IDENTIFYING PATTERNS AND TRENDS IN CAMPUS PLACEMENT DATA USING MACHINE LEARNING

PROJECT REPORT

Submitted by

TEAM MEMBERS :

TEAM ID : NM2023TMID32544

DEEPIKA. V

SARANYA. R

JEEVITHA. D

SUWETHA. S

TAMILSELVAN. S

BHARATHKUMAR. A

In partial fulfilment of the requirements for the award of the degree of Bachelor of Computer Science of Bharathiar University, Coimbatore - 46.

Under the Guidance of

Prof. B. HEMALATHA MCA., B.Ed., M.Phil., NET-UGC.,

Assistant Professor and Head of the

DEPARTMENT OF COMPUTER SCIENCE

GOVERNMENT ARTS AND SCIENCE COLLEGE (CO-ED)

(Affiliated to Bharathiar University, Coimbatore)

AVINASHI - 641 654

APRIL 2023

GOVERNMENT ARTS AND SCIENCE COLLEGE (CO-ED)

AVINASHI-641 654

(Affiliated to Bharathiar University Coimbatore)

NAAN MUDHALVAN PROJECT WORK

This is to certify that project work entitled

IDENTIFYING PATTERNS AND TRENDS IN CAMPUS PLACEMENT DATA USING MACHINE LEARNING

is the bonafide record of project work done by the above Students of III B.Sc., (CS) degree **NAAN MUDHALVAN PROJECT** during the year 2022-2023

In partial fulfilment of requirement for the degree of Bachelor of Science in Computer Science of Bharathiar University

Submitted for the Naan Mudhalvan project held on _______________

Class Mentor	Head of Department
Dr.A.Geetha	**Prof.B.Hemalatha**

TABLE OF CONTENTS

CHAPTER NO	CONTENTS
1	INTRODUCTION
2	ADVANTAGE AND DISADVANTAGE
3	APPLICATIONS
4	CONCLUSION
5	APPENDICES A. TECHNICAL FLOW B. SAMPLE INPUT C. SAMPLE OUTPUT D. SAMPLE CODING

CAMPUS PLACEMENT DATA USING MACHINE LEARNING

INTRODUCTION:

OVERVIEW:

The placement both for final jobs and summer internships is an integral part of any institute's annual calendar of activities.

Campus placement is hiring young talent for internships and entry level positions.

PURPOSE :

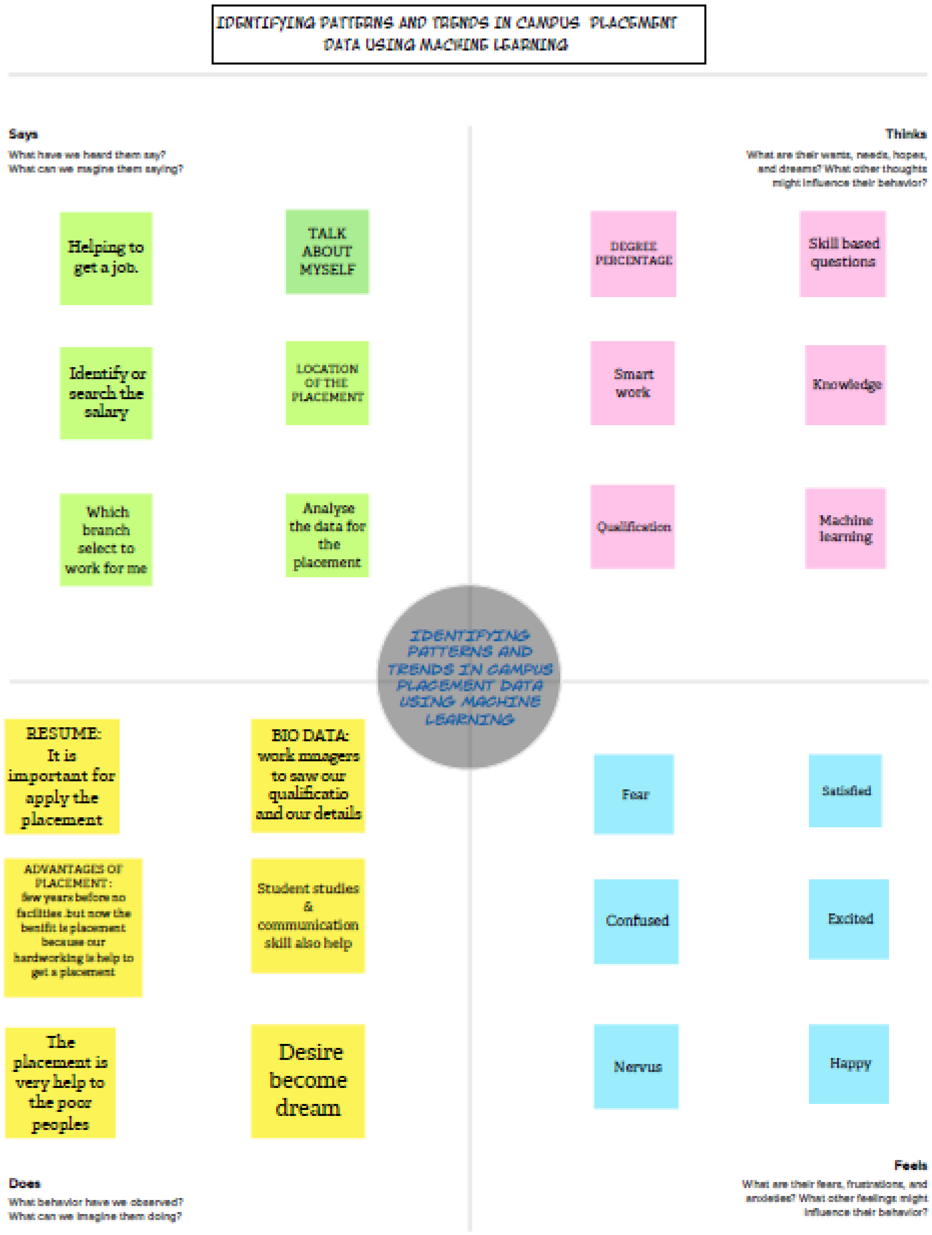
The companies will be benefited from getting wide choice of candidates to select for different job posts. Companies can select the right and talented candidate from a vast pool of young applicants within a limited time. On the other hand, students have the advantage of getting a good job according to their qualification level even before the completion of their academic course in college.

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.

PROBLEM DEFINITION AND DESIGN THINKING

EMPATHY MAP



IDENTIFYING PATTERNS AND TRENDS IN CAMPUS PLACEMENT DATA USING MACHINE LEARNING

Says

What have we heard them say?
What can we magine them saying?

Helping to get a job.

TALK ABOUT MYSELF

Identify or search the salary

LOCATION OF THE PLACEMENT

Which branch select to work for me

Analyse the data for the placement

Thinks

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

DEGREE PERCENTAGE

Skill based questions

Smart work

Knowledge

Qualification

Machine learning

IDENTIFYING PATTERNS AND TRENDS IN CAMPUS PLACEMENT DATA USING MACHINE LEARNING

RESUME: It is important for apply the placement

BIO DATA: work mnagers to saw our qualificatio and our details

ADVANTAGES OF PLACEMENT : few years before no facilities but now the benifit is placement because our hardworking is help to get a placement

Student studies & communication skill also help

The placement is very help to the poor peoples

Desire become dream

Does

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

Fear

Satisfied

Confused

Excited

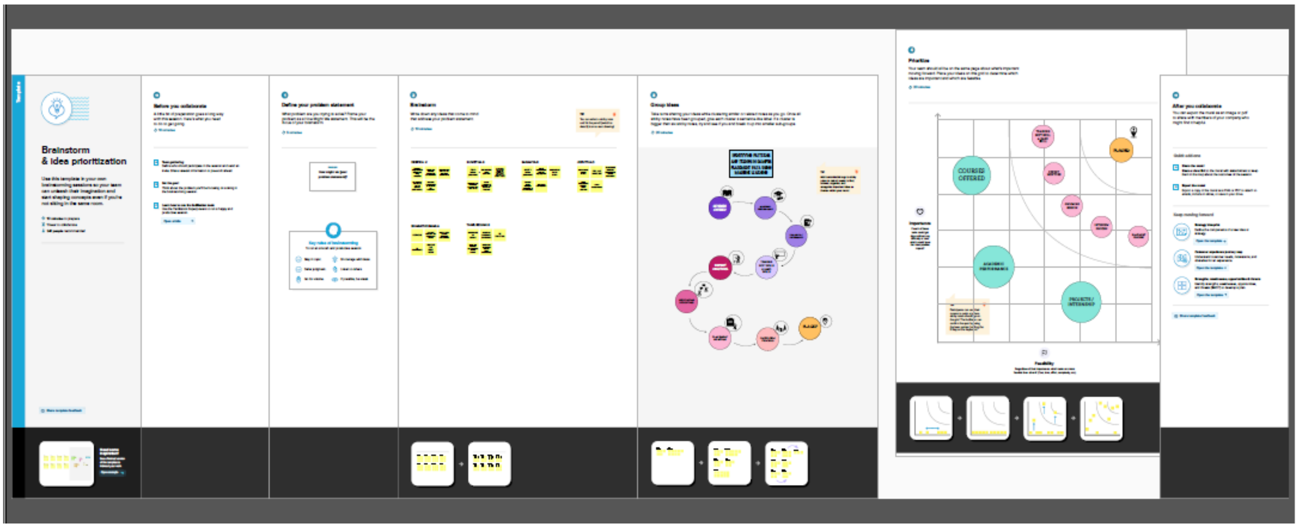
Nervus

Happy

Feels

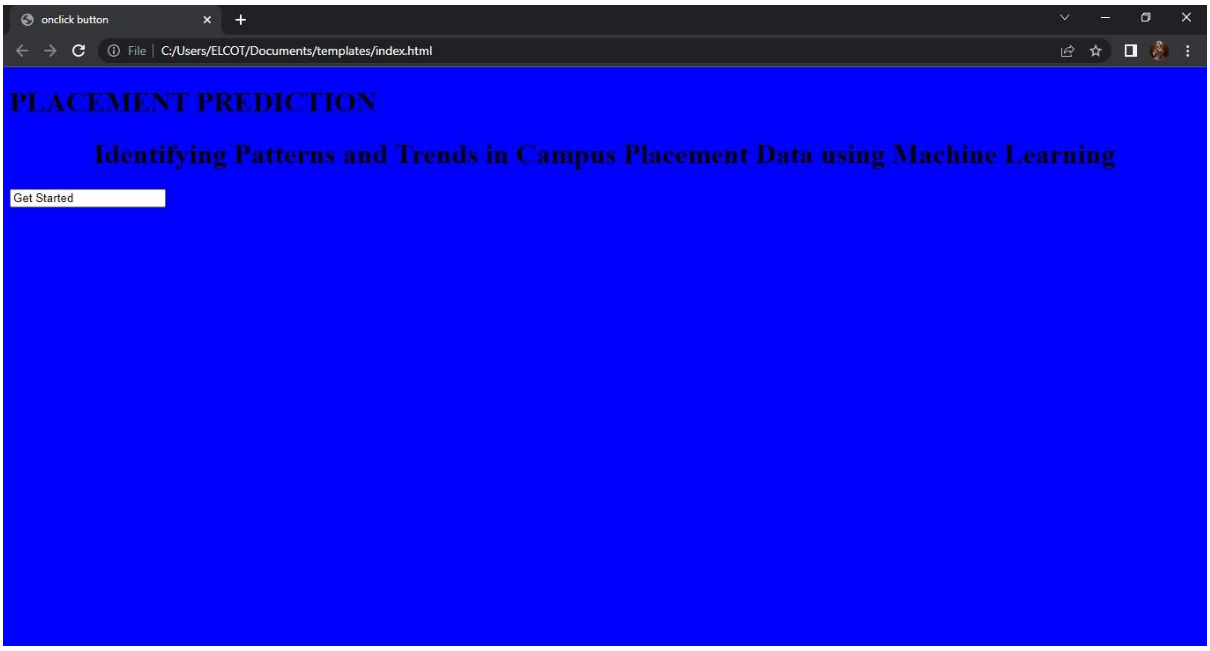
What are their fears, frustrations, and anxieties? What other feelings might influence their behavior?

BRAINSTROMING MAP



SAMPLE INPUT AND OUTPUT OF THE PROJECT:

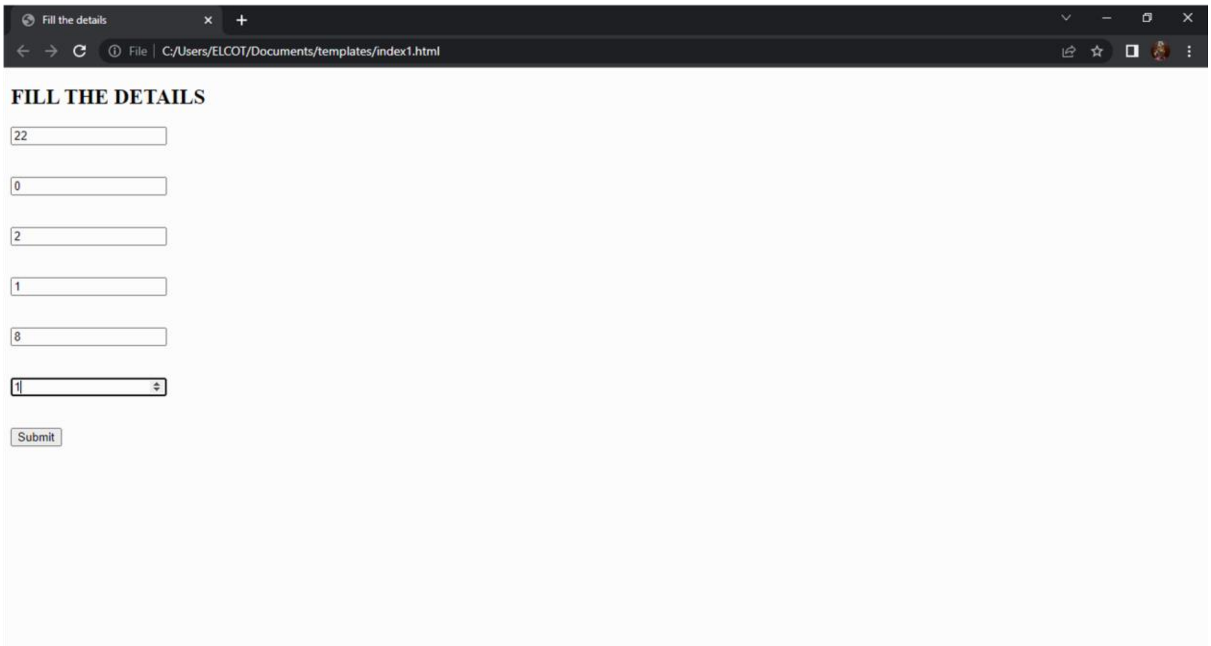
- ➢ **Index.html**



PLACEMENT PREDICTION

Identifying Patterns and Trends in Campus Placement Data using Machine Learning

➢ Index1.html



Fill the details

C:/Users/ELCOT/Documents/templates/index1.html

FILL THE DETAILS

22

0

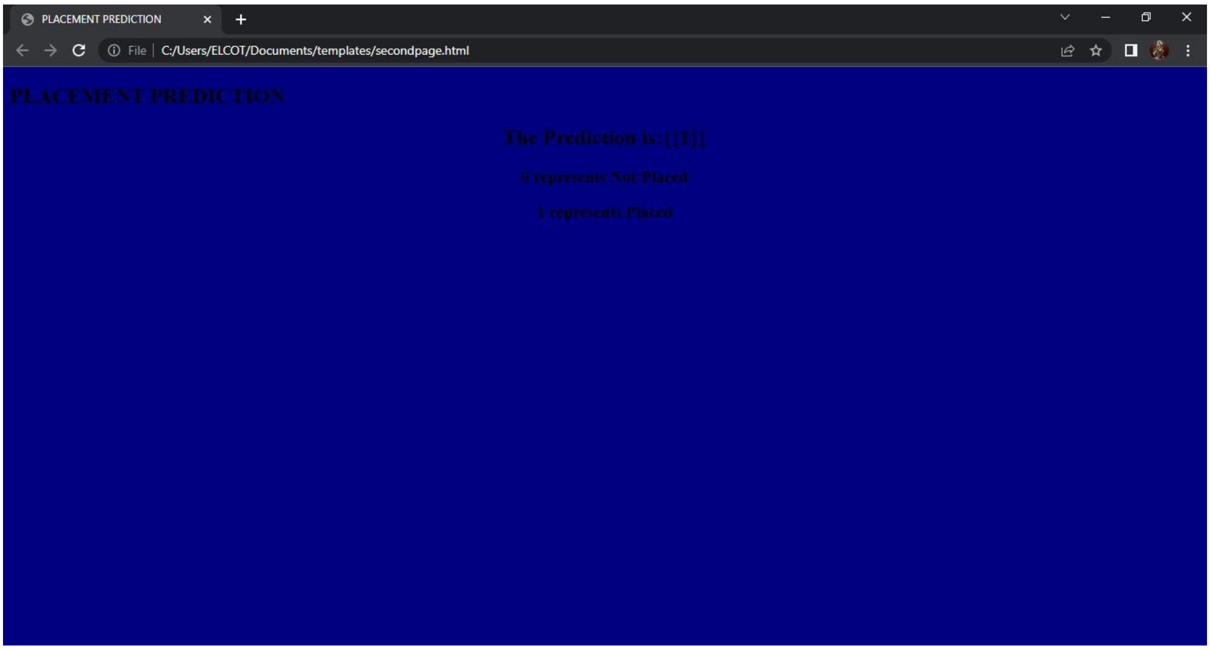
2

1

8

1

➢ Second page.html



PLACEMENT PREDICTION

C:/Users/ELCOT/Documents/templates/secondpage.html

PLACEMENT PREDICTION

The Prediction is:[[1]]

0 represents Not Placed

1 represents Placed

ADVANTAGE OF CAMPUS PLACEMENT:

- The companies will be benefited from getting wide choice of candidates to select for different jobs.
- The companies will be benefited from getting wide choice of candidates to select for different job posts.
- Companies can select the right and talented candidate from a vast pool of young applicants within a limited time.
- On the other hand, students have the advantage of getting a good job according

to their qualification level even before the completion of their academic course in college.

DISADVANTAGES OF CAMPUS PLACEMENT:

- 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.
- This is an additional expense for the company. Also, students can’t work with their dream company and will have to remain satisfied with the company that recruits them during campus selection.

APPLICATION

- Companies hold on campus recruitment drives for students in their final year, and sever large.
- You can expect questions related to coding, algorithm and machine learning.

CONCLUSION

- Goal for future placement
- Performance of student
  At the completion of placement, student and supervisors should complete the end of placement evaluation form.

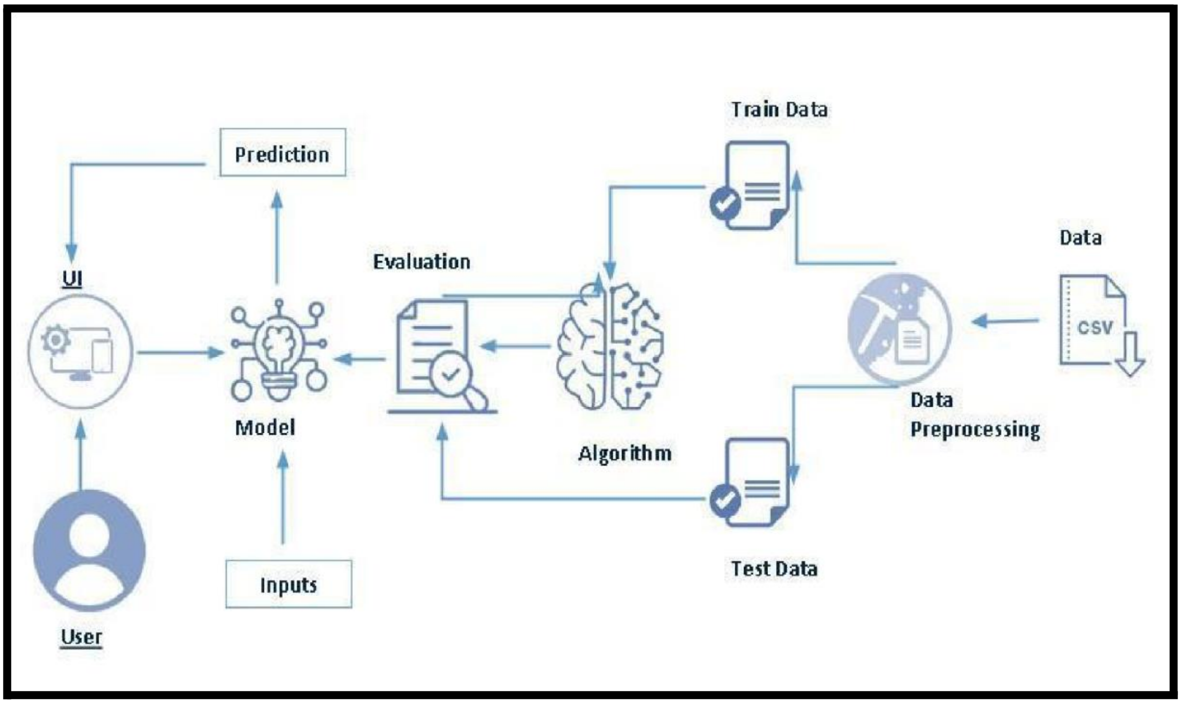
To determine what merits satisfactory or unsatisfactory performance on placement.

FUTURE SCOPE

- Given the boom in the market, college grads or undergrads have access to a wide pool of employers promising attractive roles and benefits.
- We need to use technology and creative communication strategies to stay top of mind in our target demographic.

APPENDIX

Technical Architecture:

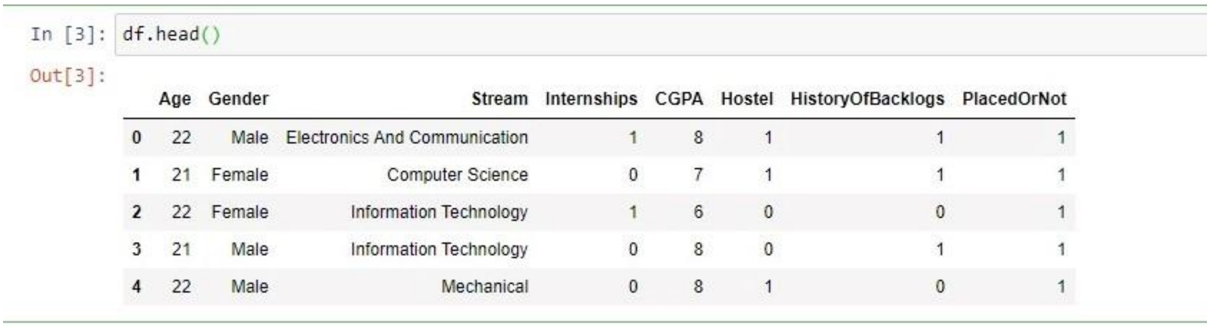


Source code

```
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('../Dataset/collegePlace.csv')
```

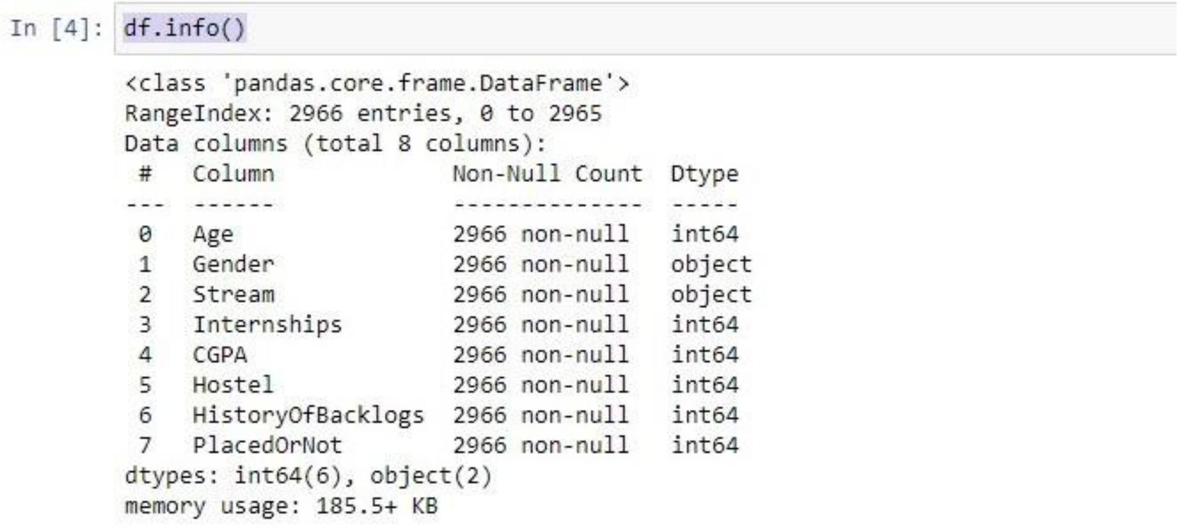
```
df.head()
```



```
In [3]: df.head()
Out[3]:
```

	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

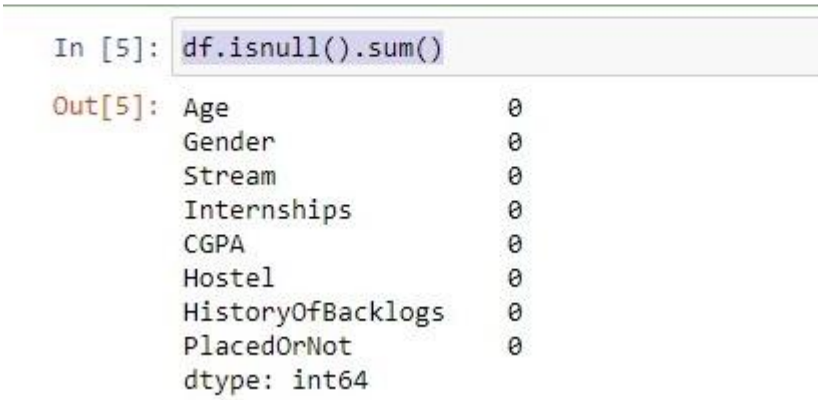
```
df.info()
```



```
In [4]: 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
```

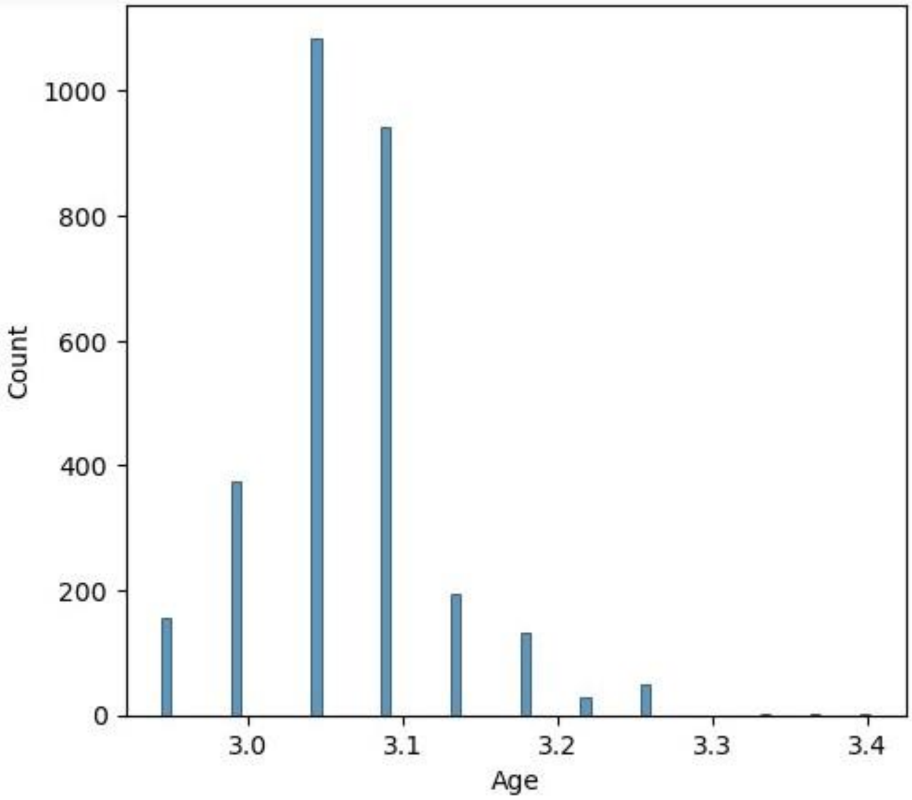
```
df.isnull().sum()
```



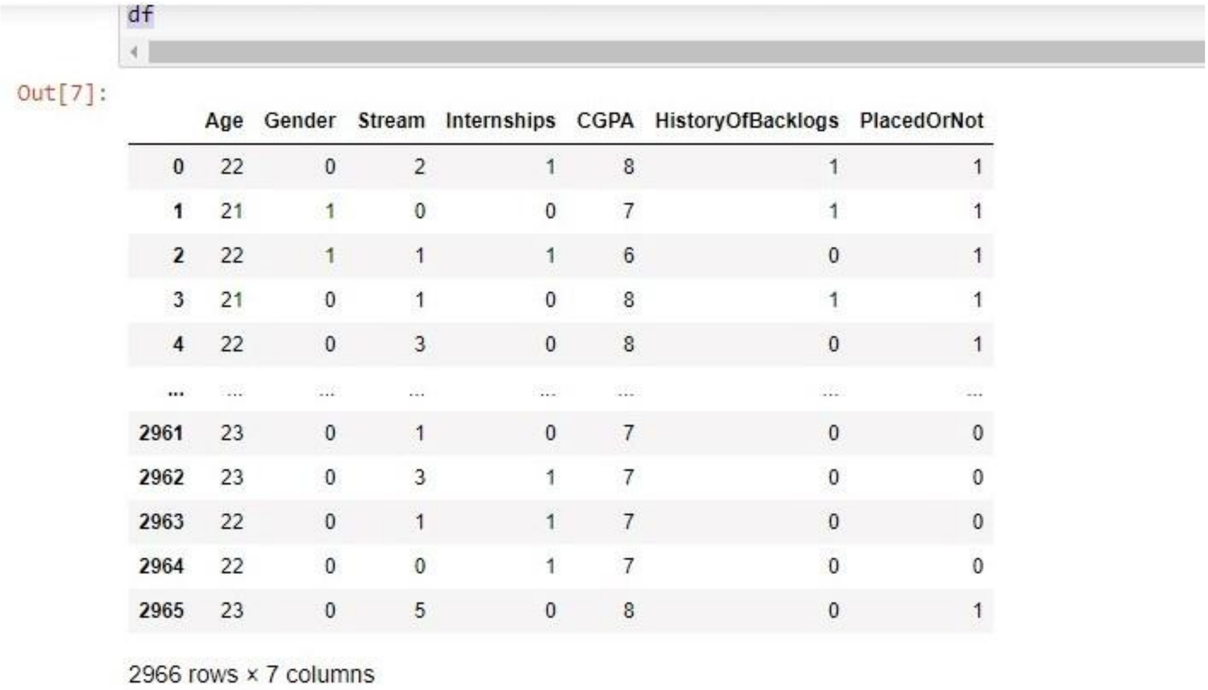
```
In [5]: df.isnull().sum()

Out[5]: 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.histplot(feature)
    transformationplot(np.log(df['Age']))
```



```
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
```



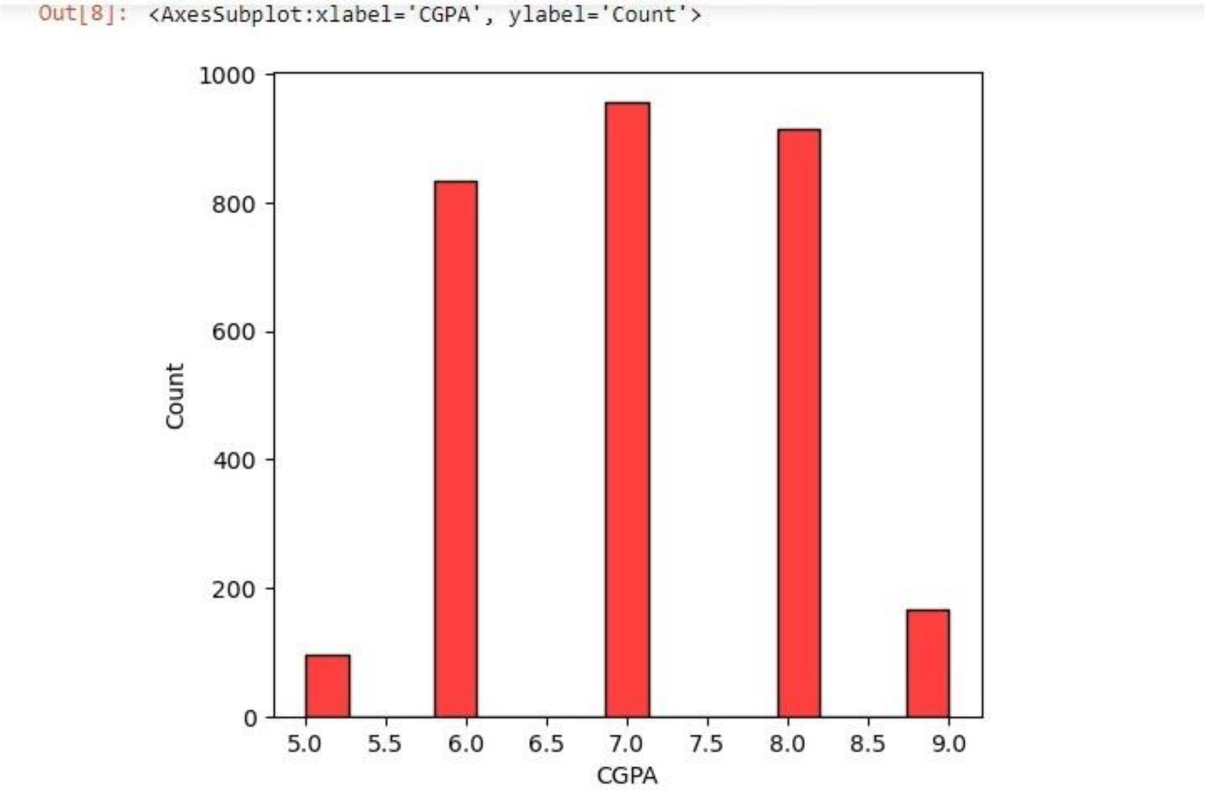
```
df
```

Out[7]:

	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

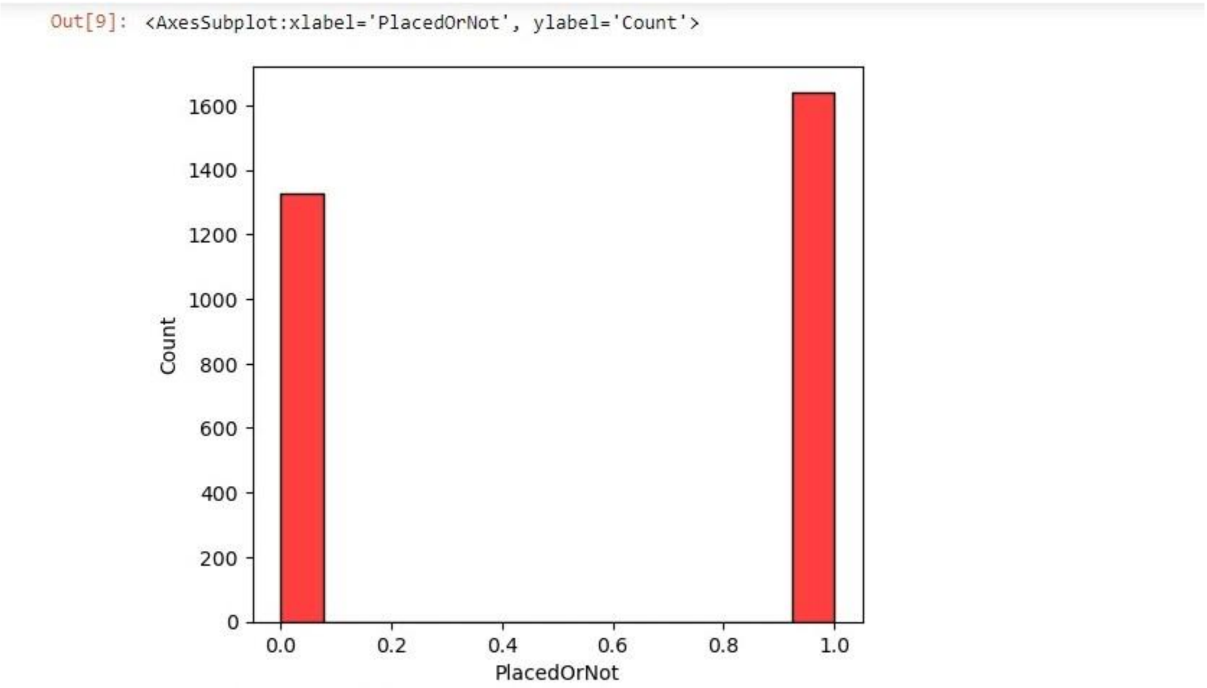
2966 rows × 7 columns

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



```
Out[8]: <AxesSubplot:xlabel='CGPA', ylabel='Count'>
```

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

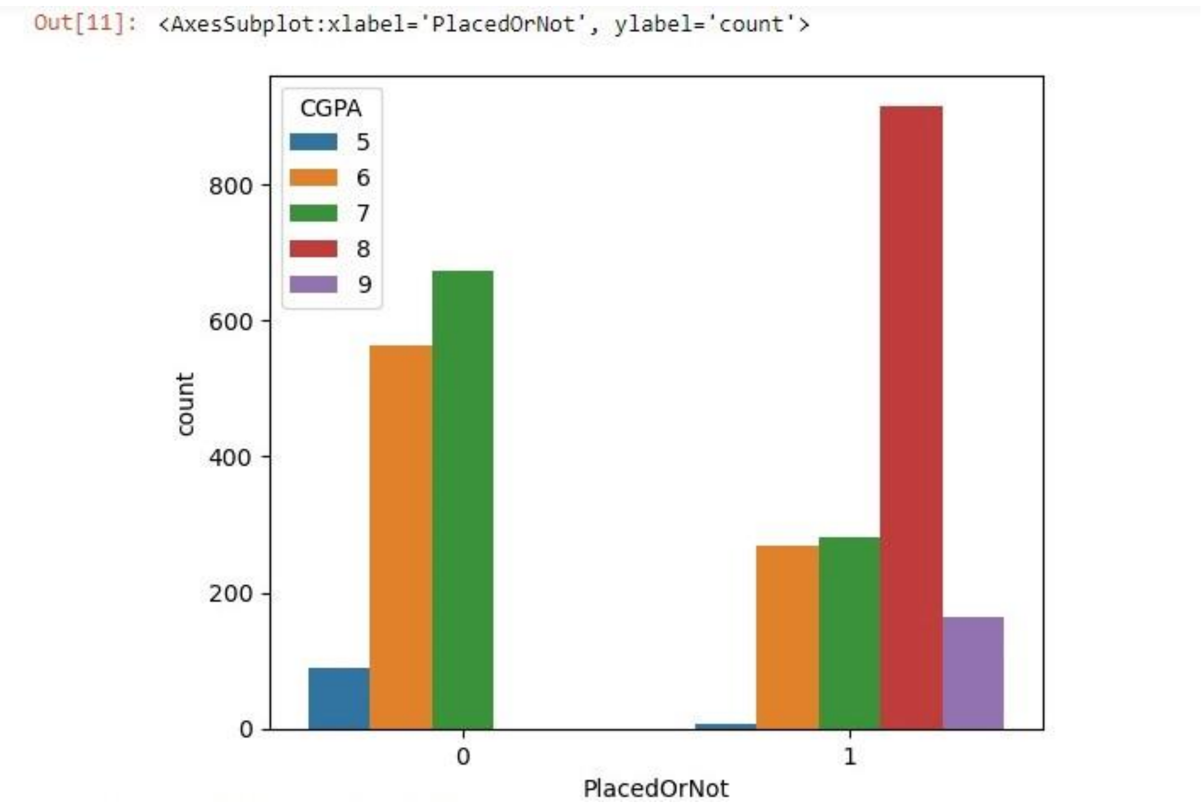


```
Out[9]: <AxesSubplot:xlabel='PlacedOrNot', ylabel='Count'>
```

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

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

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



```
Out[11]: <AxesSubplot:xlabel='PlacedOrNot', ylabel='count'>
```

```
# Feature scaling
sc = StandardScaler()
X = sc.fit_transform(df.drop(['PlacedOrNot'], axis=1))
y = df['PlacedOrNot']

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, stratify=y,
random_state=2)

# SVM model
svm_model = svm.SVC(kernel='linear')
svm_model.fit(X_train, y_train)
y_pred = svm_model.predict(X_test)
svm_accuracy = accuracy_score(y_test, y_pred)
print('Accuracy score of the SVM model: ', svm_accuracy)

from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics

best_k = {"Regular":0}
best_score = {"Regular":0}

for k in range(3, 50, 2):
    #using Regular training set
    knn_temp = KNeighborsClassifier(n_neighbors=k)  #instantiate the model
    knn_temp.fit(X_train, y_train) #Fit the model to the training set
    knn_temp_pred = knn_temp.predict(X_test) #Predict on the test set
    score = metrics.accuracy_score(y_test, knn_temp_pred)*100 #Get accuracy
    if score >= best_score["Regular"] and score < 100: #store best params
        best_score["Regular"] = score
        best_k["Regular"] = k

print("---Results---\nK: {}\nscore: {}".format(best_k,best_score))

##instantiate the models
knn = KNeighborsClassifier(n_neighbors=best_k["Regular"])

##Fit the model to the traning set
knn.fit(X_train, y_train)
knn_pred = knn.predict(X_test)
testd = metrics.accuracy_score(knn_pred, y_test)
print('Accuracy score of the KNN model: ', testd)
```

```
import numpy as np
import pandas as pd
import pickle
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn import preprocessing

# Load dataset
df = pd.read_csv('../Dataset/collegePlace.csv')

# Split into training and testing sets
x = df.drop('PlacedOrNot', axis='columns')
x = x.drop('Hostel', axis='columns')
y = df['PlacedOrNot']
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=100)

# Preprocess data
le = preprocessing.LabelEncoder()
le.fit(X_train['Gender'])
X_train['Gender'] = le.transform(X_train['Gender'])
X_test['Gender'] = le.transform(X_test['Gender'])
le.fit(X_train['Stream'])
X_train['Stream'] = le.transform(X_train['Stream'])
X_test['Stream'] = le.transform(X_test['Stream'])

# Train model
classify = KNeighborsClassifier(n_neighbors=5)
classify.fit(X_train, y_train)

# Save model
with open('../Flask/rdf.pkl', 'wb') as f:
    pickle.dump(classify, f)
```

```
# Load model and make prediction
with open('../Flask/rdf.pkl', 'rb') as f:
    model = pickle.load(f)

# Make prediction
prediction = model.predict([[1, 1, 1, 0, 0, 1]])
print(prediction)
```