INDEX

1. INTRODUCTION

1.1 OVERVIEW

This project focuses on the trend analysis of university placement. A trend analysis means an aspect of technical analysis that tries to predict future movement on past data. Students are the main stakeholders of universities and their performance plays a significant role in a country's social and economic growth by producting creative graduates, innovators and enterpreneurs. A millions of university and colleges provides the terchnical education nowadays. But few of them land in good jobs as compared to ther total. A university placements plays a very important role to get more admission and maintain the reputation in the world. Authors consider some of the important attributes of students like company, package, location, etc. A machine learning plays the most important role to find out the performance, define the trend or predict the placement for the upcoming year.

The dataset considered for this work is the CSE department in which we have initially considered four attribute of each student. For Example, 10^{th} marks, 12^{th}/Diploma marks, B.Tech and M.Tech marks. With the help of deep neural network classifier and neural networks have been known to successfully predict the future of student, out of 50 records 37 are used for training and 13 used for testing purpose and we got 77% accuracy while testing.

1.2 PURPOSE

The goal of this project is to predict whether the student will get a campus placement or not based on various parameters such as gender, SSC percentage, HSC percentage, HSC stream, degree percentage, degree type, work experience and e-test percentage**.** This research focuses on various algorithms of machine learning such as Logistic Regression, Decision Tree, K-Nearest Neighbours and Random Forest in order to produce economical and correct results for campus placement prediction. This system follows a supervised machine learning approach as it uses class labelled data for training the classification algorithm.

Nowadays the number of educational institutes is growing day by day. The aim of each higher educational institute is to help their students to get a well-paid job through their placement cell. One of the biggest challenges that higher learning institutes face these days is to uplift the placement performance of scholars.

2. PROBLEM DEFENITION & DESIGN THINKING

2.1 EMPATHY MAP

Team ID	**NM2023TMID32910**
Project Name	**Identifying Patterns and Trends in campus placement data using machine learning with python**
Maximum Marks	**5 Marks**

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.

Says
What have we heard them say?
What can we imagine them saying?

resources we
eat

subject
knowledge

eligible
percentage

required
cert/Exams

persent
of mind

Thinks
What are their wants, needs, hopes, and dreams? What other thoughts might influence their behavior?

about
salary

is it my
dream job?

is that job
suit for me?

I'm eligible
for that
position

student

asking advice
from friends
who
attend alert
interview

searched
about
company

searched
about
interview
is interval

searched
about
company
locations

asking tips
from
superior

Does
What behavior have we observed?
What can we imagine them doing?

happy

fear

sad

shivering

nervousness

Feels
What are their fears, frustrations, and anxieties? What other feelings might influence their behavior?

2.2 IDEATION AND BRAINSTORMING MAP

Team ID	NM2023TMID32910
Project Name	Identifying Patterns and Trends in campus placement data using machine learning with python
Maximum Marks	5 Marks

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.

Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

why they recruiting young people in campus interview?

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

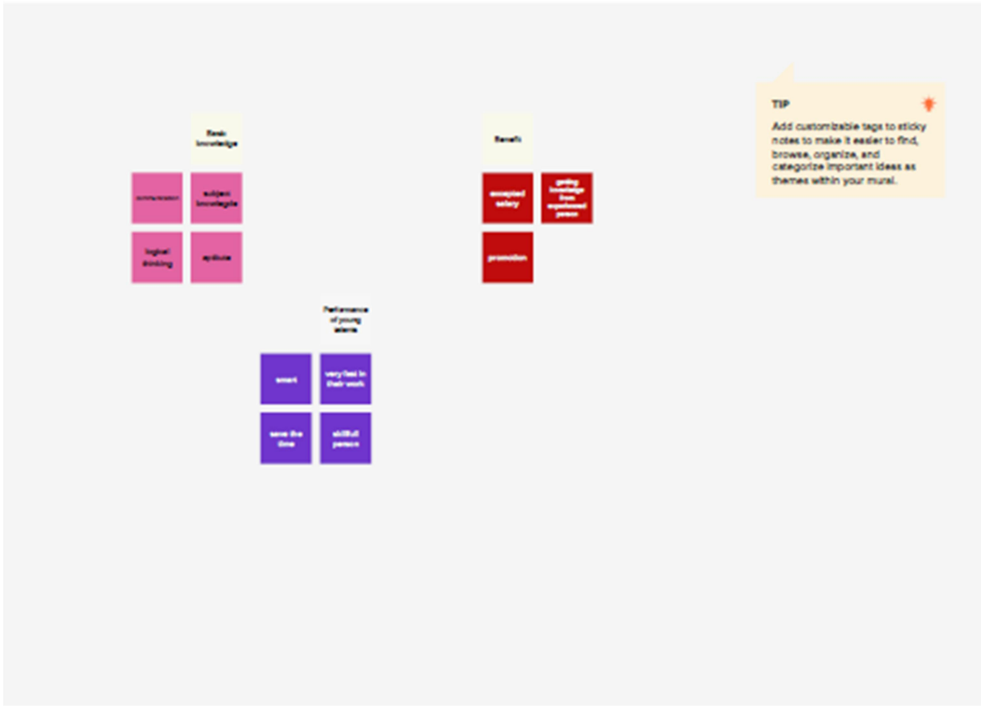
You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

3

Group Ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

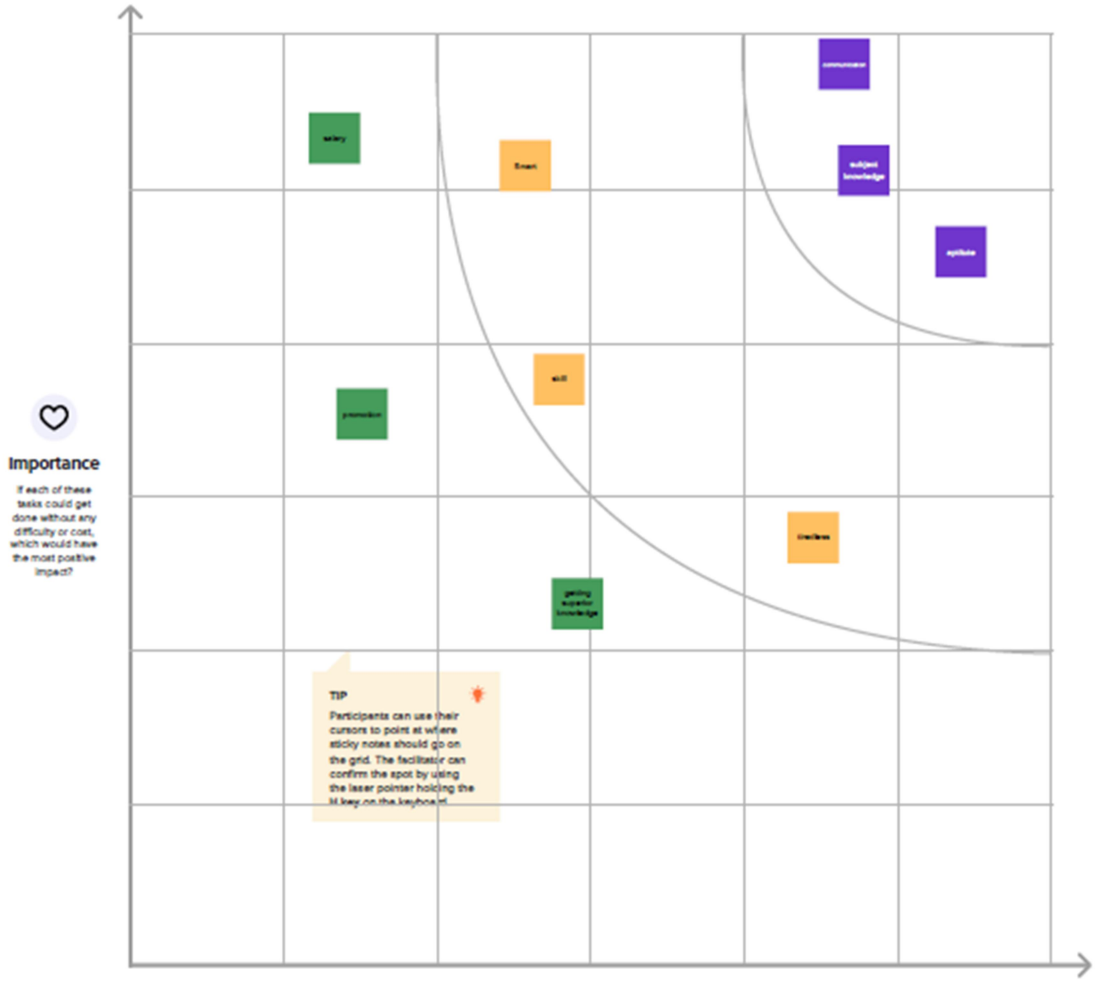


4

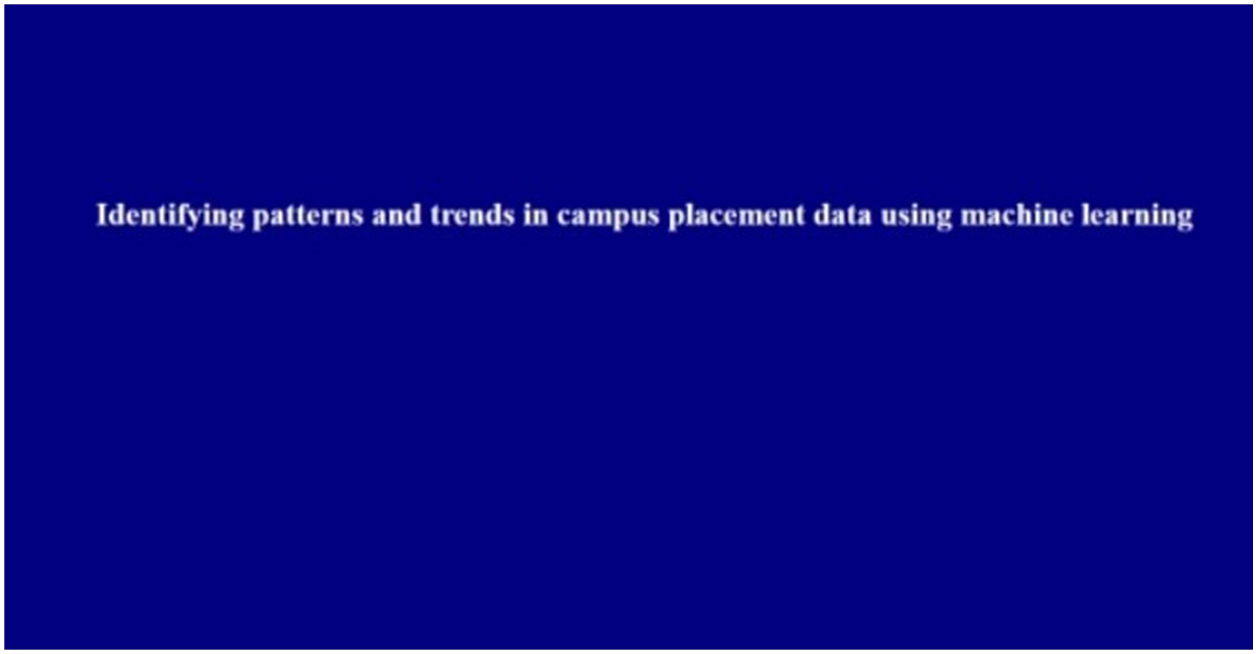
Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes



3. RESULT



Fill The Details

Enter Age

22

Enter Gender

0

Enter Stream

2

Enter Internships

1

Enter CGPA

8

Enter Backlogs

1

Submit

Placement Prediction

The prediction is:1

0 represents Not-Placed

1 represents Placed

4. ADVANTAGES & DISADVANTAGES

ADVANTAGES

- Campus recruitment helps in increased selection ratio. More number of quality candidates can be selected through this recruitment process.
- The organizations campus can built up more company loyalty through campus selection process.
- Fresh and talented graduates will work more closely with their first company. Hence, this in a way will increase the brand loyalty among different applications.
- An organization through effective campus recruitment finds an opportunity to establish a link with the next batch of students. This is turn paves way to serve the future and long term recruitment needs of the company.

DISADVANTAGES

- Campus recruitment is an expensive affair for majority of the companies as it adds up costs to the bottom line.
- Companies incur different expenses related to travel, boarding, training etc while conducting campus selection process.
- The experienced and skilled candidates having practical job exposures cannot be recruited through campus placements.
- Fresh candidates selected through campus placements require adequate training for work.

5. APPLICATIONS

- Research
- Prepare a strong Resume
- Give utmost attention to your company's presentation about itself
- Attend mock interviews
- Include a cover letter
- Stay updated
- Be punctual
- Appear presentable
- Avoid the mistakes job aspirants always commit
- Affirm your commitment
- Have a well-planned beginning and ending
- Do you know your weakness?
- Inquire about the Organization

6. CONCLUSION

The problem of campus placement prediction can be solved with the help of different machine learning algorithms such as logistic regression, decision tree, KNN and placement prediction. Here the logistic regression algorithm have the highest accuracy of 95.34% for campus placement prediction.

The algorithm of machine learning we have discussed are can used to find the trend of placement, which will be helpful for university to get more admission in future. We compared the algorithm and find out the accuracy by considering some of attributes of students. Here we used deep neural network classifier with the 1000, 2000, 5000 iteration with 71%, 77%, and 91% of accuracy.

7.FUTURE SCOPE

A machine learning algorithms are play a very important role while predicting something, in future student performance will tell the institute and student weather he or she will clear the subject or not. Accuracy may further increase by application of more advanced technics such as deep learning and exoerimenting with different activation functions of neural networks such as linear, sigmoid, tan h and ReLU.

We can also experiment with different cross validation technics such a 3 Fold, 5 Fold, 10 Fold, 15 Fold cross validation in order to analyze the change in accuracy.

8. APPENDIX

A) 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
```

```
[2] df = pd.read_csv("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

```
[3] df.info()
```

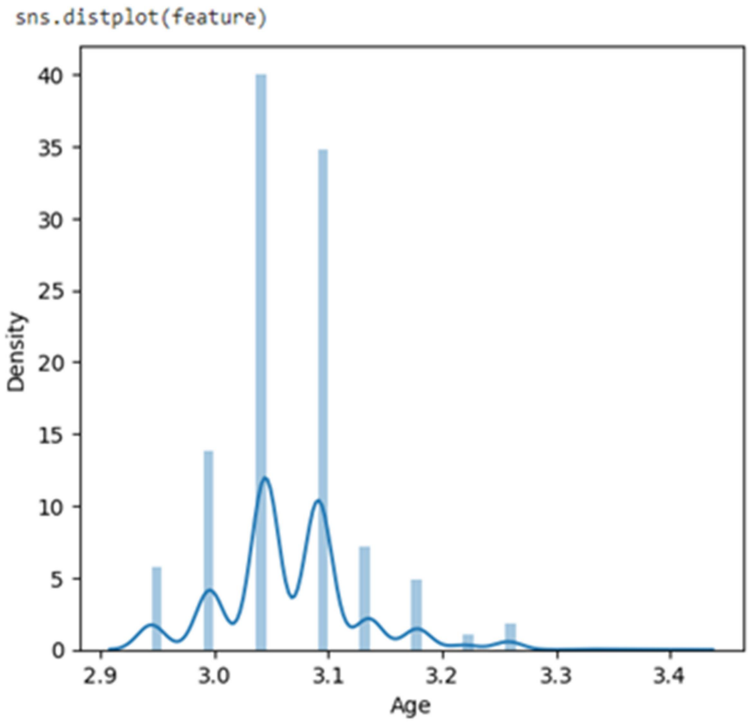
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2966 entries, 0 to 2965
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Age                2966 non-null   int64
 1   Gender             2966 non-null   object
 2   Stream             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
```

[4]
```
def transformationplot(feature):
    plt.figure(figsize=(12,5))
    plt.subplot(1,2,1)
    sns.distplot(feature)

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



[5]
```
df=df.replace(['Male'],[0])
df=df.replace(['Female'],[1])

df=df.replace(['Computer Science','Information Technology','Electronics and Communication','Mechanical','Electrical','Ci
```

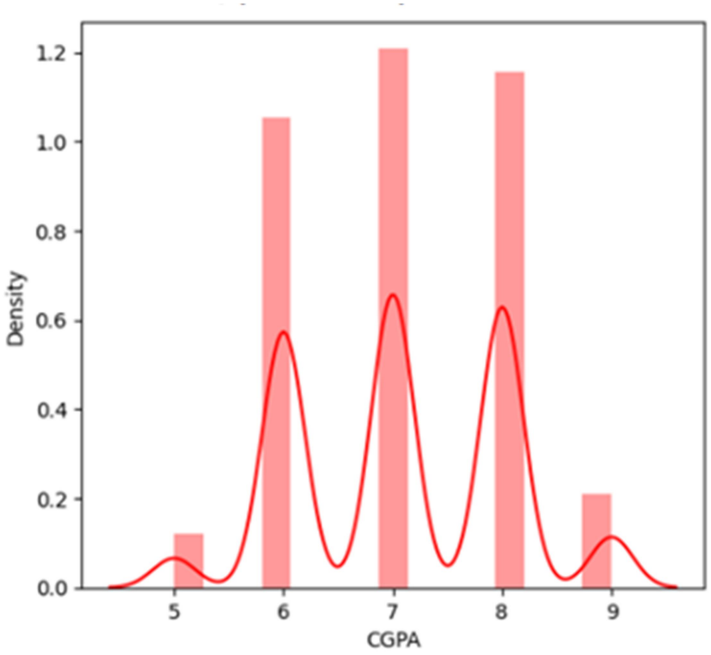
[6]
```
df=df.drop(['Hostel'],axis=1)
df
```

	Age	Gender	Stream	Internships	CGPA	HistoryOfBacklogs	PlacedOrNot
0	22	0	Electronics And Communication	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

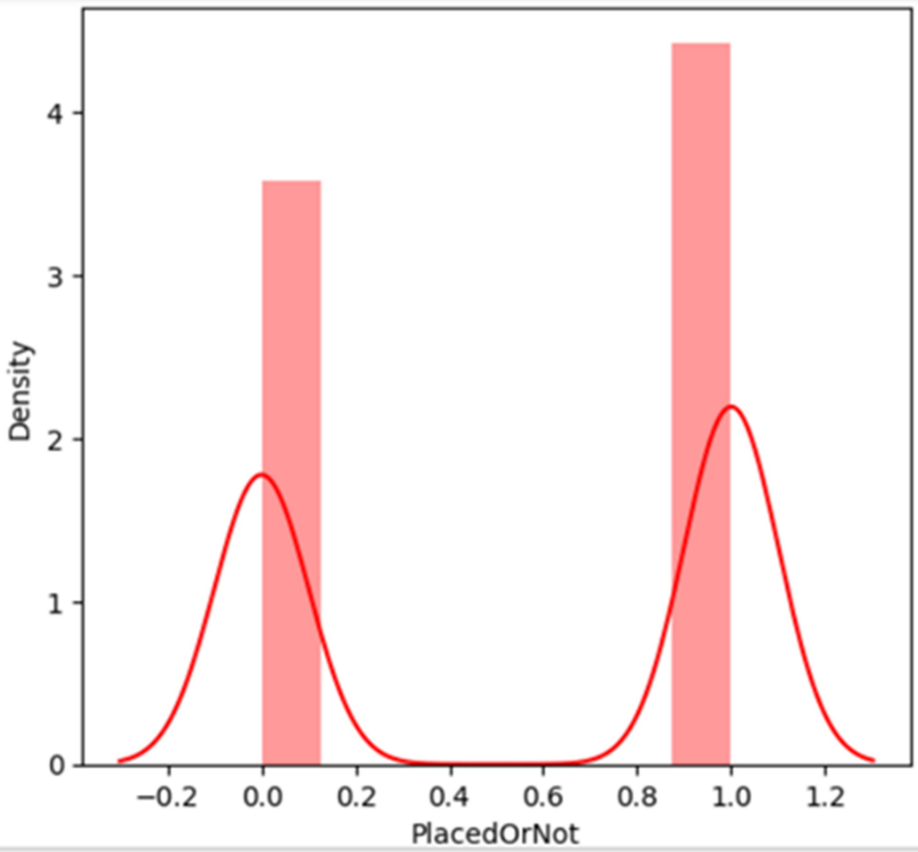
2966 rows × 7 columns

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

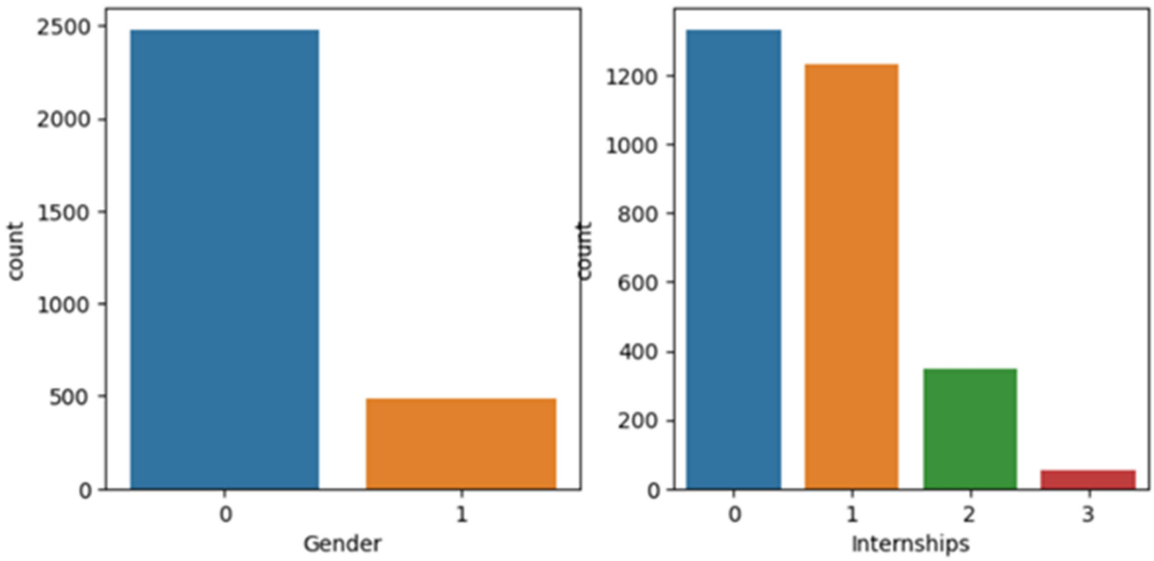
```
  sns.distplot(df['CGPA'],color='r')
<Axes: xlabel='CGPA', ylabel='Density'>
```



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

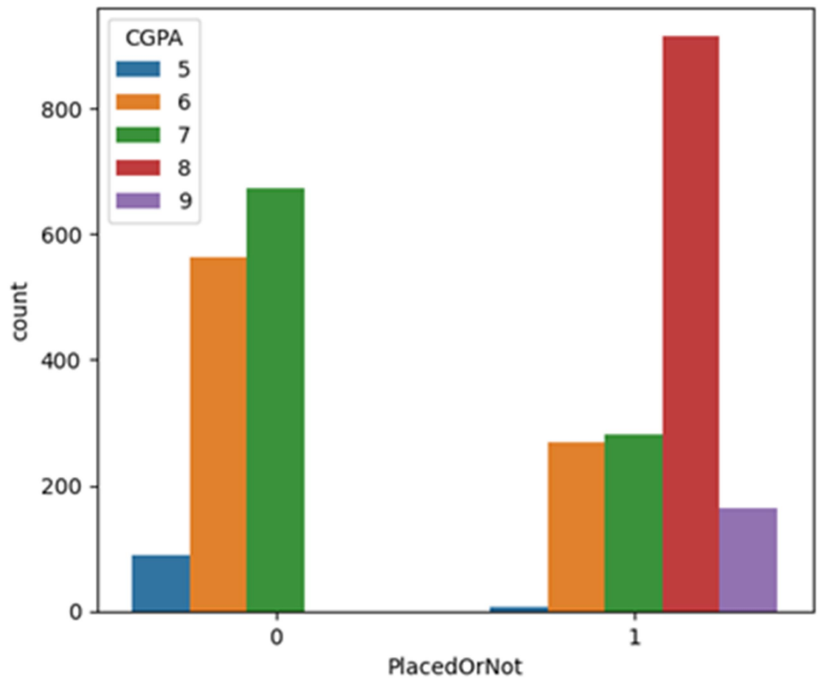



```
[ ]  plt.figure(figsize=(18,4))
     plt.subplot(1,4,1)
     sns.countplot(x=df['Gender'])
     plt.subplot(1,4,2)
     sns.countplot(x=df['Internships'])
     plt.show()
```

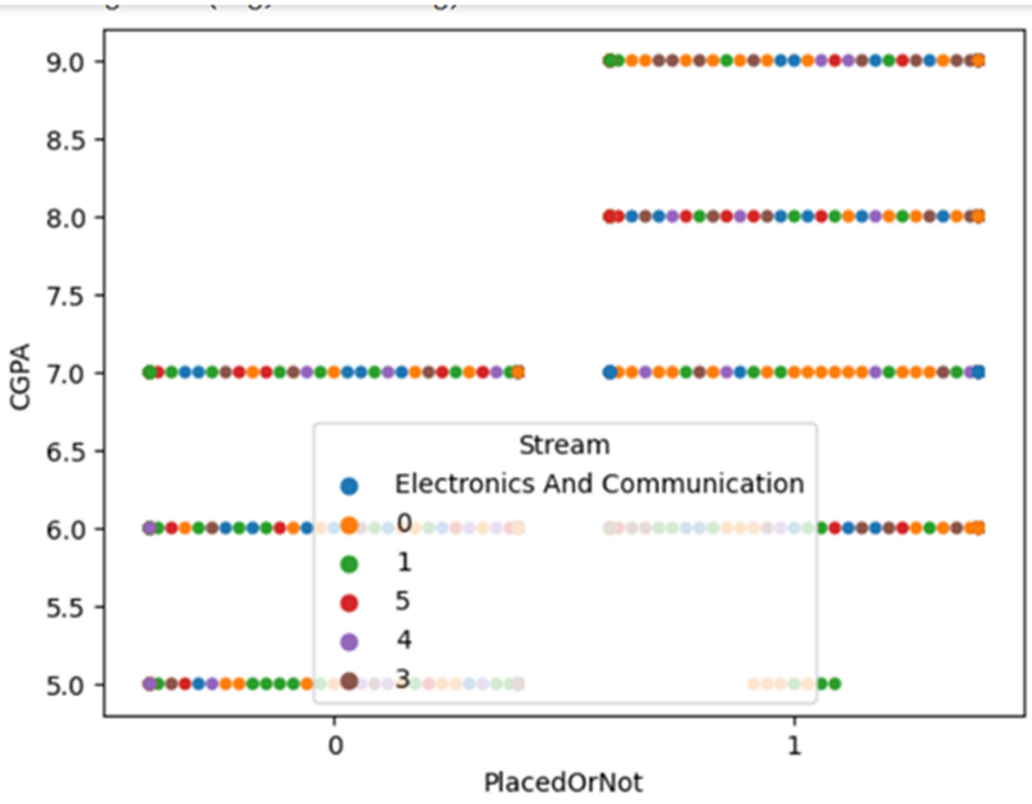



```
[ ]  plt.figure(figsize=(20,5))
     plt.subplot(131)
     sns.countplot(x=df["PlacedOrNot"],hue=df['CGPA'])
```

```
<Axes: xlabel='PlacedOrNot', ylabel='count'>
```



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




```
[ ]  from sklearn.preprocessing import StandardScaler
     sc=StandardScaler()
     df=pd.read_csv("collegePlace.csv")
     x=df[['Age']]
     df['Age']=sc.fit_transform(x)
     print(df['Age'])
```

```
0       0.388131
1      -0.366752
2       0.388131
3      -0.366752
4       0.388131
          ...
2961    1.143013
2962    1.143013
2963    0.388131
2964    0.388131
2965    1.143013
Name: Age, Length: 2966, dtype: float64
```

```
[ ] from numpy.core.fromnumeric import shape
    from sklearn.model_selection import train_test_split
    x=df.drop('PlacedOrNot',axis=1)
    y=df['PlacedOrNot']
    x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,stratify=y,random_state=0)
    df.shape
    print(x_train.shape)
    print(x_test.shape)
    print(y_train.shape)
    print(y_test.shape)
```

```
(2372, 7)
(594, 7)
(2372,)
(594,)
```

```
[ ] # Importing the necessary libraries
    import numpy as np
    from sklearn.datasets import load_digits
    from sklearn.model_selection import train_test_split
    from sklearn.svm import SVC
    from sklearn.metrics import  confusion_matrix,accuracy_score

    # Importing the dataset from the sklearn library into a local variable called dataset
    dataset = load_digits()
    x_train, x_test, y_train, y_test = train_test_split(dataset.data, dataset.target,test_size=0.30, random_state=4)
    Classifier = SVC(kernel="linear")
    Classifier.fit(x_train, y_train)
    y_pred = Classifier.predict(x_test)
    accuracy = accuracy_score(y_test,y_pred)
    confusion_mat = confusion_matrix(y_test,y_pred)
    print("Accuracy for SVM is:",accuracy)
```

```
Accuracy for SVM is: 0.9796296296296296
```

```
[ ] 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
          print("---Result---\nk:{}\n:score:{}".format(best_k,best_score))
          knn=KNeighborsClassifier(n_neighbors=best_k["Regular"])
          knn.fit(x_train,y_train)
          knn_pred=knn.predict(x_test)
          tested=accuracy_score(knn_pred,y_test)
```

```
---Result---
k:{'Regular': 3}
:score:{'Regular': 98.51851851851852}
---Result---
k:{'Regular': 5}
:score:{'Regular': 98.51851851851852}
```

```
[ ] import tensorflow as tf
    from tensorflow import keras
    from keras.models import Sequential
    from tensorflow.keras import layers
```

```
[ ] classifier=Sequential()
    classifier.add(keras.layers.Dense(6,activation='relu',input_dim=6))
    classifier.add(keras.layers.Dropout(0.50))
    classifier.add(keras.layers.Dense(6,activation='relu'))
    classifier.add(keras.layers.Dropout(0.50))
    classifier.add(keras.layers.Dense(1,activation='sigmoid'))
```

```
[ ] loss_1=tf.keras.losses.BinaryCrossentropy()
    classifier.compile(optimizer='Adam',loss=loss_1,metrics=["accuracy"])
```

```
#fitting the model
classifier. fit(x_train, y_train, batch_size= 20, epochs = 100)
```

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

app.py - C:\Users\ELCOT\Downloads\placement_prediction-master\app.py (3.7.9)

File Edit Format Run Options Window Help

```
import numpy as np
import model
from flask import Flask, request, render_template
import pickle
import pandas as pd
import sklearn as sk

app = Flask(__name__,template_folder="templates")
model = pickle.load(open('model.pkl', 'rb'))

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/predict',methods=['GET'])
def predict():

    age = request.args.get('age')
    gender = request.args.get('gender')
    stream = request.args.get('stream')
    internship = request.args.get('internship')
    cgpa = request.args.get('cgpa')
    backlogs = request.args.get('backlogs')
    arr = np.array([gender,stream,internship,cgpa,backlogs])
    brr = np.asarray(arr, dtype=float)
    output = model.predict([brr])
    if(output==1):
        out = 'Yes'
    else:
        out = 'No'
    return render_template('out.html', output=out)

if __name__ == "__main__":
    app.run(debug=True)
```

Python 3.7.9 Shell

File Edit Shell Debug Options Window Help

```
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\ELCOT\Downloads\placement_prediction-master\app.py =====

Warning (from warnings module):
  File "C:\Users\ELCOT\AppData\Local\Programs\Python\Python37\lib\site-packages\sklearn\base.py", line 451
    "X does not have valid feature names, but"
UserWarning: X does not have valid feature names, but DecisionTreeClassifier was fitted with feature names
[0]
 * Serving Flask app 'app'
 * Debug mode: on
[31m[1mWARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.[0m
 * Running on http://127.0.0.1:5000
[33mPress CTRL+C to quit[0m
 * Restarting with stat
```