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

In [3]:
```
data=pd.read_csv(r"C:\Users\user\Downloads\13_placement - 13_placement.csv")
data
```

Out[3]:

	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 [4]:
```
data.head()
```

Out[4]:

	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

In [5]:
```
data.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 [6]:
```
data.describe()
```

Out[6]:

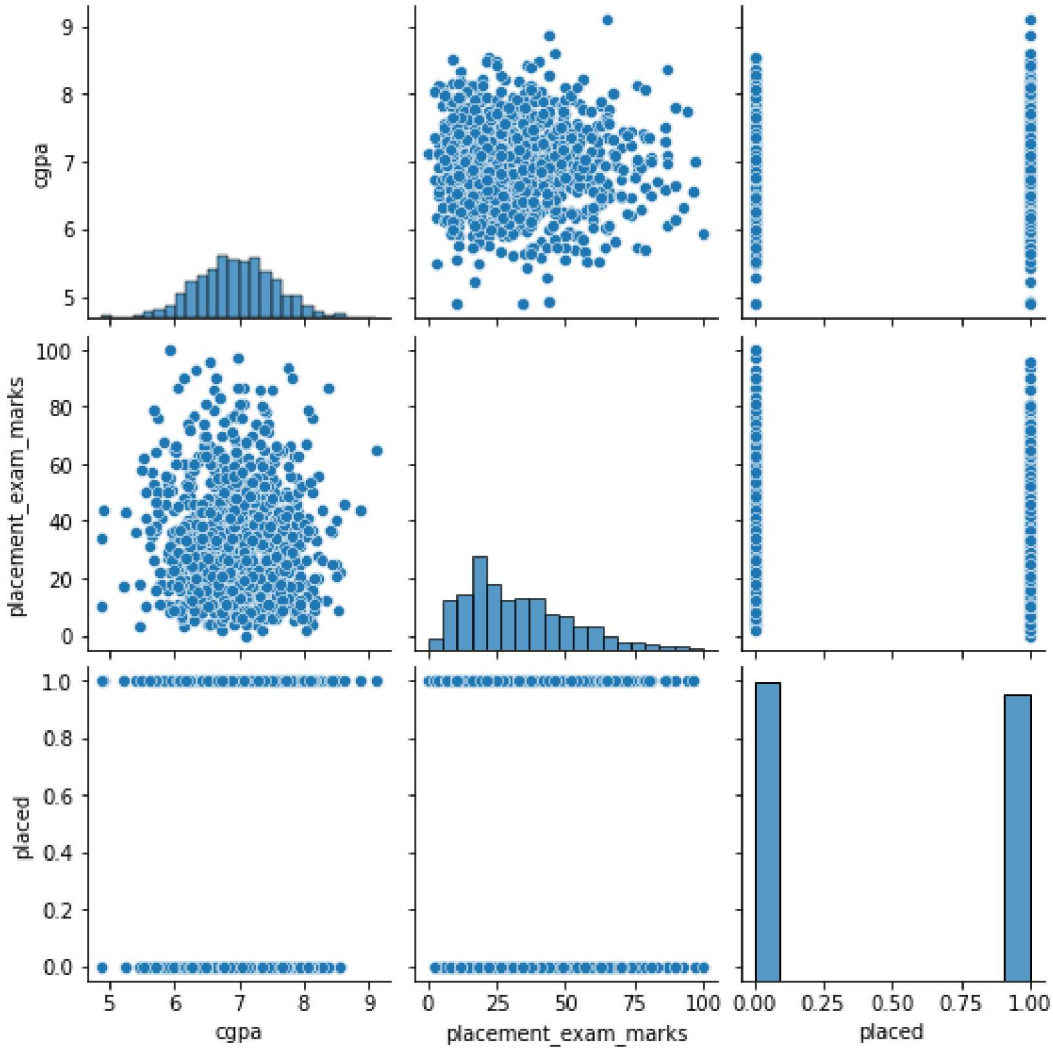
	cgpa	placement_exam_marks	placed
count	1000.000000	1000.000000	1000.000000
mean	6.961240	32.225000	0.489000
std	0.615898	19.130822	0.500129
min	4.890000	0.000000	0.000000
25%	6.550000	17.000000	0.000000
50%	6.960000	28.000000	0.000000
75%	7.370000	44.000000	1.000000
max	9.120000	100.000000	1.000000

In [7]:
```
data.columns
```

Out[7]:
```
Index(['cgpa', 'placement_exam_marks', 'placed'], dtype='object')
```

In [8]:
```
sns.pairplot(data)
```

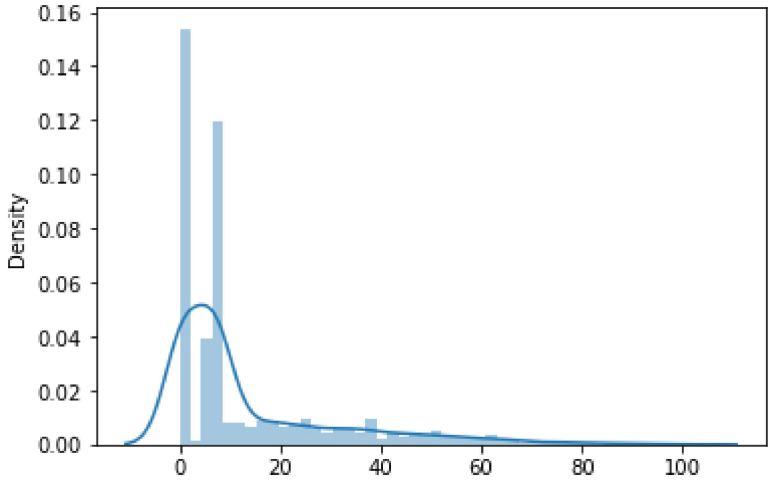
Out[8]: <seaborn.axisgrid.PairGrid at 0x18aaf6f34f0>



In [9]:
```
sns.distplot(data)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: Fut
ureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level fu
nction with similar flexibility) or `histplot` (an axes-level function for hi
stograms).
  warnings.warn(msg, FutureWarning)
```

Out[9]: <AxesSubplot:ylabel='Density'>



In [10]:
```
da=data[['cgpa', 'placement_exam_marks', 'placed']]
da
```

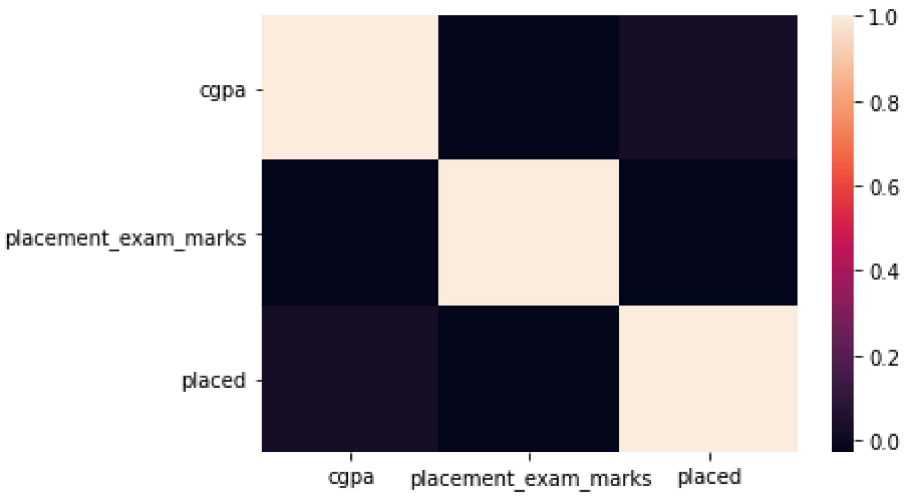
Out[10]:

	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 [11]:
```
sns.heatmap(da.corr())
```

Out[11]: <AxesSubplot:>



In [12]:
```
x=da[['cgpa', 'placement_exam_marks']]
y=da['placed']
```

In [13]:
```
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 [14]:
```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[14]: LinearRegression()

In [15]:
```
print(lr.intercept_)
```

```
0.4195198477214285
```

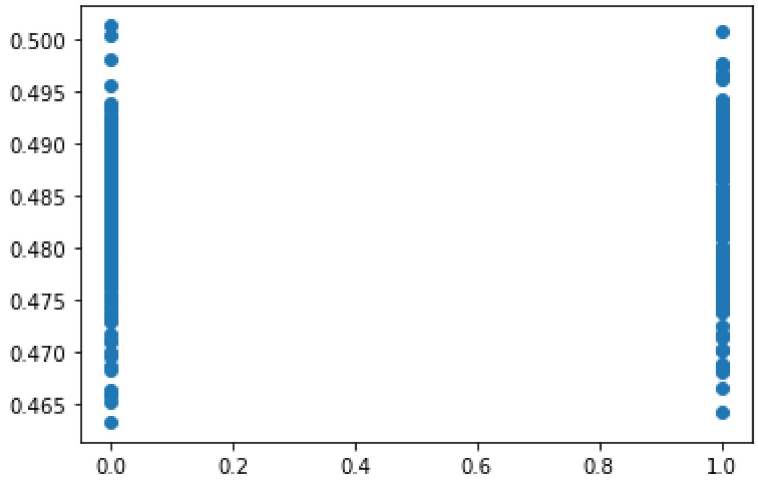
In [16]:
```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[16]:

	Co-efficient
cgpa	0.010166
placement_exam_marks	-0.000228

In [17]:
```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[17]: <matplotlib.collections.PathCollection at 0x18ab073c700>



In [18]:
```
print(lr.score(x_test,y_test))
```

```
0.0007086627321163386
```

In [19]:
```
print(lr.score(x_train,y_train))
```

```
0.0002516588653460605
```

In [20]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [21]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[21]: Ridge(alpha=10)

In [22]:
```
rr.score(x_test,y_test)
```

Out[22]: 0.0006675870524871508

In [24]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[24]: Lasso(alpha=10)

In [25]:
```
la.score(x_test,y_test)
```

Out[25]: -0.001677172046648856

In [ ]: