**UNIVERSITY COLLEGE OF**

**ENGINEERING TINDIVANAM.**

# PNT2022TMID39267

**SAKTHI. P**

**ABOORVA. S**

**PRIYADHARSHINI. M**

**SHANMATHI. U**

**APPLIED DATA SCIENCE**

**UNIVERSITY ADMIT ELIGIBILITY PREDICTOR**

**Table of Content**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Topics** | **Page No.** |
| **1** | **Introduction**  1.1 Project Overview  1.2 Purpose | **3**  **3**  **3** |
| **2** | **LITERATURE SURVEY** | **3** |
|  | 2.1 Existing problem | **3** |
|  | 2.2 References | **3** |
|  | 2.3 Problem Statement Definition | **5** |
| **3** | **IDEATION & PROPOSED SOLUTION** | **7** |
|  | 3.1 Empathy Map Canvas | **7** |
|  | 3.2 Ideation & Brainstorming | **8** |
|  | 3.3 Proposed Solution | **9** |
|  | 3.4 Problem Solution fit | **9** |
| **4** | **REQUIREMENT ANALYSIS** | **10** |
|  | 4.1 Functional requirement | **10** |
|  | 4.2 Non-Functional requirements | **10** |
| **5** | **PROJECT DESIGN** | **11** |
|  | 5.1 Data Flow Diagrams | **11** |
|  | 5.2 Solution & Technical Architecture | **11** |
|  | 5.3 User Stories | **13** |
| **6** | **PROJECT PLANNING & SCHEDULING** | **14** |
|  | 6.1 Sprint Planning & Estimation | **14** |
|  | 6.2 Sprint Delivery Schedule | **15** |

|  |  |  |
| --- | --- | --- |
| **S.No** | **Topics** | **Page No.** |
|  | 6.3 Reports from JIRA | **16** |
| **7** | **CODING & SOLUTIONING**  7.1 Feature 1- Choosing Particular Department  7.2 Feature 2- Choosing Particular Location  7.3 Feature 3- Choosing Particular University  74 Feature 4- List Of Universities In Output | **20**  **20**  **20**  **21**  **22** |
| **8** | **TESTING**  8.1 Test Cases  8.2 User Acceptance Testing | **23**  **23**  **24** |
| **9** | **RESULTS**  9.1 Performance Metrics  9.2 Output | **25**  **25**  **28** |
| **10** | **ADVANTAGES & DISADVANTAGES** | **30** |
| **11** | **CONCLUSION** | **30** |
| **12** | **FUTURE SCOPE** | **31** |
| **13** | **APPENDIX**  Source Code  Github link  Project Demo link | **31**  **31**  **43**  **43** |
| **14** | **Reference** | **43** |

## 

## CHAPTER – 1

## INTRODUCTION

### 1.1 PROJECT OVERVIEW

The admission process in the universities is usually based on the pupil’s academic performance like their high school performance and the performance of entrance exam for that particular university. Many students may have a wish for a particular location for their convenience. The dataset is collected based on the requirements of the university’s admission process. Here we propose an idea where the user can know whether they can apply for the university or not with the help of the user’s academic details like their high school percentage and their entrance percentage. Here we use the model Support Vector Machine for prediction because of its high accuracy.

### 1.2 PURPOSE

A person’s education plays a vital role in their life. While planning for education students often have several questions regarding the courses, universities, job opportunities, location, etc. Securing admission in their dream university is one of their main concerns. University Prediction would be the easiest mode to predict whether the person is applicable for the university as well as it would unbiased and totally transparent. Individually the pupil would no longer need to depend upon the consultancies who may be slightly deviated towards the list of university that may be having contract with them. Moreover, applying to only that university where the student has genuine chance would even reduce application process.

## CHAPTER - 2

## LITERATURE SURVEY

### 2.1 EXISTING PROBLEM

An application which is used to predict whether the pupil is eligible to get an admission in the

Universities with their academic details and entrance exam results. Most of the applications predict universities based on the preferred university and their scores but not based on their preferred locations. Most of the websites predict for universities that are globally recognized. Most of the websites predict for the universities that are located in various countries with their TOFEL, GRE, etc. scores. With the help of these websites the pupil can know whether the pupil can apply for their interested university. Because it would be a loss for the pupil to apply for the university without getting to know the eligibility criteria for the university both in economic and time.

### 2.2 REFERENCES

#### 2.2.1 COLLEGE ADMISSION PREDICTION USING ENSEMBLE MACHINE LEARNING MODELS [VANDIT MANISH JAIN, RIHAAN SATIA, DECEMBER 2021]

This paper aims to build a model that can help students to pick the right universities based on their profiles. So, they can judge across a wide variety of domains that include MS (international), M.Tech (India) and MBA (India and International). For the accurate predictions we plan on training a machine learning

model in order to provide results. The dataset contains information on the student profile and the university details with a field detailing if the admission was positive or not. Various algorithms have been used i.e., Ensemble Machine Learning and the predictions have been compared using key performance indicators (KPIs). The model performing the best is then used to evaluate the dependent variable i.e..The chances of admit to a university.

**2.2.2 AN AUTOMATED PREDICTION MODEL FOR COLLEGE ADMISSION**

### SYSTEM [DR. ARUNAKUMARI B. N, VISHNU SASTRY H K et.al, JUNE 2021]

At present, many students make mistakes in their preference list of colleges because of various reasons like inaccurate analysis of colleges, lack of knowledge, and apprehensive prediction. Later, they end up regretting the same after allotment. This application addresses this issue of the student admission community. The application uses data mining and data analysis techniques. Rank, category, preferred branches, preferred district, and preferred colleges are taken as input and the preference list, on thorough analysis of the last five years’ cut-off data is generated. In this paper, an attempt has been made to develop an automated web application prediction model for a college admission system which can be used to make a wise choice of college before allotmentCut-off will be different for each college, course, and category. The row headings consist of college names along with branches. The column headings consist of the various categories. The data contained in the database is of string data.

**2.2.3 GRADUATE ADMISSION PREDICTION USING MACHINE LEARNING [SARA**

### ALJASMI, ALI BOU NASSIF et.al., OCTOBER 2020]

This paper addresses machine learning models to predict the chance of a student to be admitted to a master’s program. This will assist students to know in advance if they have a chance to get accepted. The machine learning models are multiple linear regression, k-nearest neighbour,random forest, and Multilayer Perceptron. Experiments show that the Multilayer Perceptron model surpasses other models.

In this paper, machine learning models were performed to predict the opportunity of a student to get admitted to a master’s program. The machine learning models included are multiple linear regression, k- nearest neighbour, random forest, and Multilayer Perceptron. Experiments show that the Multilayer Perceptron model surpasses other models.

#### 2.2.4 GRADUATE ADMISSION CHANCE PREDICTION USING DEEP NEURAL NETWORK [MD. OMAER FARUQ GONI, MD. ABU ISMAIL SIDDIQUE et.al., SEPTEMBER 2020]

In this study, they have proposed a deep neural network (DNN) to predict the chance of getting admitted to a university according to the student portfolio. All the selection criteria are considered here to predict the chance of admission. The DNN model has been compared with existing methods in terms of different performance metrics including mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), R-squared score. It has shown the most promising result that includes R- squared score of 0.8538 and MSE of 0.0031.

In this study, the graduate admission dataset has been split into training dataset and testing dataset. Data normalization has been performed to accelerate the training process of the DNN model. Using the training dataset, the DNN model has been trained with optimal hyper parameter. It has been assessed through some standard bench markings. Normalization technique is proposed which is used to transform all the numeric features into a common scale without deforming and losing information. Without normalization these varieties of range can create problems in the learning process of machine learning (ML) algorithms.

**2.2.5 ENGINEERING & TECHNOLOGY ADMISSION ANALYSIS AND PREDICTION [MR.**

### SACHIN BHOITE, PROF. DR. AJIT MORE, FEBRUARY 2020]

The aim of this paper is to determine the factors estimating & guiding the students to select engineering college for their first-year admission. Sometimes the students who are seeking admission is not eligible to take admission into engineering program based on their past academic record. Also, sometimes the students are seeking the admission to the college for she or he are not eligible as per the merit of that college. So here researcher has built predictive model to guide the students about their admissibility in the desired college & also suggest the college where they will get the admission. So, to achieve this objective we may include machine learning capabilities that allow to improve their performance based on experience, just as humans do. As right College plays very, vital role form the students’ placement and career point of view the researcher has implemented various algorithms to achieve this objective.

**2.2.6 MULTIPLE MACHINE LEARNING CLASSIFIERS FOR STUDENT’S ADMISSION TO**

### UNIVERSITY PREDICTION [ANIL B, AKRAM PASHA, AMAN KUMAR SINGH et.al., MAY

#### 2019]

The admission predictor developed in this study uses the student’s application data that includes many features including a class variable that has binary value. This class variable is true if the student had taken admission or false if he did not. Using nominal and categorical attributes and past collected data this work is done at ease. Implementation of two different techniques on our data set; with that classification builds a predictive model and association rules which were used to find interesting hidden information in the student's records.

**2.2.7 CAPSLG: COLLEGE ADMISSION PREDICTOR AND SMART LIST GENERATOR**

### [KIRAN KUMARI, MEET KATARIA et.al., JANUARY 2019]

The CAPSLG system consists of a smart list generator working together with the help of college predictor, to aid students in the admission process. The college admission predictor uses historical colleges cut-off student admission data for predicting the most probable colleges. The system analyses student academic merits, background, and college admission criteria. Based on that, it predicts the likelihood of a university college that a student may enter. The smart list generator would enable the student to prepare the list of colleges, which could be needed to be filled in during the admission process. The system would also get feedback from the users, which would prove helpful for prediction evaluation and improving the performance factor.

**2.3 PROBLEM STATEMENT DEFINITION**



#### Fig. No. 2.1 Problem Statement1



#### Fig. No. 2.2 Problem Statement2



#### Fig. No. 2.3 Problem Statement3



**Fig. No. 2.4 Problem Statement4**

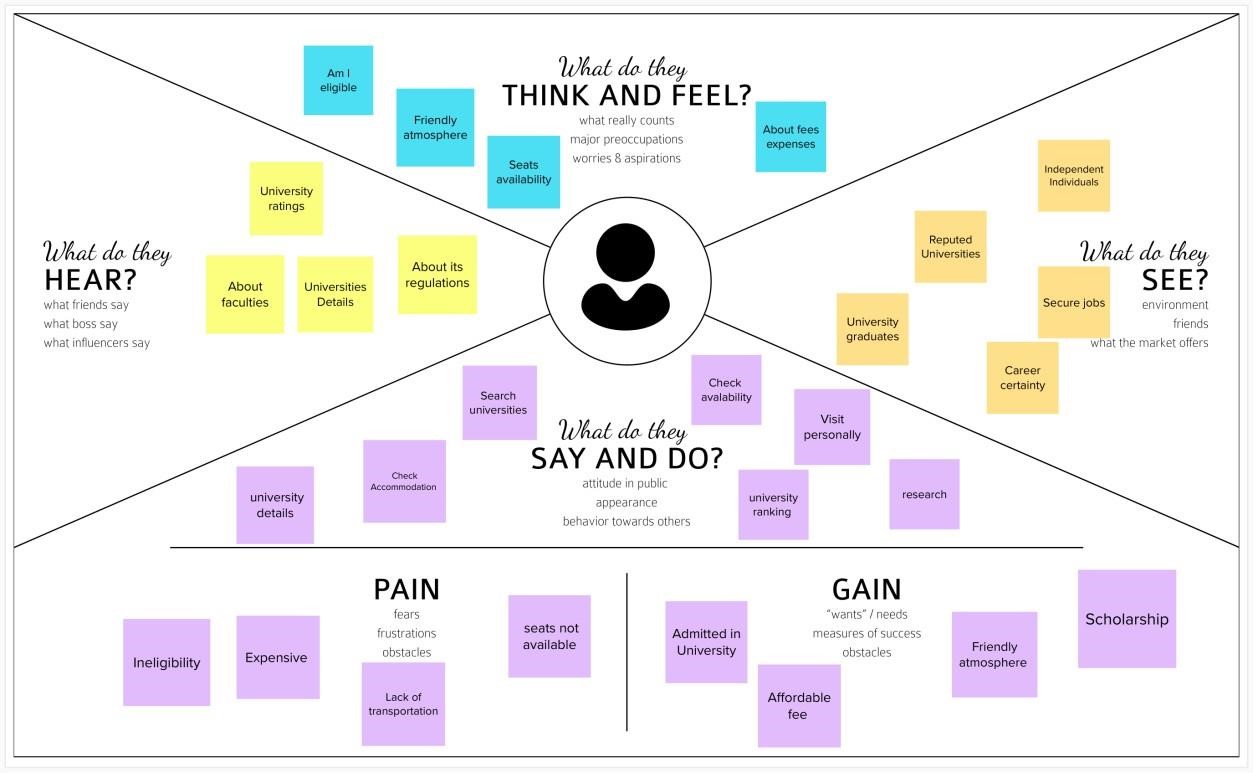
## 

## 

## CHAPTER -3

**IDEATION & PROPOSED SOLUTION**

### 3.1 Empathy Map Canvas



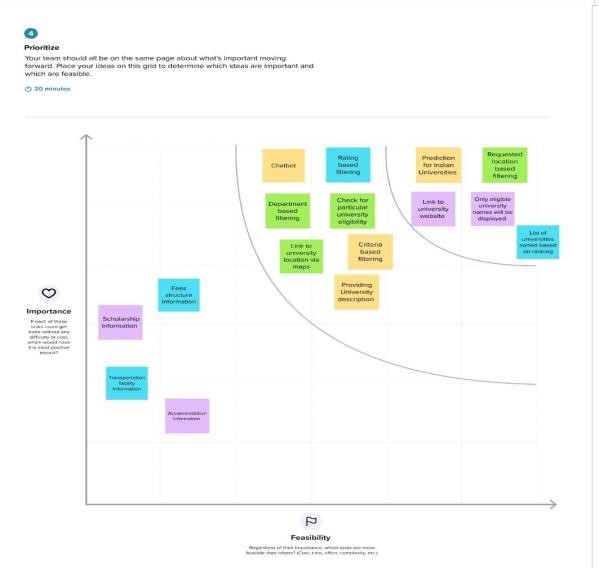
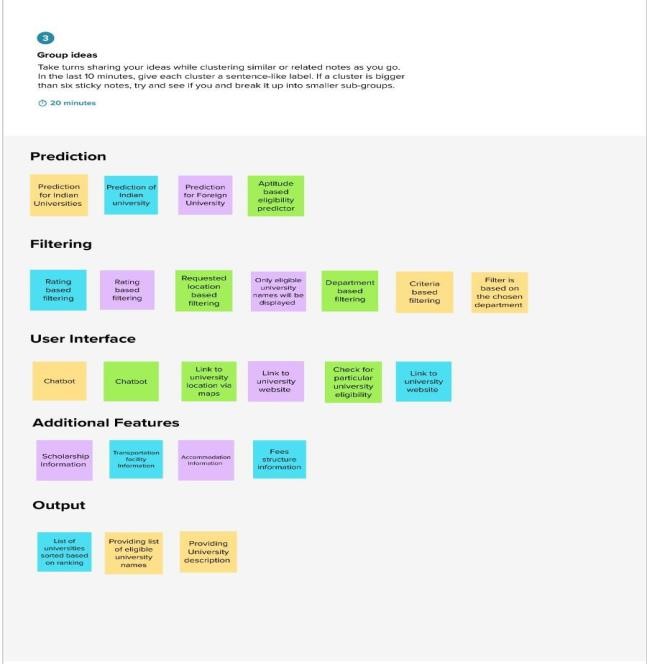
**Fig. No. 3.1 Empathy Map Canvas**

### 3.2 IDEATION & BRAINSTROMING

### 

**Fig. No. 3.3 Brainstorm and Idea Listing**

**Fig. No. 3.2 Team Gathering, Collaboration and Select the Problem Statement**



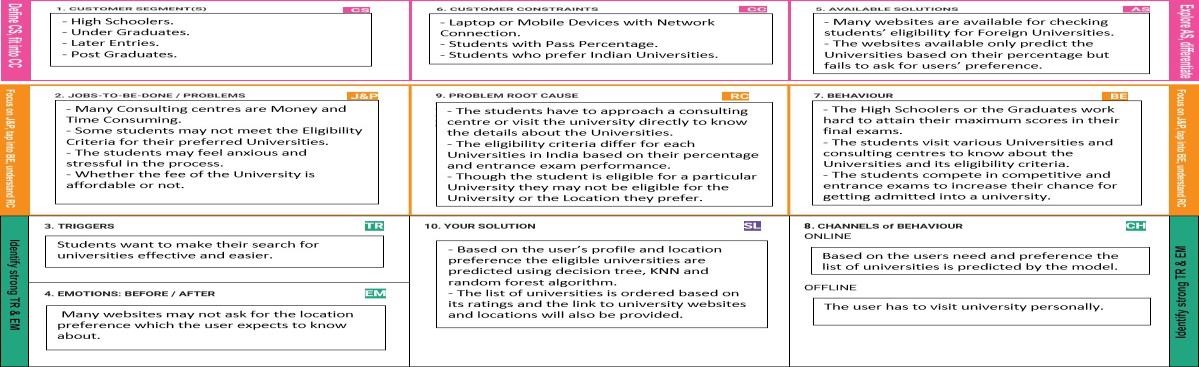
**Fig. No. 3.4 Grouping Fig. No. 3.5 Idea Prioritization**

### 3.3 PROPOSED SOLUTION

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Parameter** | **Description** |
| 1. | Problem Statement (Problem to be solved) | The list of eligible universities has to be predicted based on the user’s profile. |
| 2. | Idea / Solution description | Based on the user’s profile and location preference the eligible universities are predicted using decision tree, KNN and random forest algorithm. The list of universities is ordered based on its rankings and the link to university websites and locations will also be provided. |
| 3. | Novelty / Uniqueness | By using this predictor, the user can able to get to know about the eligible universities that falls under their preferred location. The link to the universities also be provided which helps the user to know about the universities. |
| 4. | Social Impact / Customer  Satisfaction | The final list will be based on the location preference and the order of universities will be based on rankings of university which makes the user to know about the number of their eligible universities and this prior knowledge will be useful for their preparing. |
| 5. | Business Model (Revenue Model) | Based on the user’s profile and preferred location the eligible universities will be predicted.Hence by knowing the particular universities the user will get a better idea. |
| 6. | Scalability of the Solution | The students from all over India can check their eligibility criteria for universities located in India. |

**Table 3.1 Proposed Solution**

### 3.4 PROBLEM SOLUTION FIT



**Fig. No. 3.6 Problem Solution Fit Template**

## CHAPTER - 11 REQUIREMENT ANALYSIS

### 4.1 FUNCTIONAL REQUIREMENT

|  |  |  |  |
| --- | --- | --- | --- |
| **FR No.** | **Functional (Epic)** | **Requirement** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User Profile |  | Complete user profile by providing the Student Academic details. |
| FR-2 | User Search |  | Search for desired University.  Search for Universities based on their Academic Performance and eligibility criteria. |
| FR-3 | User Preference |  | Search for Universities based on their location preference. |
| FR-4 | Result |  | The list of universities is filtered based on the eligibility of the students.  The order of the list will be based on the ratings of the university. |

**Table 4.1 Functional Requirement**

### 4.2 NON-FUNCTIONAL REQUIREMENTS

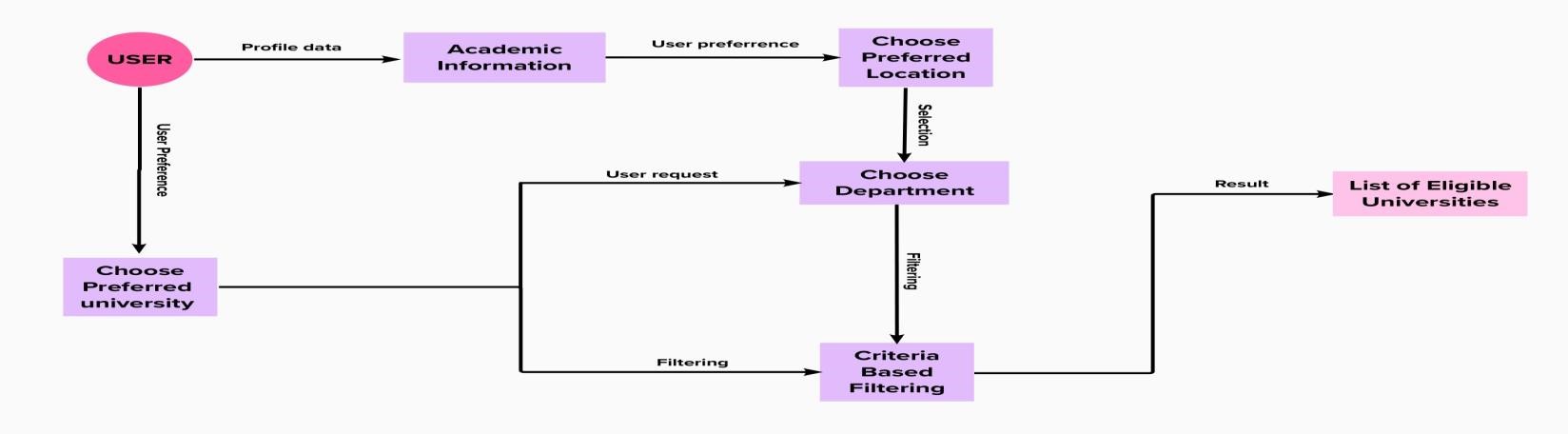
|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | Filters the universities based on the user profile. |
| NFR-2 | **Security** | User details are secured from unauthorised parties. |
| NFR-3 | **Reliability** | The users can find universities based on their preferred location and results. |
| NFR-4 | **Performance** | The website will provide the list of universities within 30 seconds. |
| NFR-5 | **Availability** | Students across India can access the website anytime. |
| NFR-6 | **Scalability** | The solution will be helpful for the students in India to know the details about universities they are eligible. |

**Table 4.2 Non-Functional Requirement**

## CHAPTER - 5

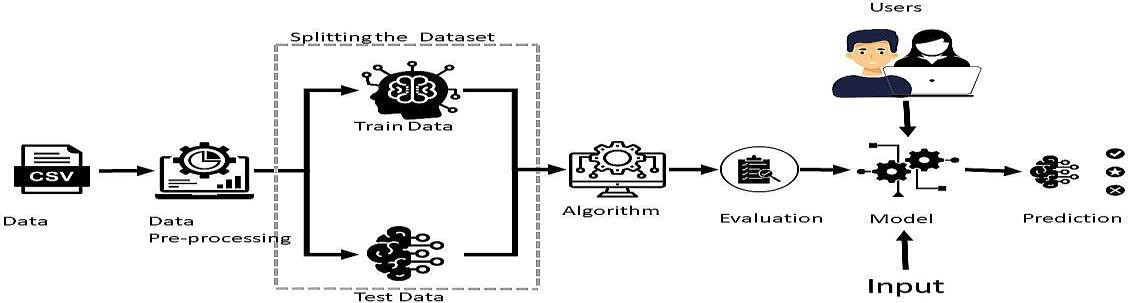
**PROJECT DESIGN**

**5.1 DATA FLOW DIAGRAMS**

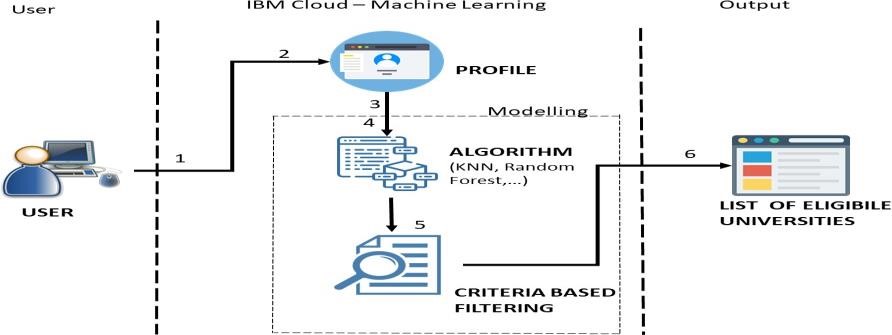


**Fig. No. 5.1 Data Flow Diagram**

### 5.2 SOLUTION AND TECHNICAL ARCHITECTURE



#### Fig. No. 5.1 Solution Architecture



#### Fig. No. 5.2 Technical Architecture

**Guidelines for Technical Architecture:**

1. Include all the processes (As an application logic / Technology Block)

1. Provide infrastructural demarcation (Local / Cloud)

1. Indicate interface to machine learning models

1. Include necessary machine learning algorithms

1. Indicate Data Storage components / services

1. Provide the list of all eligible universities along with its description

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Component** | **Description** | **Technology** |
| 1. | User Interface | How user interacts with application e.g.  Web UI, Mobile App, Chatbot etc. | HTML, CSS, JavaScript etc. |
| 2. | Application Logic-1 | Logic for a process in the application | Python (Jupyter) |
| 3. | Application Logic-2 | Logic for a process in the application | IBM Watson Assistant |
| 4. | Database | Data Type, Configurations etc. | CSV |
| 5. | External API | Purpose of External API used in the application | List of eligible  Universities |
| 6. | Machine Learning Model | Purpose of Machine Learning Model | KNN, Random Forest, Decision Tree, Support Vector Machine,etc. |
| 7. | Infrastructure (Server / Cloud) | Application Deployment on  Local System / Cloud  Local Server Configuration:  Cloud Server Configuration : | Local, Cloud etc. |

**Table 5.1 Components & Technologies**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Characteristics** | **Description** | **Technology** |
| 1. | Open-Source Frameworks | Python for Backend purpose and flask is imported for front end purpose | Python(Flask) |
| 2. | Security Implementations | The user profile will be secure | Encryptions, IAM  Controls, OWASP etc. |
| 3. | Scalable Architecture | The accurate list of eligible  universities name and its description will be provided | Support Vector Machine ML Algorithm |
| 4. | Availability | Anyone and in anytime they can visit our website | IBM Load Balancer |
| 5. | Performance | The user can have a knowledge of their eligibility for applying Universities through our website | Support Vector Machine ML Algorithm |

**Table 5.2 Application Characteristics**

**5.3 USER STORIES**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional**  **Requirement (Epic)** | **User**  **Story**  **Number** | **User Story / Task** | **Acceptance criteria** | **Priority** | **Release** |
| Customer (Web  user) | Profile | USN-1 | As a user, I can Give my academic information in the profile section | I can access my dashboard | High | Sprint- 1 |
|  |  | USN-2 | As a user, I will be able to select a location that I  prefer | I can receive  the list of locations in the dropdown  to select | High | Sprint- 1 |
|  | Search | USN-3 | As a user I can search for my  preferred university | I can use the search bar | Medium | Sprint- 2 |
|  | User  Preference | USN-4 | As a user, I can select my preferred university from the list to check my eligibility for the  particular university | I can use the dropdown list provided to  select the  university | Medium | Sprint- 2 |
|  |  | USN-5 | As a user, I can select my preferred location | I can select my preferred location | High | Sprint- 1 |
|  |  | USN-6 | As a user, I will be able to select my  preferred department | I can select a department  from the  dropdown list | Medium | Sprint- 1 |
|  | Result | USN-7 | As a user, I can view the list of universities that I am eligible in accordance to my preferred location | I can view the list of universities  filtered by the model | High | Sprint- 3 |
|  |  | USN-8 | As a user, I can access the link to the university that I am eligible from  the list | I can access the university link | Medium | Sprint- 3 |
|  |  | USN-9 | As a user, I can access the location link of the university that I  am eligible from  the list | I can access the university location link | Low | Sprint- 3 |

**Table 5.3 User Stories**

## CHAPTER - 6

**PROJECT PLANNING AND SCHEDULING**

### 6.1 SPRINT PLANNING AND ESTIMATION

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional**  **Requirement (Epic)** | **User**  **Story**  **Number** | **User Story / Task** | **Story Points** | **Priori ty** | **Team**  **Members** |
| Sprint-1 | Profile | USN-1 | As a user, I can give my academic  information in the profile section | 3 | High | Sakthi P  and  Aboorva S |
| Sprint-1 |  | USN-2 | As a user, I will be able to select a location that I prefer | 5 | High | Priyadharshini M  and  Shanmathi U |
| Sprint-2 | Search | USN-3 | As a user I can search for my preferred  university | 3 | Medium | Aboorva S  and  Priyadharshini M |
| Sprint-1 | User  Preference | USN-4 | As a user, I can select my preferred location | 3 | High | Sakthi P  and  Shanmathi U |
| Sprint-1 |  | USN-5 | As a user, I will be able to select my preferred department | 3 | Medium | Sakthi P  and  Aboorva S |
| Sprint-2 |  | USN-6 | As a user, I can select my preferred university from the list to check my eligibility for the particular university | 8 | High | Priyadharshini M  and  Shanmthi U |
| Sprint-3 | Result | USN-7 | As a user, I can view the list of universities that I am eligible in accordance to my preferred location | 3 | High | Shanmthi U  and  Aboorva S |
| Sprint-3 |  | USN-8 | As a user, I can access the link to the university that I am  eligible from the list | 2 | Medium | Priyadharshini  and  Sakthi P |
| Sprint-3 |  | USN-9 | As a user, I can access the location link of the university that I am  eligible from the list | 1 | Low | Shanmthi U  and  Aboorva S |
| Sprint-3 |  | USN-10 | From the list of universities, I can select and view the eligibility for the particular  university | 5 | High | Sakthi P  and  Aboorva S |
| Sprint-4 | Output | USN-11 | As a user, I will give  my information  accordingly asked in  the website | 5 | High | Sakthi P  and  Aboorva S |
| Sprint-4 |  | USN-12 | As a user, I will be able to view whether I’m eligible for the selected University and the list of other Universities that I’m eligible. | 8 | High | Sakthi P  and  Shanmathi U |

**Table 6.1 Product Backlog, Sprint Schedule, and Estimation**

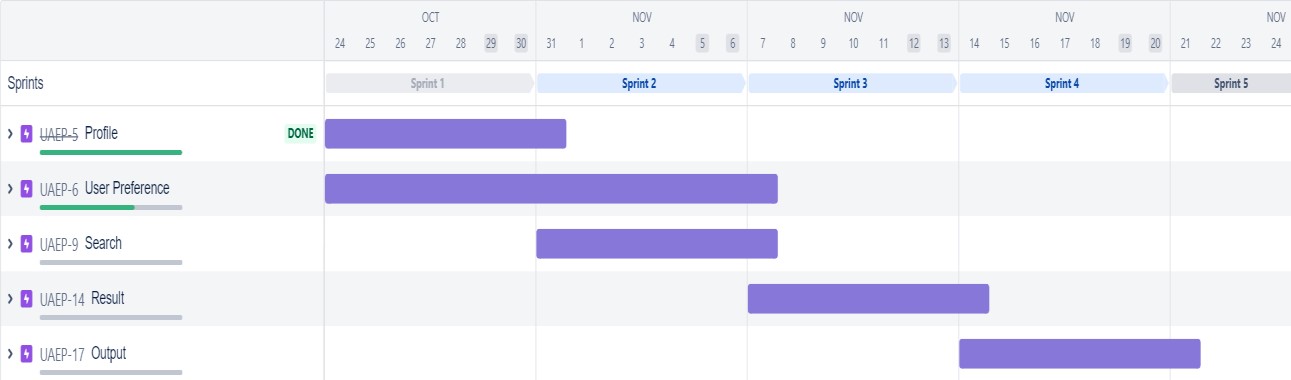
### 6.2 SPRINT DELIVERY SCHEDULE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Total**  **Story**  **Points** | **Duration** | **Sprint**  **Start Date** | **Sprint End**  **Date (Planned)** | **Story Points Completed (as on Planned End Date)** | **Sprint**  **Release**  **Date**  **(Actual)** |
| Sprint-1 | 14 | 6 Days | 24 Oct  2022 | 29 Oct 2022 | 14 | 29 Oct  2022 |
| Sprint-2 | 11 | 6 Days | 31 Oct  2022 | 05 Nov 2022 | 11 | 31 Oct  2022 |
| Sprint-3 | 11 | 6 Days | 07 Nov  2022 | 12 Nov 2022 | 11 | 07 Nov  2022 |
| Sprint-4 | 13 | 6 Days | 14 Nov  2022 | 19 Nov 2022 | 13 | 14 Nov  2022 |

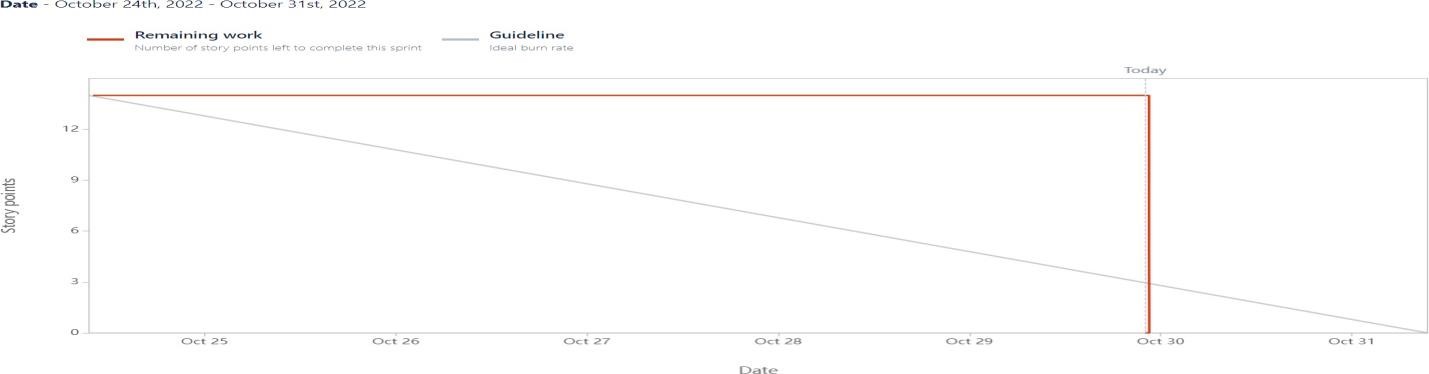
**Table 6.2 Project Tracker**

### 6.3 REPORTS FROM JIRA

#### 6.3.1 Road Map – Sprint 1



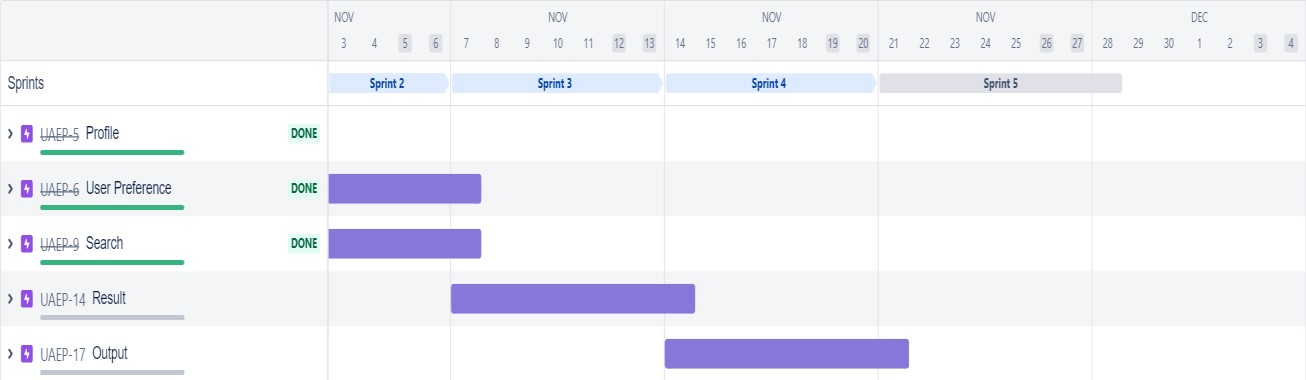
#### 6.3.2 Burn down Chart – Sprint



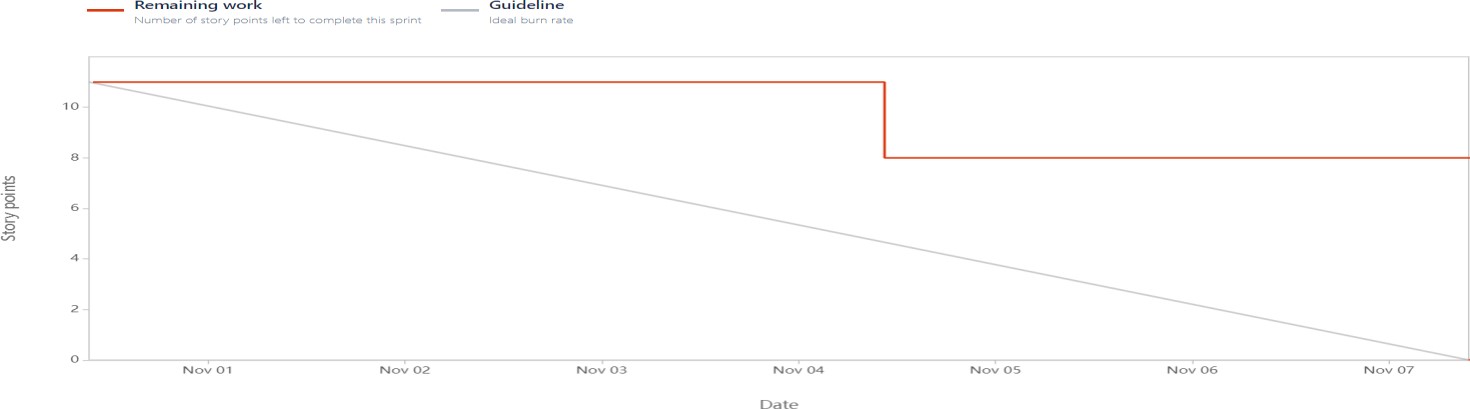
#### 6.3.3 Velocity Chart – Sprint 1



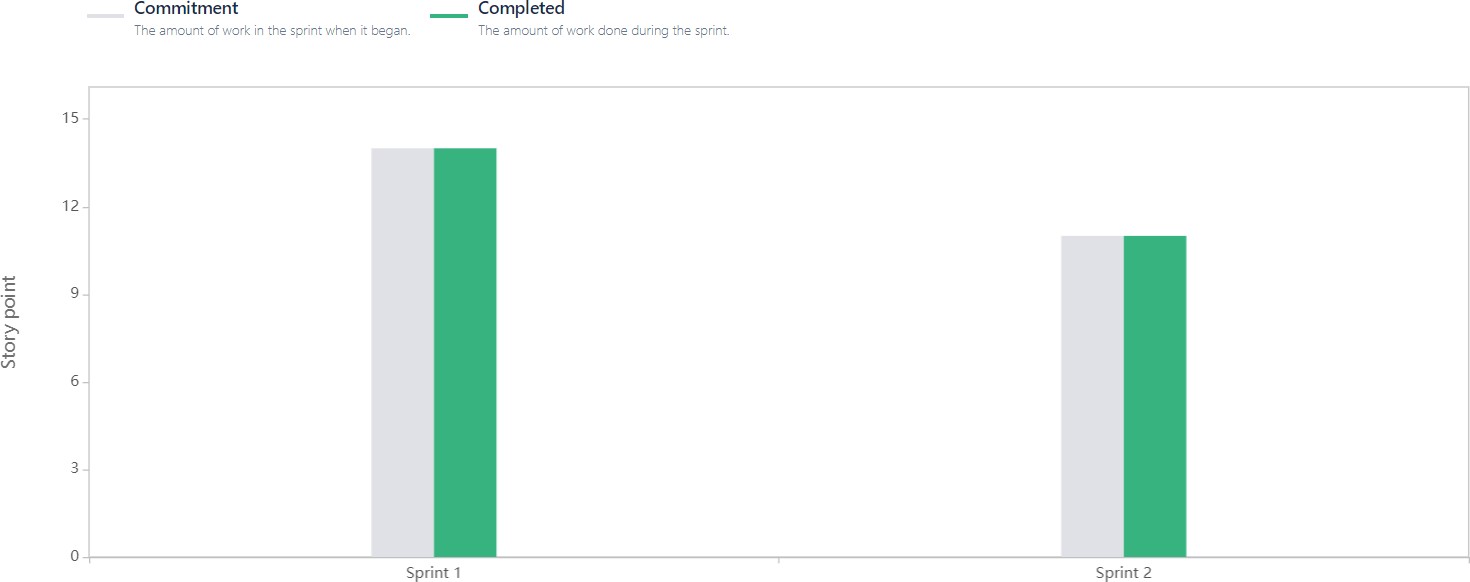
#### 6.3.4 Road Map – Sprint 18



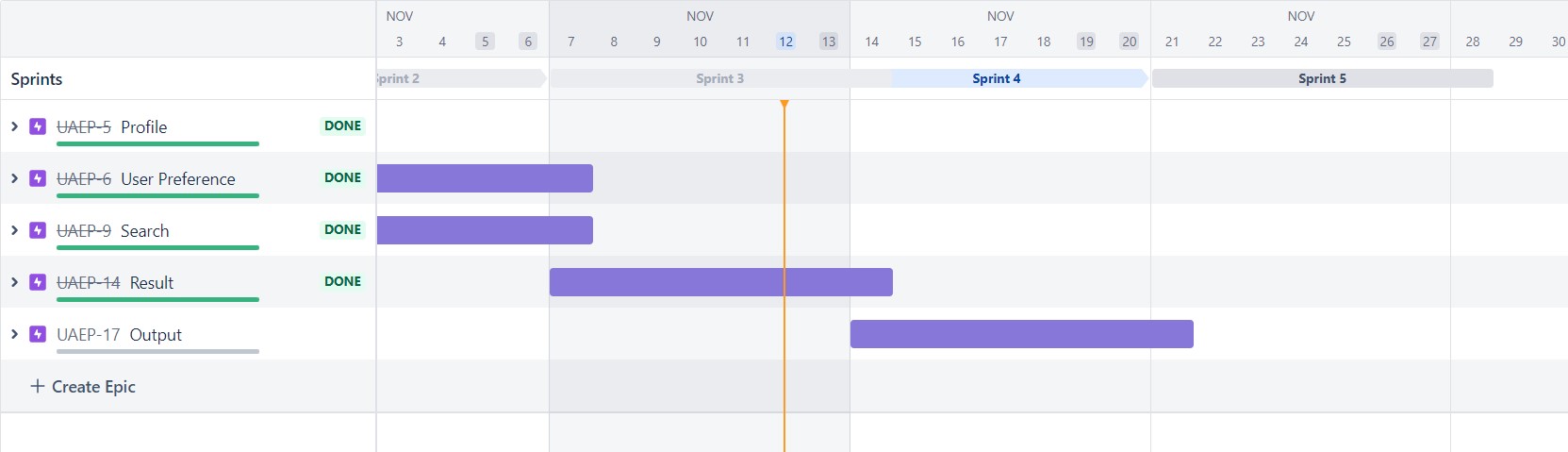
#### 6.3.5 Velocity Chart – Sprint 2



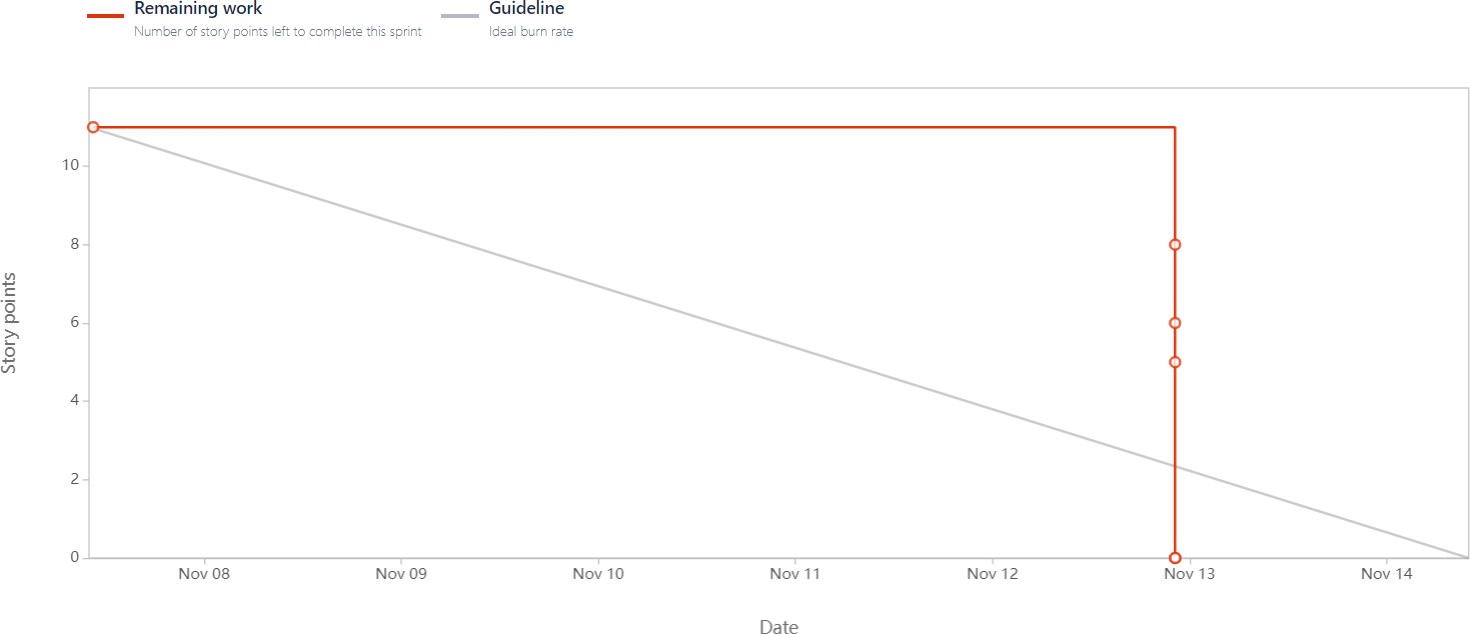
#### 6.3.6 Velocity Chart – Sprint 2



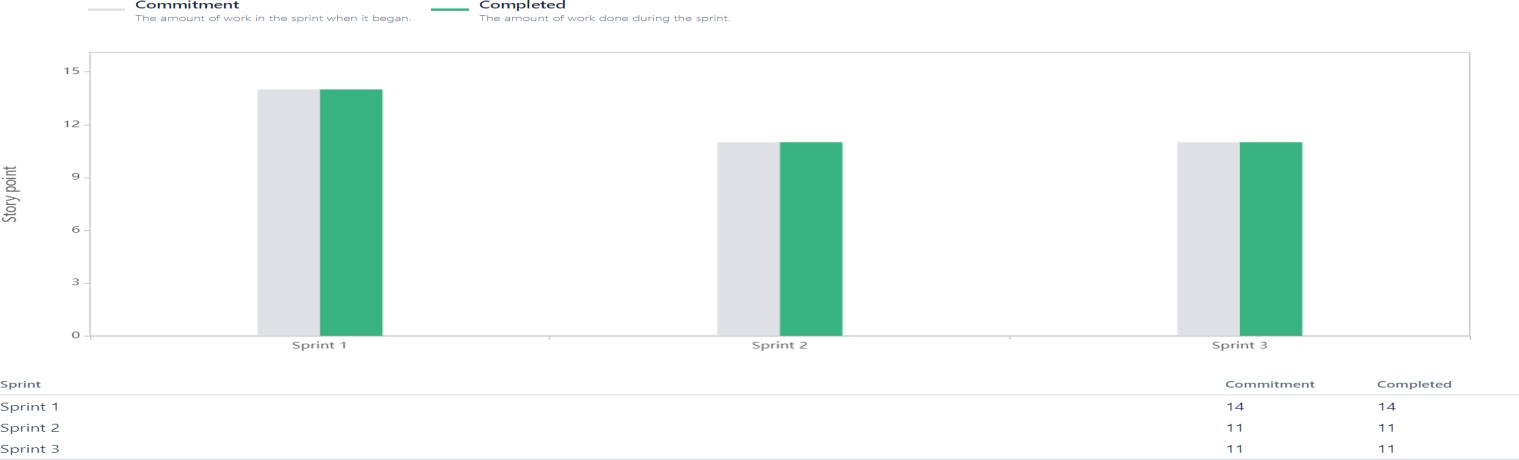
#### 6.3.7 Road Map – Sprint 19



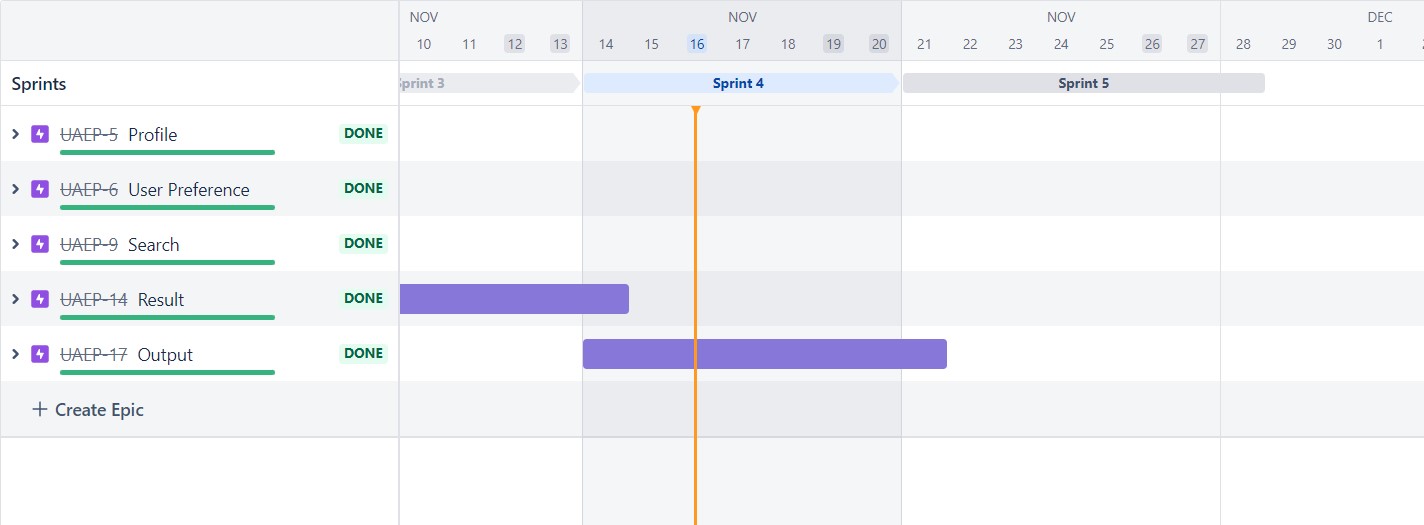
#### 6.3.8 Burn-down Chart – Sprint 3



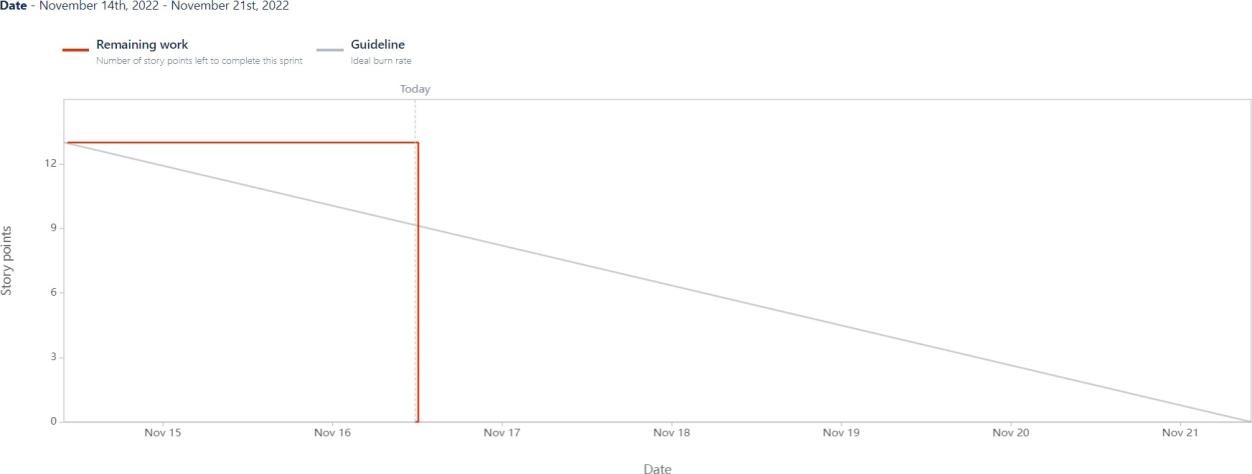
#### 6.3.9 Velocity Chart – Sprint 3



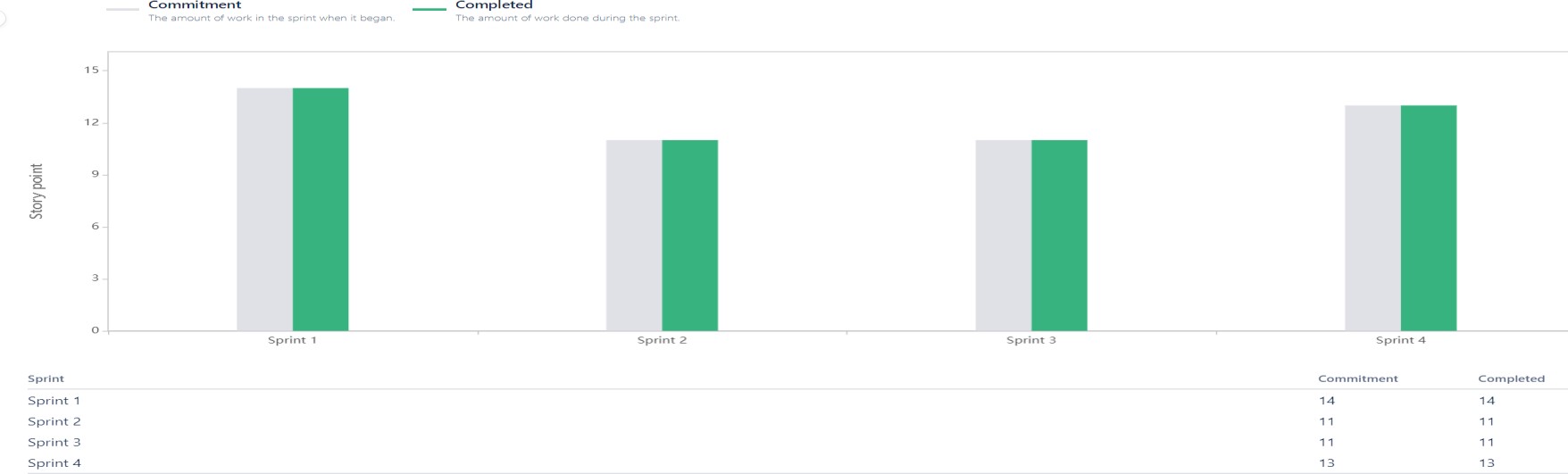
#### 6.3.10 Road Map – Sprint 4



**6.3.11 Burn-down Chart – Sprint 4**



**6.3.12 VelocityChart – Sprint 4**



## CHAPTER – 7

## CODING AND SOLUTIONING

### 7.1 FEATURE 1 – Choosing Particular Department to Check Eligibility for Admission in a Particular University

Students nowadays are entering into a university with a lot of dreams to achieve. Many candidates have a particular department which they are interested in learning. The universities have different admission criteria for getting admitted in different universities. Many candidates are unaware of the eligibility criteria of the university for that particular department which they are interested in. So, in our application we have come up with the feature where the candidates can choose a particular department which they are interested and in the redirected page the candidate can choose the particular university in which they are interested to join. Our application provides the top 20 universities for the particular department the user can choose their interested university from them. The predicted output is based on the user input and the selected university’s eligibility criteria.



#### Fig. No. 7.1 Choosing Department 7.2 FEATURE 2 – Choosing a Particular Location

The candidates wishing to join in a university in which they are interested. Some students may have a preferred location and from that location they may wish to join in a university. The candidates may be unaware of the different universities present in that particular location and their eligibility criteria. So, in our application we have come up with a feature where the user can first choose their interested department and they choose their preferred location and university from the particular location. The predicted output is based on the user input and the selected university’s eligibility criteria.



**Fig. No. 7.2 Choosing Location for Mechanical**

### 7.3 FEATURE 3 – Choosing Particular University to Check Eligibility for Admission

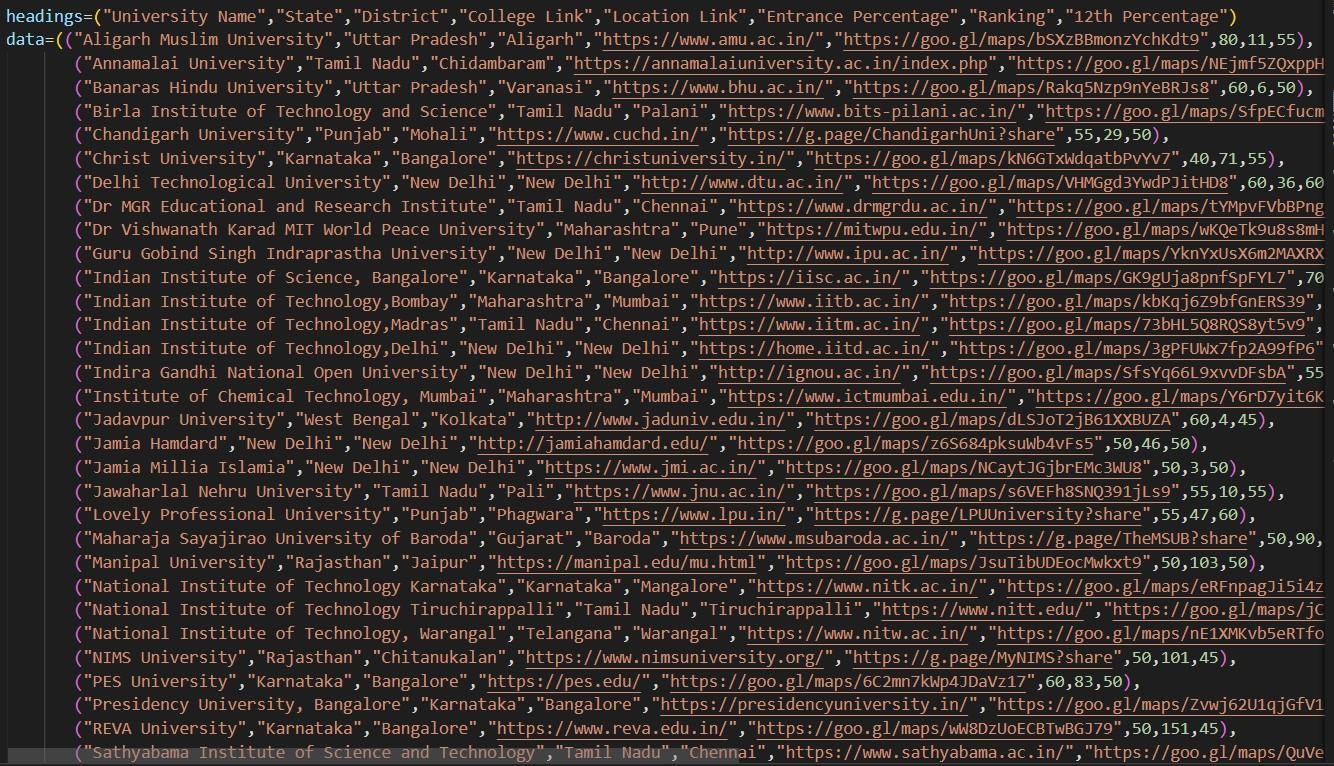
The pupils after completing their higher school studies wish to join a best university to continue their higher studies. Many candidates applying for universities have a dream university in which they are interested in learning. Different universities have different eligibility criteria for admitting candidates. The candidates have to meet the criteria set by the university to get admitted. Many candidates are unaware of the eligibility criteria set by the universities. So, we have come up with a solution with the feature where the candidates can select a particular university to check their eligibility in that particular university of their interested department.



**Fig. No. 7.8 Choosing University for EEE**

### 7.4 FEATURE 4 - Providing with Other List of Universities in Output that the User is Eligible to Apply

Many students may not be eligible for their interested university with their educational qualifications. So, in our application we are providing an additional feature where the user can view the additional list of universities that they are eligible to apply to after viewing the predicated output for their given inputs. Besides it the link to the university website is also provided for the user to check for additional information about the university.



**Fig. No. 7.12 List of Universities in Output**

## 

## 

## CHAPTER-8

## TESTING

### 8.1 TEST CASES

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Cases** | **Features** | **Description** | **Steps to Execute** | **Expected Results** |
| TC-001 | UI | Check whether all the UI elements present in the web pages are visible and functioning | 1. Enter into the webpage. 2. Verify whether the UI elements are functioning. | The UI elements to function |
| TC\_002 | Functional | Check whether the options provided in the dropdown list is redirecting when its chosen | 1. Click on the dropdown list. 2. Choose a department from the list provided. | Whether the page is redirecting when the choice is chosen |
| TC\_003 | Validation | Check whether the percentage is accepted a  both in integer in all the department pages | 1. Choose the percentage input. 2. Enter the input as integer | Check whether the in input is accepted as integer |
| TC\_004 | Validation | Check whether the percentage is accepted a both in float in all the department pages | 1. Choose the percentage input. 2. Enter the input as float | Check whether the in input is accepted as float |
| TC\_005 | Functional | Check whether the user | 1. Choose the city | The particular |
|  |  | can choose a particular | dropdown list | city should be |
|  |  | city from the give | 2. Select a particular | selected |
|  |  | dropdown list in all the department pages | city |  |
| TC\_006 | Functional | Check whether the user can choose a particular university from the give dropdown list in all the department pages | 1. Choose the city dropdown list 2. Select a particular city | The particular university should be selected |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TC\_007 | Functional | Check whether the user is redirected to the output page and the output is predicted as the user is eligible to apply for the university or not | 1. After entering the required details redirect to the output page 2. The output is produced as the user is eligible or the user is not eligible. | The eligibility of the user should be displayed. |
| TC\_008 | Functional | Check whether the list of | Check for the tale | The list of |
|  |  | other universities is given | with list of | universities is |
|  |  | after the prediction output | universities. | displayed based |
|  |  | for all the departments. |  | on the user input. |

**Table 8.1 Test Cases**

**8.2 USER ACCEPTANCE TESTING**

#### 1. Defect Analysis

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Resolution** | **Severity 1** | **Severity 2** | **Severity 3** | **Severity 4** | **Subtotal** |
| By Design | 1 | 4 | 2 | 3 | 10 |
| Duplicate | 1 | 0 | 0 | 0 | 1 |
| External | 2 | 3 | 0 | 2 | 7 |
| Fixed | 3 | 2 | 2 | 5 | 12 |
| Not Reproduced | 0 | 0 | 1 | 0 | 1 |
| Skipped | 1 | 2 | 1 | 1 | 4 |
| Won't Fix | 0 | 1 | 2 | 1 | 4 |
| Totals | 8 | 12 | 8 | 12 | 39 |

#### 2. Test Case Analysis

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Section** | **Total Cases** | **Not Tested** | **Fail** | **Pass** |
| Home Page | 8 | 0 | 0 | 8 |
| Computer Science Page | 7 | 0 | 0 | 7 |
| CIVIL Engineering Page | 8 | 0 | 0 | 8 |
| EEE Page | 5 | 0 | 0 | 5 |
| ECE Page | 7 | 0 | 0 | 7 |
| Mechanical Page | 5 | 0 | 0 | 5 |
| Version Control | 2 | 0 | 0 | 2 |
| Output Page | 7 | 0 | 2 | 5 |

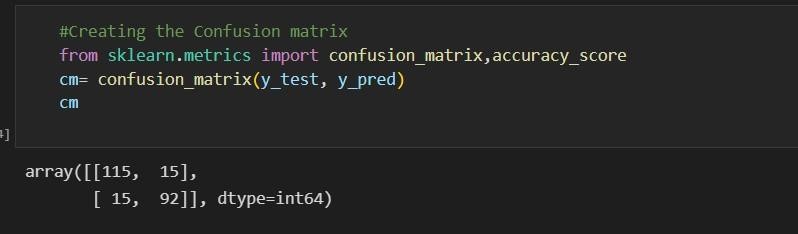
**CHAPTER – 9**

**RESULTS**

## 9.1 PERFORMANCE METRICS

A confusion matrix is a tabular representation of prediction outcomes of any binary classifier, which is used to describe the performance of the classification model on a set of test data when true values are known. The confusion matrix is simple to implement, but the terminologies used in this matrix might be confusing for beginners.

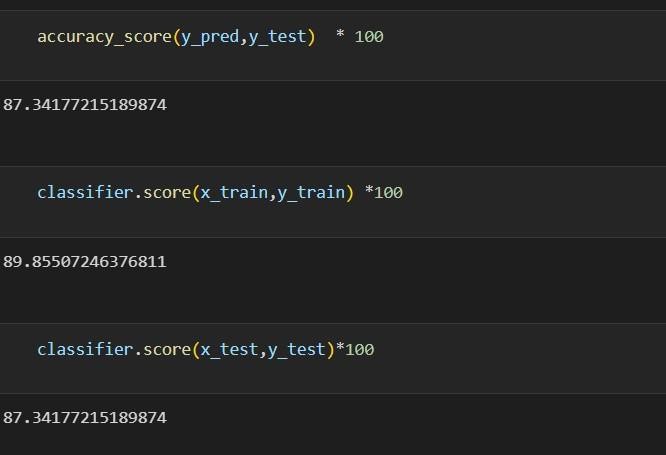
|  |  |  |
| --- | --- | --- |
| **S.No.** | **Parameter** | **Values** |
| 1. | Metrics | **Classification Model:**  Confusion Matrix – [115,15,15,92]  Accuracy Score-87.34  Classification Report – 89.85 |
| 2. | Tune the Model | Hyperparameter Tuning - 88.22  Validation Method - RandomizedCV |



#### Fig. No. 9.1 Confusion Matrix

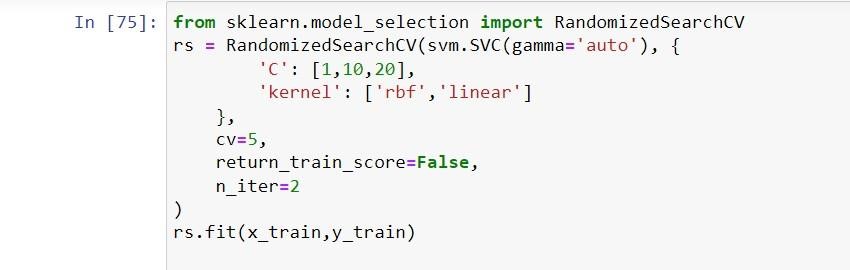
The accuracy metric is one of the simplest Classification metrics to implement, and it can be

determined as the number of correct predictions to the total number of predictions.

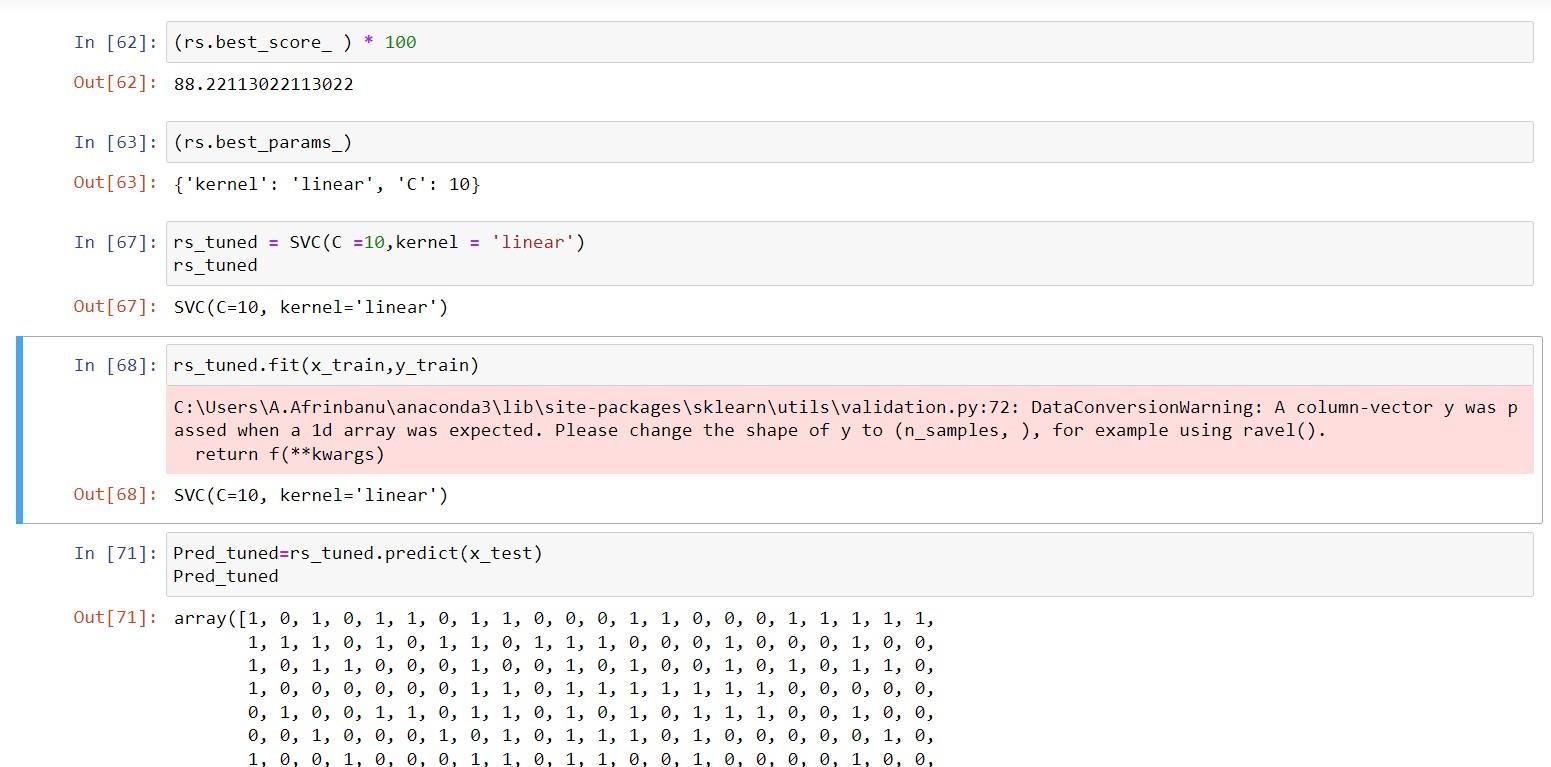


#### Fig. No. 9.2 Model Accuracy

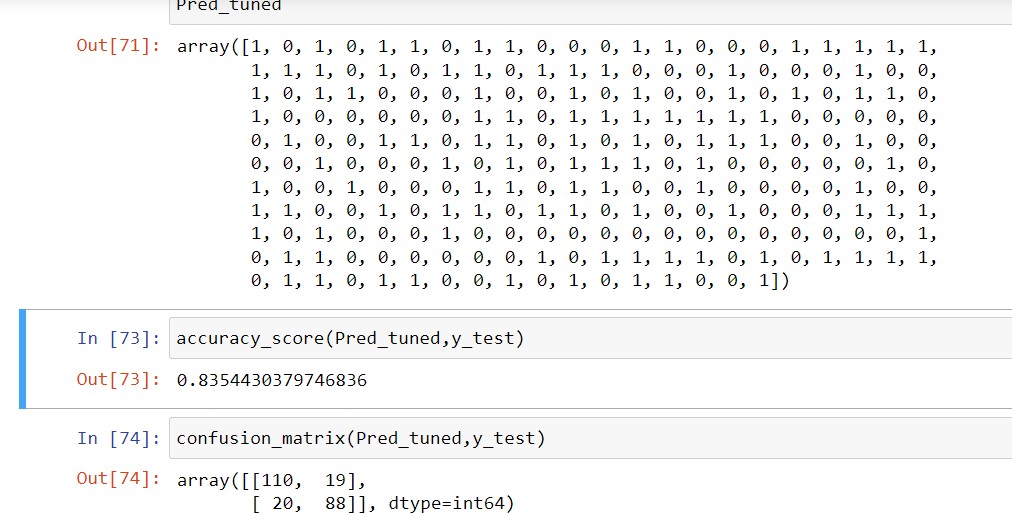
The model that has been developed using Support Vector Machine for university admit eligibility prediction provides an accuracy of 87.3 for test data and 89.85 for train data with a overall accuracy of 87.34.



#### Fig.No.9.3 Hyper Parameter Tuning



#### Fig.No.9.4 Hyper Parameter Tuning

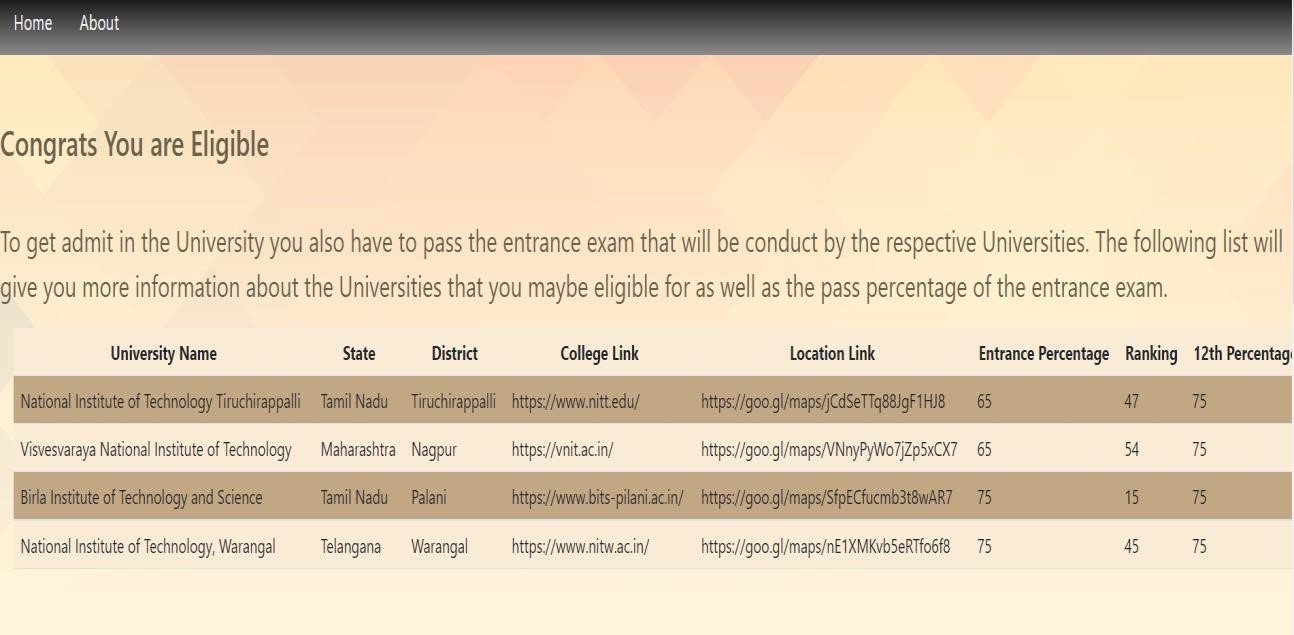


**Fig.No.9.5 Hyper Parameter Tuning**

### 9.2 OUTPUT



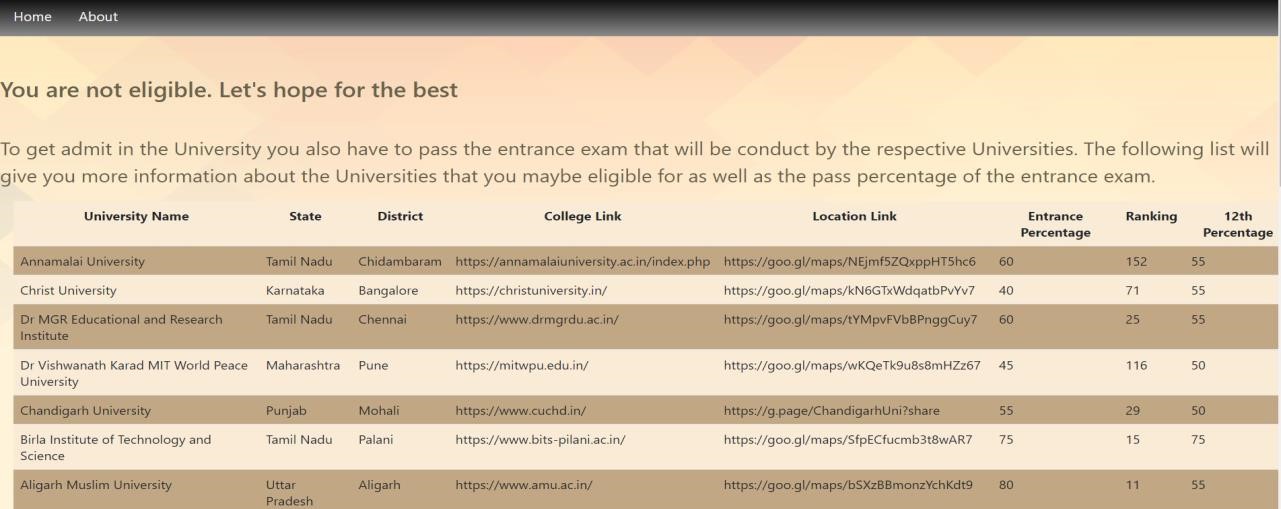
#### Fig. No. 9.6 User Input



**Fig. No. 9.7 Output for User is Eligible**



**Fig. No. 9.8 User Input**



**Fig. No. 9.9 Output for User is Not Eligible**

## CHAPTER-10

## ADVANTAGES & DISADVANTAGES

### 10.1 ADVANTAGES

* The user who are willing to get admitted in a university can choose a particular department in which they are interested to learn.
* The user can select their preferred location and the university and can ask for prediction for their eligibility for getting admitted in that university.
* The application provides them with the list of additional universities that the student is eligible to apply other than their preferred university based on their educational qualifications.
* In the list of universities, the application also provides the user with additional information like the link to the university website and location of the university and their entrance mark admission percentage

### 10.2 DISADVANTAGES

In the application output for the list of other universities the link to college website and location of the university are not directly redirected.

## CHAPTER-11

## CONCLUSION

By using this application University Admit Eligibility Predictor developed using the Machine Learning classification algorithm Support Vector Machine the output is predicted as whether the user is eligible to get an admission in the particular university or not. The students have their own preferred university, location and department. In our application the user can choose their own preferred department provided in the list in the application, location that is provided and top 20 universities based on the department is provided in the application. The user can select their own preferred department, location and university in that location. Then the user can enter their 12th percentage in the provided box then the model predicts whether the user is eligible to apply for the university or not based on the inputs given by the user. The application also provides the user with the list of other universities that the student is eligible to apply and it also provides additional information like link to the university website and location of that university. The model predicted using SVM has an accuracy 87.34.

## CHAPTER-12

## FUTURE SCOPE

In the future, the application can be enhanced in such a way that the user can get access to various departments and other universities located in India. In addition to that the user also gets the access to chatbot where the user can chat with the admin to collect more information about the universities and the information to increase their scores so that the user can improve themselves.

## CHAPTER-13

## APPENDIX

**SOURCE CODE HTML FILES:**

**Home.html**

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="wideth=device-maximum, initial-scale=1">

<link href="[https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"](https://cdn.jsdelivr.net/npm/bootstrap%405.2.2/dist/css/bootstrap.min.css) rel="stylesheet" integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi" crossorigin="anonymous"> <title>Home</title>

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font- awesome.min.css">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font- awesome/5.15.2/css/all.min.css">

<link rel="icon" type="image/x-icon" href="static/images/favicon.ico"> <link rel="stylesheet" type="text/css" href="/static/style.css">

</head>

<body>

<header><marquee>VIDYAPEETH-University Admit Eligibility Predictor</marquee></header>

<div class="topnav" id="myTopnav">

<a href="/" class="active" target="\_blank">Home</a>

<a href="/about" target="\_blank">About</a> </div>

<div class="box1">

<div class="pic">

<img src="https://thumbs.dreamstime.com/b/business-intelligence-abstract-concept-vector-

illustration-data-analysis-management-tools-enterprise-strategy-development-driven-204327386.jpg" alt="^-^" style="width:390px; height:470px; background-color: thistle;">

<p>“Keep a little fire burning; however small, however hidden.” <i>―Cormac McCarthy</i></p>

</div>

</div>

<div class="text">

<p>

<b>NAMASTE!!</b><br><br>

Hello aspiring and thirsty minds!! <br><br>

Are you eager to know about the Universities that you are eligible to get an admission with your

academic profile? Then you are at the right place.<br><br>

Lets' Quench your thirst ^-^ <br><br>

Lets' Know your academic profile, your preferred location, the department and the University

name that your thirsty mind shouting and know whether an admission is possible in the University.<br><br>

Else pen's up Here is the list of universities that you are eligible to get an admission.<br> </p>

</div>

<div class="box">

<form action="/choose\_dept" name="deptForm" method="POST">

<div class="input-icon">

<i class="fas fa-graduation-cap ic"></i>

<label class="dept" for="depts">Department :</label>

<select class="depttext" value="depts" name="depts">

<option disabled selected >Choose Department</option>

<option value="civil">Civil Engineering</option>

<option value="cse">Computer Science and Engineering</option>

<option value="ece">Electronics and Communication Engineering</option>

<option value="eee">Electrical and Eletronics Engineering</option>

<option value="mech">Mechanical Engineering</option> </select>

</div>

<input class="sub" type="submit" value="Go..!"> </form>

</div>

<div class="footer">

<h3 style="color:antiquewhite">Contact Us:</h3>

<div class="icon">

<i class="fab fa-google"><label> vidyapeethinfo@gmail.com</label></i>

</div><br><br>

<div class="icon">

<i class="fab fa-linkedin-in"><label> [www.linkedin.com/in/vidhyapeeth<](http://www.linkedin.com/in/vidhyapeeth)/label></i> </div><br><br>

<div class="icon">

<i class="fab fa-twitter"><label> https://twitter.com/Vidhyapeethinfo</label></i>

</div>

</div>

</body>

</html>

**Flask file**

import requests

from flask import Flask, render\_template, request

API\_KEY = "L2b9n\_p3zo6q3O4y9dDEELnvPDoruLIdD0lsYBSlicy2"

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}

city\_dict={ "Ali" : 18 ,"Bar" : 2 ,"Ban" : 1 ,"Ch" :3,"Chid" :22,"Chit" :0,"Hyd"

:6,"Kol":8,"Jai":15,"Moh":10,"Mum":11,"Man":7,"ND":13,

"Nag": 20,"Pal": 4,"Pali":12 ,"Pha":16 ,"Pu" :17 ,"Sa" :5 ,"Tiru":19 ,"Var":9 ,"Vel":21,"War":14 }

city\_dict1={ "Ali" : "Aligarh" ,"Bar" : "Baroda" ,"Ban" : "Bangalore" ,"Ch" :"Chennai","Chid"

:"Chidambaram","Chit":"Chitanukalan","Hyd":"Hydrabad","Kol":"Kolkata","Jai":"Jaipur","Moh":"Moha li","Mum":"Mumbai","Man":"Mangalore","ND":"New Delhi",

"Nag":"Nagpur","Pal": "Palani","Pali":"Pali" ,"Pha":"Phagwara" ,"Pu" :"Pune","Sa" :"Sawargaon"

,"Tiru":"Tiruchirappalli" ,"Var":"Varanasi","Vel":"Vellore","War":"Warangal" }

univ\_dict={"AMU" : [23 ,80],"AU" : [37 , 60],"BHU" :[28 , 60],"BITS" : [24 ,75],"CHU":[20,55],"CU" :[5,40],"DTU" : [6,60],"GGSIU" :[10,50],

"ICTM" : [16,66],"IGNOU" :[14,55],"IISB" : [18,70],"IITB" : [15,50],"IITD" : [13,75],"IITM"

:[12,55],"JH" : [17,50],"JMI" : [9,50],

"JNU" : [3 ,55],"JU" : [11,60],"LPU" :[4,55],"MGR" : [7,60],"MSUB" :[29,50],"MU" :[26 ,50 ],"NIMS" :

[0,50],"NITK" : [2 ,75],"NITT" :[25,65],

"NITW" :[19 ,75],"PES" : [38,60],"PUB" : [22 , 45],"RU":[21,50],"SIST" :[31,45],"SPPU" : [32,55],"SRM" :[30,60],"TIU" : [8,60],"UD" : [34,45],

"UH" : [35,60],"VIT":[36,55],"VKMWPU" : [33,45],"VNIT" : [1 ,65],"YCMOU" :[27,40]}

headings=("University Name","State","District","College Link","Location Link","Entrance Percentage","Ranking","12th Percentage") data=(("Aligarh Muslim University","Uttar

Pradesh","Aligarh","https://[www.amu.ac.in/"](http://www.amu.ac.in/),"https://goo.gl/maps/bSXzBBmonzYchKdt9",80,11,55), ("Annamalai University","Tamil

Nadu","Chidambaram","https://annamalaiuniversity.ac.in/index.php","https://goo.gl/maps/NEjmf5ZQxpp HT5hc6",60,152,55),

("Banaras Hindu University","Uttar

Pradesh","Varanasi","https://[www.bhu.ac.in/"](http://www.bhu.ac.in/),"https://goo.gl/maps/Rakq5Nzp9nYeBRJs8",60,6,50),

("Birla Institute of Technology and Science","Tamil Nadu","Palani","https://www.bits- pilani.ac.in/","https://goo.gl/maps/SfpECfucmb3t8wAR7",75,15,75),

("Chandigarh

University","Punjab","Mohali","https://[www.cuchd.in/"](http://www.cuchd.in/),"https://g.page/ChandigarhUni?share",55,29,50), ("Christ

University","Karnataka","Bangalore","https://christuniversity.in/","https://goo.gl/maps/kN6GTxWdqatbP vYv7",40,71,55),

("Delhi Technological University","New Delhi","New

Delhi","[http://www.dtu.ac.in/"](http://www.dtu.ac.in/),"https://goo.gl/maps/VHMGgd3YwdPJitHD8",60,36,60),

("Dr MGR Educational and Research Institute","Tamil

Nadu","Chennai","https://[www.drmgrdu.ac.in/"](http://www.drmgrdu.ac.in/),"https://goo.gl/maps/tYMpvFVbBPnggCuy7",60,25,55),

("Dr Vishwanath Karad MIT World Peace

University","Maharashtra","Pune","https://mitwpu.edu.in/","https://goo.gl/maps/wKQeTk9u8s8mHZz67" ,45,116,50),

("Guru Gobind Singh Indraprastha University","New Delhi","New

Delhi","[http://www.ipu.ac.in/"](http://www.ipu.ac.in/),"https://goo.gl/maps/YknYxUsX6m2MAXRXA",50,95,60), ("Indian Institute of Science,

Bangalore","Karnataka","Bangalore","https://iisc.ac.in/","https://goo.gl/maps/GK9gUja8pnfSpFYL7",70,

94,50),

("Indian Institute of

Technology,Bombay","Maharashtra","Mumbai","https://[www.iitb.ac.in/"](http://www.iitb.ac.in/),"https://goo.gl/maps/kbKqj6Z9b fGnERS39",50,80,60),

("Indian Institute of Technology,Madras","Tamil

Nadu","Chennai","https://[www.iitm.ac.in/"](http://www.iitm.ac.in/),"https://goo.gl/maps/73bHL5Q8RQS8yt5v9",55,153,55),

("Indian Institute of Technology,Delhi","New Delhi","New

Delhi","https://home.iitd.ac.in/","https://goo.gl/maps/3gPFUWx7fp2A99fP6",75,160,60),

("Indira Gandhi National Open University","New Delhi","New

Delhi","[http://ignou.ac.in/"](http://ignou.ac.in/),"https://goo.gl/maps/SfsYq66L9xvvDFsbA",55,71,55), ("Institute of Chemical Technology,

Mumbai","Maharashtra","Mumbai","https://[www.ictmumbai.edu.in/"](http://www.ictmumbai.edu.in/),"https://goo.gl/maps/Y6rD7yit6Kc7 dH189",66,14,55),

("Jadavpur University","West

Bengal","Kolkata","[http://www.jaduniv.edu.in/"](http://www.jaduniv.edu.in/),"https://goo.gl/maps/dLSJoT2jB61XXBUZA",60,4,45),

("Jamia Hamdard","New Delhi","New

Delhi","[http://jamiahamdard.edu/"](http://jamiahamdard.edu/),"https://goo.gl/maps/z6S684pksuWb4vFs5",50,46,50), ("Jamia Millia Islamia","New Delhi","New

Delhi","https://[www.jmi.ac.in/"](http://www.jmi.ac.in/),"https://goo.gl/maps/NCaytJGjbrEMc3WU8",50,3,50), ("Jawaharlal Nehru University","Tamil

Nadu","Pali","https://[www.jnu.ac.in/"](http://www.jnu.ac.in/),"https://goo.gl/maps/s6VEFh8SNQ391jLs9",55,10,55), ("Lovely Professional

University","Punjab","Phagwara","https://[www.lpu.in/"](http://www.lpu.in/),"https://g.page/LPUUniversity?share",55,47,60), ("Maharaja Sayajirao University of

Baroda","Gujarat","Baroda","https://[www.msubaroda.ac.in/"](http://www.msubaroda.ac.in/),"https://g.page/TheMSUB?share",50,90,40), ("Manipal

University","Rajasthan","Jaipur","https://manipal.edu/mu.html","https://goo.gl/maps/JsuTibUDEocMwkx t9",50,103,50),

("National Institute of Technology

Karnataka","Karnataka","Mangalore","https://[www.nitk.ac.in/"](http://www.nitk.ac.in/),"https://goo.gl/maps/eRFnpagJi5i4z5yS9" ,75,64,60),

("National Institute of Technology Tiruchirappalli","Tamil Nadu","Tiruchirappalli","https://[www.nitt.edu/"](http://www.nitt.edu/),"https://goo.gl/maps/jCdSeTTq88JgF1HJ8",65,47,75), ("National Institute of Technology,

Warangal","Telangana","Warangal","https://[www.nitw.ac.in/"](http://www.nitw.ac.in/),"https://goo.gl/maps/nE1XMKvb5eRTfo6f8 ",75,45,75),

("NIMS

University","Rajasthan","Chitanukalan","https://[www.nimsuniversity.org/"](http://www.nimsuniversity.org/),"https://g.page/MyNIMS?shar e",50,101,45), ("PES

University","Karnataka","Bangalore","https://pes.edu/","https://goo.gl/maps/6C2mn7kWp4JDaVz17",60,

83,50),

("Presidency University,

Bangalore","Karnataka","Bangalore","https://presidencyuniversity.in/","https://goo.gl/maps/Zvwj62U1qj

GfV113A",45,62,50),

("REVA

University","Karnataka","Bangalore","https://[www.reva.edu.in/"](http://www.reva.edu.in/),"https://goo.gl/maps/wW8DzUoECBTw BGJ79",50,151,45),

("Sathyabama Institute of Science and Technology","Tamil

Nadu","Chennai","https://[www.sathyabama.ac.in/"](http://www.sathyabama.ac.in/),"https://goo.gl/maps/QuVeBJuw7TNTMuQB7",45,43,

45),

("Savitribai Phule Pune

University","Maharashtra","Pune","[http://www.unipune.ac.in/"](http://www.unipune.ac.in/),"https://goo.gl/maps/QoBB86cSLfgA5gei 7",55,12,55),

("SRM Institute of Science and Technology","Tamil

Nadu","Chennai","https://[www.srmist.edu.in/"](http://www.srmist.edu.in/),"https://goo.gl/maps/z7BsZBg1coy96UdJ8",60,19,50), ("Techno India University","West

Bengal","Kolkata","https://[www.technoindiauniversity.ac.in/"](http://www.technoindiauniversity.ac.in/),"https://g.page/tiuwestbengal?share",60,10 9,60),

("University of Delhi","New Delhi","New

Delhi","[http://www.du.ac.in/"](http://www.du.ac.in/),"https://goo.gl/maps/FRukDt7VQXdQfUJcA",45,13,50), ("University of

Hyderabad","Telangana","Hyderabad","https://uohyd.ac.in/","https://g.page/hyderabad-central-university- hcu?share",60,10,60),

("Vellore Institute of Technology","Tamil

Nadu","Vellore","https://vit.ac.in/","https://goo.gl/maps/hgQD3vdarwFDZP1eA",55,9,55), ("Visvesvaraya National Institute of

Technology","Maharashtra","Nagpur","https://vnit.ac.in/","https://goo.gl/maps/VNnyPyWo7jZp5xCX7",6 5,54,75),

("Yashwantrao Chavan Maharashtra Open

University","Maharashtra","Sawargaon","https://[www.ycmou.ac.in/"](http://www.ycmou.ac.in/),"https://goo.gl/maps/Dc2hzzR24nu5 fjWg6",40,93,45)

)

app = Flask( name )

@app.route('/',methods=['GET']) def home(): return render\_template("Home.html") @app.route('/about') def about():

return render\_template('About.html')

@app.route('/choose\_dept', methods=['POST']) def departments(): dep = request.form["depts"] if(dep == "civil"): return render\_template("civil.html")

if(dep == "cse"): return render\_template("cse.html")

if(dep == "ece"): return render\_template("ece.html")

if(dep == "eee"):

return render\_template("eee.html") if(dep == "mech"):

return render\_template("mech.html")

@app.route('/civil',methods=['POST']) def civil(): d1\_civil=[] d2\_civil=[] u=0 e=0

percent= request.form.get('twelC',type=float) cities=request.form["citiesC"] univ = request.form["uniC"] for key in city\_dict1:

if key == cities: c=city\_dict1[key]

for key1 in univ\_dict: if key1 == univ:

x=univ\_dict[key1]

u = x[0] e = x[1]

pred = [[u,int(percent),e,0]]

payload\_scoring = {"input\_data": [{"fields": [['University Name','12th Percentage','Entrance

Percentage','Department']], "values": pred}]} response\_scoring = requests.post('https://eu-de.ml.cloud.ibm.com/ml/v4/deployments/ba6c6e10-7576-

4d5d-9c73-ca1b9b4ebf21/predictions?version=2022-11-17', json=payload\_scoring, headers={'Authorization': 'Bearer ' + mltoken})

probability = response\_scoring.json()['predictions'][0]['values'][0][0] i=0 for ds in data:

if(float(ds[7]) >= percent or c==ds[2]) and i<10:

d1\_civil.insert(i,ds) i+=1

d2\_civil = list(set(j for j in d1\_civil)) if(probability == 1):

return render\_template("output.html",prediction="Congrats You are Eligible",

headings=headings,d2=d2\_civil) else:

return render\_template("output.html",prediction="You are not eligible. Let's hope for the

best",headings=headings,d2=d2\_civil)

@app.route('/cse',methods=['POST']) def cse(): d1\_cse=[] d2\_cse=[] d=1

percent1= request.form.get('twel',type=float) cities=request.form["cities"] univ = request.form["uni"]

for key in city\_dict1: if key == cities:

c1=city\_dict1[key]

for key1 in univ\_dict:

if key1 == univ:

x=univ\_dict[key1] u1 = x[0] e1 = x[1]

pred = [[u1,int(percent1),e1,d]]

payload\_scoring = {"input\_data": [{"fields": [['University Name','12th Percentage','Entrance

Percentage','Department']], "values": pred}]} response\_scoring = requests.post('https://eu-de.ml.cloud.ibm.com/ml/v4/deployments/ba6c6e10-7576-

4d5d-9c73-ca1b9b4ebf21/predictions?version=2022-11-17', json=payload\_scoring, headers={'Authorization': 'Bearer ' + mltoken})

probability = response\_scoring.json()['predictions'][0]['values'][0][0] i1 = 0 for ds in data:

if(float(ds[7]) >= percent1 or c1==ds[2]) and i1 < 10:

d1\_cse.insert(i1,ds) i1 += 1

d2\_cse = list(set(j for j in d1\_cse)) if(probability == 1):

return render\_template("output.html",prediction="Congrats You are Eligible",

headings=headings,d2=d2\_cse) else:

return render\_template("output.html",prediction="You are not eligible. Let's hope for the

best",headings=headings,d2=d2\_cse)

@app.route('/ece',methods=['POST']) def ece(): d1\_ece=[] d2\_ece=[] d=3

percent2= request.form.get('twelEC',type=float) cities=request.form["citiesEC"] univ = request.form["uniEC"] for key in city\_dict1: if key == cities: c2=city\_dict1[key]

for key1 in univ\_dict: if key1 == univ:

x=univ\_dict[key1] u2 = x[0] e2 = x[1]

pred = [[u2,int(percent2),e2,d]]

payload\_scoring = {"input\_data": [{"fields": [['University Name','12th Percentage','Entrance Percentage','Department']], "values": pred}]}

response\_scoring = requests.post('https://eu-de.ml.cloud.ibm.com/ml/v4/deployments/ba6c6e10-7576-

4d5d-9c73-ca1b9b4ebf21/predictions?version=2022-11-17', json=payload\_scoring, headers={'Authorization': 'Bearer ' + mltoken}) probability = response\_scoring.json()['predictions'][0]['values'][0][0]

i2 = 0 for ds in data:

if(float(ds[7]) >= percent2 or c2==ds[2]) and i2 < 10: d1\_ece.insert(i2,ds) i2 += 1

d2\_ece = list(set(j for j in d1\_ece)) if(probability == 1):

return render\_template("output.html",prediction="Congrats You are Eligible",

headings=headings,d2=d2\_ece) else:

return render\_template("output.html",prediction="You are not eligible. Let's hope for the

best",headings=headings,d2=d2\_ece)

@app.route('/eee',methods=['POST']) def eee(): d1\_eee=[] d2\_eee=[] d=2

percent3= request.form.get('twelE',type=float) cities=request.form["citiesE"] univ = request.form["uniE"]

for key in city\_dict1: if key == cities: c3=city\_dict1[key]

for key1 in univ\_dict: if key1 == univ:

x=univ\_dict[key1] u3 = x[0] e3 = x[1]

pred = [[u3,int(percent3),e3,d]] payload\_scoring = {"input\_data": [{"fields": [['University Name','12th Percentage','Entrance

Percentage','Department']], "values": pred}]} response\_scoring = requests.post('https://eu-de.ml.cloud.ibm.com/ml/v4/deployments/ba6c6e10-7576-

4d5d-9c73-ca1b9b4ebf21/predictions?version=2022-11-17', json=payload\_scoring, headers={'Authorization': 'Bearer ' + mltoken})

probability = response\_scoring.json()['predictions'][0]['values'][0][0] i3=0 for ds in data:

if(float(ds[7]) >= percent3 or c3==ds[2]) and i3 < 10: d1\_eee.insert(i3,ds) i3 += 1

d2\_eee = list(set(j for j in d1\_eee)) if(probability == 1): return render\_template("output.html",prediction="Congrats You are Eligible",

headings=headings,d2=d2\_eee) else:

return render\_template("output.html",prediction="You are not eligible. Let's hope for the

best",headings=headings,d2=d2\_eee)

@app.route('/mech',methods=['POST']) def mech(): d1\_mech=[] d2\_mech=[] d=4

percent4= request.form.get('twelM',type=float) cities=request.form["citiesM"] univ4 = request.form["uniM"] for key in city\_dict1:

if key == cities: c4=city\_dict1[key]

for key1 in univ\_dict:

if key1 == univ4: x=univ\_dict[key1] u = x[0]

e = x[1]

pred = [[u,int(percent4),e,d]]

payload\_scoring = {"input\_data": [{"fields": [['University Name','12th Percentage','Entrance

Percentage','Department']], "values": pred}]} response\_scoring = requests.post('https://eu-de.ml.cloud.ibm.com/ml/v4/deployments/ba6c6e10-7576-

4d5d-9c73-ca1b9b4ebf21/predictions?version=2022-11-17', json=payload\_scoring, headers={'Authorization': 'Bearer ' + mltoken})

probability = response\_scoring.json()['prediction'][0]['values'][0][0] i4=0 for ds in data:

if(float(ds[7]) >= percent4 or c4==ds[2]) and i4 < 10:

d1\_mech.insert(i4,ds) i4 += 1

d2\_mech = list(set(j for j in d1\_mech)) if(probability == 1):

return render\_template("output.html",prediction="Congrats You are Eligible",

headings=headings,d2=d2\_mech) else:

return render\_template("output.html",prediction="You are not eligible. Let's hope for the

best",headings=headings,d2=d2\_mech)

if name == ' main ':

app.run(debug=True)

### ipynb file

# # Import Libraries

import pandas as pd import numpy as np import seaborn as sns import matplotlib.pyplot as mtp import sklearn

from scipy.stats import iqr

df = pd.read\_csv(r"C:\Users\Laptop\Desktop\flask-example\project example\university.csv") df.head()

df1 = df df1.head()

df1.isnull().sum() df1.shape df1.describe()

# # Univariate Analysis

sns.distplot(df1['12th Percentage']) #univariate

df = df1.drop(['District'],axis=1) df.head()

# # Descriptive Statistics df1.mean() df1.median() df1.mode() df1.var()

df1.std()

df1.min()

q = df1.quantile([0.75,0.25]) q

iqr = q.iloc[0] - q.iloc[1] iqr

u = q.iloc[0] + (1.5 \*iqr) u

l = q.iloc[1] - (1.5\*iqr) l

print(df1.skew()) # # outliers sns.boxplot(df1['Entrance Percentage'])

# # handling outliers df1['Entrance Percentage'] = np.where(df1['Entrance Percentage'] >70,40,df1['Entrance Percentage']) sns.boxplot(df1['Entrance Percentage'])

# # Encoding

from sklearn.preprocessing import LabelEncoder le = LabelEncoder() df['University Name'] = le.fit\_transform(df1['University Name']) df['District'] = le.fit\_transform(df1['District']) df['Department'] = le.fit\_transform(df1['Department']) df['Output'] = le.fit\_transform(df1['Output'])

df.head()

1. = df.iloc[:,1:5]

x.head()

1. = df.iloc[:,5:6] y.head()

# # Scaling

from sklearn.preprocessing import StandardScaler sc = StandardScaler() x\_scaled = sc.fit\_transform(x) x\_scaled

# # Splitting dataset into train and test

from sklearn.model\_selection import train\_test\_split x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,test\_size = 0.3,random\_state =0) x\_train

y\_test

# # Training And Test the model

from sklearn.svm import SVC # "Support vector classifier" classifier = SVC(kernel='linear', random\_state=0) classifier.fit(x\_train, y\_train)

#Predicting the test set result

y\_pred= classifier.predict(x\_test) y\_pred

#Creating the Confusion matrix

from sklearn.metrics import confusion\_matrix,accuracy\_score cm= confusion\_matrix(y\_test, y\_pred) cm

accuracy\_score(y\_pred,y\_test) \* 100

classifier.score(x\_train,y\_train) \*100 classifier.score(x\_test,y\_test)\*100

import pickle

pickle.dump(classifier,open("project.pkl","wb"))

### GITHUB & PROJECT DEMO LINK

GitHub Link - https://github.com/IBM-EPBL/IBM-Project-20939-1659767325.git

Project Demo Link - https://youtu.be/7DHUPKit0hs

#### REFERENCES

1. Vandit Manish Jain, Rihaan Satia**, “**College Admission Prediction Using Ensemble Machine Learning Models”, December 2021
2. Dr. Arunakumari B. N, Vishnu Sastry H K et.al, “An Automated Prediction Model for College Admission System”, June 2021
3. Sara Aljasmi, Ali Bou Nassif et.al., “Graduate Admission Prediction Using Machine Learning”, October 2020
4. Md. Omaer Faruq Goni, Md. Abu Ismail Siddique et.al., “Graduate Admission Chance Prediction Using Deep Neural Network”, September 2020
5. Mr. Sachin Bhoite, Prof. Dr. Ajit More, “Engineering & Technology Admission Analysis and Prediction”, February 2020
6. Anil B, Akram Pasha, Aman Kumar Singh et.al., “Multiple Machine Learning Classifiers for Student’s Admission to University Prediction”, May 2019
7. Kiran Kumari, Meet Kataria et.al., “CAPSLG: College Admission Predictor and Smart List Generator”, January 2019