A Project Report on

Identifying Patterns And Trends In Campus Placement Data Using Machine Learning

**by**

**PINJARI SHAMSHEER (21AT5A0419)**

**Under the Guidance of**

**Dr.T.Tirupal (M.Tech.,Ph.D)**

**Associate Professor**

****

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**G. PULLAIAH COLLEGE OF ENGINEERING AND TECHNOLOGY**

**(Autonomous)**

(Approved by AICTE | NAAC Accreditation with ‘A’ Grade | Accredited by NBA (ECE, CSE, EEE, CE) |

Permanently Affiliated to JNTUA)

ACKNOWLEDGEMENTS

I Am extremely grateful to Chairman, Sri G. M. Mohan Kumar, of G. Pullaiah College of Engineering and Technology, Kurnool, Andhra Pradesh for their good blessings. I owe indebtedness to our Principal Dr. C. Srinivasa Rao, M.E., Ph.D. G. Pullaiah College of Engineering and Technology, Kurnool for providing us the required facilities. I Would like to express our deep sense of gratitude and our sincere thanks to HOD Dr. T. TIRUPAL M. Tech., Ph.D. Department of Electronics and Communication Engineering, G. Pullaiah College of Engineering and Technology, Kurnool for providing the necessary facilities and encouragement towards the project work. I thank our project supervisor, Dr. T. TIRUPAL M. Tech., Ph.D.. for his guidance, valuable suggestions and support in the completion of the project.

**Contents**

**Abstract** iv

**Contents** v

**List of Figures and Tables** vi

**CHAPTER 1 INTRODUCTION 1**

**CHAPTER 2 PROJECT FLOW 6**

2.1 Data Collection

2.2 Data Preperation

2.3 Exploratory data analysis

2.4 Model Building

**CHAPTER 3 PROJECT CODE 12**

**CHAPTER 4 HTML AND FLASK CODE 22**

**CHAPTER 5 OUTPUTS 24**

**CHAPTER 5 APPLICATIONS 25**

**CHAPTER 6 PHOTOS 26**

ABSTRACT

Campus recruitment is a strategy for sourcing, engaging and hiring young talent for internship and entry-level positions. College recruiting is typically a tactic for medium- to large-sized companies with high-volume recruiting needs, but can range from small efforts (like working with university career centers to source potential candidates) to large-scale operations (like visiting a wide array of colleges and attending recruiting events throughout the spring and fall semester).Campus recruitment often involves working with university career services centers and attending career fairs to meet in-person with college students and recent graduates.Our solution revolves around the placement season of a Business School in India. Where it has various factors on candidates getting hired such as work experience,exam percentage etc., Finally it contains the status of recruitment and remuneration details.We will be using algorithms such as KNN, SVM and ANN. We will train and test the data with these algorithms. From this the best model is selected and saved in .pkl format. We will be doing flask integration and IBM deployment.

**CHAPTER 1**

**INTRODUCTION TO MACHINE LEARNING**

Machine learning is a subfield of artificial intelligence that involves the development of algorithms and statistical models that enable computers to improve their performance in tasks through experience. These algorithms and models are designed to learn from data and make predictions or decisions without explicit instructions. There are several types of machine learning, including supervised learning, unsupervised learning, and reinforcement learning. Supervised learning involves training a model on labeled data, while unsupervised learning involves training a model on unlabeled data. Reinforcement learning involves training a model through trial and error. Machine learning is used in a wide variety of applications, including image and speech recognition, natural language processing, and recommender systems.

Machine learning can be broadly categorized into three main types: supervised learning, where models are trained on labeled data to make predictions or classifications; unsupervised learning, which involves finding patterns or structures in unlabeled data through techniques like clustering and dimensionality reduction; and reinforcement learning, wherein agents learn to make sequential decisions by interacting with an environment and receiving feedback. These types encompass a wide range of algorithms and approaches that form the foundation of modern AI systems and applications across various domains.

**CHAPTER 2**

**Project Flow:**

* **Data collection**:

There are many popular open sources for collecting the data. Eg: kaggle.com Link: <https://www.kaggle.com/code/neesham/prediction-of-placements/data>

* **Data Preparation**:

The download data set is not suitable for training the machine learning model as it might have so much randomness so we need to clean the dataset properly in order to fetch good results. This activity includes the following steps

* **Handling Missing data**:

Handling missing values is a crucial step in data preprocessing and analysis. Missing values can occur in datasets due to various reasons, such as data entry errors, equipment malfunction, or participants not providing certain information. Dealing with missing values appropriately is essential to avoid biased or incorrect results in data analysis. Here are some common methods for handling missing values

Deletion of missing values

Imputation of missing values

Mean, Median, or Mode

Create indicator variables

Domain-specific knowledge

* **Handling outliers:**

Outliers are the observations in a dataset that deviate significantly from the rest of the data. In any data science project, it is essential to identify and handle outliers, as they can have a significant impact on many statistical methods, such as means, standard deviations, etc., and the performance of ML models.

* **Handling Categorical data:**

Handling categorical data is an important part of data preprocessing and analysis. Categorical data represents qualitative variables with discrete categories or labels

some popular methods in handling categorical data include:

label encoding one hot encoding

**Exploratory Data Analysis**:

Exploratory Data Analysis (EDA) is an approach to analysing and summarizing large datasets to gain insights and understand the underlying patterns, relationships, and trends in the data

**Visual Analysis**:

Visual analysis is the process of using visual representations, such as charts, plots, and graphs, to explore and understand data.

* **Univariate analysis:**

In simple words, univariate analysis is understanding the data with a single feature.

* **Bivariate analysis:**

Count plot is used here. As a 1st parameter we are passing x value and as a 2nd parameter we are passing hue value.

* **Multivariate analysis:**

In simple words, multivariate analysis is to find the relation between multiple features.

### Scaling The Data:

### Scaling data is an essential preprocessing step in many machine learning algorithms, especially those that involve distance-based calculations or gradient-based optimization. Scaling ensures that all features in the dataset are on a similar scale, preventing one feature from dominating others due to its larger magnitude.

Min-Max Scaling (Normalization)

Standardization (Z-score Scaling)

Robust Scaling

Max Abs Scaling

### Splitting The Data into Train and Test:

### Splitting the dataset is crucial for ML model training and evaluation. It involves creating three subsets: training, used to train the model; validation (optional), used for hyperparameter tuning; and test, to assess model generalization. The typical split ratio is 60-80% for training, 10-20% for validation (if used), and 10-20% for testing. Data should be split randomly and represent a fair distribution of samples. Consistency must be maintained to avoid data duplication. Libraries like Scikit-learn aid in easy and randomized dataset splitting, while k-fold cross-validation is used for more reliable performance estimate.

### Model Building:

### Building a model in machine learning is creating a mathematical representation by generalizing and learning from training data. Then, the built machine learning model is applied to new data to make predictions and obtain results.

### Training The Model in Multiple Algorithms:

We can train our data on different algorithms. For this project we are applying four classification algorithms. The best model is saved based on its performance.

**SVM model:**

A function named Support vector machine is created and train and test data are passed as the parameters. Inside the function, SVMClassifier algorithm is initialized and training data is passed to the model with. Fit () function. Test data is predicted with. predict () function and saved in a new variable. For evaluating the model, a confusion matrix and classification report is done.

**KNN model:**

A function named KNN is created and train and test data are passed as the parameters. Inside the function, KNeighborsClassifier algorithm is initialized and training data is passed to the model with. fit () function. Test data is predicted with. predict () function and saved in new variable. For evaluating the model, confusion matrix and classification report is done.

**Artificial neural network model:**

We will also be using a neural network to train the model.

### Performance Testing & Hyperparameter Tuning:

### Performance testing and hyperparameter tuning are crucial steps in the development of machine learning models.Performance testing involves evaluating the model's performance on a separate test dataset to assess its ability to generalize to new, unseen data. This helps identify issues like overfitting or underfitting and ensures the model's reliability in real-world scenarios.Hyperparameter tuning is the process of selecting the best hyperparameters for the model to achieve optimal performance. Hyperparameters are configuration settings that are not learned during training, such as learning rate, number of layers, or number of trees.Grid search and random search are common techniques for hyperparameter tuning, where different combinations of hyperparameters are tested to find the best combination.Both performance testing and hyperparameter tuning are iterative processes that require experimentation and careful analysis to build an accurate and robust machine learning model.

### Model Deployment:

Model deployment is the process of deploying a machine learning model into a production environment, where it can be used to make predictions on new data. It involves taking the trained model and integrating it into an application or system, and making it available to end-users.

**Save The Best Model:**

Saving the best model after comparing its performance using different evaluation metrics means selecting the model with the highest performance and saving its weights and configuration. This can be useful in avoiding the need to retrain the model every time it is needed and also to be able to use it in the future.

**Integrate With Web Framework:**

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

**CHAPTER 3**

**PROJECT CODE**

**Importing the libraries:**

import numpy as np

import pandas as pd

import os

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.neighbors import KNeighborsClassifier

from sklearn import svm

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

from sklearn import preprocessing

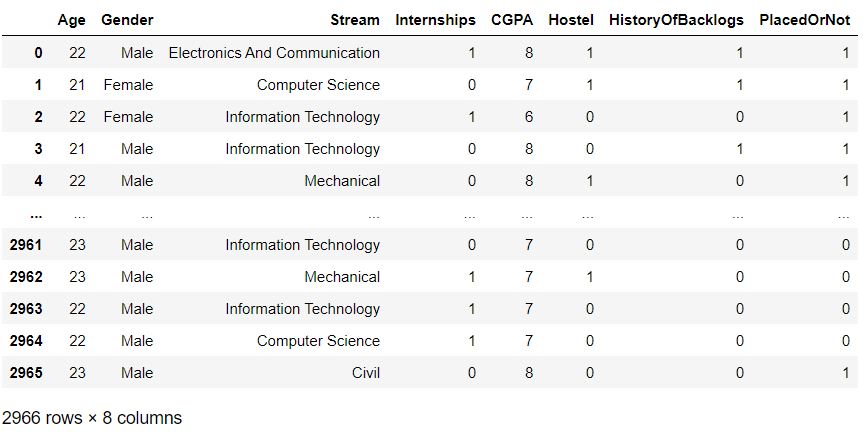
from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import cross\_val\_score

**Read or Load data:**

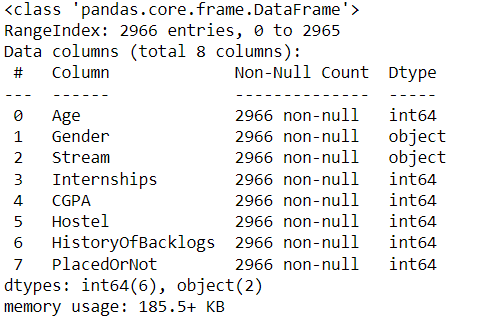
data=pd.read\_csv("collegePlace.csv")

data



**The info() method prints information about the DataFrame.**

data.info()



**The purpose of the given code is to create a transformation plot for the 'Age' feature using the logarithmic transformation.**

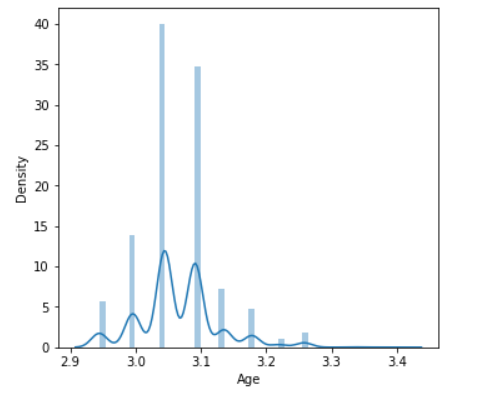
def transformationplot(feature):

plt.figure(figsize=(12,5))

plt.subplot(1,2,1)

sns.distplot(feature)

transformationplot(np.log(data['Age']))



performing ordinal encoding on the 'Gender' and 'Stream' columns in the 'data' DataFrame.

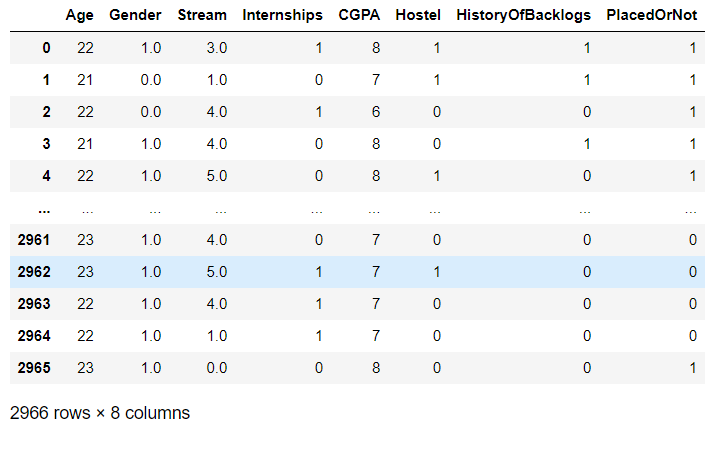
from sklearn.preprocessing import OrdinalEncoder

oe=OrdinalEncoder()

data[['Gender']]=oe.fit\_transform(data[['Gender']])

data[['Stream']]=oe.fit\_transform(data[['Stream']])

data



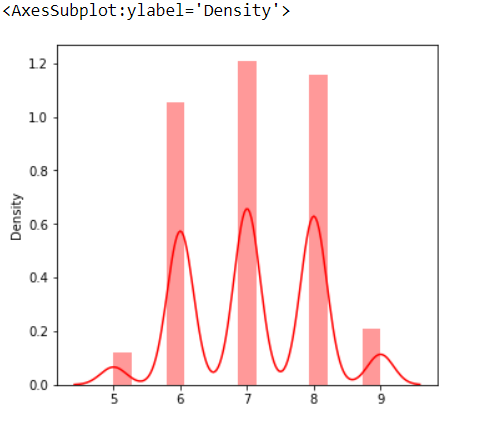
**Univariate Analysis:**

create a visualization of the distribution of the 'CGPA' column and ‘PlacedOrNot’ column from the 'data' DataFrame using a histogram.

plt.figure(figsize=(12,5))

plt.subplot(121)

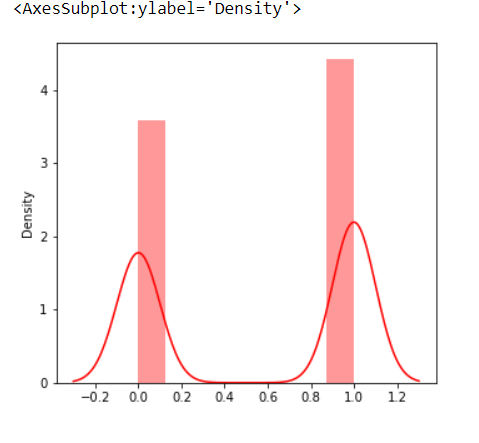
sns.distplot(data[['CGPA']],color='r')



plt.figure(figsize=(12,5))

plt.subplot(121)

sns.distplot(data[['PlacedOrNot']],color='r')



**Bivariate Analysis:**

creates a visualization with two subplots

plt.figure(figsize=(18,4))

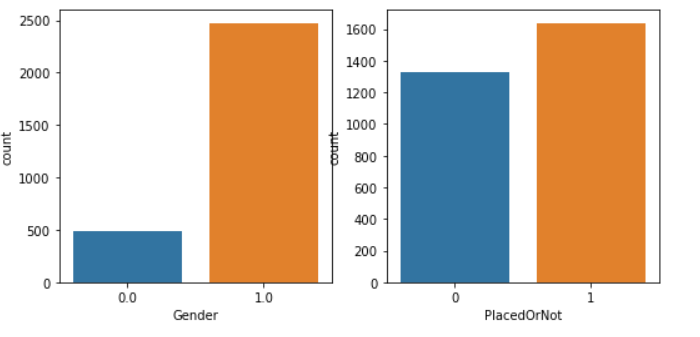
plt.subplot(1,4,1)

sns.countplot(data['Gender'])

plt.subplot(1,4,2)

sns.countplot(data['PlacedOrNot'])

plt.show()



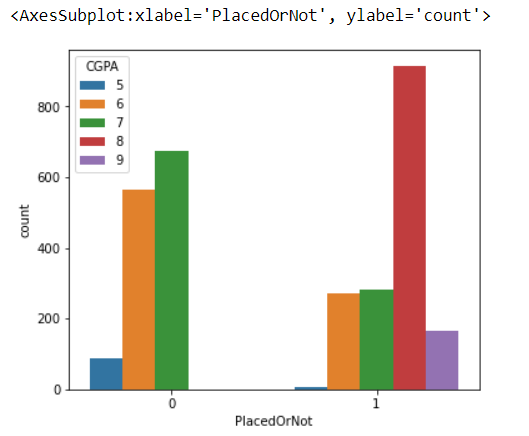
**Multivariate Analysis:**

creates a visualization with three subplots

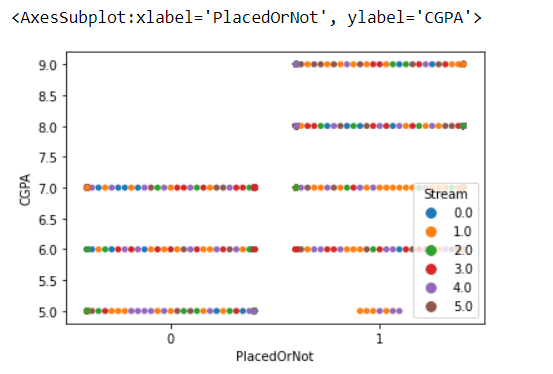
plt.figure(figsize=(20,5))

plt.subplot(131)

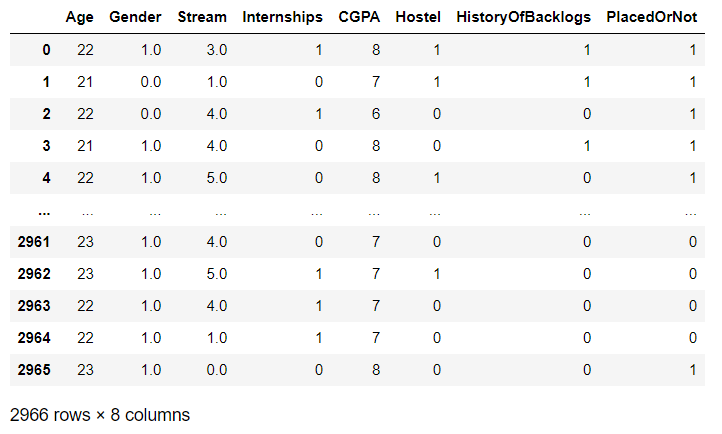
sns.countplot(data['PlacedOrNot'],hue=data['CGPA'])



sns.swarmplot(data['PlacedOrNot'],data['CGPA'],hue=data['Stream'])



data

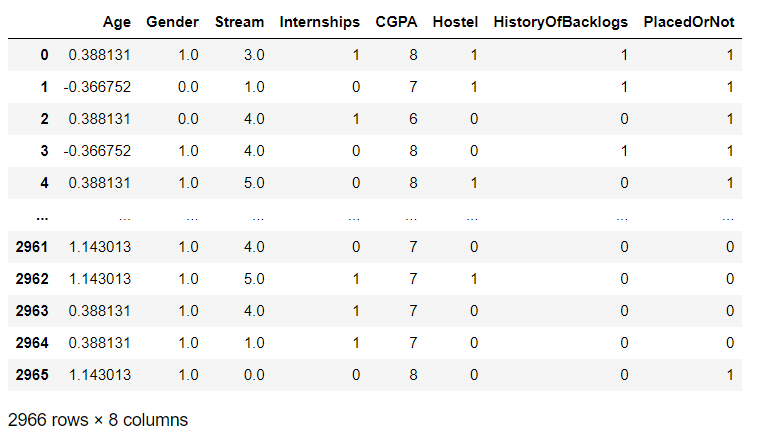


**Scaling:**

sc=StandardScaler()

data[['Age']]=sc.fit\_transform(data[['Age']])

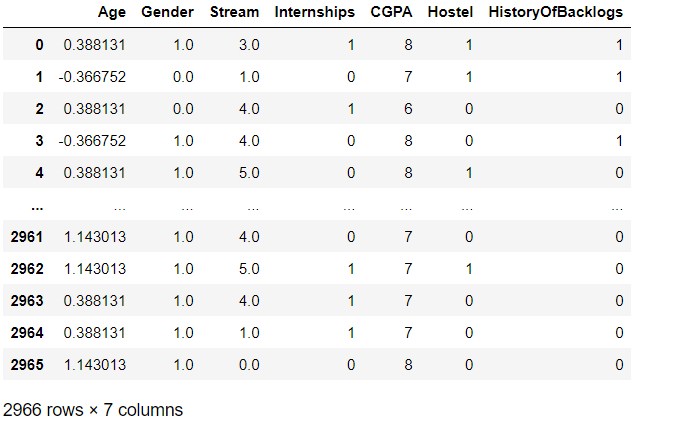
data



**Splitting the data into train and test:**

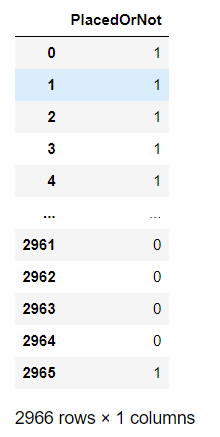
X=data.iloc[:,0:-1]

X



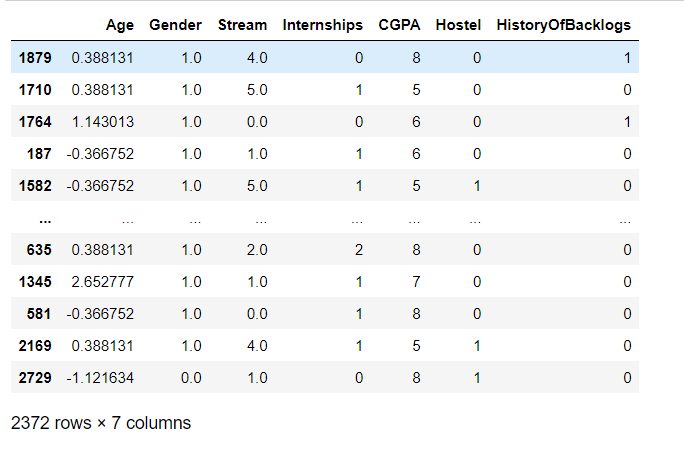
y=data.iloc[:,-1:]

y

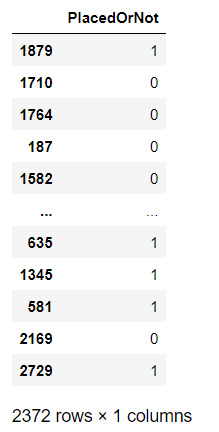


X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,y,train\_size=0.8,random\_state=16)

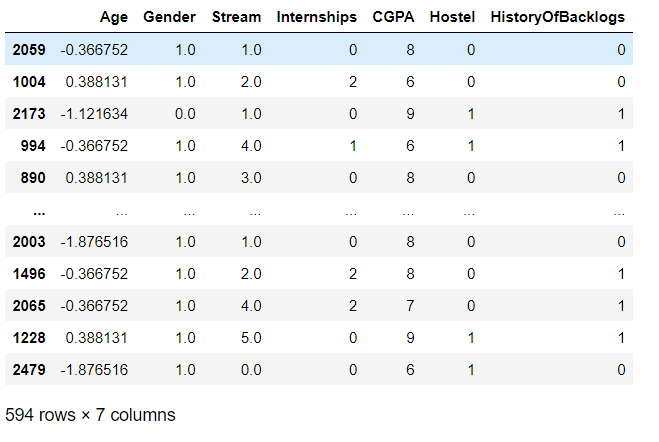
X\_train



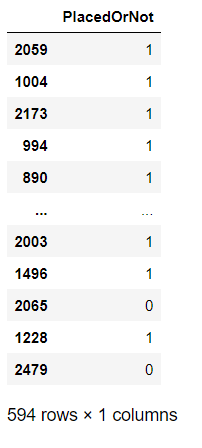
y\_train



X\_test



y\_test



**Model Building:**

**SVM**

from sklearn.svm import SVC

svm=SVC(kernel='linear')

svm.fit(X\_train,y\_train)

**OUTPUT: SVC(kernel='linear')**

pred\_test=svm.predict(X\_test)

pred\_train=svm.predict(X\_train)

train\_accuracy=accuracy\_score(pred\_train,y\_train)

test\_accuracy=accuracy\_score(pred\_test,y\_test)

print("accuracy on training data is",train\_accuracy)

print("accuracy on testing data is",test\_accuracy)

**OUTPUT: accuracy on training data is 0.7765598650927488**

**accuracy on testing data is 0.7474747474747475**

**CHAPTER 4**

**HTML code and Flask code**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css"></head>

<body>

    <section id="hero" class="d-flex-column justify-content-center">

        <div class="container">

            <div class="row justify-content-center">

                <div class ="col-xl-8">

                    <a href ="index1.html">

                    <h1>Identifying Patterns And Trends In Campus Placement Data Using Machine Learning</h1>

                    </a>

                </div>

            </div>

        </div>

    </section>

    <section id="about" class="about">

        <div class="container">

            <div class="section-title">

                <h2>Fill the details</h2>

            </div>

            <div class="row content">

                <div class="first">

                    <form action="{{url\_for('y\_predict')}}" method="POST">

                        <input type="number" id="sen1" name="sen1" placeholder="Age">

                        <input type="number" id="sen2" name="sen2" placeholder="Gender M(1),F(0)">

                        <input type="number" id="sen3" name="sen3" placeholder="Civil(0),CS(1),EEE(2),ECE(3),IT(4),Mech(5)" style="width:300px">

                        <input type="number" id="sen4" name="sen4" placeholder="Internships">

                        <input type="number" id="sen5" name="sen5" placeholder="CGPA">

                        <input type="number" id="sen6" name="sen6" placeholder="Hostel">

                        <input type="number" id="sen7" name="sen7" placeholder="Number of Backlogs">

                        <input type="submit" value="Submit">

                    </form>

                </div>

            </div>

        </div>

    </section>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>OUTPUT</title>

    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css"></head>

</head>

<body>

    <section id="hero" class="d-flex flex-column justify-content-center">

        <div class="container">

            <div class="row justify-content-center">

                <div class="col-xl-8">

                    <h1>The prediction is : {{y}}</h1>

                    <h3> 0 represents Not-placed</h3>

                    <h3> 1 represents Placed</h3>

                </div>

            </div>

        </div>

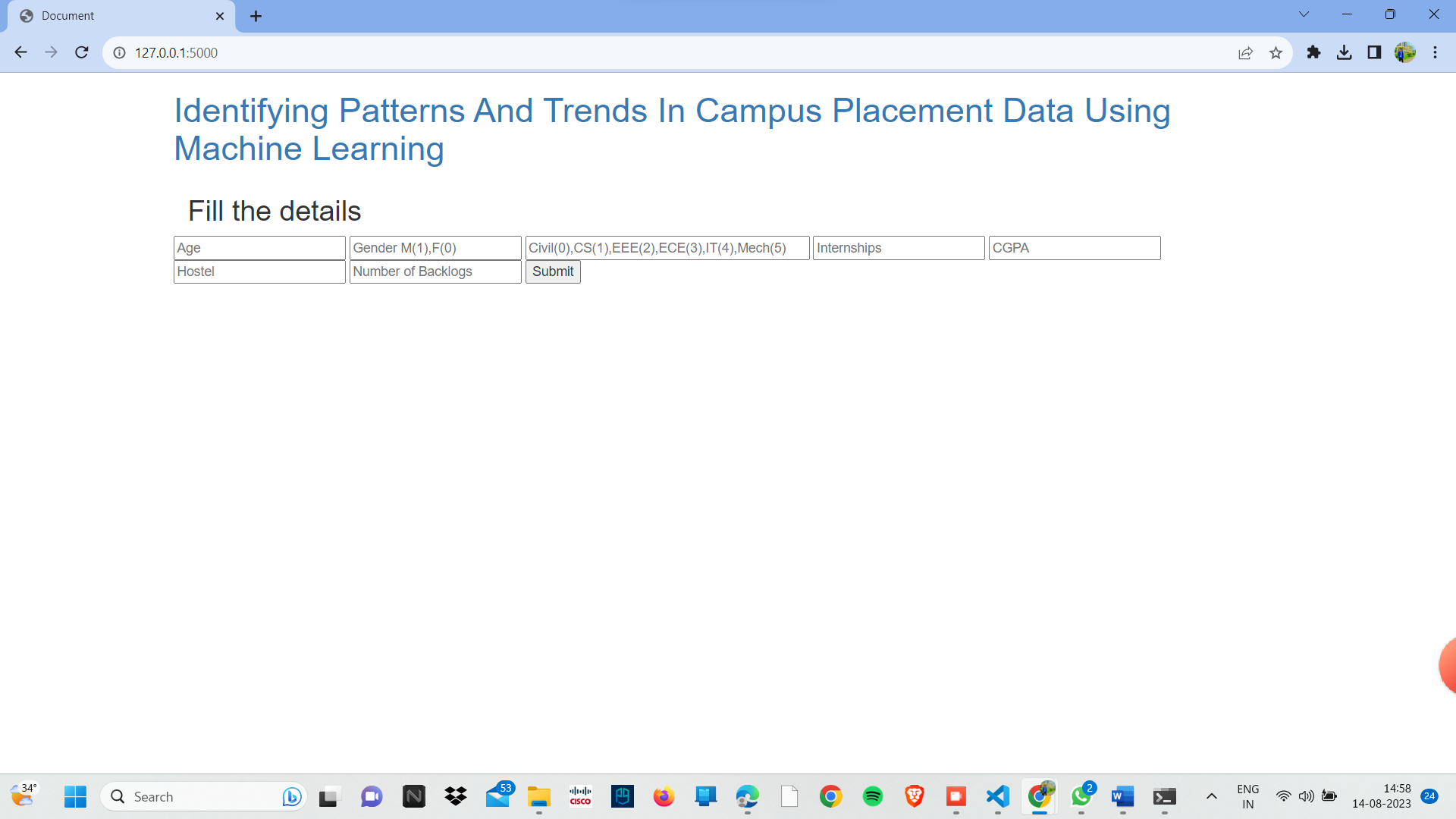
    </section>

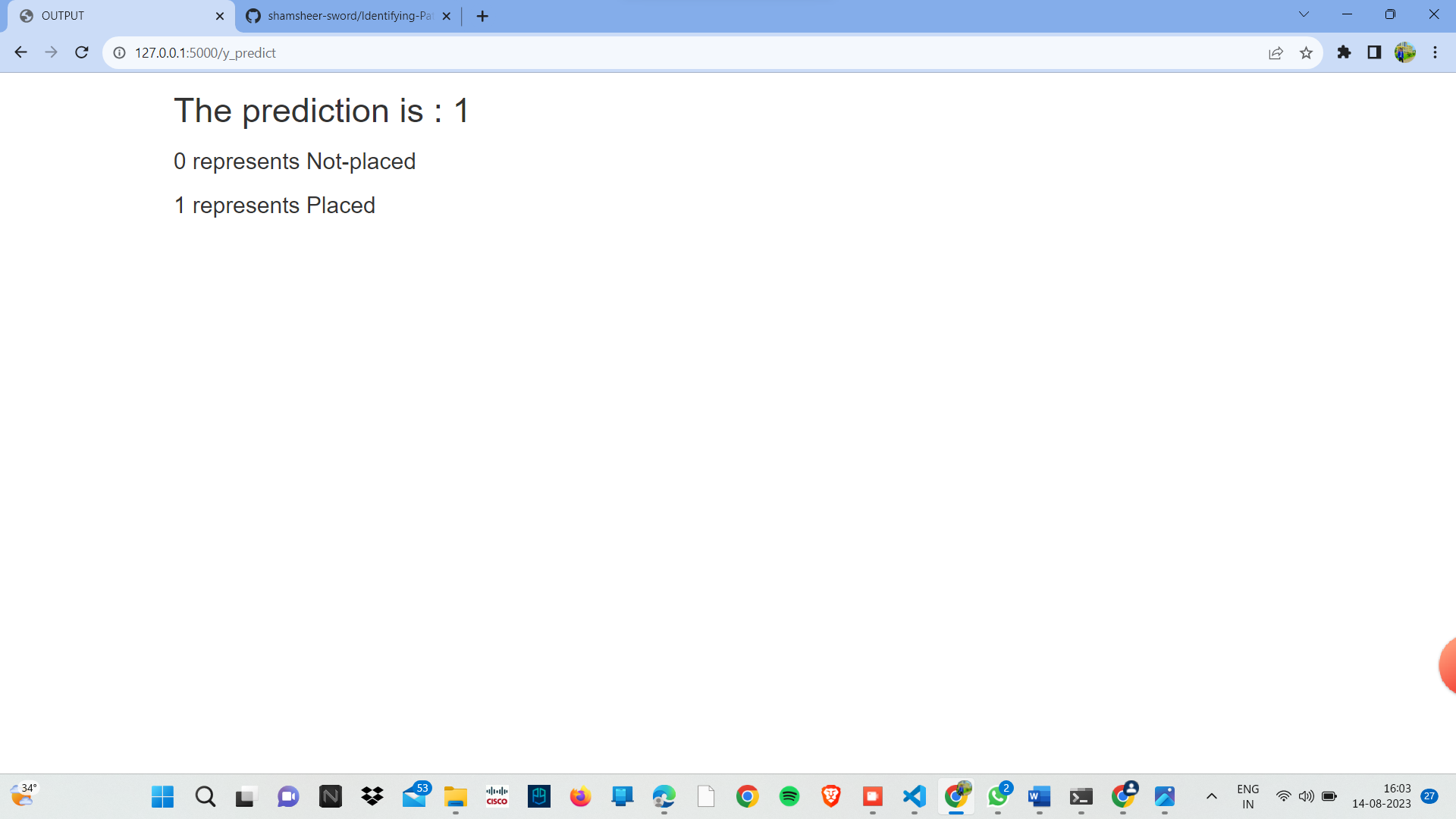
</body>

</html>

**CHAPTER 5**

**Outputs**





**CHAPTER 6**

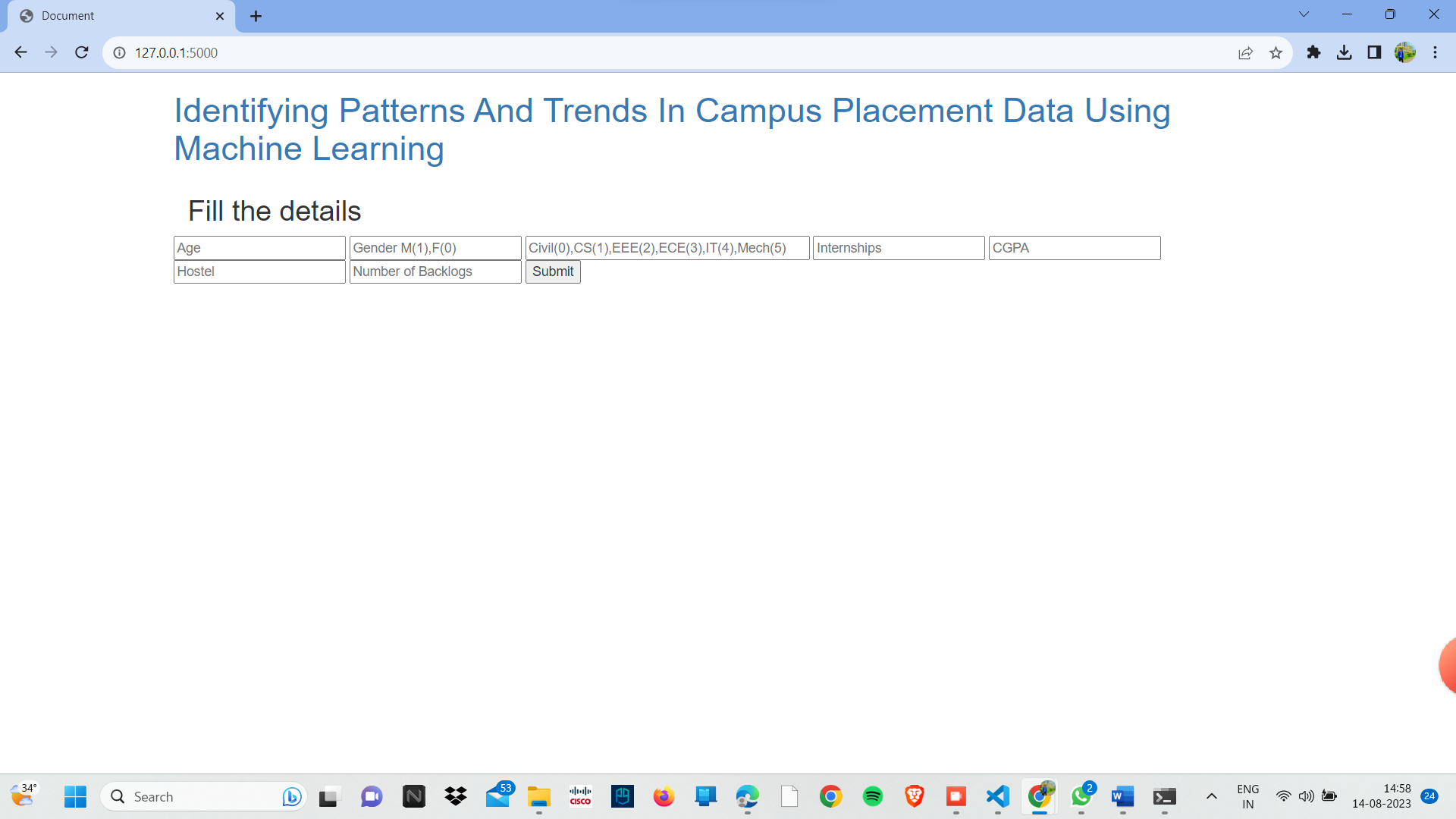
**APPLICATIONS**

The project "Identifying Patterns and Trends in Campus Placement Data Using Machine Learning" aims to leverage advanced machine learning techniques to analyze and uncover insights from a dataset of campus placement records. By applying various data preprocessing, feature engineering, and predictive modeling methods, the project seeks to identify meaningful patterns and trends in factors influencing student placements, such as academic performance, skillsets, and demographic information. The project's outcomes will not only provide valuable insights for educational institutions to enhance their placement strategies but also showcase the power of machine learning in extracting actionable insights from complex real-world datasets.

**CHAPTER 7**

**Photos**

**Fig1:**



**Fig2:**

