	0.640000
50%	317.000000	107.000000	3.000000	3.500000	3.500000	8.610000	1.000000	0.730000
75%	325.000000	112.000000	4.000000	4.000000	4.000000	9.062500	1.000000	0.830000
max	340.000000	120.000000	5.000000	5.000000	5.000000	9.920000	1.000000	0.970000

In [21]:
```
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
```

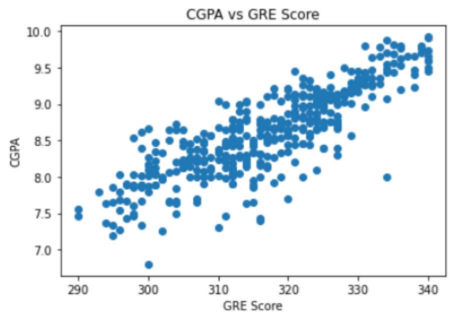
Handling Missing Values

In [22]:
```
data.isnull().sum()
```

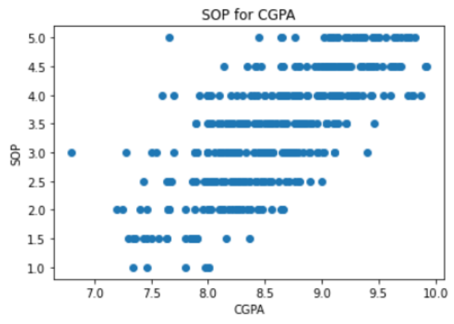
Out[22]:
```
GRE Score            0
TOEFL Score          0
University Rating    0
SOP                  0
LOR                  0
CGPA                 0
Research             0
Chance of Admit      0
dtype: int64
```

Visualization

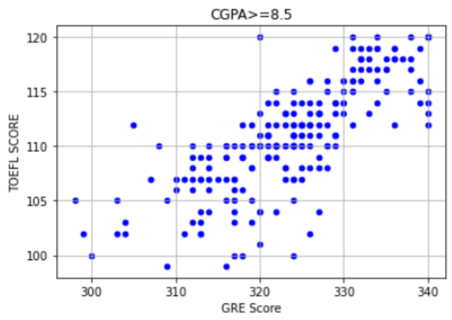
In [23]:
```
plt.scatter(data['GRE Score'],data['CGPA'])
plt.title('CGPA vs GRE Score')
plt.xlabel('GRE Score')
plt.ylabel('CGPA')
plt.show()
```



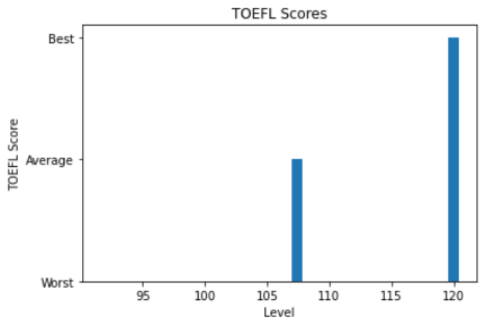
In [24]:
```
plt.scatter(data['CGPA'],data['SOP'])
plt.title('SOP for CGPA')
plt.xlabel('CGPA')
plt.ylabel('SOP')
plt.show()
```



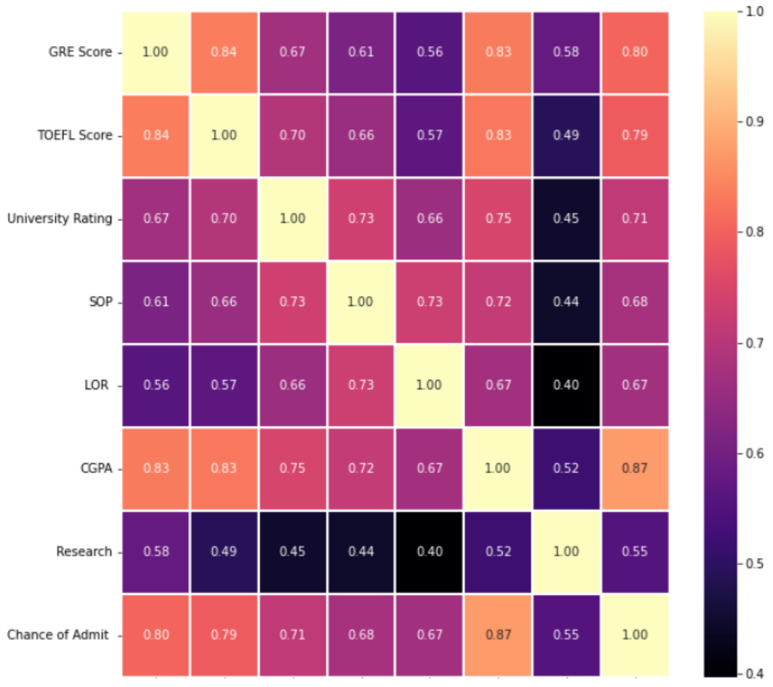
In [25]:
```
data[data.CGPA >= 8.5].plot(kind='scatter', x='GRE Score', y='TOEFL Score',color="BLUE")

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

plt.show()
```



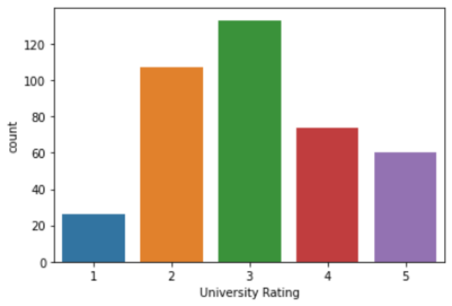
In [26]:
```
p = np.array([data["TOEFL Score"].min(),data["TOEFL Score"].mean(),data["TOEFL Score"].max()])
r = ["Worst","Average","Best"]
plt.bar(p,r)

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

plt.show()
```



```
In [12]: plt.figure(figsize=(10, 10))

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

         plt.show()
```



```
In [28]: data.Research.value_counts()

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

Out[28]: <AxesSubplot:xlabel='University Rating', ylabel='count'>
```



Trainning and test split

```
In [29]: X=data.drop(['Chance of Admit '],axis=1) #input data_set
         y=data['Chance of Admit '] #output labels
```

```
In [30]: 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)
```

MODELING AND TRAINING

In [31]:
```
from sklearn.ensemble import GradientBoostingRegressor
rgr = GradientBoostingRegressor()
rgr.fit(X_train,y_train)
```

Out[31]:
```
GradientBoostingRegressor()
```

In [32]:
```
rgr.score(X_test,y_test)
```

Out[32]:
```
0.7345575572947072
```

In [33]:
```
y_predict=rgr.predict(X_test)
```

In [34]:
```
from sklearn.metrics import mean_squared_error, r2_score,mean_absolute_error
import numpy as np
print('Mean Absolute Error:', mean_absolute_error(y_test, y_predict))
print('Mean Squared Error:', mean_squared_error(y_test, y_predict))
print('Root Mean Squared Error:', np.sqrt(mean_squared_error(y_test, y_predict)))
```

```
Mean Absolute Error: 0.048898416300819064
Mean Squared Error: 0.00443403167036226
Root Mean Squared Error: 0.06658852506522622
```

In [35]:
```
y_train = (y_train>0.5)
y_test = (y_test>0.5)
```

In [36]:
```
from sklearn.linear_model._logistic import LogisticRegression

lore = LogisticRegression(random_state=0, max_iter=1000)

lr = lore.fit(X_train, y_train)
```

In [38]:
```
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: 0.9333333333333333
Recall Score: 1.0
ROC AUC Score: 0.6666666666666667
Confussion Matrix:
 [[ 2  4]
 [ 0 54]]
```

SAVING THE MODEL

In [39]:
```
import pickle
```

In [40]:
```
pickle.dump(lr, open("university.pkl", "wb")) #logistic regression model
```

IBM Deployment

```
In [1]: 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
(1.0.257)
Requirement already satisfied: ibm-cos-sdk==2.11.* in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ib
m-watson-machine-learning) (2.11.0)
Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from
ibm-watson-machine-learning) (1.3.4)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-mac
hine-learning) (2022.9.24)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-ma
chine-learning) (2.26.0)
Requirement already satisfied: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-m
achine-learning) (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: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-mac
hine-learning) (1.26.7)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-ma
chine-learning) (0.8.9)
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-mach
ine-learning) (0.3.3)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (fr
om ibm-cos-sdk==2.11.*->ibm-watson-machine-learning) (2.11.0)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from
ibm-cos-sdk==2.11.*->ibm-watson-machine-learning) (0.10.0)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packag
es (from ibm-cos-sdk==2.11.*->ibm-watson-machine-learning) (2.11.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages
(from ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.*->ibm-watson-machine-learning) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas<1.
5.0,>=0.24.2->ibm-watson-machine-learning) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas
<1.5.0,>=0.24.2->ibm-watson-machine-learning) (1.20.3)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateut
il<3.0.0,>=2.1->ibm-cos-sdk-core==2.11.0->ibm-cos-sdk==2.11.*->ibm-watson-machine-learning) (1.15.0)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (f
rom requests->ibm-watson-machine-learning) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests-
>ibm-watson-machine-learning) (3.3)
Requirement already satisfied: zipp>=0.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from importlib-me
tadata->ibm-watson-machine-learning) (3.6.0)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (fr
om packaging->ibm-watson-machine-learning) (3.0.4)
Note: you may need to restart the kernel to use updated packages.
```

Authenticate and set space

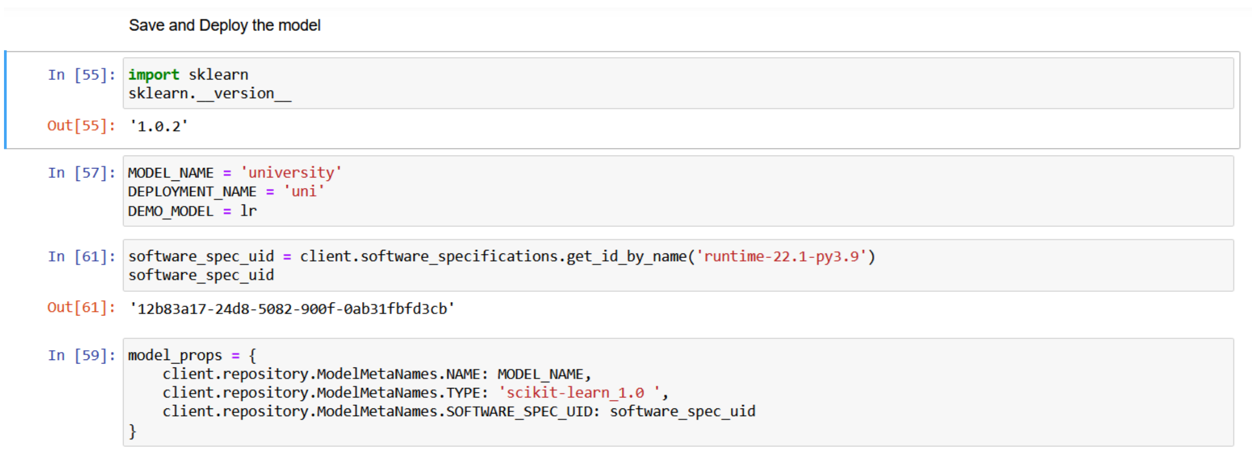
```
In [2]: from ibm_watson_machine_learning import APIClient
        import json
```

```
In [49]: uml_credentials = {
             "url": "https://us-south.ml.cloud.ibm.com",
             "apikey": "PsONFf0TD_cZHNECesojE_TSP7JdrzmnKS2IgxvaVTE1"
         }

         client = APIClient(uml_credentials)
```

```
In [50]: def guid_from_space_name(client, space_name):
             space = client.spaces.get_details()
             idr = []
             for i in space['resources']:
                 idr.append(i['metadata']['id'])
             return idr
```

```
In [68]: space_uid = guid_from_space_name(client, "university")
         print("Space Id:",space_uid[0])

         Space Id: dbdb029e-0578-4249-a7ef-a9afb6207be9
```

```
In [70]: client.software_specifications.list()

         ------------------------------  ------------------------------------  ----
         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
         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
         ------------------------------  ------------------------------------  ----
```



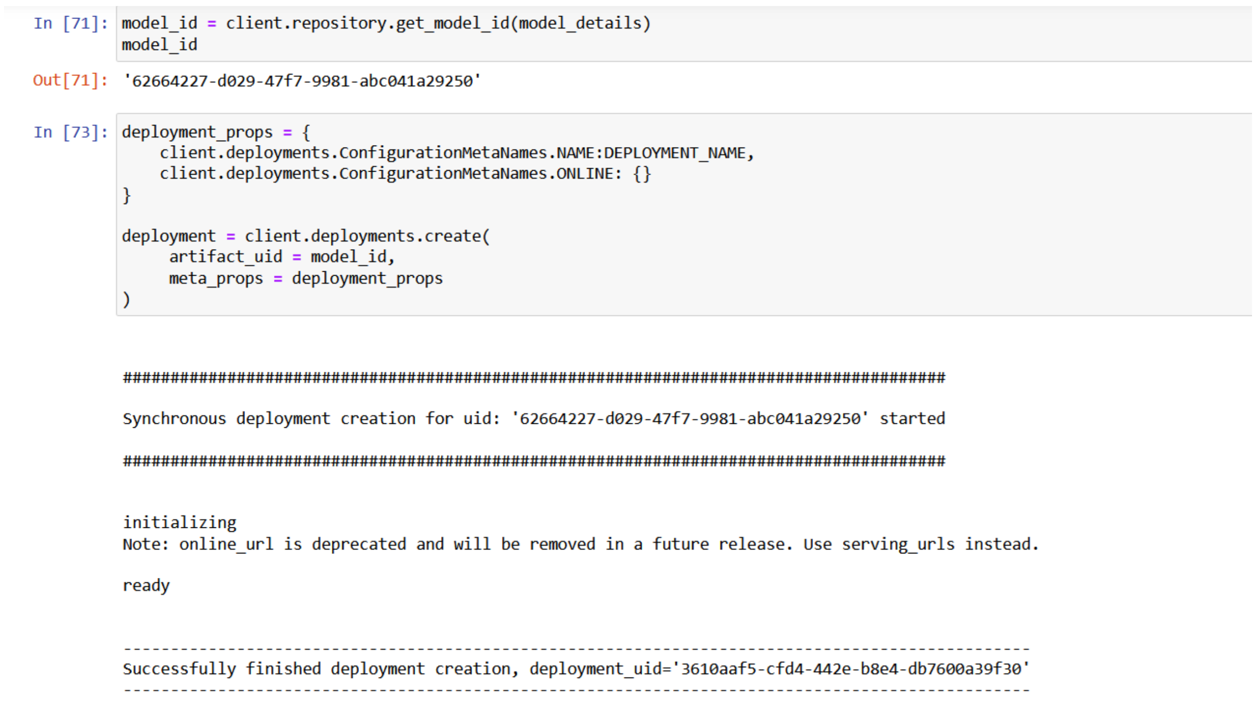
Save and Deploy the model

```
In [55]: import sklearn
         sklearn.__version__

Out[55]: '1.0.2'
```

```
In [57]: MODEL_NAME = 'university'
         DEPLOYMENT_NAME = 'uni'
         DEMO_MODEL = lr
```

```
In [61]: software_spec_uid = client.software_specifications.get_id_by_name('runtime-22.1-py3.9')
         software_spec_uid

Out[61]: '12b83a17-24d8-5082-900f-0ab31fbfd3cb'
```

```
In [59]: model_props = {
             client.repository.ModelMetaNames.NAME: MODEL_NAME,
             client.repository.ModelMetaNames.TYPE: 'scikit-learn_1.0 ',
             client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid
         }
```



```
In [71]: model_id = client.repository.get_model_id(model_details)
         model_id

Out[71]: '62664227-d029-47f7-9981-abc041a29250'
```

```
In [73]: deployment_props = {
             client.deployments.ConfigurationMetaNames.NAME:DEPLOYMENT_NAME,
             client.deployments.ConfigurationMetaNames.ONLINE: {}
         }

         deployment = client.deployments.create(
              artifact_uid = model_id,
              meta_props = deployment_props
         )
```

```


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

Synchronous deployment creation for uid: '62664227-d029-47f7-9981-abc041a29250' 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='3610aaf5-cfd4-442e-b8e4-db7600a39f30'
------------------------------------------------------------------------------------------------
```

Project Demo Link:
https://drive.google.com/file/d/10Nr5LMu5uA5kKIKTS8l4C-koUeE6heCt/view?usp=share_link
Github Link:
https://github.com/IBM-EPBL/IBM-Project-21048-1659771001