**ANNAI COLLEGE OF ENGINEERING AND TECHNOLOGY**

(8205)

**IBM - NALAYA THIRAN PROJECT**

### **University Admit Eligibility Predictor**

**submited by**

**TEAM ID : PNT2022TMID52352**

**TEAM SIZE : 4**

**TEAM LEADER: AKASH P**

**TEAM MEMBER: ABBAS S**

**TEAM MEMBER: AFEESGAN R**

**TEAM MEMBER: HARIHARAN K P**

**INDUSTRY MENTOR FACULTY MENTOR**

**Sandesh P NADHIYA BANU R**

**INDEX**

S No. TOPIC

**PROBLEM STATEMENT**

**1. INTRODUCTION**

1.1. Purpose of Document

1.2. Project Summary

1.3. Motivation

1.4. Significance

1.5. Project Scope

1.6. System Purpose

1.5.1 Users

1.5.2 Location

1.5.3 Responsibilities

1.5.4 Need

1.7. Limitations and Future Work

1.8. Beneficiaries

1.9. Overview of Document

**2. FUNCTIONAL OBJECTIVES**

2.1. High Priority

2.2. Medium Priority

2.3. Low Priority

**3. NON-FUNCTIONAL OBJECTIVES**

3.1 Reliability

3.2 Usability

3.3 Performance

3.4 Security

3.5 Supportability

3.6 Online user Documentation and Help

3.7 Purchased Components

3.8 Interfaces

**4. THE CONTEXT MODEL**

4.1 Goal Statement

4.2 System Externals

**5. THE USE CASE MODEL**

5.1 System Use Case Diagram

5.2 Use Case Descriptions (for selected cases)

**6. THE CLASS MODE**L

6.1 System Class Diagram

6.2 System Object Diagram

.**7. THE ENTITY RELATIONSHIP MODEL**

**8. METHODOLOGY FOR DEVELOPING SOFTWARE**

**9. SOFTWARE AND HARDWARE TECHNOLOGIES USED**

9.1 Hardware

9.2 Software

**10.CODING & SOLUTIONING**

**11.TESTING**

**12.ADVANDAGE & DISADVANDAGES**

**13.CONCLUTION**

**14.FUTURE SCOPE**

**15.APPENDIX**

**PROBLEM STATEMENT:**

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.

**1. INTRODUCTION**

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.

**1.2. Project Summary**

### Project Name: University Admit Eligibility Predictor

### Project type: Web Application

Team member: Akash P , Abbas S , Afeesgan R , Hariharan K P

Languages used: Python, HTML, Javascript, CSS

Database: Mysql

Development Platform : Jupitar

**1.3. Motivation**

In the current world scenario, it is not enough for a student to just have an Under Graduate

degree. Most employers now look for higher qualifications in their new recruits. As a result,

the demands for a good higher education are at an all time high. A lot of students from India

prefer to continue their higher education with foreign universities, especially in the United

States.

In order to get admitted to these foreign universities, a set of academic requirements are

needed. However, because of the sheer number of universities of different levels, students are

often stuck in a dilemma till the very last minute as to whether or not their applications will

be accepted or not as no concrete documentation is available which lists the requirements.

**1.4. Significance**

We at University Admit Eligibility Predictor are here to provide a solution to that problem. Not only do we provide a

single platform that documents all the requirements as well as the different tiers of

universities, but our website also incorporates an AI Model that was built after considering

many leading Machine Learning Algorithms, to provide the most accurate prediction of how

much of a chance of admissions does a student’s current grades and other academic

transcripts allow them in the tier of universities of their choice.

**1.5. Project Scope**

The scope of this project is a web application that allows users to enter their academic data

and get predictions of their chances of admissions in the university tier of their choosing. It

also provides them answers to the most common FAQ’s that arise when thinking of

admissions abroad for Post Graduate studies.

It also provides an analysis based on the data set used that shows how the different

parameters affect chances of admissions.

A Database will also be implemented for the system so that students can save their data and

review and edit it as they progress with the most recent predictions being saved with their

profile.

Issues of web security other than password protection within the website are not part of this

project.

**1.6. System Purpose**

**1.6.1. Users**

Students- The people who will benefit the most from using this system are Indian students.

Especially students looking to pursue their higher education from foreign universities,

particularly in the United States.

Administrators- The administrator shall be able to access all the data stored in the application.

**1.6.2. Locations**

The system will be available to all users from any location as long as they have an Internet

connection. The administrator can also access the website from any location as long as he has

the correct login credentials and access to the Internet.

**1.6.3. Responsibilities**

The primary responsibilities of the system are:

• Provide customers access to the prediction model

• Provide answers to most common FAQs regarding PG Admissions abroad

• Provide administrator access to all records

• Provide analysis of how the various academic factors affect university admission

Other desired features of the system:

• Maintaining a profile for each user

• Password protection for each account.

**1.6.4. Need**

This system is needed so as to answer the queries of students in a compete and concise

manner as well as to provide them an as accurate as possible analysis of their chances of

admissions to their dream universities.

**1.7. Limitations and Future Work**

The system is built on a limited data set, this could affect the accuracy of the predictions as a

whole.

The system cannot guarantee that our predictions will be a 100% guarantee of admissions

because a lot other factors such as the Personal Interview also plays a major role in the

admissions procedure.

Other factors such as changes in policies by the university or by the country can also affect

chances of admissions in a way that is beyond the scope of this project.

Admissions also depend on the individual university’s policy regarding the intake of foreign

students and is not modelled by our system.

Future work in the project could include weighing in the features that have been ignored as of

yet like percentage seats for Foreign Students. Other criterions like Co-curricular

achievements, Leadership positions held, job experience etc can also be included as metrics

for the model.

**1.8. Beneficiaries**

The people who will benefit the most from using this system are Indian students. Especially

students looking to pursue their higher education from foreign universities, particularly in the

United States.

**1.9. Overview of Document**

The rest of this document gives the detailed specifications for the new sales system. It is

organized as follows:

• Section 2: Functional Objectives

Each objective gives a desired behaviour for the system and a measure to determine if the

final system has successfully met the objective. These objectives are organized by priority. In

order for the new system to be considered successful, all high priority objectives must be met.

• Section 3: Non-Functional Objectives

This section is organized by category. Each objective specifies a technical requirement or

constraint on the overall characteristics of the system. Each objective is measurable.

• Section 4: Context Model

This section gives a text description of the goal of the system, and a pictorial description of

the scope of the system in a context diagram. Those entities outside the system that interact

with the system are described.

• Section 5: Use Case Model

The specific behavioural requirements of the system are detailed in a series of use cases.

Each use case accomplishes a business task and shows the interaction between the system and some outside actor. The system use case diagram depicts the interactions between all use

cases and system actors. A sequence diagram is also provided to show the flow of the

application.

• Section 6: Class Model

A class is a collection of objects in the system that have the same data and behaviour. All

analysis classes and their relationships are shown on the class diagram. An Object Diagram is

also provided to further explain the relations.

• Section 7: Entity Relationship Model

An entity relationship model is provided to showcase how the database will be modelled.

• Section 8: Methodology for developing system

The RAD Model of the SDLC Project Development Methodology was used for developing

this project. The system explains the RAD model and the reasons it was chosen.

• Section 9: Software and Hardware used

This section details the Hardware, Software and Online tools used for developing the

University Admit Eligibility Predictor project.

• Section 10: Code Snippets

This section provides some of the major code snippets used to develop the project.

• Section 11: Testing Report

This section details the different test cases needed for the project and how they were

implemented.

Section 12: Validation.

This Section contains a final validation of the system where the system is checked to see if all the promised functional and non-functional requirements as mentioned in the SRS have been fulfilled or not.

**2. FUNCTIONAL OBJECTIVES**

**2.1. High Priority**

1. The system shall provide the user access to the AI predictor, wherein the user will be able to

fill in a form with their academic transcripts data (GRE score, TOEFL Score, CGPA, SOP

Score, LOR Score, Research experience), choose the tier of university they wish to apply to

(1-5(top level)) and then get a prediction of their chances of admissions to that level

university based on the mapping between their requirements and the student’s results.

2. The system shall provide the administrator access to all the records in the database on a

“read-only” basis.

**2.2. Medium Priority**

1. The system shall provide all users with answers to the most common FAQ’s like –

“Distribution of University Tiers”, “University Admissions Criteria”

2. The system shall allow the user’s details to be stored for the next time they return to the

website. If the user chooses to take a new evaluation, the most recent inputs as well as

prediction shall replace any previous data.

**2.3. Low Priority**

The system shall provide users an analysis of how the various factors mentioned in the form

affect their chances of admissions as well as what is the general trend of applications to the

various tiers of universities.

**3. NON-FUNCTIONAL REQUIREMENTS**

**3.1. Reliability**

• The system shall be completely operational all hours of the day unless system failure or

upgradation work is to be performed

• Down time after a failure shall not exceed 24 hours

**3.2. Usability**

• No training is required to use the website.

• The form, home, about, FAQ and analysis pages load up within 10 seconds.

• The results from the predictor should not take more than 30 seconds.

**3.3. Performance**

• The system can support any number of users at a time.

• The mean time to view a web page over a 56Kbps modem connection shall not exceed 5

seconds.

**3.4. Security**

• The system shall provide password protected access to the website to all users – students and

admins both

**3.5. Supportability**

• The system will be able to incorporate more features without major reengineering.

• The system web site shall be viewable from Internet Explorer 4.0 or later, Netscape

Navigator/Communicator 3.0 or later and the America Online web browser version 3.0 or

later.

**3.6. Online user documentation and help**

• The system shall provide a web page that explains how to navigate the site. This page should

be customized based on what pages that user is allowed to access.

• This help page should be accessible from all other pages.

**3.7. Purchased Components**

• No purchased components

3.8. Interfaces

The system must interface with the MongoDB database and the web search engine it will be

launched on.

**4. THE CONTEXT MODEL**

**4.1. Goal Statement**

The goal of the system is to provide help to students who are looking for PG Abroad. The

system proposes to achieve this by –

• Providing answers to the most commonly asked questions regarding university admissions

for PG Studies Abroad.

• Providing an as accurate as possible prediction for the student’s chances of admissions to the

universities of their choice based on their academic transcripts.

• Providing an analysis conducted over our dataset to the user in order to help them understand

the weightage of various academic data values on their chances of admissions.

**4.2. System Externals**

Students

A student is a user who has access to the system as well as to the university

admissions predictor. He/ She may access all web pages and the model but can only

access his/her own record data.

Administrator

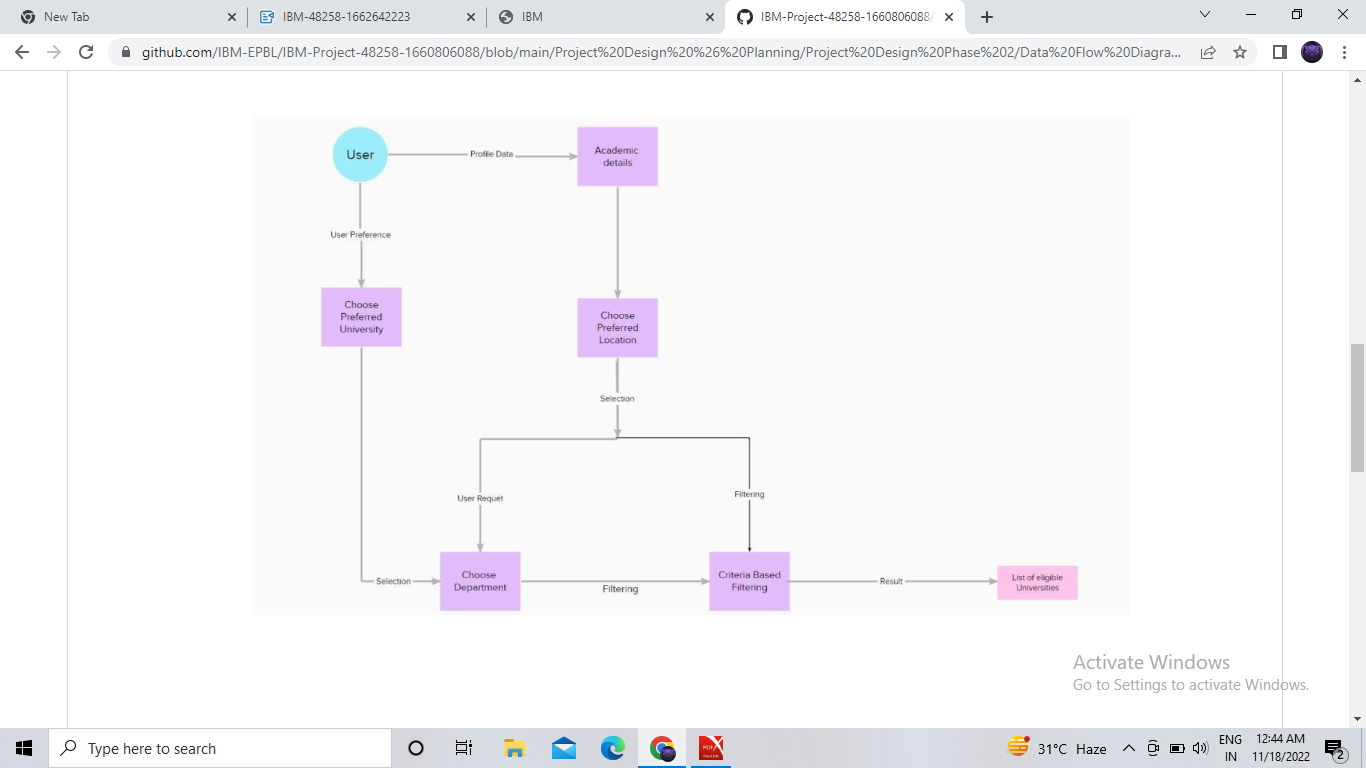
An administrator is a user that has access to all the web pages except for the predictor

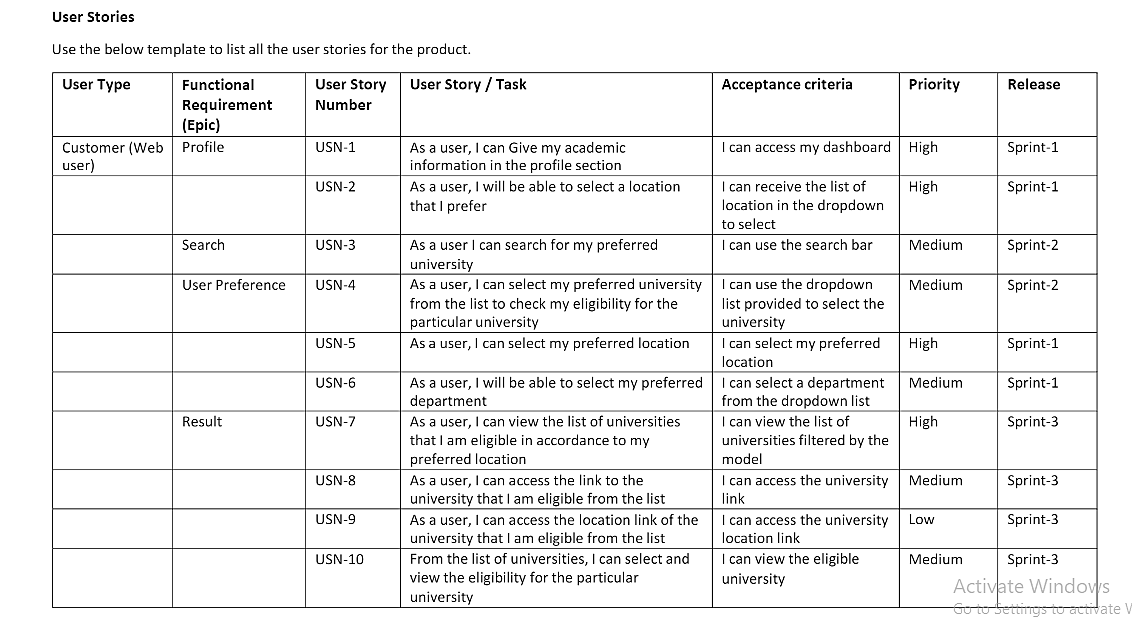
as he/she has no use of it. The administrator can access the data pertaining to all users

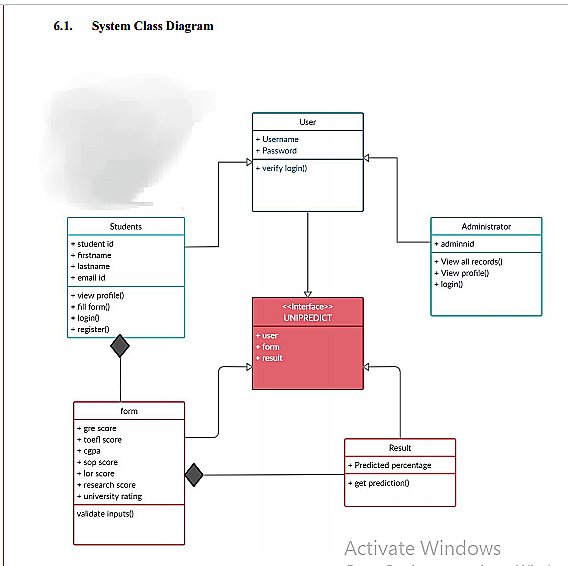
of the system.

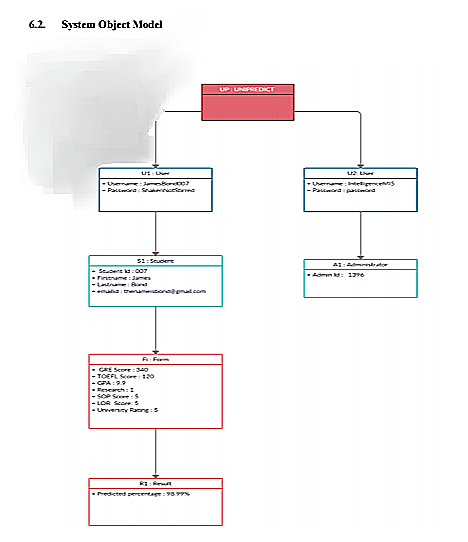
**5. THE USE CASE MODEL**

**5.1. System Use Case Diagram**

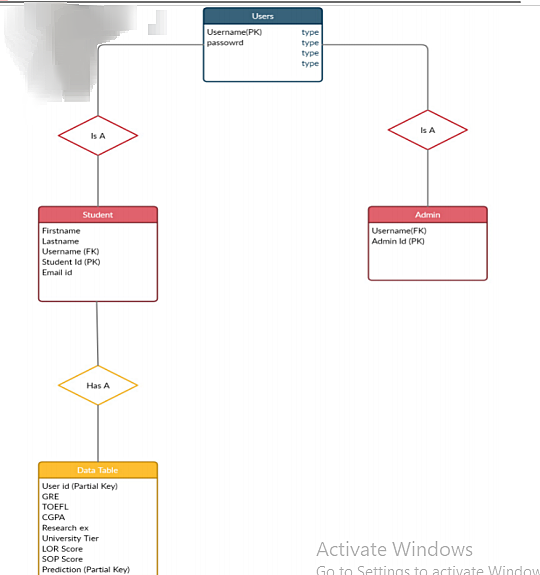


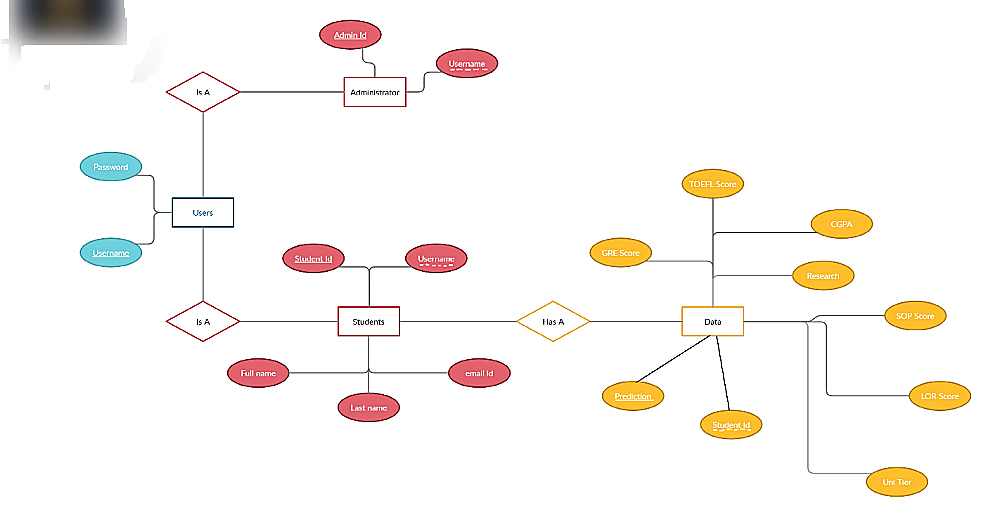






**7. ER Diagram**





**8. METHODOLOGY FOR DEVELOPING THE SYSTEM**

In order to develop the system, the Software Development Life Cycle (SDLC) was followed.

Of all the different models available under SDLC, the methodology selected was the Rapid

Application Development (RAD) methodology.

**Models considered**: Waterfall Model, Prototype Model, Agile Model and RAD Model

**Model Selected**: RAD Model

RAD model is Rapid Application Development model. It is a type of incremental model. In

RAD model the components or functions are developed in parallel as if they were mini

projects. The developments are time boxed, delivered and then assembled into a working

prototype.

The project was built using the RAD Model (Rapid Application Development Model).

While the project is not too big and does not require a lot of the features of using the RAD

Model like the powerful auto code generators and other RAD tools, the benefits of using the

RAD Model far outweighed those of any other model considered.

Rapid application development is a software development methodology that uses minimal

planning in favour of rapid prototyping. A prototype is a working model that is functionally

equivalent to a component of the product.

RAD projects follow iterative and incremental model and have small teams comprising of

developers, domain experts, customer representatives and other IT resources working

progressively on their component or prototype.

The most important aspect for this model to be successful is to make sure that the prototypes

developed are reusable.

Reasons for choosing the RAD Model –

1. The project is modular –The project could be divided into smaller modules. The

modules are as follows –

a. Front End

a.1. Student Dashboard

a.2. Admin Dashboard

b. AI Model

c. Backend - Database

RAD Models work best with modular projects. They allow for all the modules to be

developed in parallel and then finally assembled together in a single working prototype.

2. The project was time bound – It needed to be delivered before the end of Semester.

RAD model enables rapid delivery as it reduces the overall development time due to

the reusability of the components and parallel development. The RAD model allows

for Rapid development and Prototyping. Due to these reasons, it was decided that this

model would work well for the time bounded project.

3. The project is small and does not have too high a risk factor.

4. Changes in the requirements can be easily accommodated while working with a RAD

Model.

5. Progress can be measured. Thus, at each step, there was a deliverable that could be

presented to the client if needed. It also allows for regular and early feedback and

reviews from the client.

6. Productivity with fewer people in a short time. While the RAD Model requires good

levels of skill in all the different areas of work for the different modules, yet its well

defined and structured approach helps increase productivity with even a small group

of developers. Since the project was developed by a single developer, the RAD model

made the most sense.

7. High reusability of components. The model follows an incremental approach

according to which any components developed for a previous prototype can be reused in making the next one. The modular approach also makes the code much more

organized and cleaner.

These along with many other advantages of the RAD model in regards to the UNIPREDICT

project made this model the preferred choice.

**9. SOFTWARE AND HARDWARE PLATFORMS USED**

The following section details the Software and Hardware platforms used to develop the

UNIVERSITY ADMIT ELIGIBLITY PREDICTER Application.

**9.1. Hardware**

A home PC – capable of handling light ML processing.

Device Specifications:

1. I5 10th Gen processor

2. 8 GB RAM

3. 64 bit Operating System

**9.2. Software**

1. Visual Studios Code

Visual Studios is a free source-code editor made by Microsoft for Windows,

Linux and MacOS. Features include support for debugging, syntax

highlighting, intelligent code completion, snippets, code refactoring, and

embedded Git. Users can change the theme, keyboard shortcuts, preferences,

and install extensions that add additional functionality.

2. Anaconda (Jupyter Notebook)

Project Jupyter is a non-profit organization created to "develop open-source

software, open-standards, and services for interactive computing across dozens

of programming languages".[2] Spun off from IPython in 2014 by Fernando

Pérez, Project Jupyter supports execution environments in several dozen

languages.

Project Jupyter's name is a reference to the three core programming

languages supported by Jupyter, which are Julia, Python and R, and also

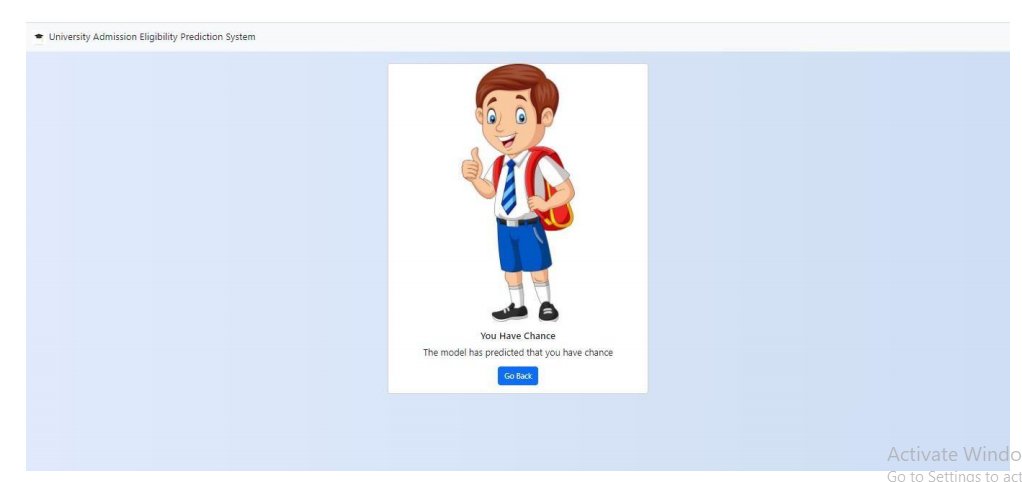
a homage to Galileo's notebooks recording the discovery of the moons of

Jupiter.

**10.CODING & SOLUTIONING**

**Feature 1;**

The new feature will predict the chances in the admission of the university. The feature was designed in the html code connected with app.py as the backend.



**Source Code :**

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

**{% block body %}**

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

**<div class="d-flex justify-content-center">**

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

**<img src="..\static\img\chance.png" class="card-img-top" alt="...">**

**<div class="card-body">**

**<h5 class="card-title">You Have Chance</h5>**

**<p class="card-text">The model has predicted that you have chance</p>**

**<a href="/home" class="btn btn-primary">Go Back</a>**

**</div>**

**</div>**

**</div>**

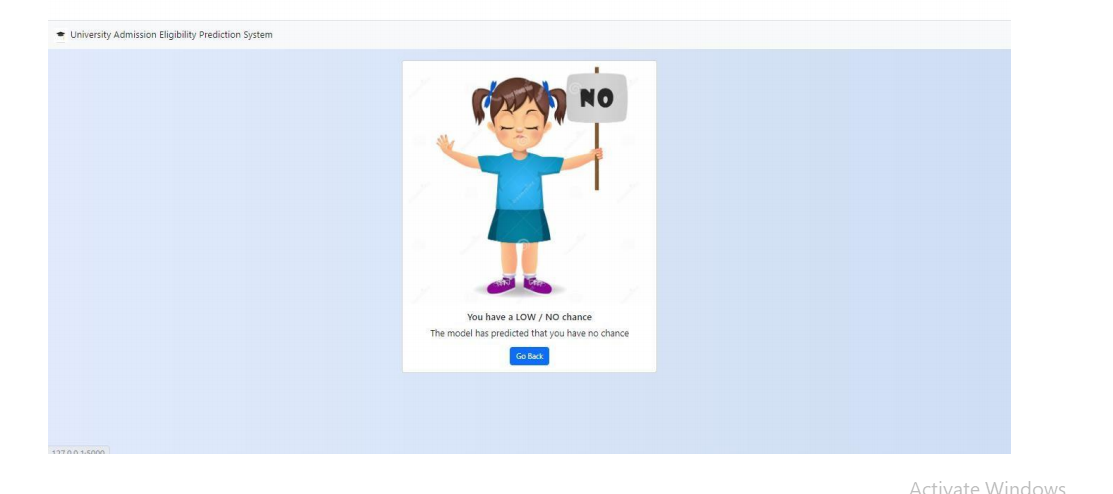
**</div>**

**{% endblock %}**

**Feature 2**

The new feature will predict the low chances in the admission of the university. The

feature was designed in the html code connected with APP.py as the backend.



**Source Code:**

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

**{% block body %}**

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

**<div class="d-flex justify-content-center">**

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

**<img src="..\static\img\Nochance.jpg" class="card-img-top" alt="...">**

**<div class="card-body">**

**<h5 class="card-title">You have a LOW / NO chance</h5>**

**<p class="card-text">The model has predicted that you have no chance</p>**

**<a href="/home" class="btn btn-primary">Go Back</a>**

**</div>**

**</div>**

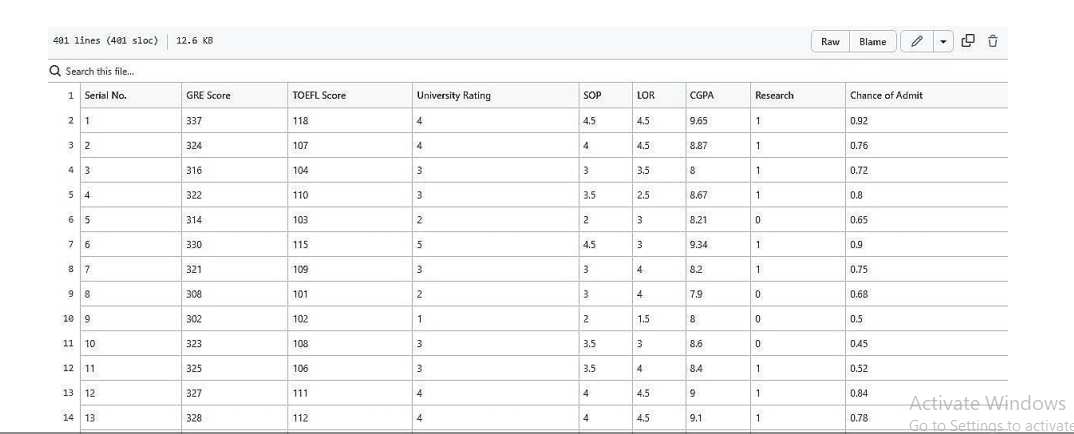
**</div>**

**</div>**

**{% endblock %}**

**Database Schema**

The database used here in this project was Admission\_Predict.csv. The sample screenshot of the database are,

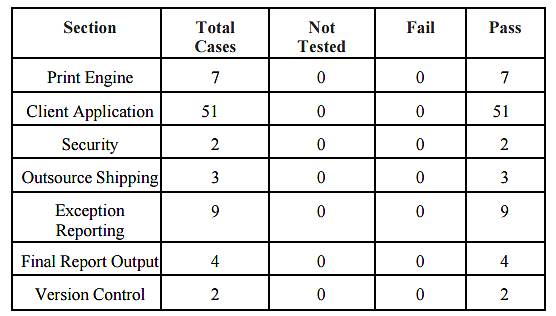


**11 .TESTING**

**Test Cases**

**Test Case Analysis**

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



**User Acceptance Testing**

● Purpose of Document

The purpose of this document is to briefly explain the test coverage and open

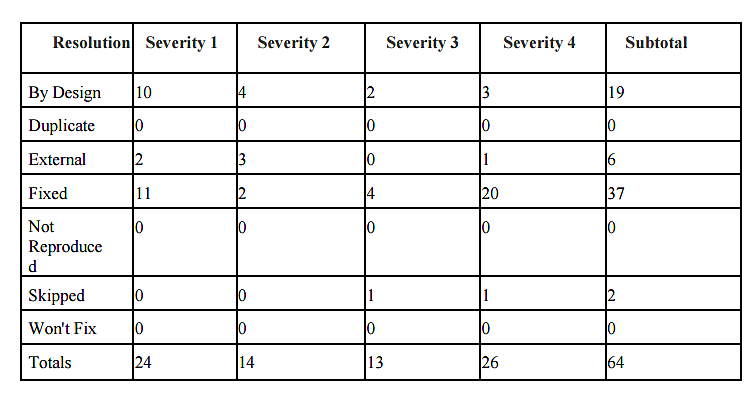
issues of the University Admit Eligibility Predictor project at the time of the

release to User Acceptance Testing (UAT).

● Defect Analysis

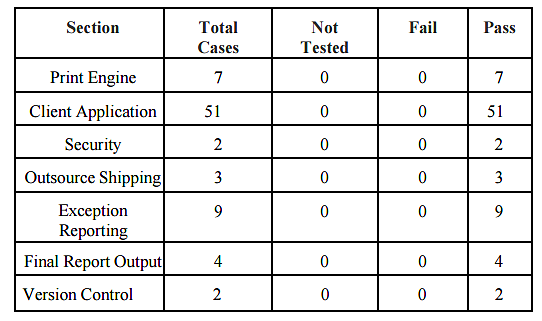
This report shows the number of resolved or closed bugs at each severity level, and

how they were resolved.



**Test Case Analysis**

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



**12. ADVANTAGES & DISADVANTAGES**

**12.1 Advantages**

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

● Here the chance of occurrence of error is less when compared with the existing system.

● It is fast, efficient and reliable.

● Avoids data redundancy and inconsistency.

● Very user-friendly.

● Easy accessibility of data.

**12.2 Dis-Advantages**

● Required active internet connection.

● System will provide inaccurate results if data entered incorrectly.

**13. CONCLUSION**

This system ,being the first we have created in Python using ML algorithms and other front end

languages such as html, css, java script , has proven more difficult than originally imagined.

While it may sound simple to fill out a few forms and process the information, much more is

involved in the selection of applicants than this. Every time progress was made and features were added, ideas for additional features or methods to improve the usability of the system made themselves apparent. Furthermore, adding one feature meant that another required feature was now possible, and balancing completing these required features with the ideas for improvement as well as remembering everything that had to be done was a project in itself. Debugging can sometimes be a relatively straight forward process, or rather rather finding out what you must debug can be. Since so many parts of the admissions system are integrated into one another, if an error occurs on one page, it may be a display error, for example; it may be the information is not correctly read from the database; or even that the information is not correctly stored in the database initially, and all three must be checked on each occasion. This slows down the process and can be frustrating if the apparent cause of a problem is not obvious at first. Language used must be simple and easy to understand and compatibility is paramount. If this system were not designed as an entirely web based application, it would not have been possible to recreate its current state of portability. Overall, the system performs well,and while it does not include all of the features that may have been desired, it lives up to initial expectations. The majority of features that are included work flawlessly and the errors that do exist are minor or graphical.

**14. FUTURE SCOPE**

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

● 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.

**15. APPENDIX**

**15.1 Source Code**

**PYTHON CODE**

Uploading the python code ,

**IMPORT STATEMENTS**

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

In [9]:

**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**=**'Kd7OVsyC1wCRI3\_hxIElXNrzYfwF5T82wLtLwzWjVIz8',  
 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 **=** 'universityadmiteligiplitypredicto-donotdelete-pr-czvbv27zokwayb'  
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[9]:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **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 [10]:

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

In [11]:

data**.**describe()

Out[11]:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **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 [12]:

data**.**info()

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

In [13]:

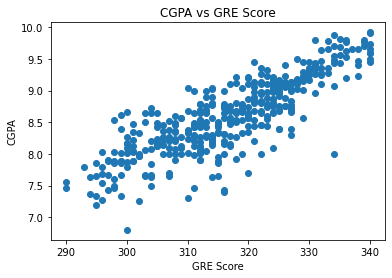
data**.**isnull()**.**sum()

Out[13]:

GRE Score 0  
TOEFL Score 0  
University Rating 0  
SOP 0  
LOR 0  
CGPA 0  
Research 0  
Chance of Admit 0  
dtype: int64

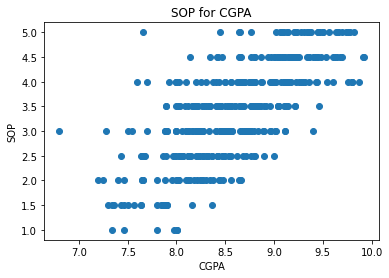
In [14]:

plt**.**scatter(data['GRE Score'],data['CGPA'])  
plt**.**title('CGPA vs GRE Score')  
plt**.**xlabel('GRE Score')  
plt**.**ylabel('CGPA')  
plt**.**show()



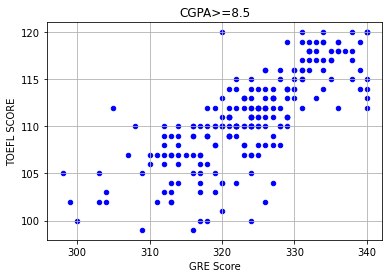
In [15]:

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



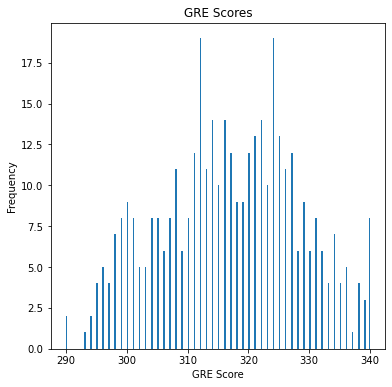
In [16]:

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()



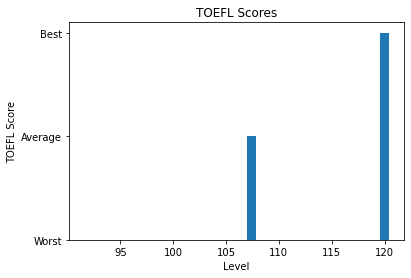
In [17]:

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



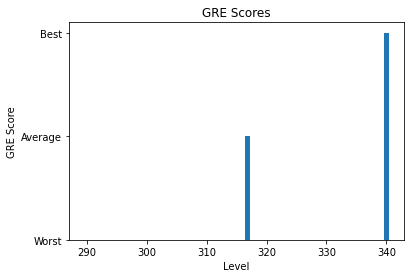
In [18]:

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()



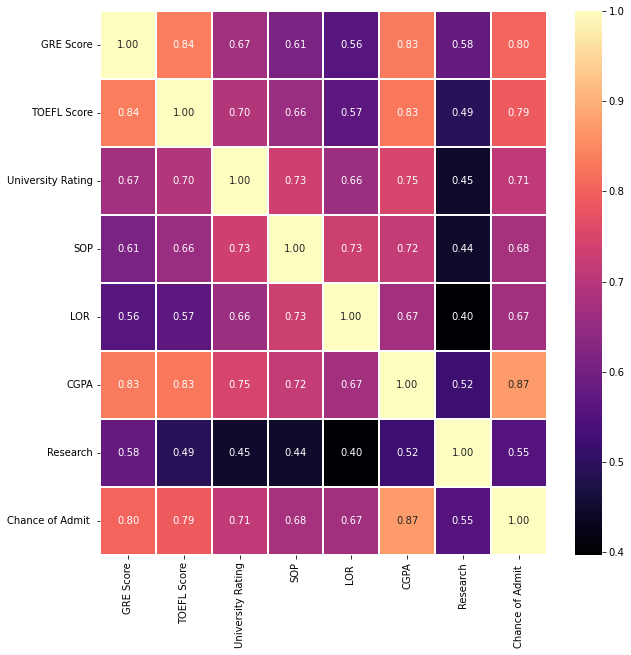
In [19]:

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 [20]:

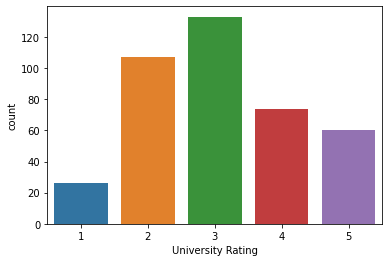
plt**.**figure(figsize**=**(10, 10))  
  
sns**.**heatmap(data**.**corr(), annot**=True**, linewidths**=**0.05, fmt**=** '.2f',cmap**=**"magma")  
  
plt**.**show()



In [21]:

data**.**Research**.**value\_counts()  
  
sns**.**countplot(x**=**"University Rating",data**=**data)

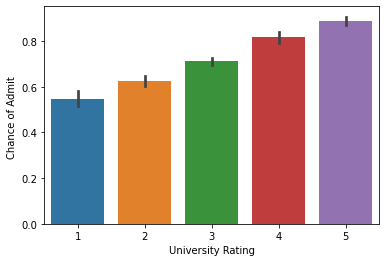
Out[21]:



In [22]:

sns**.**barplot(x**=**"University Rating", y**=**"Chance of Admit ", data**=**data)

Out[22]:



In [23]:

X**=**data**.**drop(['Chance of Admit '],axis**=**1) *#input data\_set*  
y**=**data['Chance of Admit '] *#output labels*

In [24]:

**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 [25]:

**from** sklearn.ensemble **import** GradientBoostingRegressor  
rgr **=** GradientBoostingRegressor()  
rgr**.**fit(X\_train,y\_train)

Out[25]:

GradientBoostingRegressor()

In [26]:

rgr**.**score(X\_test,y\_test)

Out[26]:

0.7530574145678719

In [27]:

y\_predict**=**rgr**.**predict(X\_test)

In [28]:

**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: 0.051805109421916004  
Mean Squared Error: 0.0049473849468724945  
Root Mean Squared Error: 0.07033764956886528

In [29]:

y\_train **=** (y\_train**>**0.5)  
y\_test **=** (y\_test**>**0.5)

In [30]:

**from** sklearn.linear\_model.\_logistic **import** LogisticRegression  
  
lore **=** LogisticRegression(random\_state**=**0, max\_iter**=**1000)  
  
lr **=** lore**.**fit(X\_train, y\_train)

In [31]:

y\_pred **=** lr**.**predict(X\_test)

In [32]:

**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: 0.9833333333333333  
Recall Score: 1.0  
ROC AUC Score: 0.9  
Confussion Matrix:  
 [[ 4 1]  
 [ 0 55]]

# save the model

In [33]:

**import** pickle

In [34]:

pickle**.**dump(lr, open("university.pkl", "wb")) *#logistic regression model*

# HOSTING THE MODEL

In [35]:

**import** pickle

In [36]:

lr **=** pickle**.**load(open("university.pkl", "rb")) *#logistic regression model*

In [37]:

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: packaging in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-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: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (0.8.9)  
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: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (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-watson-machine-learning) (1.3.4)  
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2.26.0)  
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (1.26.7)  
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-watson-machine-learning) (2022.9.24)  
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-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk==2.11.\*->ibm-watson-machine-learning) (2.11.0)  
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (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-dateutil<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 (from 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-metadata->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 (from packaging->ibm-watson-machine-learning) (3.0.4)  
Note: you may need to restart the kernel to use updated packages.

In [38]:

**from** ibm\_watson\_machine\_learning **import** APIClient  
**import** json

In [39]:

uml\_credentials **=** {  
 "url": "https://us-south.ml.cloud.ibm.com",  
 "apikey": "LmZMR10X36aP4p-l0ElaXn2WAI13QdbXmJgPvSPp0Se3"  
}  
  
client **=** APIClient(uml\_credentials)

In [40]:

**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 [42]:

space\_uid **=** guid\_from\_space\_name(client, "models")  
print(space\_uid[0])

372c6d3c-6ebc-4e59-8d73-813d8c7becfa

In [43]:

client**.**set**.**default\_space(space\_uid[0])

Out[43]:

'SUCCESS'

In [44]:

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  
runtime-22.2-py3.10-xc 5e8cddff-db4a-5a6a-b8aa-2d4af9864dab base  
autoai-kb\_3.1-py3.7 632d4b22-10aa-5180-88f0-f52dfb6444d7 base  
----------------------------- ------------------------------------ ----  
Note: Only first 50 records were displayed. To display more use 'limit' parameter.

In [45]:

**import** sklearn  
sklearn**.**\_\_version\_\_

Out[45]:

'1.0.2'

In [46]:

MODEL\_NAME **=** 'university'  
DEPLOYMENT\_NAME **=** 'uni'  
DEMO\_MODEL **=** lr

In [47]:

software\_spec\_uid **=** client**.**software\_specifications**.**get\_id\_by\_name('runtime-22.1-py3.9')

In [48]:

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 [49]:

model\_details **=** client**.**repository**.**store\_model(  
 model **=** DEMO\_MODEL,  
 meta\_props **=** model\_props,  
 training\_data **=** X\_train,  
 training\_target **=** y\_train   
)  
model\_details

Out[49]:

{'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': '2022-11-18T13:38:36.488Z',  
 'id': '534f1bfd-c90e-46a0-9a74-421185119171',  
 'modified\_at': '2022-11-18T13:38:39.204Z',  
 'name': 'university',  
 'owner': 'IBMid-6640042KCK',  
 'resource\_key': 'c0764487-efac-451e-b135-dc308d058500',  
 'space\_id': '372c6d3c-6ebc-4e59-8d73-813d8c7becfa'},  
 'system': {'warnings': []}}

Deployment creation failed. Error: 400. {"trace":"-qoc6ygirf8w7","errors":[{"code":"space\_lacks\_compute","message":"Space: 372c6d3c-6ebc-4e59-8d73-813d8c7becfa is not associated with a WML instance which is mandatory for create\_deployment operation"}]}

In [50]:

model\_id **=** client**.**repository**.**get\_model\_id(model\_details)  
model\_id

Out[50]:

'534f1bfd-c90e-46a0-9a74-421185119171'

In [52]:

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: '534f1bfd-c90e-46a0-9a74-421185119171' 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='e7f9d063-bc56-4b87-b70a-7c80ec0ce1f3'  
------------------------------------------------------------------------------------------------