Practical No 1 Regression Analysis and Plot Interpretation.

Simple Linear Regression

Importing the libraries

In [2]:

```
import pandas as pd
import matplotlib.pyplot as plt
```

Importing the dataset

In [3]:

```
dataset = pd.read_csv('Salary_Data.csv')
```

In [4]:

```
dataset
```

Out[4]:

	YearsExperience	Salary
0	1.1	39343
1	1.3	46205
2	1.5	37731
3	2.0	43525
4	2.2	39891
5	2.9	56642
6	3.0	60150
7	3.2	54445
8	3.2	64445
9	3.7	57189
10	3.9	63218
11	4.0	55794
12	4.0	56957
13	4.1	57081
14	4.5	61111
15	4.9	67938
16	5.1	66029
17	5.3	83088
18	5.9	81363
19	6.0	93940
20	6.8	91738
21	7.1	98273
22	7.9	101302
23	8.2	113812
24	8.7	109431
25	9.0	105582
26	9.5	116969
27	9.6	112635
28	10.3	122391
29	10.5	121872

In [10]:

```
x= dataset.iloc[:, :-1].values
#x=dataset[["YearsExperience"]]
```

In [11]:

```
x
```

Out[11]:

```
array([[ 1.1],
       [ 1.3],
       [ 1.5],
       [ 2. ],
       [ 2.2],
       [ 2.9],
       [ 3. ],
       [ 3.2],
       [ 3.2],
       [ 3.7],
       [ 3.9],
       [ 4. ],
       [ 4. ],
       [ 4.1],
       [ 4.5],
       [ 4.9],
       [ 5.1],
       [ 5.3],
       [ 5.9],
       [ 6. ],
       [ 6.8],
       [ 7.1],
       [ 7.9],
       [ 8.2],
       [ 8.7],
       [ 9. ],
       [ 9.5],
       [ 9.6],
       [10.3],
       [10.5]])
```

In [12]:

```
y = dataset.iloc[:, 1].values
#y=dataset[["salary"]]
```

In [13]:

```
y
```

Out[13]:

```
array([ 39343,  46205,  37731,  43525,  39891,  56642,  60150,  54445,
        64445,  57189,  63218,  55794,  56957,  57081,  61111,  67938,
        66029,  83088,  81363,  93940,  91738,  98273, 101302, 113812,
       109431, 105582, 116969, 112635, 122391, 121872], dtype=int64)
```

Splitting the dataset into the Training set and Test set

In [14]:

```
from sklearn.model_selection import train_test_split
```

In [19]:

```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 1/3, random_state = 0
```

In [20]:

```
X_train
```

Out[20]:

```
array([[ 2.9],
       [ 5.1],
       [ 3.2],
       [ 4.5],
       [ 8.2],
       [ 6.8],
       [ 1.3],
       [10.5],
       [ 3. ],
       [ 2.2],
       [ 5.9],
       [ 6. ],
       [ 3.7],
       [ 3.2],
       [ 9. ],
       [ 2. ],
       [ 1.1],
       [ 7.1],
       [ 4.9],
       [ 4. ]])
```

In [21]:

```
y_train
```

Out[21]:

```
array([ 56642,  66029,  64445,  61111, 113812,  91738,  46205, 121872,
        60150,  39891,  81363,  93940,  57189,  54445, 105582,  43525,
        39343,  98273,  67938,  56957], dtype=int64)
```

In [22]:

```
X_test
```

Out[22]:

```
array([[ 1.5],
       [10.3],
       [ 4.1],
       [ 3.9],
       [ 9.5],
       [ 8.7],
       [ 9.6],
       [ 4. ],
       [ 5.3],
       [ 7.9]])
```

In [23]:

```
y_test
```

Out[23]:

```
array([ 37731, 122391,  57081,  63218, 116969, 109431, 112635,  55794,
        83088, 101302], dtype=int64)
```

Fitting Simple Linear Regression to the Training set

In [24]:

```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

Out[24]:

```
LinearRegression()
```

Predicting the Test set results

In [25]:

```
regressor.predict(X_test)
```

Out[25]:

```
array([ 40835.10590871, 123079.39940819,  65134.55626083,  63265.36777221,
       115602.64545369, 108125.8914992 , 116537.23969801,  64199.96201652,
        76349.68719258, 100649.1375447 ])
```

In [26]:

```
salary_pred=regressor.predict([[6]])
```

In [27]:

```
salary_pred
```

Out[27]:

```
array([82891.84690277])
```

In [29]:

```
regressor.score(X_test,y_test)
```

Out[29]:

```
0.9749154407708353
```

In [30]:

```
regressor.coef_
```

Out[30]:

```
array([9345.94244312])
```

In [31]:

```
regressor.intercept_
```

Out[31]:

```
26816.19224403119
```

In [32]:

```
salarypd=9345.94244312*6+26816.19224403119
```

In [33]:

```
salarypd
```

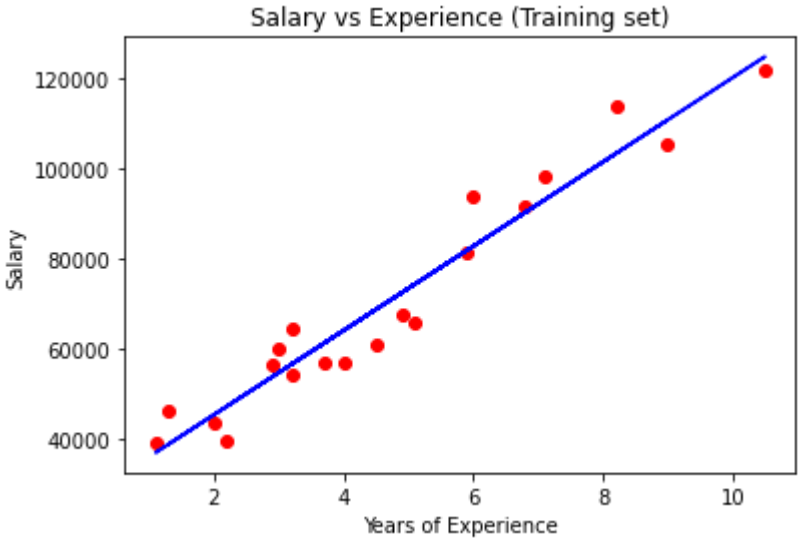
Out[33]:

```
82891.84690275119
```

Visualising the Training set results

In [36]:

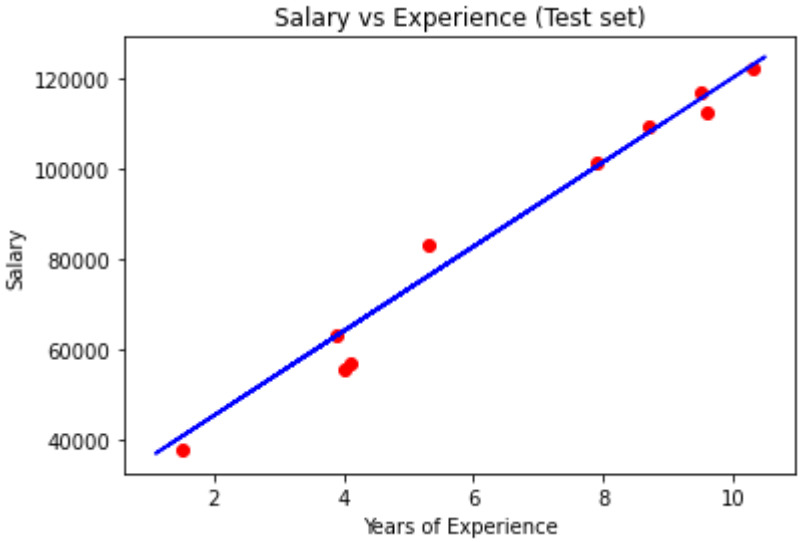
```
plt.scatter(X_train, y_train, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Salary vs Experience (Training set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```



Visualising the Test set results

In [37]:

```
plt.scatter(X_test, y_test, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Salary vs Experience (Test set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```



In [ ]:

```

```