Scikit-learn is a library for Python that provides machine learning developers with many unsupervised and supervised learning algorithms.

In [1]:

```
# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

```
Bad key "text.kerning_factor" on line 4 in
C:\Users\91920\anaconda3\lib\site-packages\matplotlib\mpl-data\stylelib\_cla
ssic_test_patch.mplstyle.
You probably need to get an updated matplotlibrc file from
https://github.com/matplotlib/matplotlib/blob/v3.1.3/matplotlibrc.template
 (https://github.com/matplotlib/matplotlib/blob/v3.1.3/matplotlibrc.templat
e)
or from the matplotlib source distribution
```

In [3]:

```
dataset = pd.read_csv('E:/Machine Learning/Linear Regression/Salary_Data.csv')
```

In [5]:

```
dataset
```

Out[5]:

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0
5	2.9	56642.0
6	3.0	60150.0
7	3.2	54445.0
8	3.2	64445.0
9	3.7	57189.0
10	3.9	63218.0
11	4.0	55794.0
12	4.0	56957.0
13	4.1	57081.0
14	4.5	61111.0
15	4.9	67938.0
16	5.1	66029.0
17	5.3	83088.0
18	5.9	81363.0
19	6.0	93940.0
20	6.8	91738.0
21	7.1	98273.0
22	7.9	101302.0
23	8.2	113812.0
24	8.7	109431.0
25	9.0	105582.0
26	9.5	116969.0
27	9.6	112635.0
28	10.3	122391.0
29	10.5	121872.0

In [6]:

```
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 1].values
```

In [6]:

```
dataset.iloc[:, 1]
```

Out[6]:

```
0      39343.0
1      46205.0
2      37731.0
3      43525.0
4      39891.0
5      56642.0
6      60150.0
7      54445.0
8      64445.0
9      57189.0
10     63218.0
11     55794.0
12     56957.0
13     57081.0
14     61111.0
15     67938.0
16     66029.0
17     83088.0
18     81363.0
19     93940.0
20     91738.0
21     98273.0
22    101302.0
23    113812.0
24    109431.0
25    105582.0
26    116969.0
27    112635.0
28    122391.0
29    121872.0
Name: Salary, dtype: float64
```

In [7]:

```
dataset.iloc[:, :-1]
```

Out[7]:

	YearsExperience
0	1.1
1	1.3
2	1.5
3	2.0
4	2.2
5	2.9
6	3.0
7	3.2
8	3.2
9	3.7
10	3.9
11	4.0
12	4.0
13	4.1
14	4.5
15	4.9
16	5.1
17	5.3
18	5.9
19	6.0
20	6.8
21	7.1
22	7.9
23	8.2
24	8.7
25	9.0
26	9.5
27	9.6
28	10.3
29	10.5

In [7]:

```
dataset.head()
```

Out[7]:

	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0

In [8]:

```
dataset.shape
```

Out[8]:

```
(30, 2)
```

In [9]:

```
X.shape
```

Out[9]:

```
(30, 1)
```

In [10]:

```
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, train_size = 0.80)
```

In [11]:

```
print(X_train.shape)
print(X_test.shape)
```

```
(24, 1)
(6, 1)
```

In [13]:

```
print(y_train.shape)
print(y_test.shape)
```

```
(24,)
(6,)
```

In [12]:

```
# Fitting Simple Linear Regression to the Training set
from sklearn.linear_model import LinearRegression
cdac = LinearRegression()
cdac.fit(X_train, y_train)
```

Out[12]:

```
LinearRegression()
```

In [13]:

```
print(cdac.intercept_)
print(cdac.coef_)
```

```
26563.554306667575
[9456.23662661]
```

In [15]:

```
# Predicting the Test set results
y_pred = cdac.predict(X_test)
```

In [16]:

```
y_pred
```

Out[16]:

```
array([ 63442.87715046, 123962.79156079,  65334.12447578,  45476.02755989,
       117343.42592216,  64388.50081312])
```

In [17]:

```
y_test
```

Out[17]:

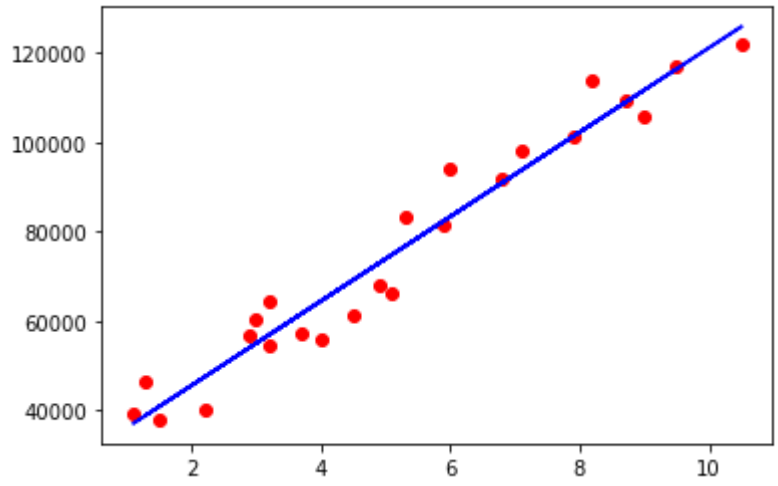
```
array([ 63218., 122391.,  57081.,  43525., 112635.,  56957.])
```

In [18]:

```
# Visualising the Training set results
plt.scatter(X_train, y_train, color = 'red')
plt.plot(X_train, cdac.predict(X_train), color = 'blue')
```

Out[18]:

```
[<matplotlib.lines.Line2D at 0x1987edcc048>]
```



In [19]:

```
# Visualising the Test set results
plt.scatter(X_test, y_test, color = 'red')
plt.plot(X_test, cdac.predict(X_test), color = 'blue')
plt.title('Salary vs Experience (Test set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```

In [20]:

```
import sklearn.metrics as metrics
```

In [21]:

```
mse = metrics.mean_squared_error(y_test,y_pred)
print("Mean Squared Error {}".format(mse))
```

```
Mean Squared Error 25306358.26590493
```

In [24]:

```
from sklearn.metrics import r2_score
r2_score(y_test, y_pred)
```

Out[24]:

```
0.9628229021977678
```

There are four assumptions associated with a linear regression model:

Linearity: The relationship between X and the mean of Y is linear.

Homoscedasticity: The variance of residual is the same for any value of X.

Independence: Observations are independent of each other.

Normality: For any fixed value of X, Y is normally distributed.

In [ ]:

```

```