

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [10]: df=pd.read_csv(r"C:\Users\Admin\Downloads\13_placement - 13_placement.csv")
df
```

Out[10]:

	cgpa	placement_exam_marks	placed
0	7.19	26	1
1	7.46	38	1
2	7.54	40	1
3	6.42	8	1
4	7.23	17	0
...
995	8.87	44	1
996	9.12	65	1
997	4.89	34	0
998	8.62	46	1
999	4.90	10	1

1000 rows × 3 columns

```
In [19]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 3 columns):
#   Column                Non-Null Count  Dtype
---  -
0   cgpa                  1000 non-null  float64
1   placement_exam_marks 1000 non-null  int64
2   placed                1000 non-null  int64
dtypes: float64(1), int64(2)
memory usage: 23.6 KB
```

```
In [20]: df.info()
```

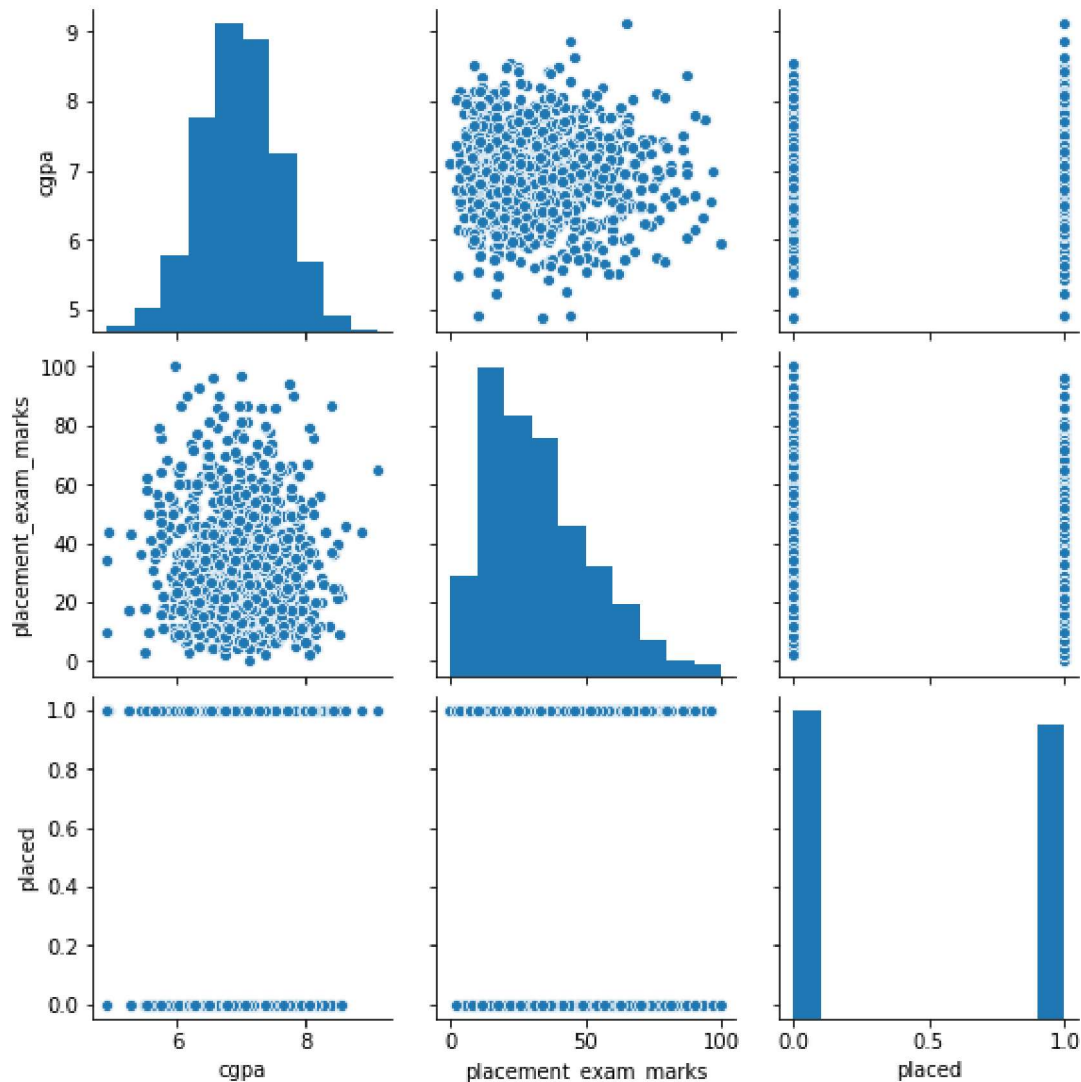
```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 1000 entries, 0 to 999  
Data columns (total 3 columns):  
#   Column                Non-Null Count  Dtype    
---  ---                  
0   cgpa                  1000 non-null   float64  
1   placement_exam_marks 1000 non-null   int64    
2   placed                1000 non-null   int64    
dtypes: float64(1), int64(2)  
memory usage: 23.6 KB
```

```
In [21]: df.columns
```

```
Out[21]: Index(['cgpa', 'placement_exam_marks', 'placed'], dtype='object')
```

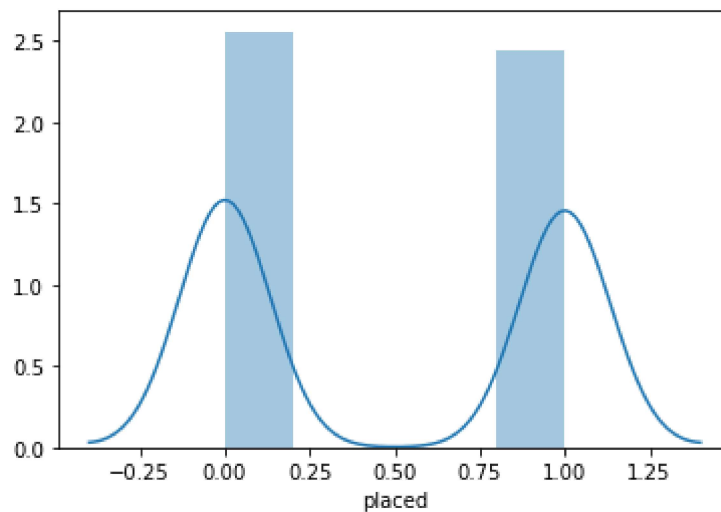
```
In [22]: sns.pairplot(df)
```

```
Out[22]: <seaborn.axisgrid.PairGrid at 0x19c0526f160>
```



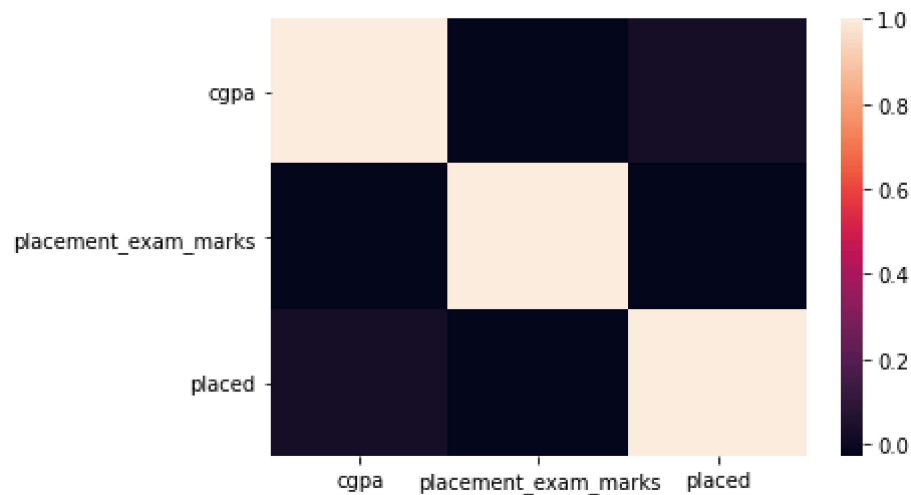
```
In [23]: sns.distplot(df['placed'])
```

```
Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x19c05701e50>
```



```
In [24]: sns.heatmap(df.corr())
```

```
Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0x19c058aaa60>
```



```
In [25]: x=df[['cgpa', 'placement_exam_marks']]
y=df[['placed']]
```

```
In [26]: from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3)
```

```
In [27]: from sklearn.linear_model import LinearRegression
lr= LinearRegression()
lr.fit(x_train,y_train)
```

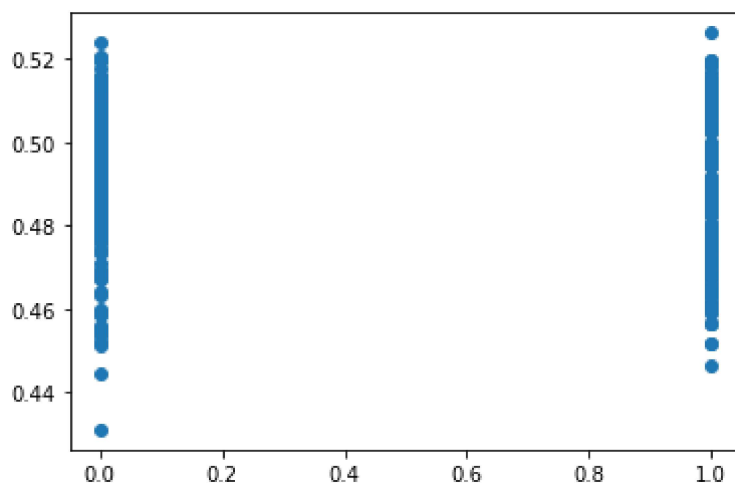
```
Out[27]: LinearRegression()
```

```
In [28]: print(lr.intercept_)
```

```
[0.62263853]
```

```
In [29]: prediction= lr.predict(x_test)  
plt.scatter(y_test,prediction)
```

Out[29]: <matplotlib.collections.PathCollection at 0x19c063787f0>



```
In [30]: print(lr.score(x_test,y_test))
```

-0.00545978267267877

```
In [ ]:
```