Day-10

Placement Statement

In [1]:

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

In [2]:

```
d=pd.read_csv(r"C:\Users\user\Downloads\placement.csv")
d
```

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]:

```
d.columns
```

Out[3]:

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

In [4]:

```
d.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]:

```
x=d[['cgpa', 'placement_exam_marks']]
y=d['placed']
```

In [6]:

```
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 [7]:

```
from sklearn.linear_model import LinearRegression

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

Out[7]:

```
LinearRegression()
```

In [8]:

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

```
0.2573738696988293
```

In [9]:

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

```
-0.009032572115204 03
```

In [10]:

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

```
0.004028706285135186
```

Ridge Regression

In [11]:

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

In [12]:

```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
rr.score(x_test,y_test)
```

Out[12]:

```
-0.008808585617939402
```

Lasso Regression

In [13]:

```
la=Lasso(alpha=10)
```

In [14]:

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

Out[14]:

```
Lasso(alpha=10)
```

In [15]:

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

Out[15]:

```
-0.001677172046648634
```

Elastic Regreesion

In [16]:

```
from sklearn.linear_model import ElasticNet
en=ElasticNet()
en.fit(x_train,y_train)
```

Out[16]:

```
ElasticNet()
```

In [17]:

```
predict=(en.predict(x_test))
print(predict)
```

```
[0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714
 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714 0.48285714]
```

In [18]:

```
print(en.score(x_test,y_test))
```

```
-0.001677172046648634
```

Evaluation Method

In [19]:

```
from sklearn import metrics
```

In [20]:

```
print("Mean Absolute Error:",metrics.mean_absolute_error(y_test,predict))
```

```
Mean Absolute Error: 0.5001142857142857
```

In [21]:

```
print("Root Mean Square Error:",np.sqrt(metrics.mean_squared_error(y_test,predict)))
```

```
Root Mean Square Error: 0.500407996803914
```

In [22]:

```
print("Mean Square Error:",metrics.mean_squared_error(y_test,predict))
```

```
Mean Square Error: 0.25040816326530607
```

In [ ]:

```

```