

IDENTIFYING PATTERNS AND TRENDS IN CAMPUS PLACEMENT DATA USING MACHINE LEARNING.....

TEAM ID:61896B44EFD1F71B0F
04A99ACDE4CA72

TEAM SIZE :4

TEAM LEADER:SAKTHIVEL.V(20UCS2544)

TEAM MEMBERS:SIVAMUTHU.S(20UCS2545)
THANGAARASU.P(20UCS2547)
VALLARASU(20UCS2548)

PROJECT REPORT

1. INTRODUCTION :

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.

1.1 OVERVIEW :

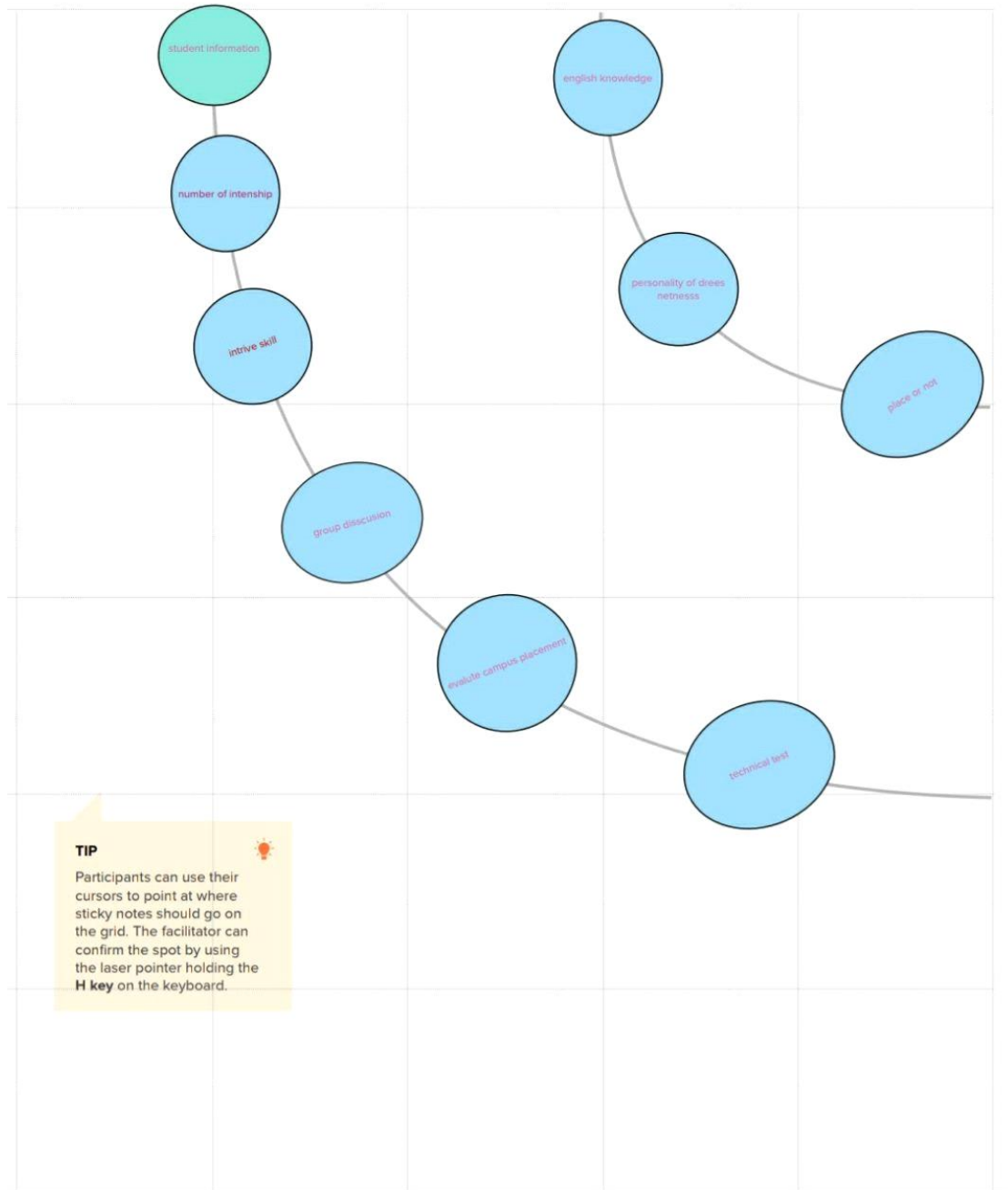
It refers to recruitment from educational institutions. Some big organisations remain in touch with the educational institutions with the purpose of recruiting young talented people. These persons come from colleges, universities, management institutes, technical institutes, etc.

1.2 PURPOSE :

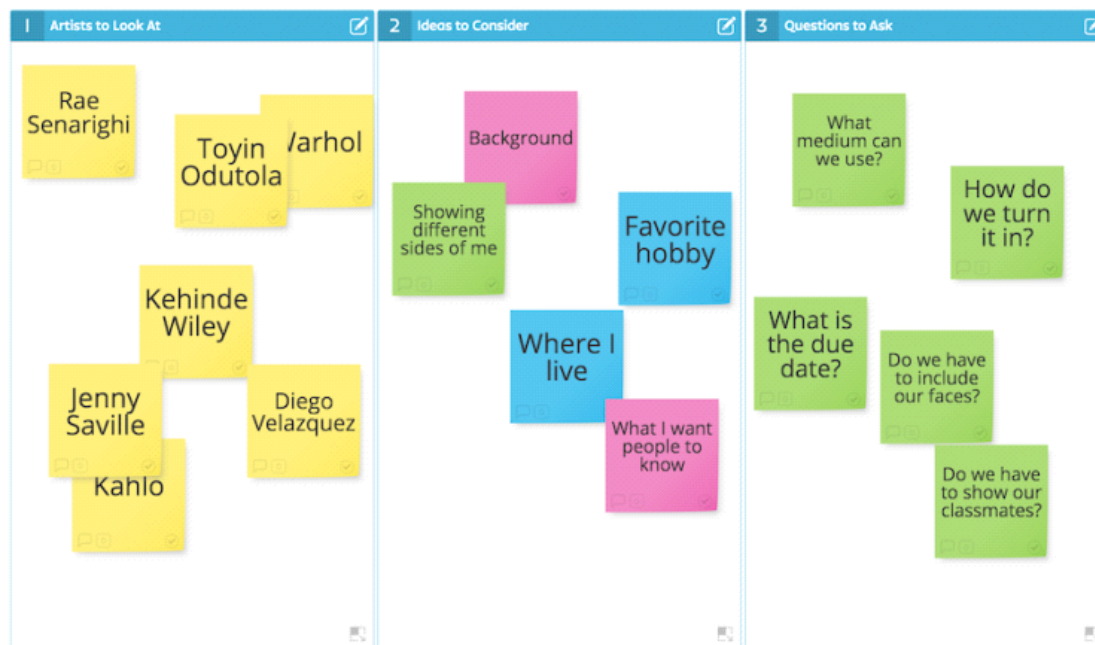
If you do not get placed through campus then first thing you should understand that there is a world full of opportunities out of the college and yes there are thousands of companies for you so no need to lose hope and feel belittle.

2. PROBLEM DEFINITION & DESIGN THINKING

2.1 EMPATHY MAP :



2.2 IDEATION & BRAIN STROMING :



3. RESULT :

	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

(2966, 8)

```

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

```

Computer Science 776

Information Technology 691

Electronics And Communication 424

Mechanical 424

Electrical 334

Civil 317

Name: Stream, dtype: int64

CSE 776

IT 691

ECE 424

MECH 424

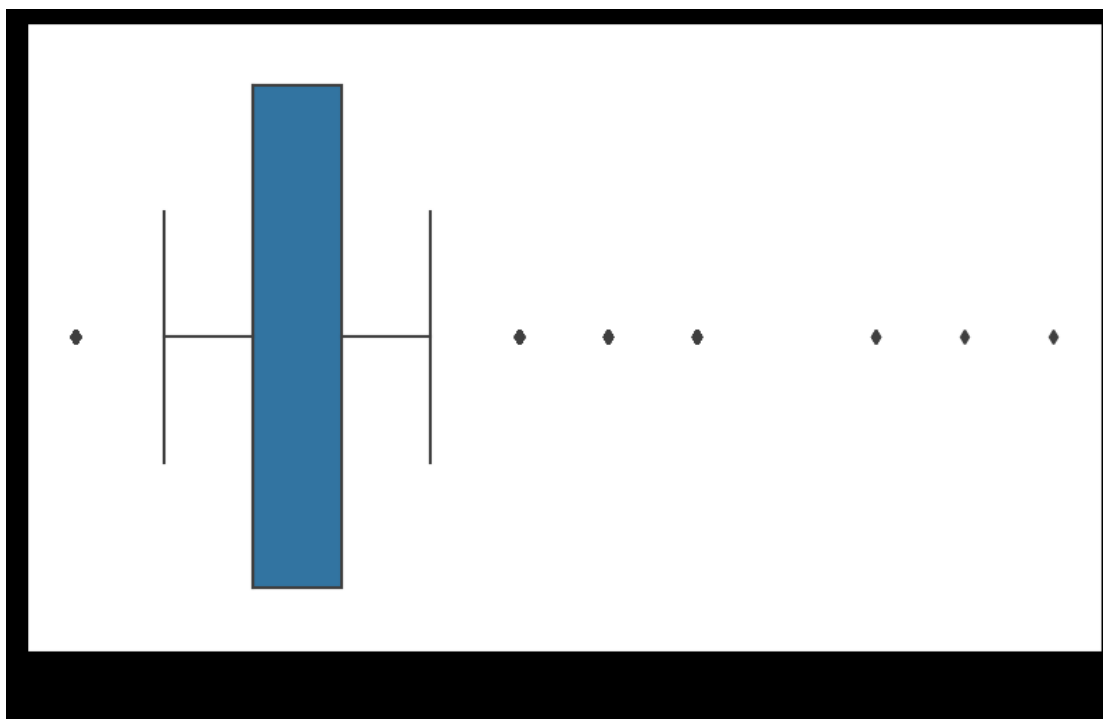
EC 334

Civil 317

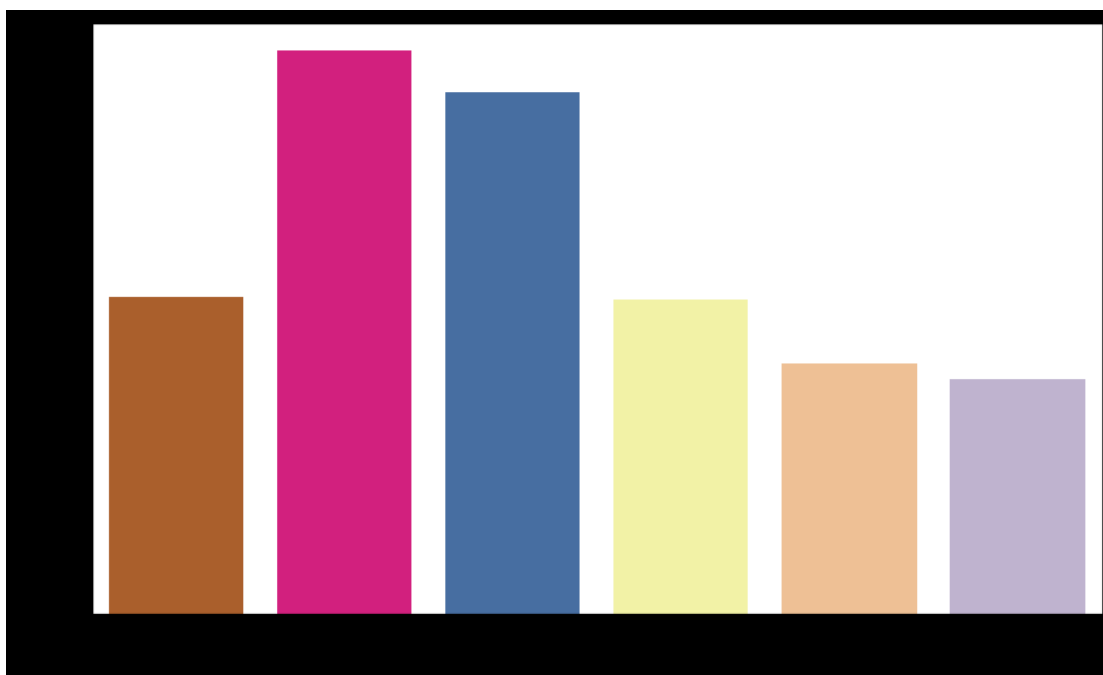
Name: Stream, dtype: int64

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

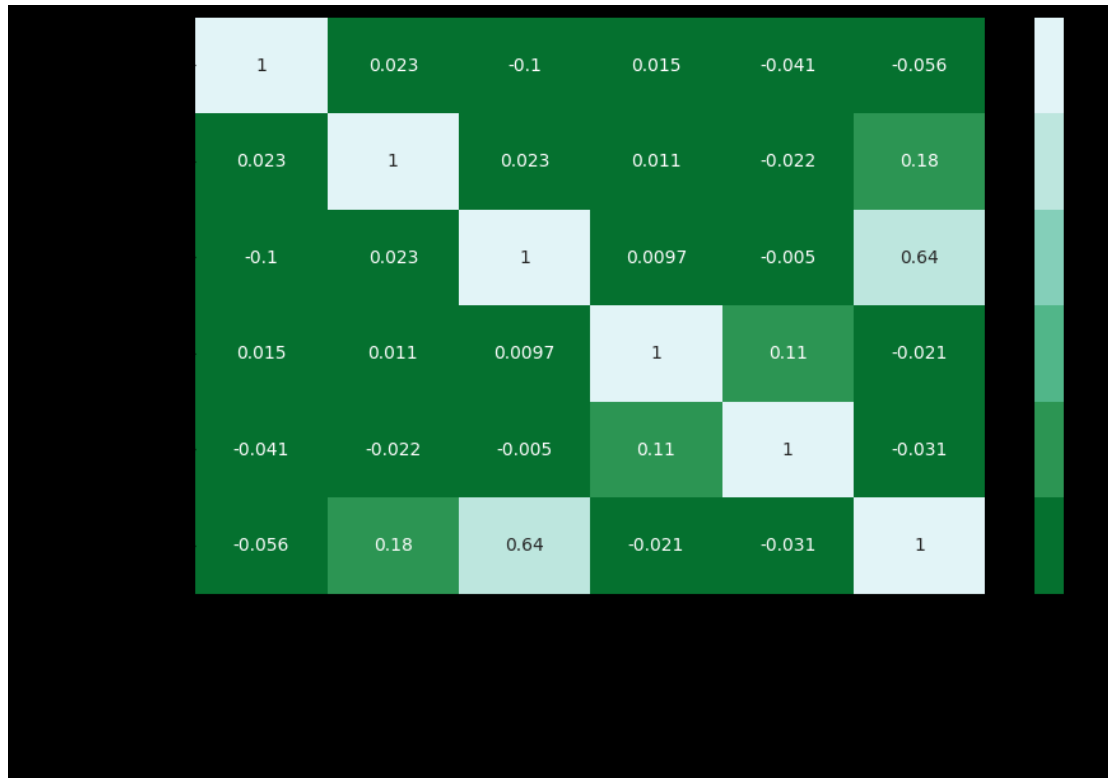
<AxesSubplot:xlabel='Age'>



<AxesSubplot:xlabel='Stream', ylabel='count'>



<AxesSubplot:>



0.8857509627727856

0.8741976893453145.

4. ADVANTAGES & DISADVANTAGES

ADVANTAGE :

1. Saves Time & Efforts
2. Improved Retention Rates
3. Getting New Knowledge & Skills
4. Quick Learners & Multi-tasking candidates
5. Good relationship between Organization & Campus

DISADVANTAGE :

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.

5. APPLICATIONS

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.

The educational institutions partner with corporations who wish to recruit from the student population.

6. CONCLUSION

At the completion of placement, students and supervisors should complete the end of placement evaluation form.

Let us know how you found your placement and what skills and knowledge you think you have gained.

7. FUTURE SCOPE

In the progress will starting to end period we all satisfy

we will get long time so more information before giving time of underworld

8. APPENDIX

a) source code

```
import numpy as np

import pandas as pd

import os

import seaborn as sns

import matplotlib.pyplot as plt

import plotly.express as px

import plotly.graph_objects as go

from plotly.subplots import make_subplots

from sklearn import preprocessing

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression

from sklearn.ensemble import RandomForestClassifier

from xgboost import XGBClassifier
```

```

from sklearn.metrics import accuracy_score

import plotly.express as px

df = pd.read_csv("/kaggle/input/engineering-placements-prediction/collegePlace.csv")

df.head()

df.shape

df.info()

df.isnull().sum()

df['Stream'].value_counts()

mapping = {"Electronics And Communication": "ECE", "Computer Science": "CSE", "Information
Technology": "IT", "Mechanical": "MECH", "Civil": "Civil", "Electrical": "EC"}

df["Stream"] = df["Stream"].map(mapping)

df['Stream'].value_counts()

df['Stream'].value_counts()

df.describe()


# I tried all the columns and find out that only age column has some outliers.


plt.figure(figsize = (10, 6), dpi = 100)

sns.boxplot(x = "Age", data = df)

max_threshold = df['Age'].quantile(0.95)

print(max_threshold)


min_threshold = df['Age'].quantile(0.01)

print(min_threshold)

```



```
df = df[(df['Age'] < max_threshold) & (df['Age'] > min_threshold)]
```

```
24.0
```

```
19.0
```

```
plt.figure(figsize = (10, 6), dpi = 100)
```

```
color_palette = sns.color_palette("Accent_r")
```

```
sns.set_palette(color_palette)
```

```
sns.countplot(x = "Stream", data = df)
```

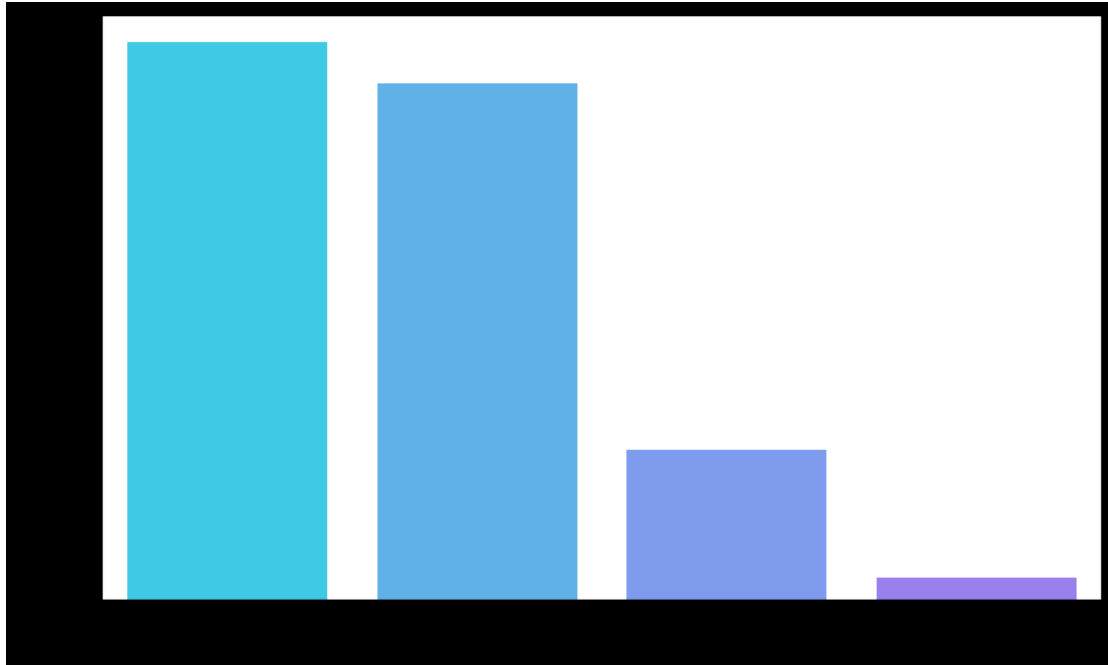
```
plt.figure(figsize = (10, 6), dpi = 100)
```

```
color_palette = sns.color_palette("cool")
```

```
sns.set_palette(color_palette)
```

```
sns.countplot(x = "Internships", data = df)
```

```
plt.show()
```



```
plt.figure(figsize = (10, 6), dpi = 100)
```

```
grp = dict(df.groupby('CGPA').groups)
```

```
m = {}
```

```
for key, val in grp.items():
```

```
    if key in m:
```

```
        m[key] += len(val)
```

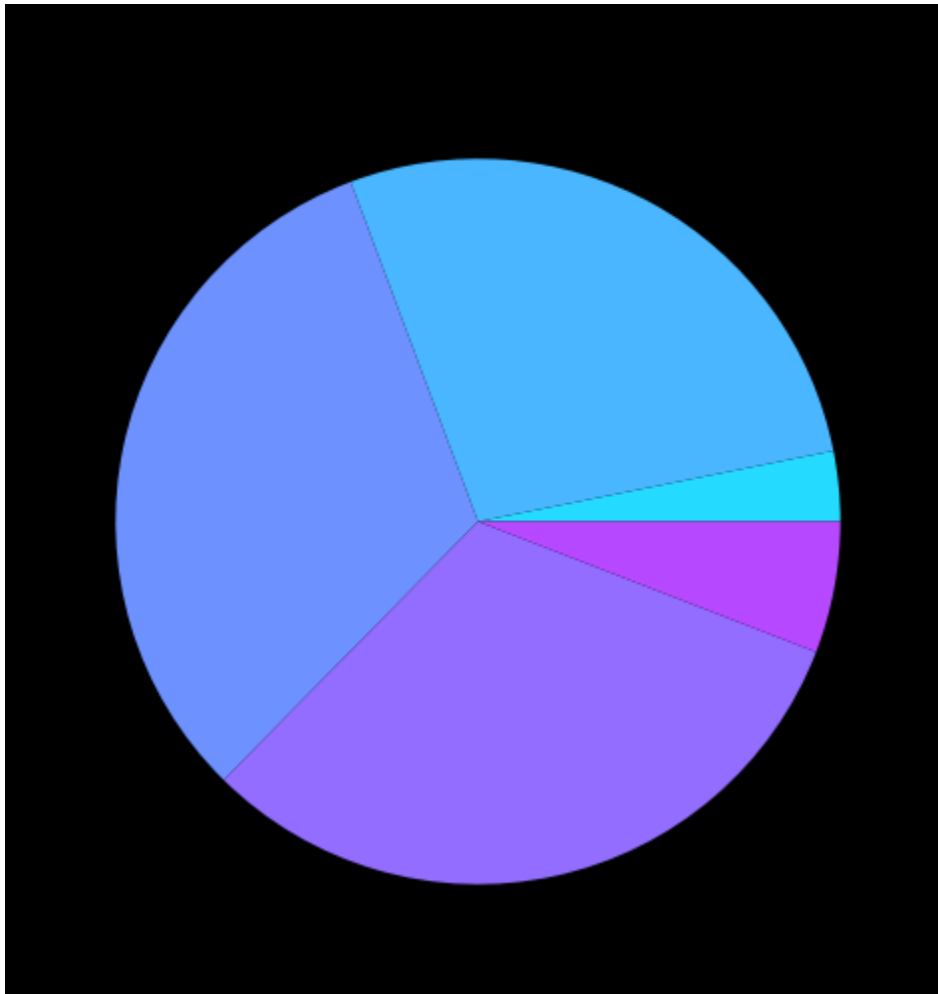
```
    else:
```

```
        m[key] = len(val)
```

```
plt.title("Distribution of CGPA")
```

```
plt.pie(m.values(), labels = m.keys())
```

```
plt.show()
```



```
plt.figure(figsize = (10, 6), dpi = 100)
```

```
# setting the different color palette
```

```
color_palette = sns.color_palette("Accent_r")
```

```
sns.set_palette(color_palette)
```

```
sns.countplot(x = "Gender", data = df)
```

```
plt.show()
```



```
plt.figure(figsize = (10, 6), dpi = 100)
```

```
# setting the different color palette
```

```
color_palette = sns.color_palette("plasma")
```

```
sns.set_palette(color_palette)
```

```
sns.barplot(x = "PlacedOrNot", y = "Gender", data = df)
```

```
plt.show()
```



```
plt.figure(figsize = (10, 6), dpi = 100)
```

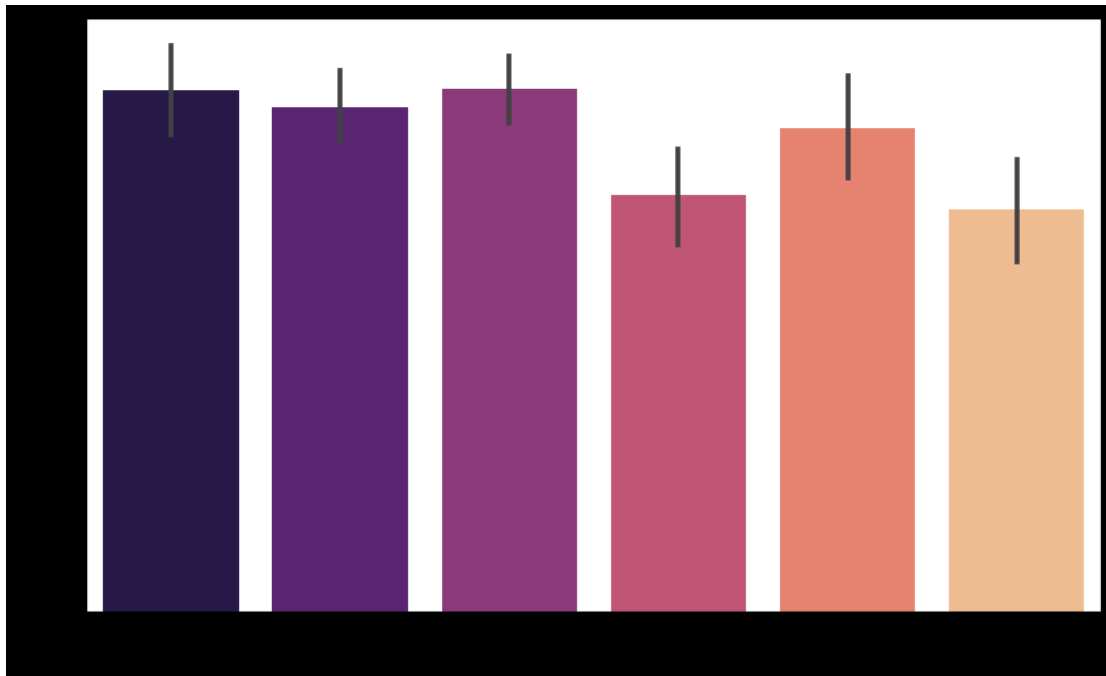
```
# setting the different color palette
```

```
color_palette = sns.color_palette("magma")
```

```
sns.set_palette(color_palette)
```

```
sns.barplot(x = "Stream", y = "PlacedOrNot", data = df)
```

```
plt.show()
```



```
plt.figure(figsize = (10, 6), dpi = 100)
```

```
# setting the different color palette
```

```
color_palette = sns.color_palette("BuGn_r")
```

```
sns.set_palette(color_palette)
```

```
sns.countplot(x = "PlacedOrNot", data = df)
```

```
plt.show()
```



```
plt.figure(figsize = (10, 6), dpi = 100)

color = sns.color_palette("BuGn_r")

sns.heatmap(df.corr(), vmax=0.9, annot=True,cmap = color)

le = preprocessing.LabelEncoder()

df["Gender"] = le.fit_transform(df["Gender"])

df["Stream"] = le.fit_transform(df["Stream"])

acc = accuracy_score(y_test, pred)

acc

from catboost import CatBoostClassifier

X = df[['Age', 'Gender', 'Internships', 'CGPA', 'Hostel',
        'HistoryOfBacklogs', 'Stream']]

y = df["PlacedOrNot"]
```

```
x_train, x_test, y_train, y_test = train_test_split(X,y, test_size=0.30, random_state=100)
```

```
clf = CatBoostClassifier(
```

```
    iterations = 5,
```

```
    learning_rate = 0.1,
```

```
    loss_function='CrossEntropy',
```

```
).fit(x_train, y_train)
```

```
pred = clf.predict(x_test)
```

```
0:   learn: 0.6410786      total: 56.9ms   remaining: 228ms
```

```
1:   learn: 0.5947409      total: 58ms     remaining: 86.9ms
```

```
2:   learn: 0.5558258      total: 58.7ms   remaining: 39.1ms
```

```
3:   learn: 0.5267040      total: 59.4ms   remaining: 14.8ms
```

```
4:   learn: 0.5015275      total: 59.9ms   remaining: 0us
```

```
acc = accuracy_score(y_test, pred)
```

```
acc
```