Day-9

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 [4]:

```
d.columns
```

Out[4]:

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

In [5]:

```
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 [6]:

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

In [7]:

```
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 [8]:

```
from sklearn.linear_model import LinearRegression

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

Out[8]:

```
LinearRegression()
```

In [9]:

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

```
0.49414191144655434
```

In [10]:

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

```
-0.011936529299728438
```

In [11]:

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

```
0.002748581875469336
```

Ridge Regression

In [12]:

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

In [13]:

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

Out[13]:

```
-0.011974434362202357
```

In [14]:

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

Out[14]:

```
-0.011974434362202357
```

Lasso Regression

In [16]:

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

In [17]:

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

Out[17]:

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

In [18]:

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

Out[18]:

```
-0.0069095270958625665
```