

IDENTIFYING PATTERNS
AND TRENDS IN
CAMPUS PLACEMENT
DATA USING
MEACHINE LEARNING

SUBMITTED BY,

➤ TEAM LEADER: J.KIRUTHIGA

➤ TEAM MATES:

M . NANDHINI

T.NISHANTHI

S.NITHIYAVATHI

INTRODUCTION:

OVERVIEW

- 1.PURPOSE
- 2.ADVANTAGES
- 3.DISADVANTAGES
- 4.CONCLUSION
- 5.FUTURE SCOPE

➤ PROJECT OVERVIEW :

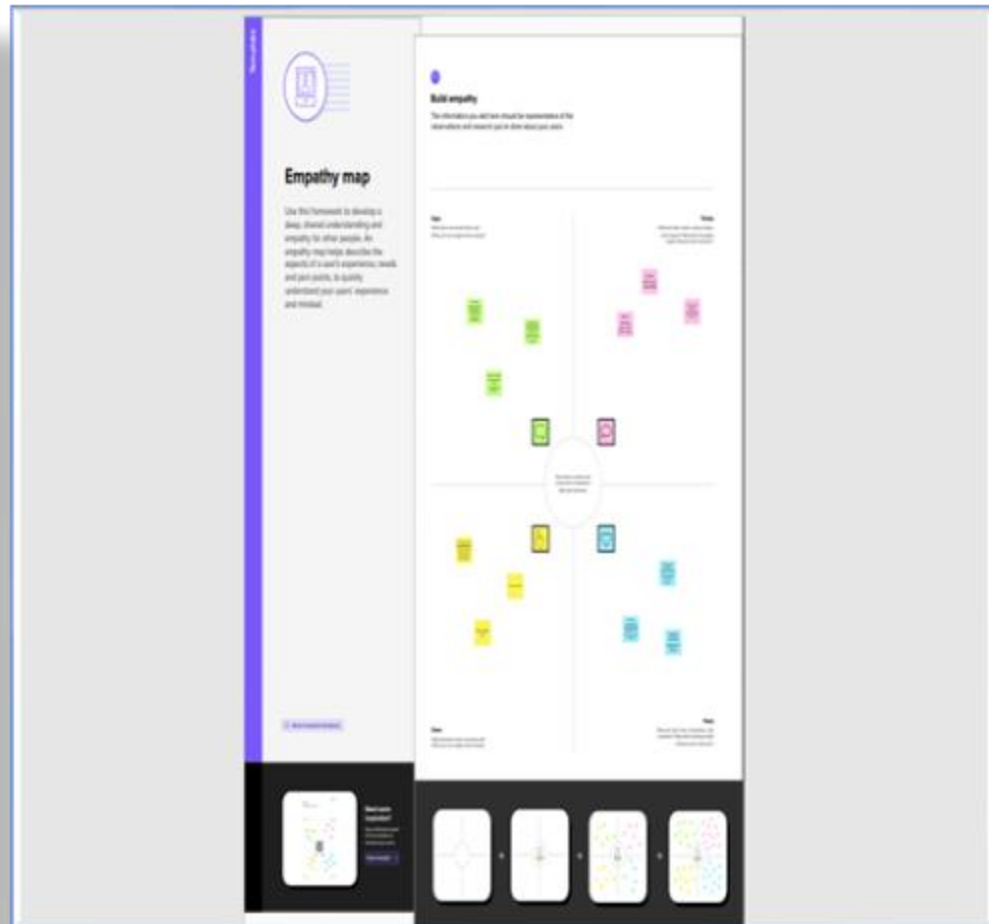
- ❖ Campus recruitment is a strategy for sourcing, engaging and hiring young talent for internship and entry level positions.
- ❖ College recruiting is typically a tactic for medium to large sized companies with high volume recruiting needs, but can range from small efforts to large scale operations.
- ❖ Campus recruitment often involves working with university career services centers and attending career fairs to meet in person with college students and recent graduates.
- ❖ 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.
- ❖ Based on this a final data set is created and the interested candidates will be registered automatically by the system.

PURPOSE:

- ❖ Campus placement or campus recruiting is a program conducted within universities or other educational institutions to provide jobs to students nearing completion of their studies .
- ❖ In this type of program, the educational institutions partner with corporations who wish to recruit from the student population .
- ❖ Placement Management System manages student information in the college with regard to placement .
- ❖ It improves existing system .
- ❖ It has the facility of maintaining the details of the student , thereby reducing the manual work .
- ❖ It will save time and energy which are spending in making reports and collecting data.
- ❖ Placement Management System can be accessed throughout the college with proper login provided.

PROBLEM DEFINITION & DESIGN THINKING:

➤ EMPATHY MAP:



IDEATION & BRAINSTORMING MAP:

2

Brainstorm solo

Have each participant begin in the "solo brainstorm space" by silently brainstorming ideas and placing them into the template. This "silent-storming" avoids group-think and creates an inclusive environment for introverts and extroverts alike. Set a time limit. Encourage people to go for quantity.

🕒 10 minutes

kiruthiga

The specification word used in the lesson matter to model	The problem as through you were describe it to a friend or colleague	
Use a simple framework when solving a new problem to address with iterative learning	Application of machine learning can bring including external application recommended	

nithavathi

intensity to help business expense product or speed up manual	time consuming process	understanding which process need automation
easiest process to automate		


nanthini

informal description	a clear statement description the initial state of a problem	proactive then reactive approach to problem solving
executive summary		

nisanthi

customer issue that a proposed product or service aims to solving	design thinking improve experience	collect information from the research and previous experience with the problem
talk to multiple stakeholder who are impact by the problem		

RESULT :

 [Open in Colab](#)

```
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

Untitled3.ipynb - Colaboratory

colab.research.google.com/drive/1VnBjF5V6CynS24krOaKKHJE0Xz1S77q0

Untitled3.ipynb

File Edit View Insert Runtime Tools Help Last saved at April 22

+ Code + Text

```
[ ] 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.shape
```

(2966, 8)

```
[ ] 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
```

Untitled3.ipynb - Colaboratory

colab.research.google.com/drive/1VnBjF5V6CynS24krOaKKHJE0Xz1S77q0

Untitled3.ipynb

File Edit View Insert Runtime Tools Help Last saved at April 22

+ Code + Text

memory usage: 185.5+ KB

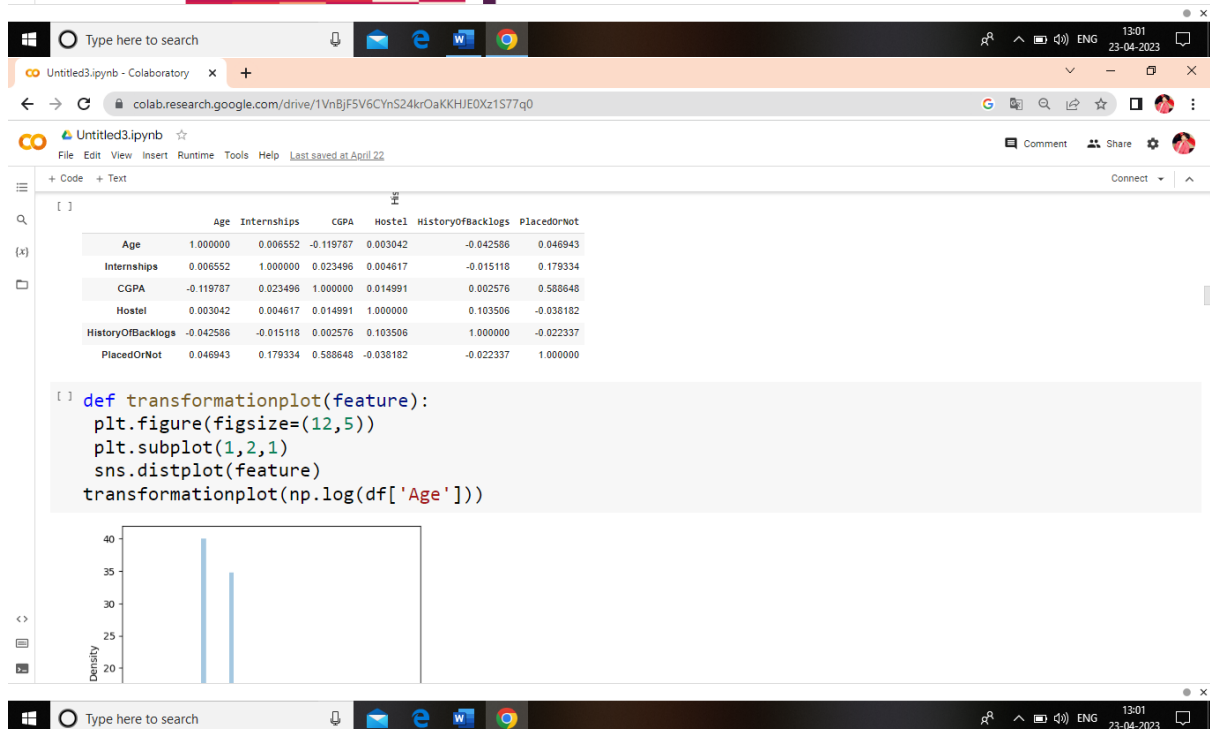
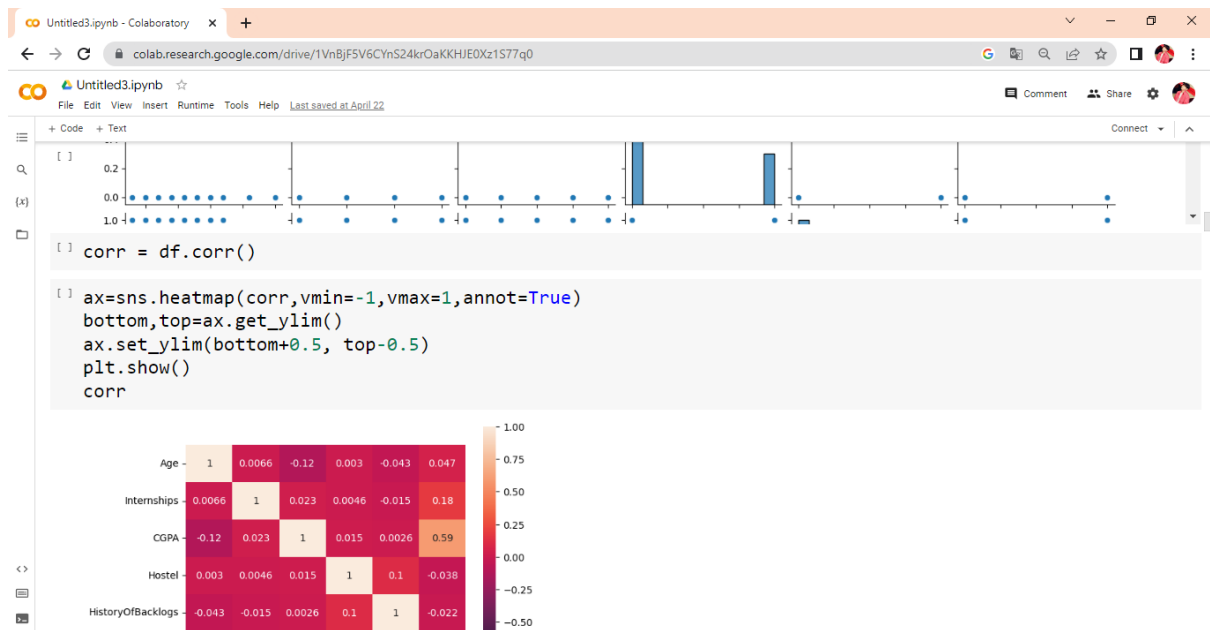
```
[ ] df.isnull().sum()
```

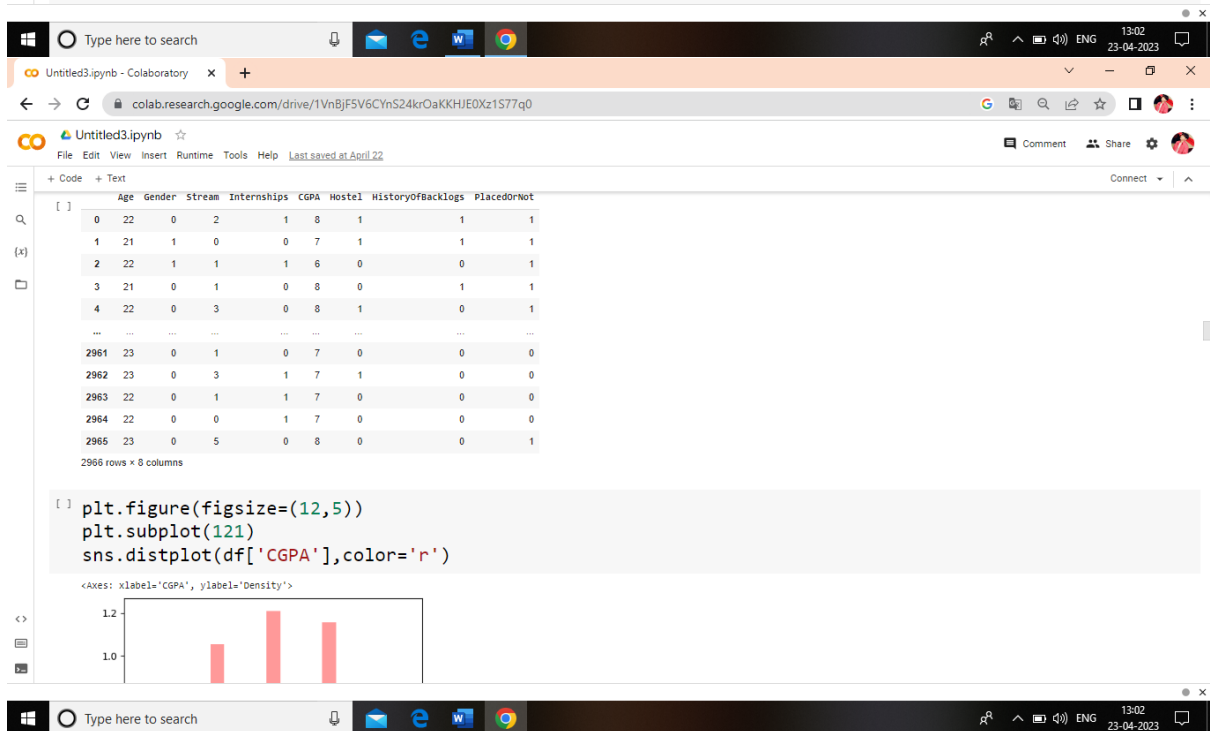
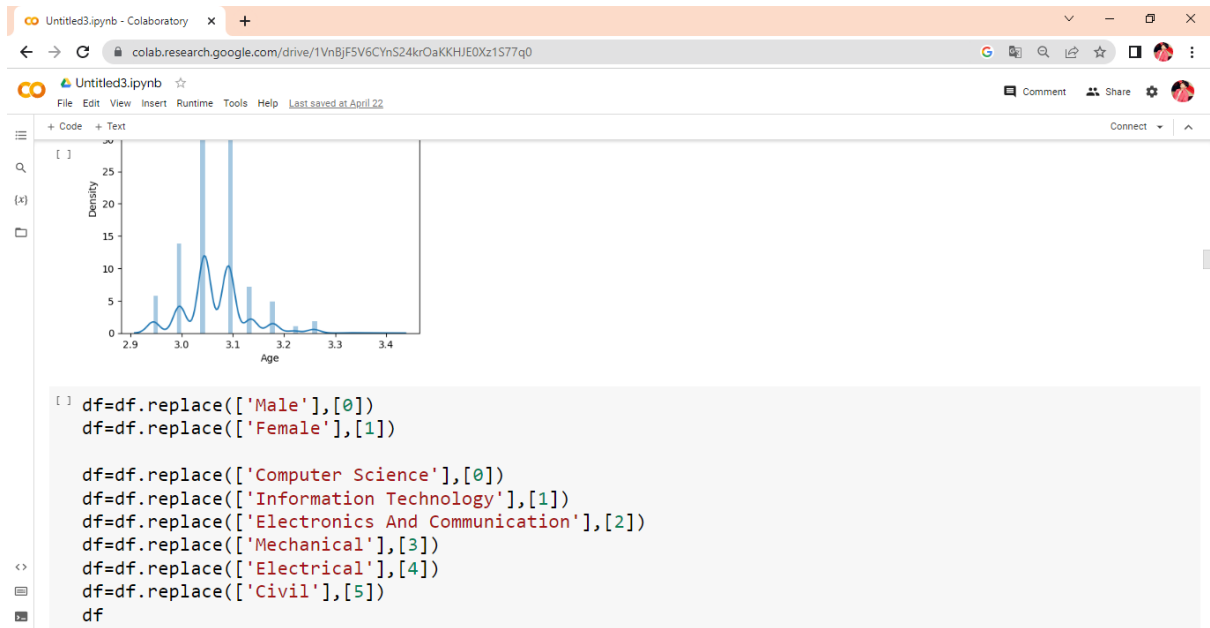
	Age	Gender	Stream	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot
Age	0	0	0	0	0	0	0	0
Gender	0	0	0	0	0	0	0	0
Stream	0	0	0	0	0	0	0	0
Internships	0	0	0	0	0	0	0	0
CGPA	0	0	0	0	0	0	0	0
Hostel	0	0	0	0	0	0	0	0
HistoryOfBacklogs	0	0	0	0	0	0	0	0
PlacedOrNot	0	0	0	0	0	0	0	0

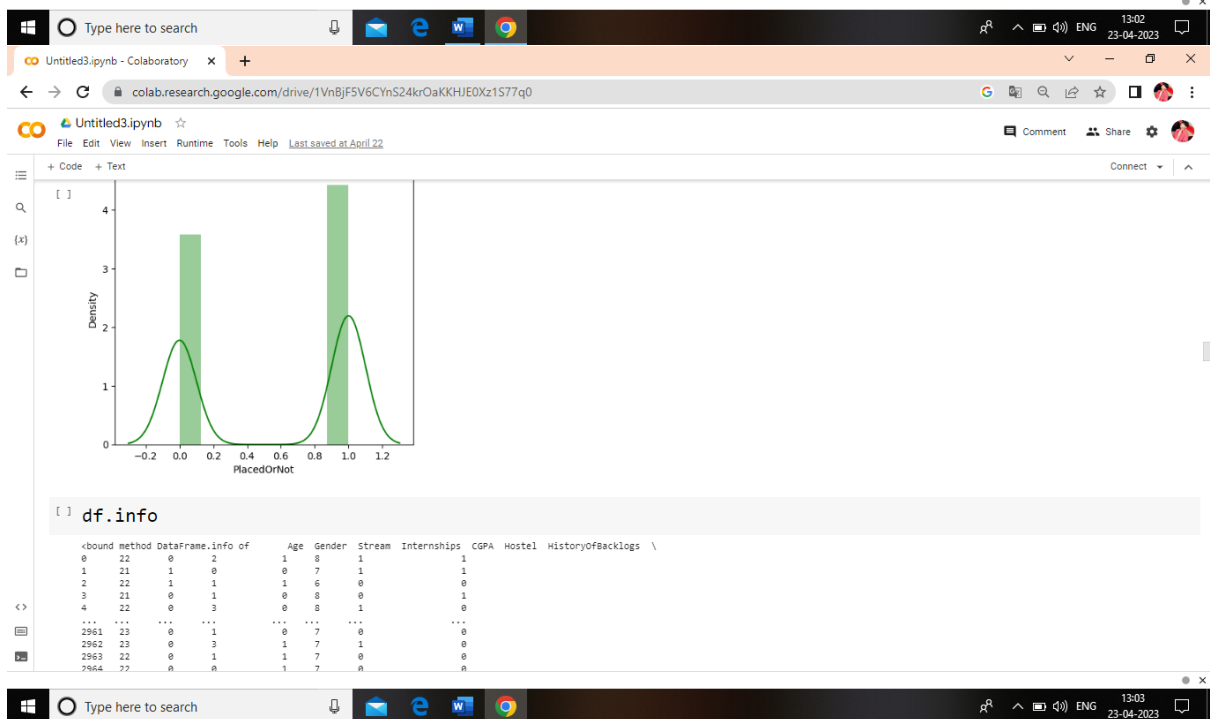
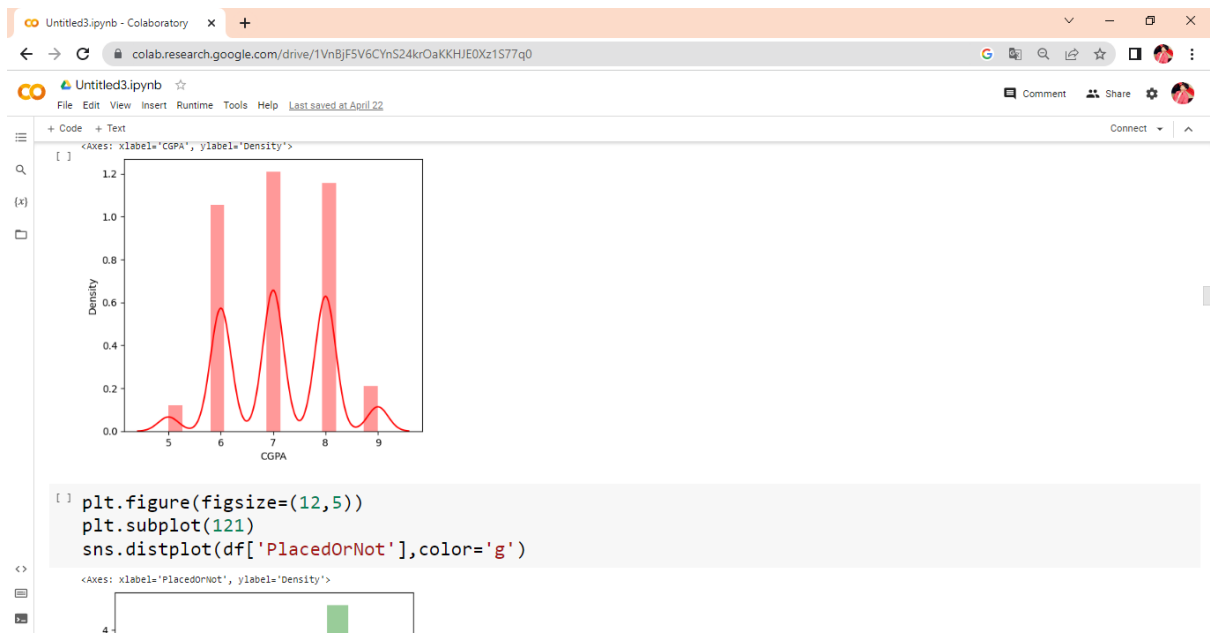
dtype: int64

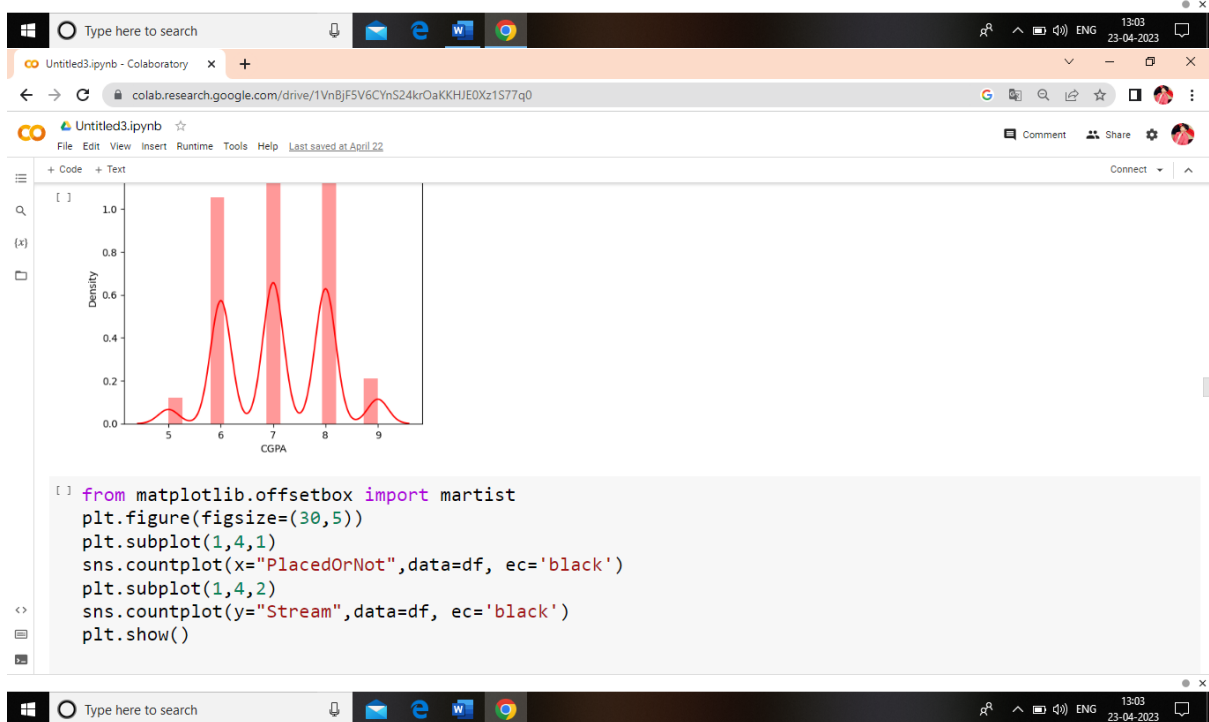
```
[ ] sns.pairplot(df)
```

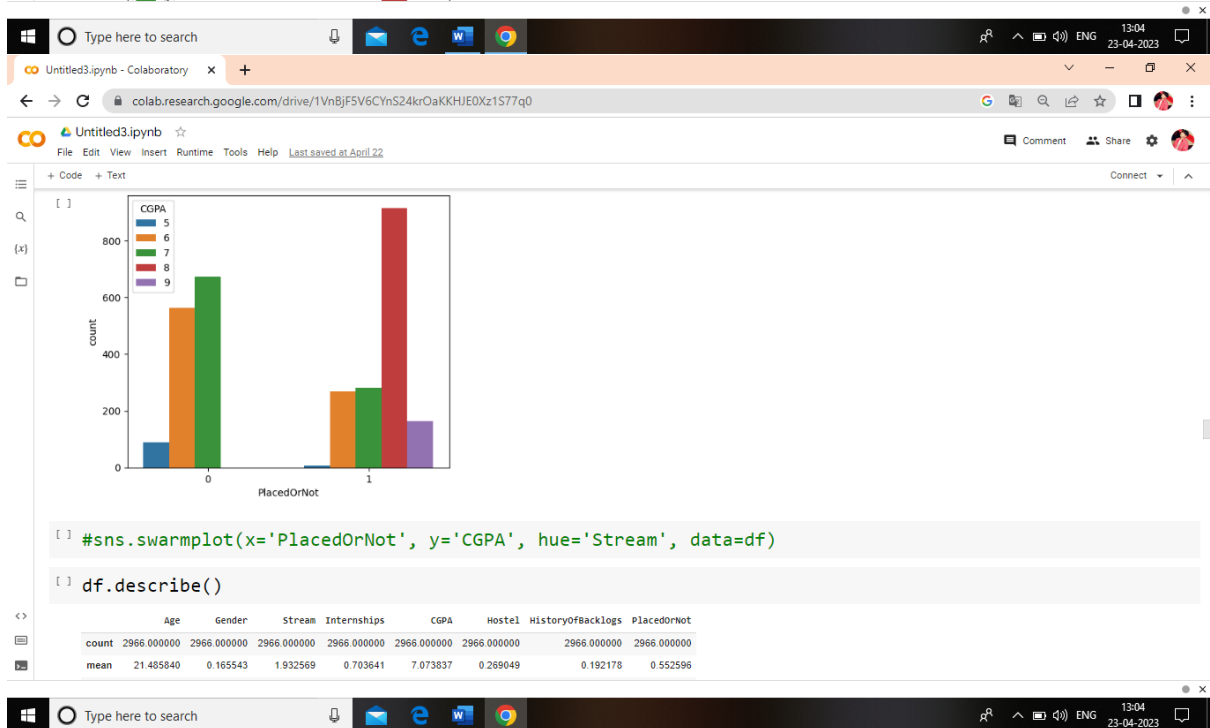
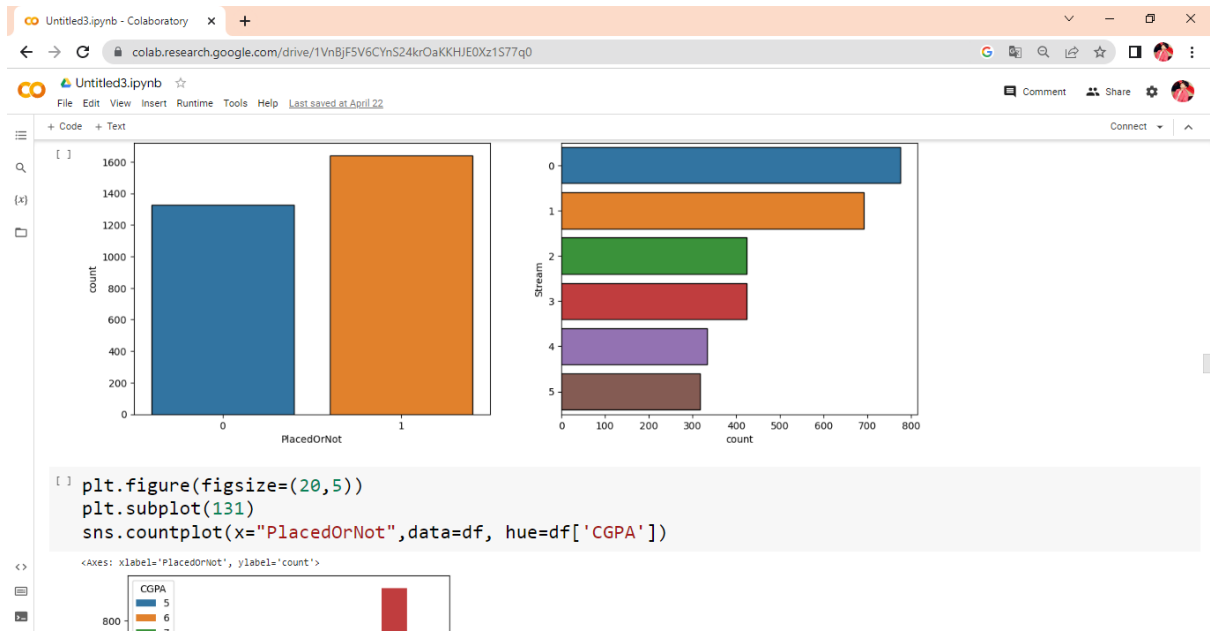
<seaborn.axisgrid.PairGrid at 0x7ff031523c10>











Untitled3.ipynb - Colaboratory

colab.research.google.com/drive/1VnBjF5V6CYnS24krOaKKHJE0Xz1S77q0

Untitled3.ipynb

File Edit View Insert Runtime Tools Help Last saved at April 22

+ Code + Text

```
[ ] df.describe()
```

	Age	Gender	Stream	Internships	CGPA	Hostel	HistoryOfBacklogs	PlacedOrNot
count	2966.000000	2966.000000	2966.000000	2966.000000	2966.000000	2966.000000	2966.000000	2966.000000
mean	21.485840	0.165543	1.932569	0.703641	7.073837	0.269049	0.192178	0.552596
std	1.324933	0.371732	1.682618	0.740197	0.967748	0.443540	0.394079	0.497310
min	19.000000	0.000000	0.000000	0.000000	5.000000	0.000000	0.000000	0.000000
25%	21.000000	0.000000	0.000000	0.000000	6.000000	0.000000	0.000000	0.000000
50%	21.000000	0.000000	2.000000	1.000000	7.000000	0.000000	0.000000	1.000000
75%	22.000000	0.000000	3.000000	1.000000	8.000000	1.000000	0.000000	1.000000
max	30.000000	1.000000	5.000000	3.000000	9.000000	1.000000	1.000000	1.000000

```
[ ] 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 int64
2 Stream 2966 non-null int64
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(8)
memory usage: 185.5 KB

Untitled3.ipynb - Colaboratory

colab.research.google.com/drive/1VnBjF5V6CYnS24krOaKKHJE0Xz1S77q0

Untitled3.ipynb

File Edit View Insert Runtime Tools Help Last saved at April 22

+ Code + Text

```
[ ] df['Gender'].value_counts()
```

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

```
[ ] x=df.drop('PlacedOrNot',axis=1)
y=df['PlacedOrNot']
import joblib
joblib.dump(x,"placement")
print(x)
print(y)
```

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

[2966 rows x 6 columns]

```
Untitled3.ipynb - Colaboratory x +
colab.research.google.com/drive/1VnBjF5V6CynS24krOaKKHJE0Xz1S77q0

Untitled3.ipynb ☆
File Edit View Insert Runtime Tools Help Last saved at April 22
+ Code + Text
Connect ^

[ ] [2966 rows x 6 columns]
0      1
1      1
2      1
3      1
4      1
..
2961    0
2962    0
2963    0
2964    0
2965    1
Name: PlacedOrNot, Length: 2966, dtype: int64

[ ] sc=StandardScaler()

[ ] print(sc)
sc.fit(x)
StandardScaler()
StandardScaler()
StandardScaler()

[ ] sd=sc.transform(x)
#x=sc.fit_transform()
#x = pd.DataFrame(x)
print(sd)

[[ 0.38813858 -0.44540301  0.04008175  0.40044544  0.95719868  2.05024603]
 [ 0.36675158  2.24515772 -1.14074288 -0.95077319 -0.07631043  2.05024603]
 [ 0.38813858  2.24515772 -0.55433057  0.40044544 -1.10901154 -0.48774634]
 ...
 [ 0.38813858 -0.44540301 -0.55433057  0.40044544 -0.07631043 -0.48774634]
 [ 0.38813858 -0.44540301 -1.14074288  0.40044544 -0.07631043 -0.48774634]
 [ 1.14301273 -0.44540301  1.02331069 -0.95077319  0.95719868 -0.48774634]]
```

```
Type here to search
Untitled3.ipynb - Colaboratory x +
colab.research.google.com/drive/1VnBjF5V6CynS24krOaKKHJE0Xz1S77q0

Untitled3.ipynb ☆
File Edit View Insert Runtime Tools Help Last saved at April 22
+ Code + Text
Connect ^

[ ] [[ 0.38813858 -0.44540301  0.04008175  0.40044544  0.95719868  2.05024603]
 [-0.36675158  2.24515772 -1.14074288 -0.95077319 -0.07631043  2.05024603]
 [ 0.38813858  2.24515772 -0.55433057  0.40044544 -1.10901154 -0.48774634]
 ...
 [ 0.38813858 -0.44540301 -0.55433057  0.40044544 -0.07631043 -0.48774634]
 [ 0.38813858 -0.44540301 -1.14074288  0.40044544 -0.07631043 -0.48774634]
 [ 1.14301273 -0.44540301  1.02331069 -0.95077319  0.95719868 -0.48774634]]

[ ] x=sd
y=df['PlacedOrNot']

[ ] y

0      1
1      1
2      1
3      1
4      1
..
2961    0
2962    0
2963    0
2964    0
2965    1
Name: PlacedOrNot, Length: 2966, dtype: int64

[ ] x_train, x_test, y_train, y_test=train_test_split(x,y, test_size=0.2, stratify=y, random_state=2)

[ ] print(x.shape,x_train.shape,x_test.shape)
```

```
Untitled3.ipynb - Colaboratory
colab.research.google.com/drive/1VnBjF5V6CYnS24krOaKKHJE0Xz1S77q0

Untitled3.ipynb
File Edit View Insert Runtime Tools Help Last saved at April 22
+ Code + Text
(2966, 6) (2372, 6) (594, 6)

[ ] classifier =svm.SVC(kernel='linear')
classifier.fit(x_train,y_train)
svm.SVC(kernel='linear')
x_train_prediction=classifier.predict(x_train)
training_data_accuracy=accuracy_score(x_train_prediction,y_train)

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

Accracy score of the training data: 0.7685497479469039

[ ] x_train
array([[ 1.89789488, -0.44540301,  1.82331869,  0.40044544, -1.10981154,
        -0.40774634],
       [-1.12163373, -0.44540301, -0.55433057,  0.40044544, -0.07631043,
        -0.40774634],
       [-0.36675158, -0.44540301, -0.55433057,  0.40044544, -0.07631043,
        -0.40774634],
       ...,
       [-1.12163373, -0.44540301, -1.14874288,  1.75166407, -0.07631043,
        -0.40774634],
       [ 0.36813058, -0.44540301, -0.55433057,  0.40044544,  1.99069179,
        -0.40774634],
       [ 1.89789488, -0.44540301, -1.14874288,  0.40044544, -1.10981154,
        2.05024603]])

[ ] best_k={"Regular":0}
```

```
Untitled3.ipynb - Colaboratory
colab.research.google.com/drive/1VnBjF5V6CYnS24krOaKKHJE0Xz1S77q0

Untitled3.ipynb
File Edit View Insert Runtime Tools Help Last saved at April 22
+ Code + Text
(2966, 6) (2372, 6) (594, 6)

[ ] x_train
array([[ 1.89789488, -0.44540301,  1.82331869,  0.40044544, -1.10981154,
        -0.40774634],
       [-1.12163373, -0.44540301, -0.55433057,  0.40044544, -0.07631043,
        -0.40774634],
       [-0.36675158, -0.44540301, -0.55433057,  0.40044544, -0.07631043,
        -0.40774634],
       ...,
       [-1.12163373, -0.44540301, -1.14874288,  1.75166407, -0.07631043,
        -0.40774634],
       [ 0.36813058, -0.44540301, -0.55433057,  0.40044544,  1.99069179,
        -0.40774634],
       [ 1.89789488, -0.44540301, -1.14874288,  0.40044544, -1.10981154,
        2.05024603]])

[ ] best_k={"Regular":0}
best_score={"Regular":0}
for k in range(3,50,2):
    knn_temp = KNeighborsClassifier(n_neighbors=k)
    knn_temp.fit(x_train, y_train)
    knn_temp_pred = knn_temp.predict(x_test)
    score=metrics.accuracy_score(y_test, knn_temp_pred)*100
    if score >= best_score["Regular"]and score<100:
        best_score["Regular"]=score
        best_k["Regular"] = k
```



```
Untitled3.ipynb - Colaboratory x +
colab.research.google.com/drive/1VnBjF5V6CYnS24krOaKKHJE0Xz1S77q0

Untitled3.ipynb ☆
File Edit View Insert Runtime Tools Help Last saved at April 22
+ Code + Text
[ ] classifier = Sequential()
#add input layer and first hidden layer
classifier.add(keras.layers.Dense(6,activation = 'relu',input_dim =6))
classifier.add(keras.layers.Dropout(0.50))

#add second hidden layer
classifier.add(keras.layers.Dense(6,activation = 'relu'))
classifier.add(keras.layers.Dropout(0.50))

#final or output layer
classifier.add(keras.layers.Dense(1,activation = 'sigmoid'))

[ ] #compiling the model
loss_1 =tf. keras.losses.BinaryCrossentropy()
classifier.compile(optimizer= 'Adam', loss= loss_1, metrics = ['accuracy'])

[ ] classifier.fit(x_train, y_train, batch_size = 20, epochs = 100)

Epoch 1/100
119/119 [=====] - 1s 3ms/step - loss: 0.6994 - accuracy: 0.5270
```

```
Windows Type here to search
Untitled3.ipynb - Colaboratory x +
colab.research.google.com/drive/1VnBjF5V6CYnS24krOaKKHJE0Xz1S77q0

Untitled3.ipynb ☆
File Edit View Insert Runtime Tools Help Last saved at April 22
+ Code + Text
[ ] Epoch 1/100
119/119 [=====] - 1s 3ms/step - loss: 0.6994 - accuracy: 0.5270
Epoch 2/100
119/119 [=====] - 0s 3ms/step - loss: 0.6373 - accuracy: 0.6935
Epoch 3/100
119/119 [=====] - 0s 3ms/step - loss: 0.5442 - accuracy: 0.7707
Epoch 4/100
119/119 [=====] - 0s 3ms/step - loss: 0.4644 - accuracy: 0.7985
Epoch 5/100
119/119 [=====] - 0s 3ms/step - loss: 0.4177 - accuracy: 0.8111
Epoch 6/100
119/119 [=====] - 0s 2ms/step - loss: 0.3917 - accuracy: 0.8212
Epoch 7/100
119/119 [=====] - 0s 2ms/step - loss: 0.3750 - accuracy: 0.8288
Epoch 8/100
119/119 [=====] - 0s 2ms/step - loss: 0.3637 - accuracy: 0.8360
Epoch 9/100
119/119 [=====] - 0s 2ms/step - loss: 0.3559 - accuracy: 0.8423
Epoch 10/100
119/119 [=====] - 0s 2ms/step - loss: 0.3499 - accuracy: 0.8461
Epoch 11/100
119/119 [=====] - 0s 2ms/step - loss: 0.3450 - accuracy: 0.8449
Epoch 12/100
119/119 [=====] - 0s 2ms/step - loss: 0.3414 - accuracy: 0.8461
Epoch 13/100
119/119 [=====] - 0s 2ms/step - loss: 0.3375 - accuracy: 0.8470
Epoch 14/100
119/119 [=====] - 0s 2ms/step - loss: 0.3350 - accuracy: 0.8470
Epoch 15/100
119/119 [=====] - 0s 2ms/step - loss: 0.3320 - accuracy: 0.8478
Epoch 16/100
119/119 [=====] - 0s 2ms/step - loss: 0.3299 - accuracy: 0.8512
Epoch 17/100
119/119 [=====] - 0s 2ms/step - loss: 0.3275 - accuracy: 0.8533
Epoch 18/100
119/119 [=====] - 0s 2ms/step - loss: 0.3264 - accuracy: 0.8520
Epoch 19/100
119/119 [=====] - 0s 2ms/step - loss: 0.3240 - accuracy: 0.8537
Epoch 20/100
```

[illegible]

Untitled3.ipynb - Colaboratory x +

colab.research.google.com/drive/1VnBjF5V6CYnS24krOaKKHJE0Xz1S77q0

Untitled3.ipynb ☆

File Edit View Insert Runtime Tools Help Last saved at April 22

Comment Share Settings

+ Code + Text

Connect

```
[ ] from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test, pred)
cm

array([[248, 18],
       [ 68, 260]])

[ ] input_data=[[23,0,2,1,8,1]]
prediction=knn.predict(input_data)
print(prediction)
if(prediction[0]==0):
    print('Not placed')
else:
    print('placed')

[1]
placed

[ ] input_data=[[23,0,1,0,7,0]]
prediction=knn.predict(input_data)
print(prediction)
if(prediction[0]==0):
    print('Not placed')
else:
    print('placed')
```

Type here to search

13:10 23-04-2023

Untitled3.ipynb - Colaboratory x +

colab.research.google.com/drive/1VnBjF5V6CYnS24krOaKKHJE0Xz1S77q0

Untitled3.ipynb ☆

File Edit View Insert Runtime Tools Help Last saved at April 22

Comment Share Settings

+ Code + Text

Connect

```
[ ] [1]
placed

[ ] [ ] input_data=[[23,0,1,0,7,0]]
prediction=knn.predict(input_data)
print(prediction)
if(prediction[0]==0):
    print('Not placed')
else:
    print('placed')

[1]
placed

[ ] [ ] import pickle
pickle.dump(knn,open("placement.pkl","wb"))
model = pickle.load(open('placement.pkl','rb'))
```

Type here to search

13:23 23-04-2023

ADVANTAGES:

- ❖ **Work placement allows students to have a real time experience of the job at hand**
- ❖ **At times, what is taught in theory in a school or university may not be the same as a real time experience**
- ❖ **For example, a textbook may teach you theoretically, how to build a box**
- ❖ **Offering you an insight into the 'world of work' and allows you to get a feel for what you do and don't like doing**
- ❖ **Adds invaluable work experience to your cv which is attractive to graduate employers**
- ❖ **Provides the opportunity to work with a more diverse group of people**

DISADVANTAGES:

- ❖ **Adapting to work placement might not be easy**
- ❖ **You'll have to communicate, get to know a new group of people and work under more pressure that you're used to**
- ❖ **As a result, work placements can be daunting and more stressful than your established student life**

APPLICATIONS:

- ✓ **Operating system: Windows 10**
- ✓ **Front end : html, python**
- ✓ **Software :anaconda, pycharm**
communication.

CONCLUSION:

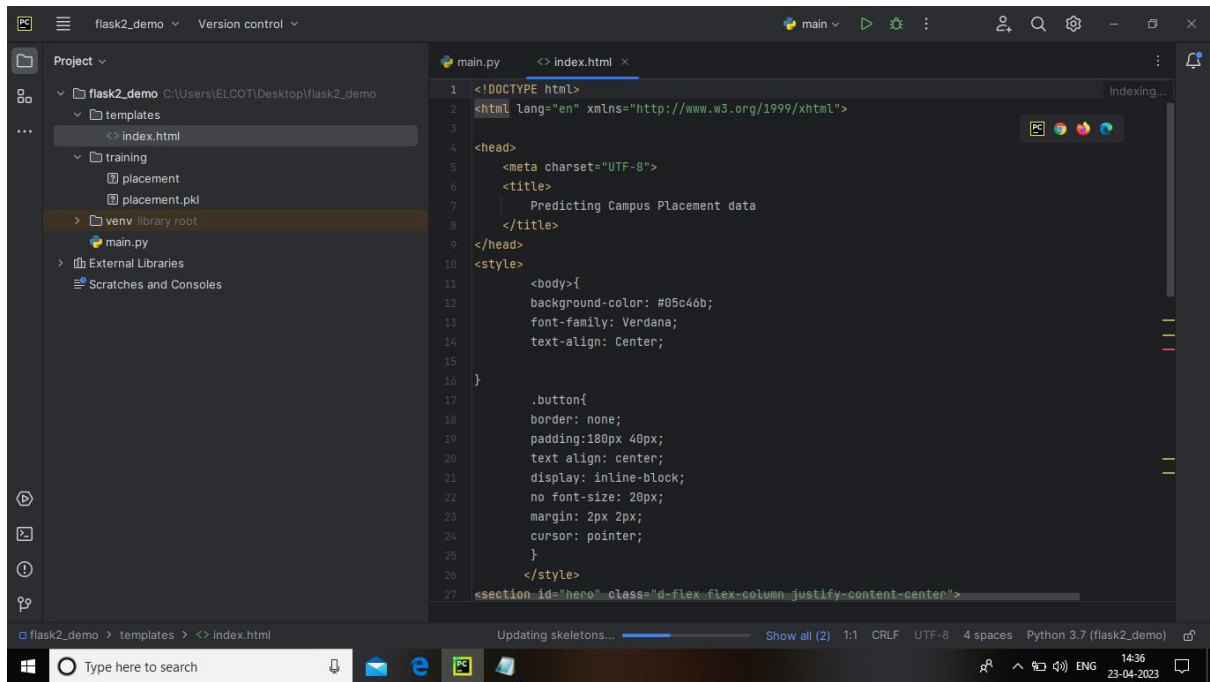
- ❖ **Placement is understood as the allocation of people to jobs**
- ❖ **If the number of individuals is large in relation to available jobs, only the best qualified persons can be selected and placed**
- ❖ **Once we establish this unique profile for each individual, people and jobs can be matched optimally within the constraints set by available jobs and available people**

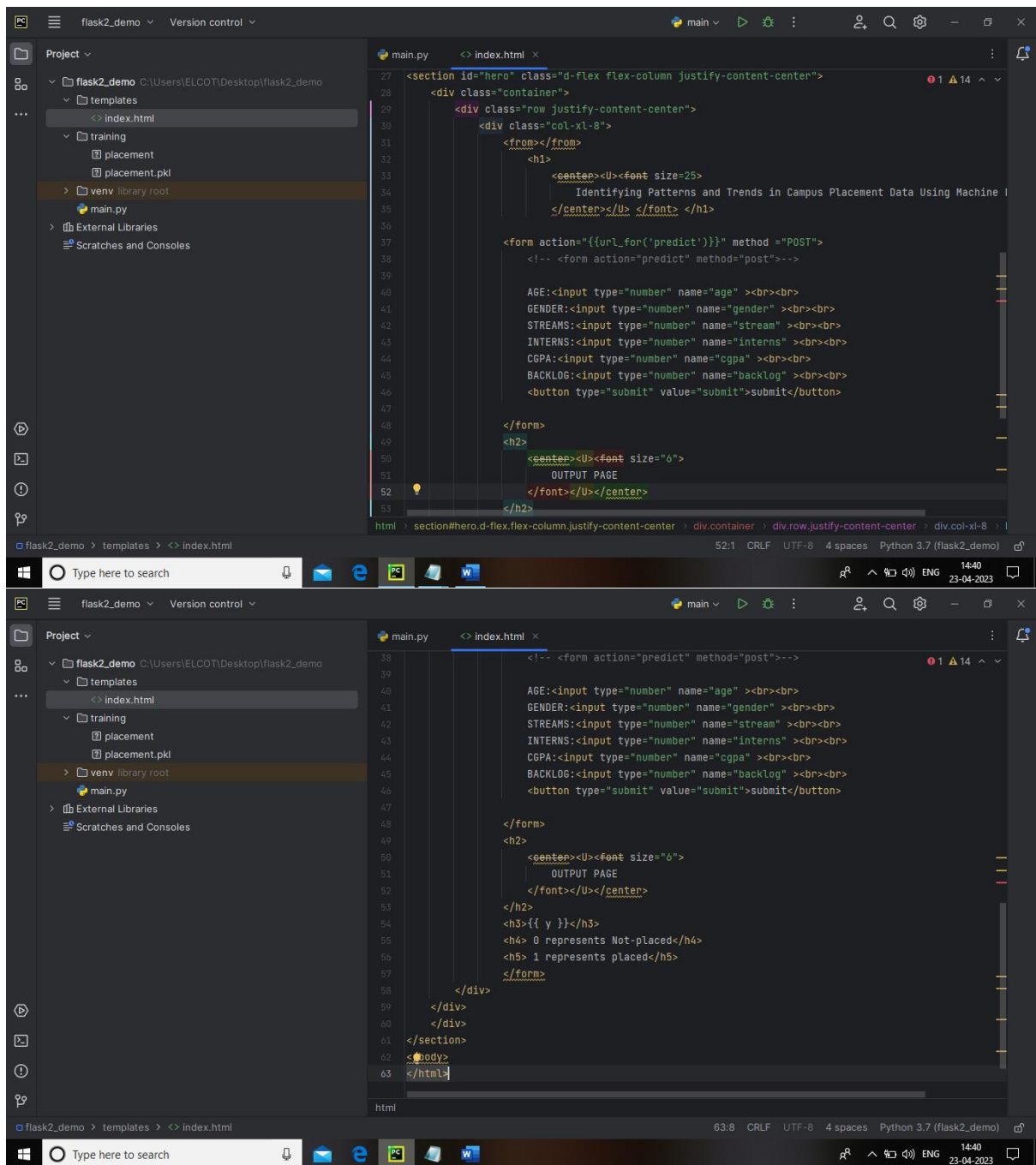
FUTURE SCOPE:

- ❖ **The project has a wide scope**
- ❖ **Our project mainly helps in improving productivity and makes use of utilization of resources**
- ❖ **Which provides the up to date information of all the students in the collage**

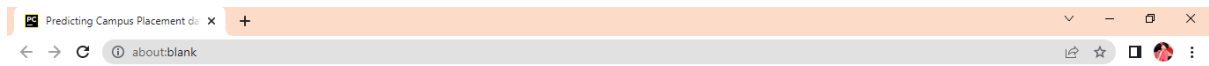
APPENDIX:

SOURCE CODE:






```
1 from flask import Flask, render_template, request
2 import pickle
3 app = Flask(__name__)
4
5 model = pickle.load(open(r"C:\Users\ELCOT\Desktop\flask2_demo\training\placement.pkl", 'rb'))
6 @app.route('/')
7 def home():
8     return render_template('index.html')
9
10 @app.route('/predict', methods=['POST'])
11 def predict():
12     x_test = [int(yo) for yo in request.form.values()]
13     # sts = [np.array(x_test)]
14     prediction = model.predict(x_test)
15     #opt = prediction[0]
16     return render_template('index.html', y='prediction is {}'.format(prediction))
17
18 if __name__ == "__main__":
19     app.run(debug=False)
20
21
```



Identifying Patterns and Trends in Campus Placement Data Using Machine Learning

AGE:

GENDER:

STREAMS:

INTERNS:

CGPA:

BACKLOG:

OUTPUT PAGE

{{ y }}

0 represents Not-placed

1 represents placed

