DATA COLLECTION

In [1]:

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```python
a=pd.read_csv(r"C:\Users\user\Downloads\13_placement.csv")
a
```

Out[2]:

	cgpa	placement_exam_marks	placed
0	7.19	26.0	1
1	7.46	38.0	1
2	7.54	40.0	1
3	6.42	8.0	1
4	7.23	17.0	0
...	...	...	...
995	8.87	44.0	1
996	9.12	65.0	1
997	4.89	34.0	0
998	8.62	46.0	1
999	4.90	10.0	1

1000 rows × 3 columns

In [3]:

```python
a.head(5)
```

Out[3]:

	cgpa	placement_exam_marks	placed
0	7.19	26.0	1
1	7.46	38.0	1
2	7.54	40.0	1
3	6.42	8.0	1
4	7.23	17.0	0

DATA CLEANING AND PRE-PROCESSING

In [4]:

```python
a.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   float64
 2   placed                1000 non-null   int64
dtypes: float64(2), int64(1)
memory usage: 23.6 KB
```

In [5]:

```python
a.describe()
```

Out[5]:

	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 [6]:

```python
a.columns
```

Out[6]:

```
Index(['cgpa', 'placement_exam_marks', 'placed'], dtype='object')
```
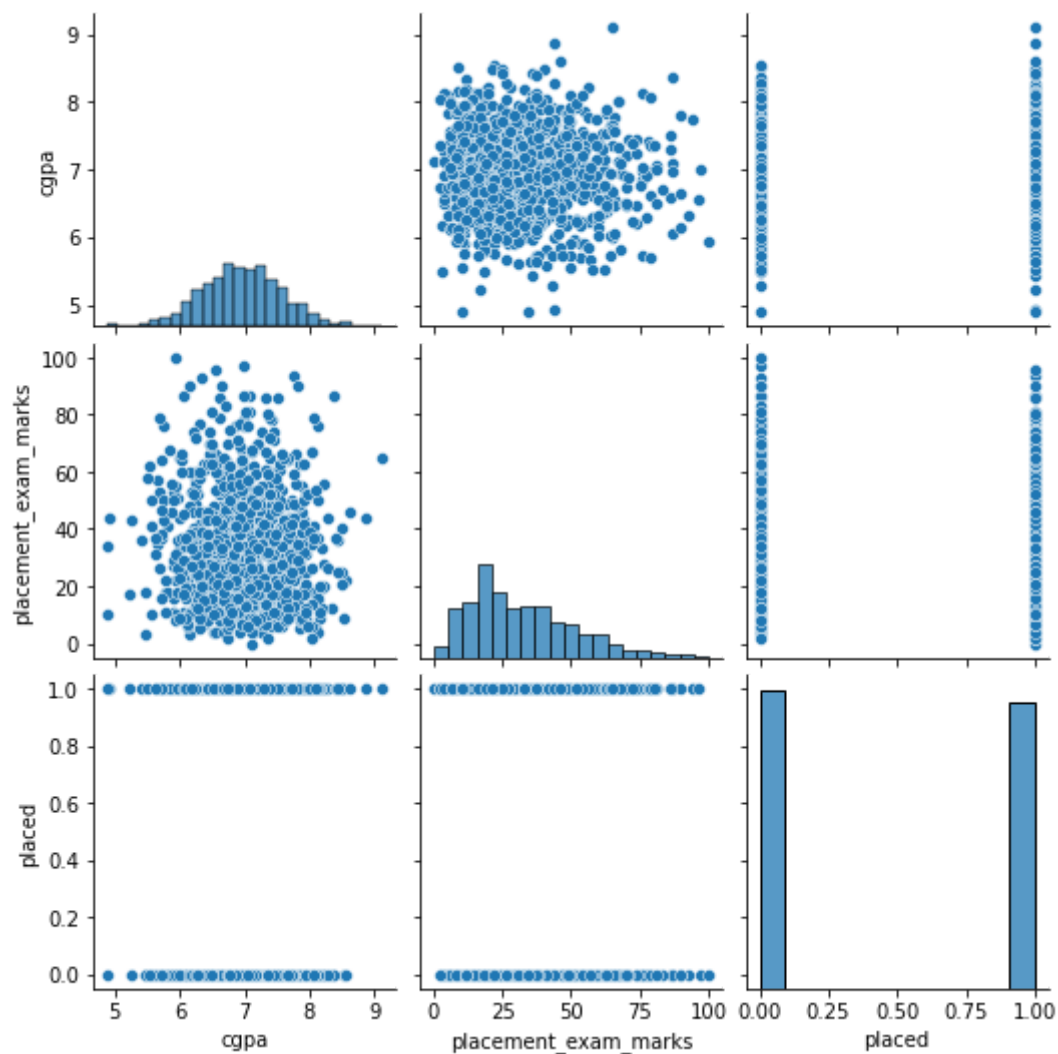
EDA and VISUALIZATION

In [8]:

```
sns.pairplot(a)
```

Out[8]:

```
<seaborn.axisgrid.PairGrid at 0x2a282a31f10>
```
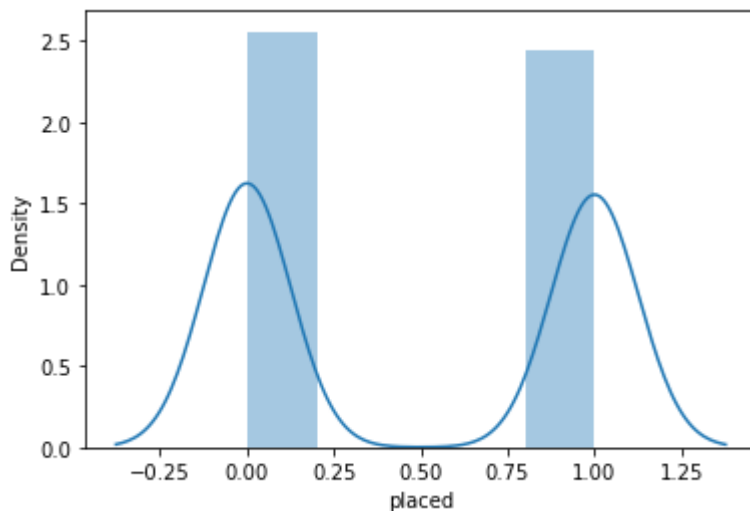
In [9]:

```python
sns.distplot(a['placed'])
```

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

Out[9]:

```
<AxesSubplot:xlabel='placed', ylabel='Density'>
```
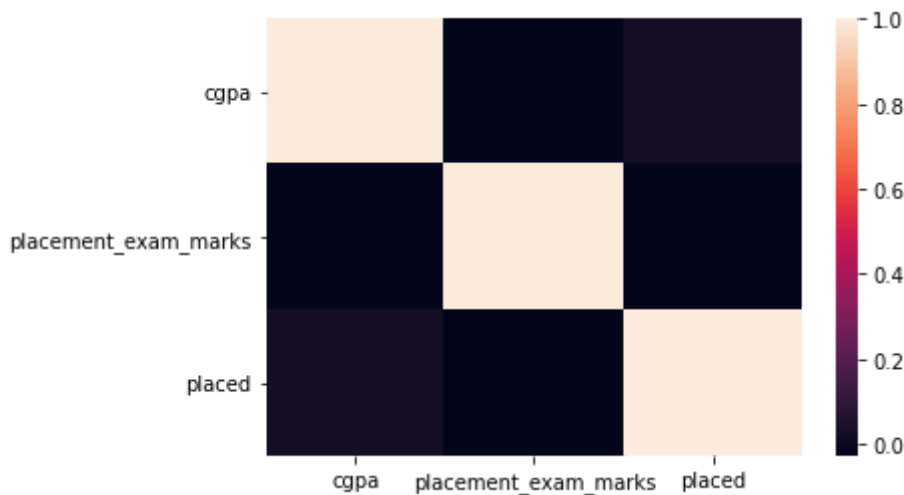

In [11]:

```python
f=a[['cgpa', 'placement_exam_marks', 'placed']]
```

In [12]:

```python
sns.heatmap(f.corr())
```

Out[12]:

```
<AxesSubplot:>
```

To train the model-model building

In [13]:

```python
x=f[['cgpa', 'placement_exam_marks']]
y=f['placed']
```

In [14]:

```python
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 [15]:

```python
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[15]:

```
LinearRegression()
```

In [16]:

```python
print(lr.intercept_)
```

```
0.3467134828157735
```

In [17]:

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

Out[17]:

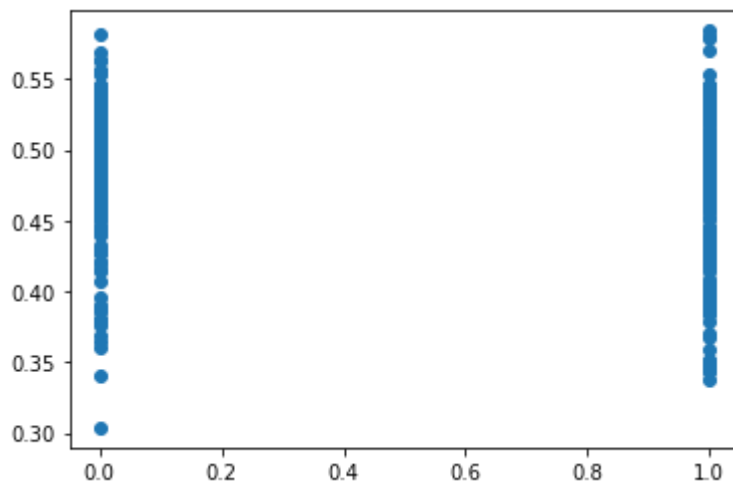|	Co-efficient
cgpa	0.030158
placement_exam_marks	-0.002514

In [18]:

```python
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[18]:

```
<matplotlib.collections.PathCollection at 0x2a2838b3820>
```


In [19]:

```python
print(lr.score(x_test,y_test))
```

```
-0.04522231741239935
```

In [20]:

```python
lr.score(x_train,y_train)
```

Out[20]:

```
0.010449796984944193
```

RIDGE REGRESSION

In [21]:

```python
from sklearn.linear_model import Ridge,Lasso
```

In [22]:

```python
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[22]:

```
Ridge(alpha=10)
```

In [23]:

```
rr.score(x_test,y_test)
```

Out[23]:

-0.04512963894431876

LASSO REGRESSION

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]:

-0.007857469911041193

In [ ]: