Simple Linear Regression

In [30]:

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
df=pd.read_csv("placements.csv")
import seaborn as sns
```

In [2]:

```
df.head()
```

Out[2]:

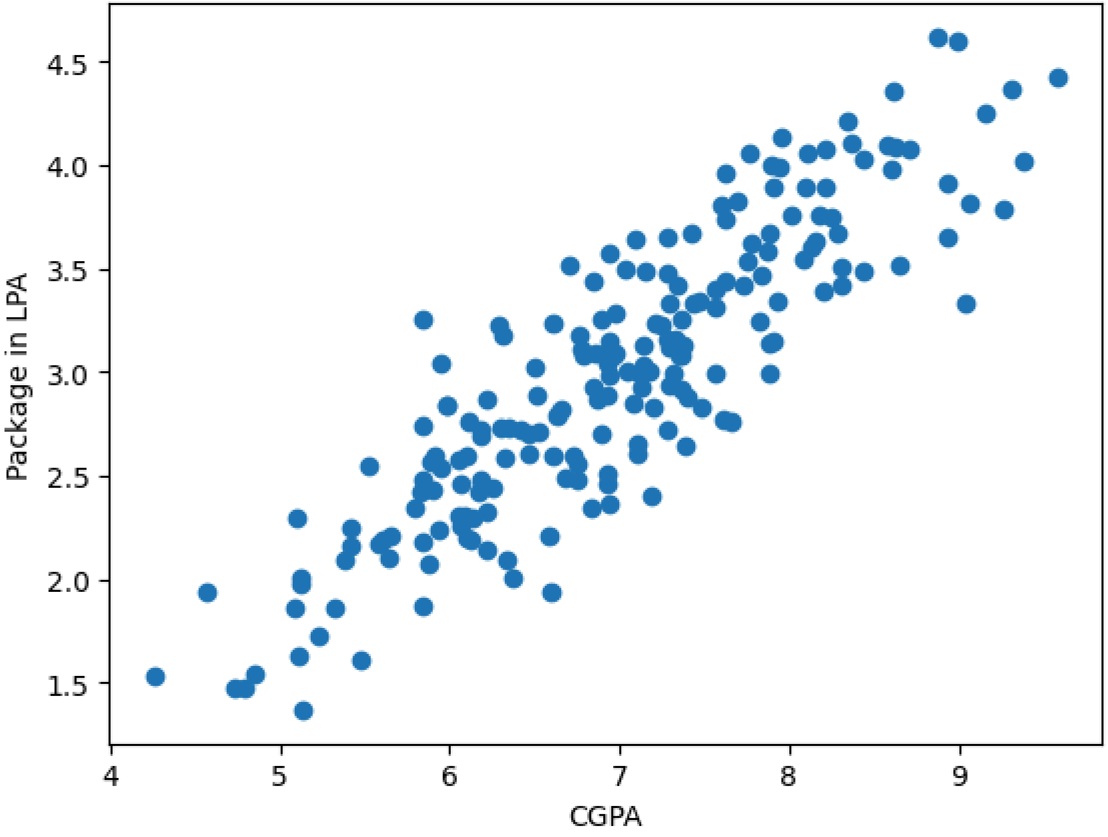
	cgpa	package
0	6.89	3.26
1	5.12	1.98
2	7.82	3.25
3	7.42	3.67
4	6.94	3.57

In [3]:

```
plt.scatter(df["cgpa"],df["package"])
plt.xlabel("CGPA")
plt.ylabel("Package in LPA")
```

Out[3]:

```
Text(0, 0.5, 'Package in LPA')
```



In [4]:

```
X=df.iloc[:,0:1]
y=df.iloc[:,-1]
```

In [5]:

```
X
```

Out[5]:

	cgpa
0	6.89
1	5.12
2	7.82
3	7.42
4	6.94
...	...
195	6.93
196	5.89
197	7.21
198	7.63
199	6.22

200 rows × 1 columns

In [6]:

```
y
```

Out[6]:

```
0      3.26
1      1.98
2      3.25
3      3.67
4      3.57
       ...
195    2.46
196    2.57
197    3.24
198    3.96
199    2.33
Name: package, Length: 200, dtype: float64
```

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=0.2,random_state=2)
```

In [8]:

```
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
```

In [9]:

```
lr.fit(X_train,y_train)
```

Out[9]:

```
LinearRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [10]:

```
X_test
```

```
Out[10]:       cgpa
         112   8.58
          29   7.15
         182   5.88
         199   6.22
         193   4.57
          85   4.79
          10   5.32
          54   6.86
         115   8.35
          35   6.87
          12   8.94
          92   7.90
          13   6.93
         126   5.91
         174   7.32
           2   7.82
          44   5.09
           3   7.42
         113   6.94
          14   7.73
          23   6.19
          25   7.28
           6   6.73
         134   7.20
         165   8.21
         173   6.75
```

	cgpa
45	7.87
65	7.60
48	8.63
122	5.12
178	8.15
64	7.36
9	8.31
57	6.60
78	6.59
71	7.47
128	7.93
176	6.29
131	6.37
53	6.47

```
In [11]: y_test
```

```
Out[11]: 112    4.10
         29     3.49
         182    2.08
         199    2.33
         193    1.94
         85     1.48
         10     1.86
         54     3.09
         115    4.21
         35     2.87
         12     3.65
         92     4.00
         13     2.89
         126    2.60
         174    2.99
         2      3.25
         44     1.86
         3      3.67
         113    2.37
         14     3.42
         23     2.48
         25     3.65
         6      2.60
         134    2.83
         165    4.08
         173    2.56
         45     3.58
         65     3.81
         48     4.09
         122    2.01
         178    3.63
         64     2.92
         9      3.51
         57     1.94
         78     2.21
         71     3.34
         128    3.34
         176    3.23
         131    2.01
         53     2.61
         Name: package, dtype: float64
```

In [12]:
```
lr.predict(X_test.iloc[0].values.reshape(1,1))
```

```
c:\Users\lucius seneca\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:493: UserWarning: X
does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

Out[12]:
```
array([3.89111601])
```

In [13]:
```
lr.predict(X_test.iloc[1].values.reshape(1,1))
```

```
c:\Users\lucius seneca\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:493: UserWarning: X
does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

Out[13]:
```
array([3.09324469])
```

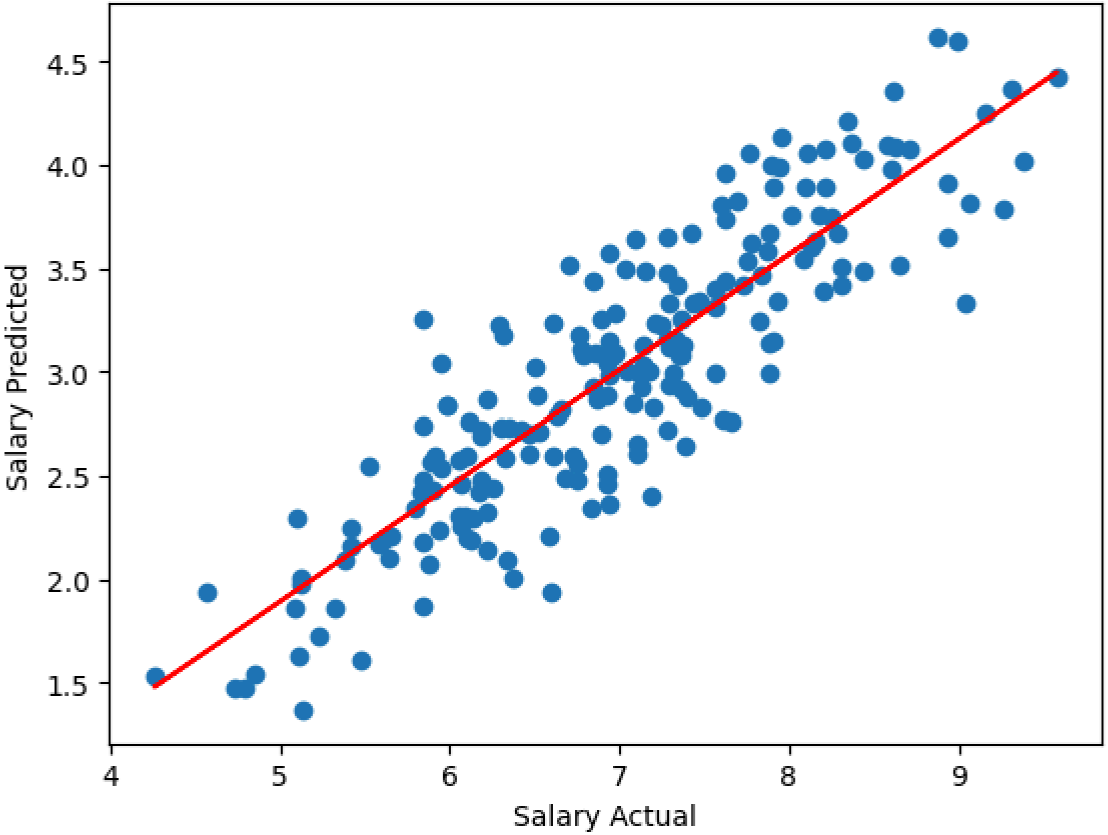
In [14]:
```
lr.predict(X_test.iloc[2].values.reshape(1,1))
```

```
c:\Users\lucius seneca\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:493: UserWarning: X
does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

Out[14]:
```
array([2.38464568])
```

```
In [15]: plt.scatter(df["cgpa"],df["package"])
         plt.plot(X_train,lr.predict(X_train),color="red")
         plt.xlabel("Salary Actual")
         plt.ylabel("Salary Predicted")
```

```
Out[15]: Text(0, 0.5, 'Salary Predicted')
```



```
In [16]: m=lr.coef_
```

```
In [17]: b=lr.coef_
In [18]: b
```

```
Out[18]: array([0.55795197])
```

```
In [19]: m
```

```
Out[19]: array([0.55795197])
```

```
In [20]: from sklearn.metrics import mean_absolute_error,mean_squared_error,r2_score
```

```
In [22]: y_pred=lr.predict(X_test)
```

```
In [23]: y_test.values
```

```
Out[23]: array([4.1 , 3.49, 2.08, 2.33, 1.94, 1.48, 1.86, 3.09, 4.21, 2.87, 3.65,
               4.  , 2.89, 2.6 , 2.99, 3.25, 1.86, 3.67, 2.37, 3.42, 2.48, 3.65,
               2.6 , 2.83, 4.08, 2.56, 3.58, 3.81, 4.09, 2.01, 3.63, 2.92, 3.51,
               1.94, 2.21, 3.34, 3.34, 3.23, 2.01, 2.61])
```

```
In [24]: print("MAE",mean_absolute_error(y_test,y_pred))
```

```
MAE 0.2884710931878175
```

```
In [26]: print("MSE",mean_squared_error(y_test,y_pred))
```

```
MSE 0.12129235313495527
```

```
In [35]: print("r2",r2_score(y_test,y_pred))
         r2=r2_score(y_test,y_pred)
```

```
r2 0.780730147510384
```

##Adjusted R2 score

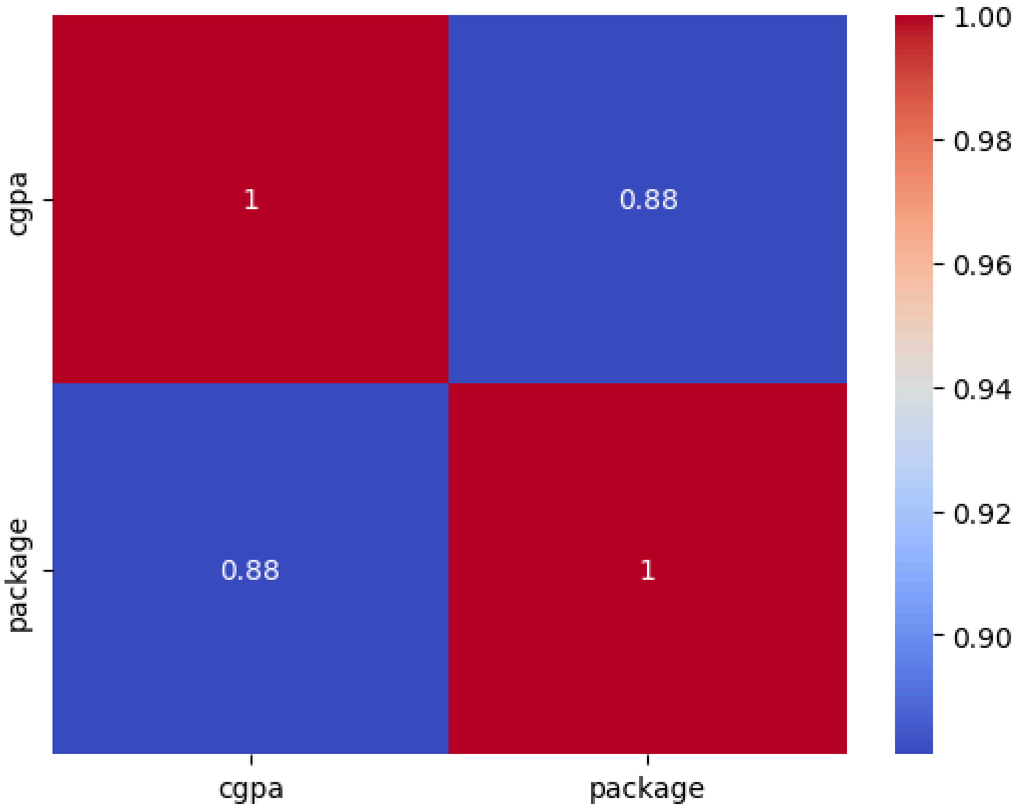
```
In [29]: df
```

```
Out[29]:
        cgpa  package
   0    6.89     3.26
   1    5.12     1.98
   2    7.82     3.25
   3    7.42     3.67
   4    6.94     3.57
 ...     ...      ...
 195    6.93     2.46
 196    5.89     2.57
 197    7.21     3.24
 198    7.63     3.96
 199    6.22     2.33

200 rows × 2 columns
```

```
In [32]: sns.heatmap(df.corr(),annot=True,cmap="coolwarm")
```

```
Out[32]: <Axes: >
```



```
In [33]: X_test.shape
```

```
Out[33]: (40, 1)
```

```
In [37]: 1-((1-r2)*(40-1)/(40-2))
```

```
Out[37]: 0.7749598882343415
```

In [38]:
```
df.shape
```

Out[38]:
```
(200, 2)
```

In [39]:
```
df1=df.copy()
df1["new"]=np.random.random(200)
```

In [40]:
```
df1
```

Out[40]:

	cgpa	package	new
0	6.89	3.26	0.372946
1	5.12	1.98	0.449844
2	7.82	3.25	0.950117
3	7.42	3.67	0.821743
4	6.94	3.57	0.624586
...	...	...	...
195	6.93	2.46	0.094940
196	5.89	2.57	0.613568
197	7.21	3.24	0.918439
198	7.63	3.96	0.800360
199	6.22	2.33	0.140589

200 rows × 3 columns

In [41]:
```
df1=df1[["cgpa","new","package"]]
```

In [42]:

```
df1
```

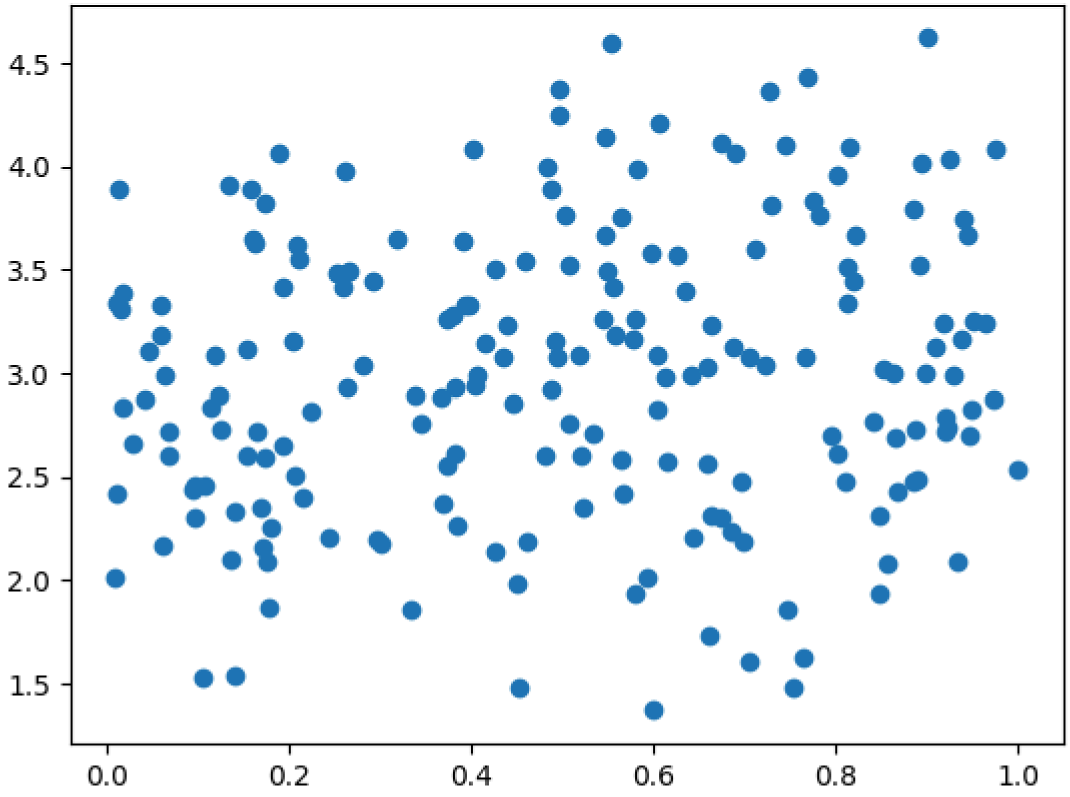
Out[42]:

	cgpa	new	package
0	6.89	0.372946	3.26
1	5.12	0.449844	1.98
2	7.82	0.950117	3.25
3	7.42	0.821743	3.67
4	6.94	0.624586	3.57
...	...	...	...
195	6.93	0.094940	2.46
196	5.89	0.613568	2.57
197	7.21	0.918439	3.24
198	7.63	0.800360	3.96
199	6.22	0.140589	2.33

200 rows × 3 columns

```
In [44]: plt.scatter(df1["new"],df1["package"])
```

```
Out[44]: <matplotlib.collections.PathCollection at 0x21f8b1afdf0>
```



```
In [46]: X-df1.iloc[:,0:2]
         y=df1.iloc[:,-1]
```

```
In [47]: X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=2)
```

```
In [48]: lr=LinearRegression()
```

In [49]:
```
lr.fit(X_train,y_train)
```

Out[49]:
```
LinearRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [50]:
```
y_pred=lr.predict(X_test)
```

In [52]:
```
r21=r2_score(y_test,y_pred)
```

In [55]:
```
r21
```

Out[55]:
```
0.780730147510384
```

In [56]:
```
1-((1-r21)*(40-1)/(40-2))
```

Out[56]:
```
0.7749598882343415
```

In [ ]:
```

```