Linear Regression:

It is a Machine learning Algorythm based on supervised ~~alg~~ Learning. It is a Statistical method that is used for predictive Analysis. L.R. makes predictions for continious or real numberic variables.

L.R. show a linear Relationship betn Independent and dependent Variables.

For eg. height, weight, Salary, aye. etc.

~~Br~~

eg.

Exp (x)	Salary (y)	$(y_i - \bar{y})$	$(x_i - \bar{x})$	$(x_i - \bar{x})^2$	$(y_i - \bar{y})(x_i - \bar{x})$
1	5	-4.6	-2	4	9.2
2	7	-2.6	-1	1	2.6
3	9	-0.6	0	0	0
4	12	2.4	1	1	2.4
5	15	5.4	2	4	10.8
$\bar{x} = 3$	$\bar{y} = 9.6$			10	25

$$y = mx_i + c$$

$$m = \frac{\Sigma (x_i - \bar{x})(y_i - \bar{y})}{\Sigma (x_i - \bar{x})^2}$$

Date
Page

$m = \frac{25}{10}$

$\boxed{m = 2.5}$

$\bar{y} = m\bar{x} + c$

$c = \bar{y} - m\bar{x}$

$= 9.6 - (2.5 \times 3)$

$\boxed{c = 2.1}$

$y_{p_1} = mx_1 + c = (2.5 \times 1) + 2.1 = \underline{4.6}$

$y_{p_2} = mx_2 + c = (2.5 \times 2) + 2.1 = \underline{7.1}$

$y_{p_3} = mx_3 + c = (2.5 \times 3) + 2.1 = \underline{9.6}$

$y_{p_4} = mx_4 + c = (2.5 \times 4) + 2.1 = \underline{12.1}$

$y_{p_5} = mx_5 + c = (2.5 \times 5 + 2.1) = \underline{14.6}$

exp	salary (y)	y_{pred}	$(y_i - y_p)$	$(y_i - y_p)^2$
1	5	4.6	0.4	0.16
2	7	7.1	-0.1	0.01
3	9	9.6	-0.6	0.36
4	12	12.1	-0.1	0.01
5	15	14.6	0.4	0.16
				0.7

square error:

$$error = \Sigma(y - y_p)^2$$

$$=$$

$$\boxed{error = 0.7}$$

mean square error:

$$mse = \sqrt{\frac{(y - y_p)^2}{n}}$$

$$mse = \frac{\Sigma(y - y_p)^2}{n}$$

$$= \frac{0.7}{5}$$

$$\boxed{mse = 0.14}$$

root mean square error:

$$RMSE = \sqrt{\frac{\Sigma(y - y_p)^2}{n}}$$

$$= \sqrt{\frac{0.7}{5}}$$

$$\boxed{RMSE = 0.374}$$

#

$$R^2 = 1 - \frac{se}{sst}$$

$$\because \; sst = (y_i - \bar{y})^2$$

$$= 63.19$$

$$R^2 = 1 - \frac{0.7}{63.19}$$

$$\boxed{R^2 = 0.989}$$

In [1]:
```python
import pandas as pd
```

In [2]:
```python
salary=pd.read_excel('salary.xlsx')
```

In [3]:
```python
salary
```

Out[3]:

	exp	salary
0	1	5
1	2	7
2	3	9
3	4	12
4	5	15

In [39]:
```python
x=salary.iloc[:,:-1]
```

In [40]:
```python
y=salary.iloc[:,-1]
```

In [10]:
```python
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
ypred=lr.fit(x,y)
```

In [11]:
```python
ypred
```

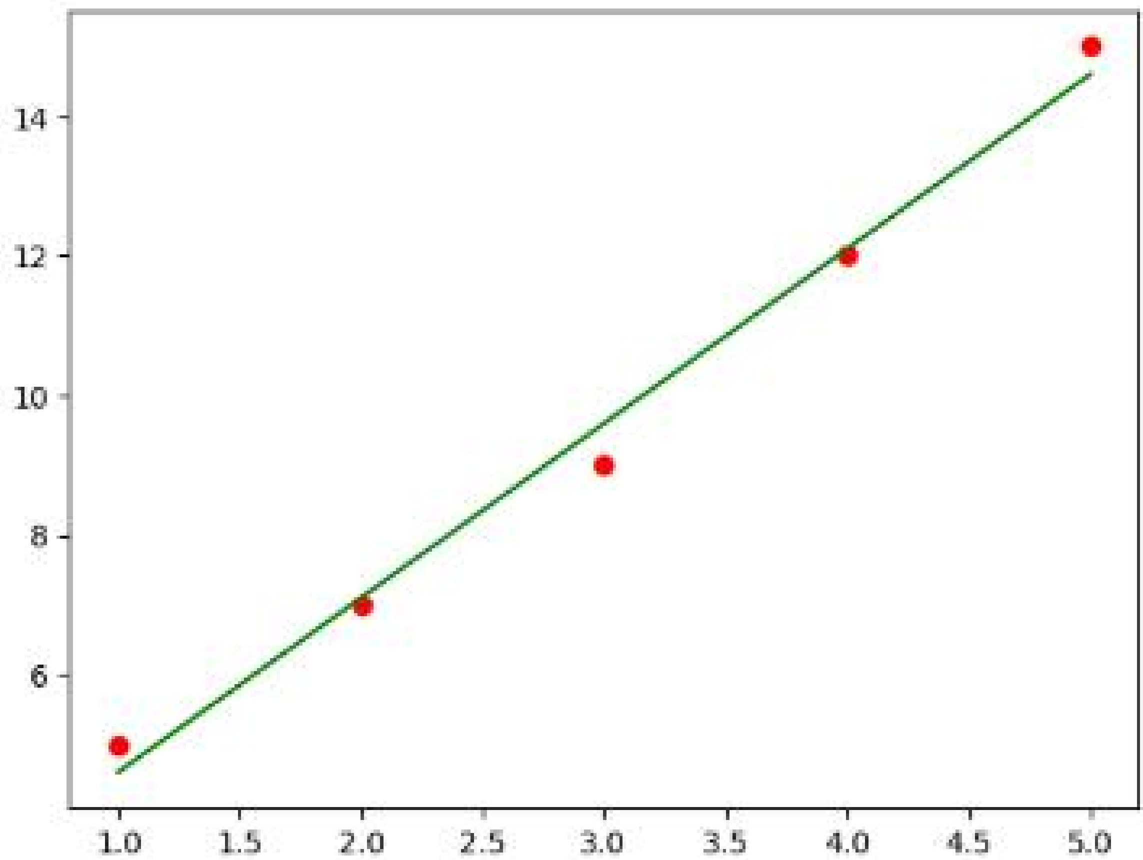
Out[11]:
```
LinearRegression()
```

In [12]:
```
lr.coef_
```
Out[12]: array([2.5])

In [13]:
```
lr.intercept_
```
Out[13]: 2.1000000000000014

In [16]:
```
pred=lr.predict(x)
pred
```
Out[16]: array([4.6, 7.1, 9.6, 12.1, 14.6])

In [27]:
```
plt.scatter(salary['exp'],salary['salary'],c='r')
plt.plot(x,pred,c='g')
plt.legend()
```
Out[27]: [<matplotlib.lines.Line2D at 0x27e92dc1670>]



```
In [28]: import numpy as np

In [33]: error=y-pred
         se=np.sum(error**2)
         se

Out[33]: 0.6999999999999987

In [34]: n=np.size(x)
         MSE=se/n
         MSE

Out[34]: 0.13999999999999974

In [36]: RMSE=np.sqrt(MSE)
         RMSE

Out[36]: 0.3741657386773938

In [37]: ymean=np.mean(y)
         sst=np.sum((y-ymean)**2)
         sst

Out[37]: 63.199999999999996

In [38]: R2=1-(se/sst)
         R2

Out[38]: 0.9889240506329114
```