Identifying Patterns and Trends in Campus Placement Data using Machine Learning

TEAM SIZE : 4

TEAM LEADER : NAVEEN S (20UCS4725)

TEAM MEMBERS : SHREEGANTH D (20UCS4730)

SURYA K (20UCS4737)

ANANDHANKANNAN (20UCS4706)

INTRODUCTION

Overview

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.

Purpose

The business requirements for a project aimed at "Identifying Patterns and Trends in Campus Placement Data using Machine Learning" would likely include the following:

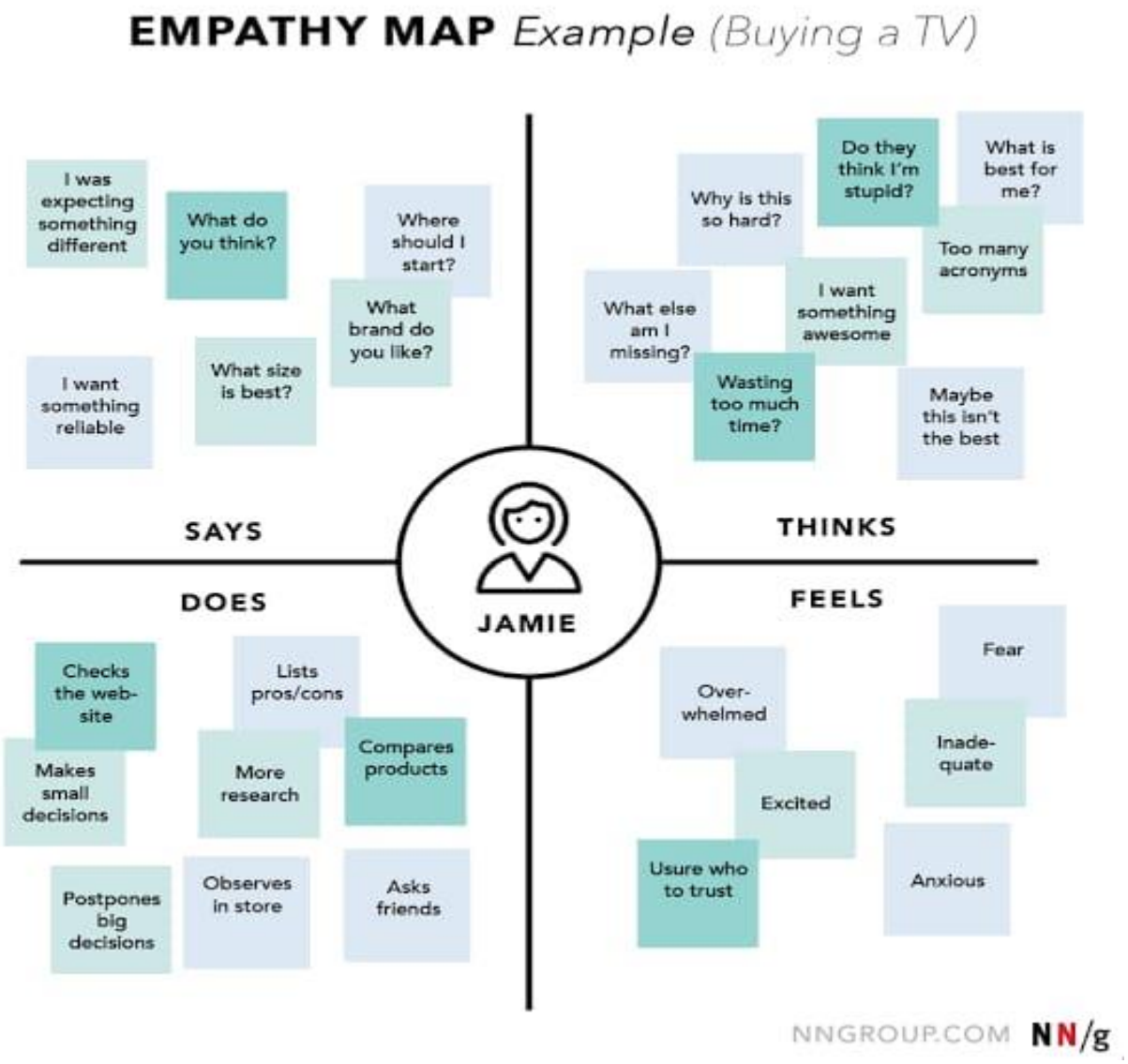
- Access to campus placement data: The project would require access to data on student performance, qualifications, and job placement outcomes. This data would need to be collected, cleaned, and prepared for analysis

- Machine learning expertise: The project would require individuals with expertise in machine learning, data science and statistical analysis to develop and implement the algorithms and models needed to analyze the data.

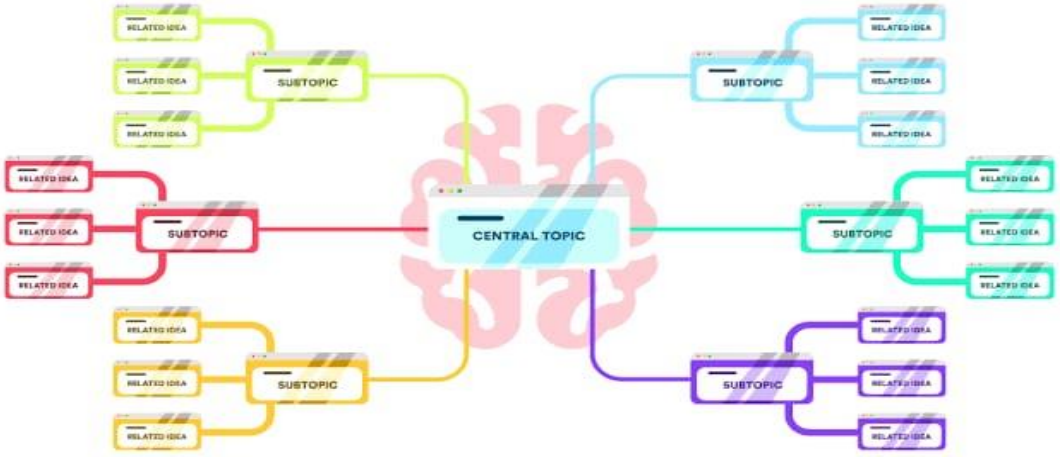
- Infrastructure for model deployment: The project would require infrastructure for deploying the models and algorithms developed, including hardware, software, and cloud-based resources.

Problem definition & design thinking

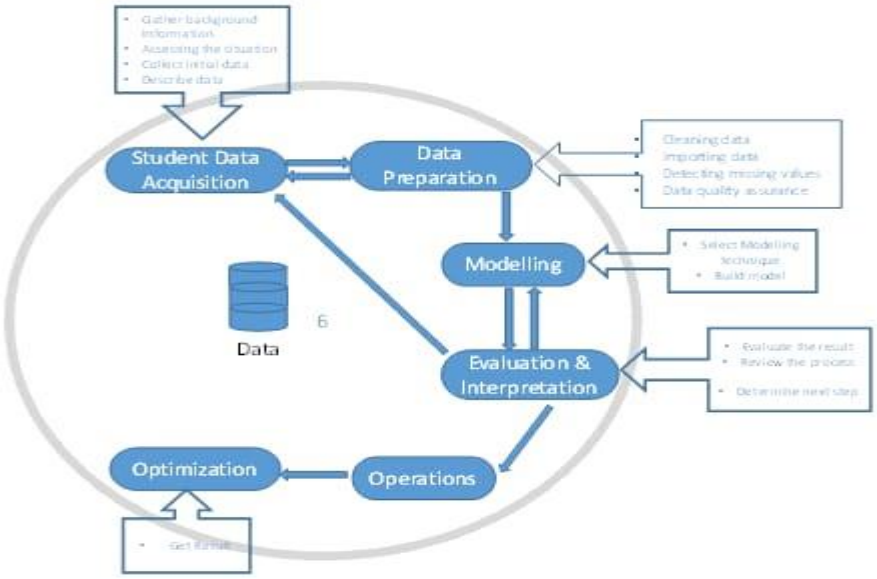
Empathy Map



Ideation & Brainstorming Map



RESULT



ADVANTAGES & DISADVANTAGES

Advantages :

- Easily predicts and analyses lot of student data set for predefined classes by using fuzzy logic
- Provide an efficient single point management system which will give all the data of the students of the college at the same place
- This system can be used in various colleges and institutes for overall growth.
- It used to predict the eligibility of student for placement so to prepare for only those.
- It used to Predicting the placement of a student gives an idea to the Placement Office as well as the student on where they stand

Disadvantages :

- studies are required to investigate new hybrid models of fuzzy classification algorithms to improve the performance of prediction

system**.**

- This system could address a wide range of problems by distilling data from any combination of education records maintenance system.
- This system is work on previous record not consider current academic record.
- Need to improved to a great extent using this prediction model in all institutions.
- This system not take diff data of different streams of engineering.

APPLICATION

Placements hold great importance for students and educational institutions. It helps a student to build a strong foundation for the professional career ahead as well as a good placement record gives a competitive edge to a college/university in the education market.

study focuses on a system that predicts if a student would be placed or not based on the student's qualifications, historical data, and experience. This predictor uses a machine-learning algorithm to give the result.

CONCLUSION

The campus placement activity is incredibly a lot of vital as institution point of view as well as student point of view. In this regard to improve the student's performance, a work has been analyzed and predicted using the classification algorithms Decision Tree and the Random forest algorithm to validate the approaches. The algorithms are applied on the data set and attributes used to build the model. The accuracy obtained after analysis for Decision tree is 84% and for the Random Forest is 86%. Hence, from the above said analysis and prediction it's better if the Random Forest algorithm is used to predict the placement results

FUTURE SCOPE

- It would of great help if we revise and update our curriculum and other extra activities for each semester in accordance with the public, private and government sector requirement. We can also predict which company picks which category of students. Make a list of skill a particular company looking for, then on the basis of that we can train our student. These traits will make prediction process more accurate.

APPENDIX

SOURCE CODE:

```
import numpy as np
import pandas as pd
import os

import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import svm
from sklearn.metrics import accuracy_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
from sklearn.model_selection import cross_val_score
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import joblib
from sklearn.metrics import accuracy_score
```

```
df = pd.read_csv(r"/content/collegePlace.csv")
df.head()
```

	Age	Gender	Stream	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot
0	22	Male	Electronics And Communication	1	8	1	1	1
1	21	Female	Computer Science	0	7	1	1	1
2	22	Female	Information Technology	1	6	0	0	1
3	21	Male	Information Technology	0	8	0	1	1
4	22	Male	Mechanical	0	8	1	0	1

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2966 entries, 0 to 2965
Data columns (total 8 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Age                 2966 non-null   int64
 1   Gender              2966 non-null   object
 2   Stream              2966 non-null   object
 3   Internships         2966 non-null   int64
 4   CGPA                2966 non-null   int64
 5   Hostel              2966 non-null   int64
 6   HistoryOfBacklogs   2966 non-null   int64
 7   PlacedOrNot         2966 non-null   int64
dtypes: int64(6), object(2)
memory usage: 185.5+ KB
```

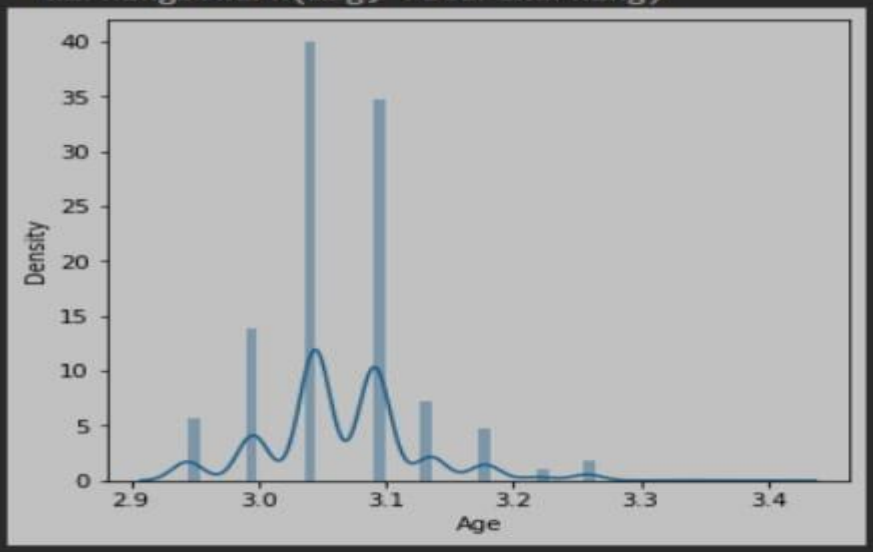
```
df.isnull().sum()
```

```
Age                 0
Gender              0
Stream              0
Internships         0
CGPA                0
Hostel              0
HistoryOfBacklogs   0
PlacedOrNot         0
dtype: int64
```

```
def transformationplot(feature):
    plt.figure(figsize=(12,5))
    plt.subplot(1,2,1)
    sns.distplot(feature)

transformationplot(np.log(df['Age']))
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/distribut
  warnings.warn(msg, FutureWarning)
```




```
df = df.replace(['Male'], [0])
df = df.replace(['Female'], [1])
```

```
df = df.replace(['Computer Science', 'Information Technology','Electronics And Communication','Mechanical','Electrical','Civil'],
                [0,1,2,3,4,5])
```

```
df = df.drop(['Hostel'], axis=1)
```

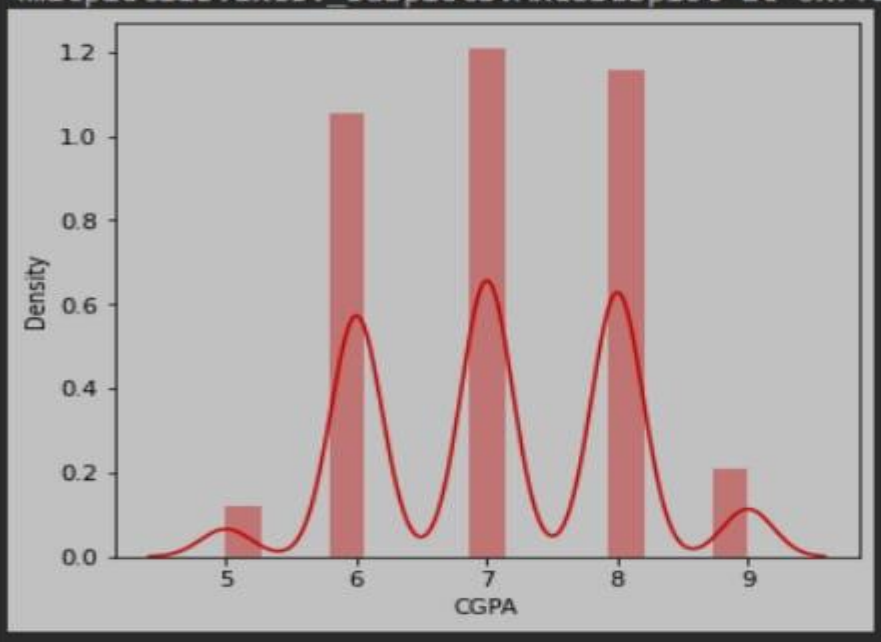
```
df
```

	Age	Gender	Stream	Internships	CGPA	HistoryOfBacklogs	PlacedOrNot
0	22	0	2	1	8	1	1
1	21	1	0	0	7	1	1
2	22	1	1	1	6	0	1
3	21	0	1	0	8	1	1
4	22	0	3	0	8	0	1
...	...	...	...	...	...	...	...
2961	23	0	1	0	7	0	0
2962	23	0	3	1	7	0	0
2963	22	0	1	1	7	0	0
2964	22	0	0	1	7	0	0
2965	23	0	5	0	8	0	1

2966 rows × 7 columns

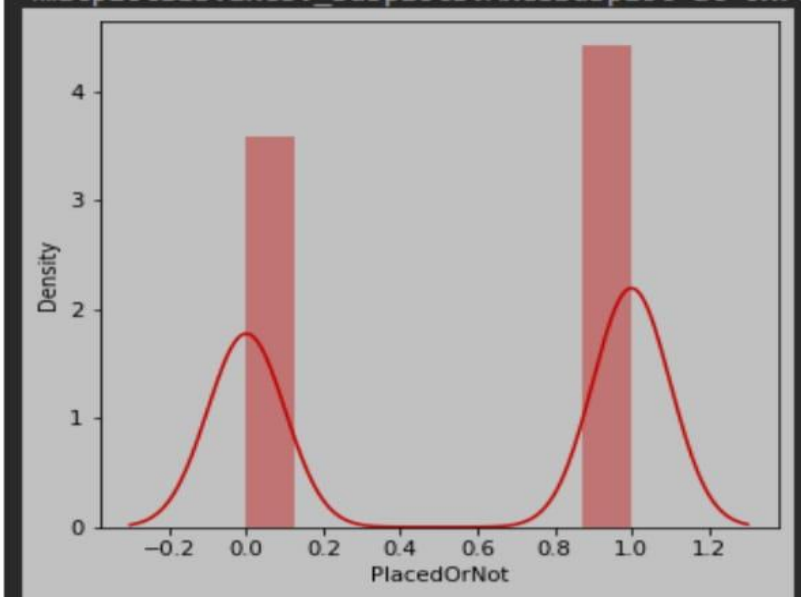
```
plt.figure(figsize=(12,5))
plt.subplot(121)
sns.distplot(df['CGPA'],color='r')
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:261
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f5463e50d00>
```



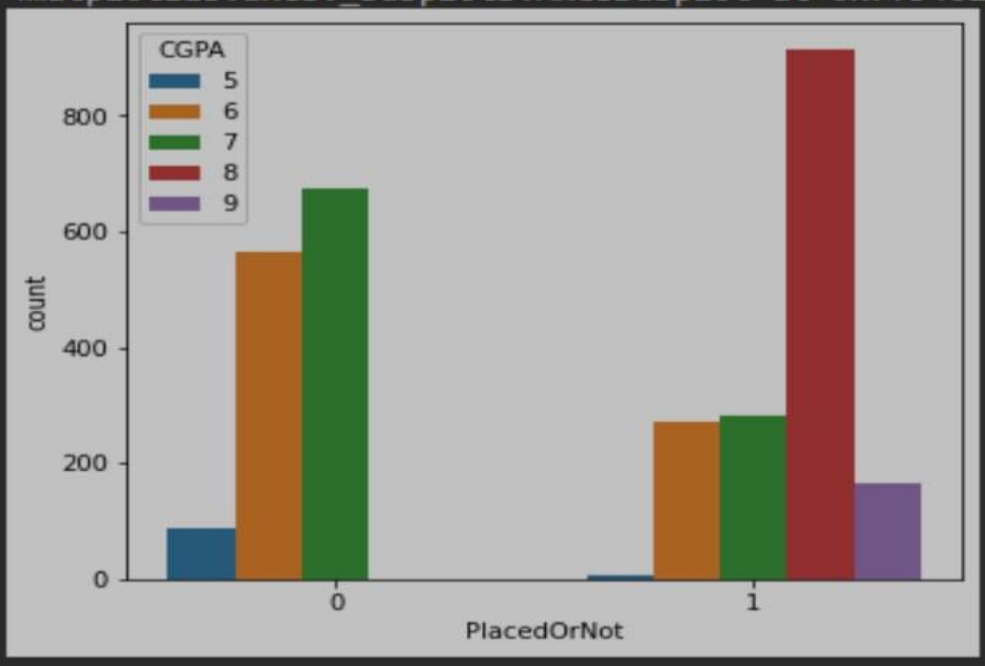

```
plt.figure(figsize=(12,5))
plt.subplot(121)
sns.distplot(df['PlacedOrNot'],color='r')
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/distributions.py:2619: Futu
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f5463d95790>
```



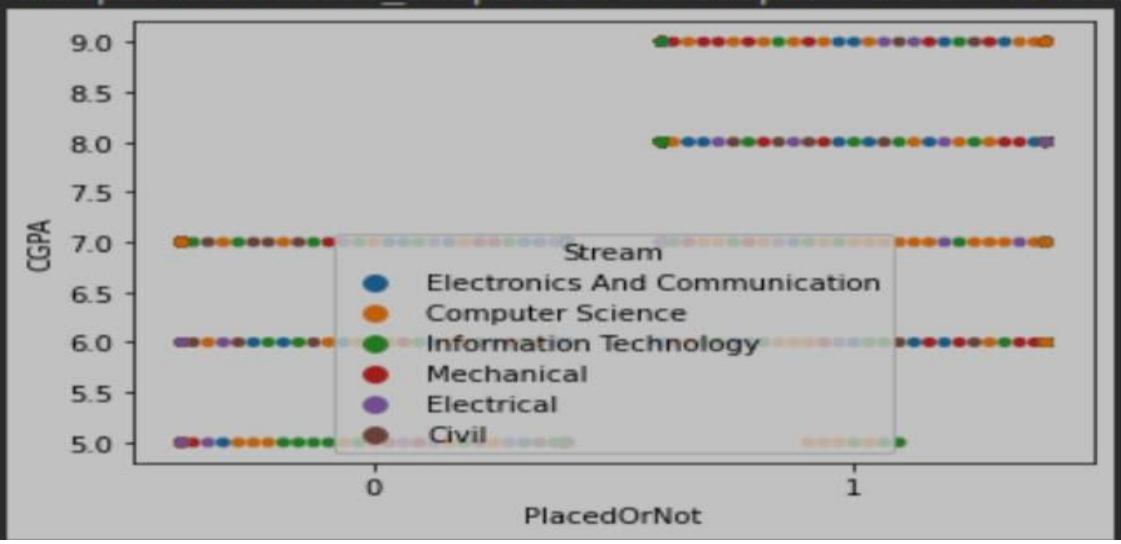
```
plt.figure(figsize=(20,5))
plt.subplot(131)
sns.countplot(df["PlacedOrNot"],hue=df['CGPA'])
```

```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36: Futur
  warnings.warn(
<matplotlib.axes._subplots.AxesSubplot at 0x7f5461cf85b0>
```



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

```
/usr/local/lib/python3.8/dist-packages/seaborn/_decorators.py:36:
  warnings.warn(
/usr/local/lib/python3.8/dist-packages/seaborn/categorical.py:129
  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.8/dist-packages/seaborn/categorical.py:129
  warnings.warn(msg, UserWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f5463d06df0>
```




```
# perfroming feature Scaling operation using standard scaller on X part of the dataset becau
# there different type of values in the columns
sc=StandardScaler()
x_bal=sc.fit_transform(x_bal)
```

```
x_bal = pd.DataFrame(x_bal,columns=names)
```

```
X = standardized_data
Y = df['PlacedOrNot']
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.2, stratify=Y, random_state=2)
```

```
classifier = svm.SVC(kernel='linear')
```

```
classifier.fit(X_train, Y_train)
```

```
SVC(kernel='linear')
```

```
X_train_prediction = classifier.predict(X_train)
training_data_accuracy = accuracy_score(X_train_prediction, Y_train)
```

```
print('Accuracy score of the training data : ', training_data_accuracy)
```

```
Accuracy score of the training data :  0.7685497470489039
```