Simple linear Regresion - HR Analytics

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

Approach for building a regression problem in python

- 1) Read and access the data
- 2) Identify the independent and dependent variables
- 3) Splitting the data into train and test
- 4) Building the model
- 5) Identify the equation using the Slope and intercept and R-Square
- 6) Predict the test data , using your model

In [2]:
```
ctc = pd.read_csv(r'C:\Users\Admin\Desktop\CTCdata (2).csv')
```

In [3]:
```
ctc
```

Out[3]:

	CTCoffered	LastCTC	Interview rating	Skill Set Index	Highest qualification	Total years of work exp
0	19	18	4	3	3	8.5
1	17	16	4	3	3	7.7
2	17	16	4	3	3	7.9
3	9	8	3	1	2	2.7
4	10	9	5	4	4	9.7
...	...	...	...	...	...	...
186	7	5	4	2	2	5.5
187	21	19	3	2	2	5.3
188	14	14	5	4	4	10.3
189	10	8	5	4	4	9.5
190	15	15	4	3	3	7.7

191 rows × 6 columns

In [4]:
```
ctc_new = ctc[['CTCoffered','LastCTC']]
ctc_new.head()
```

Out[4]:

	CTCoffered	LastCTC
0	19	18
1	17	16
2	17	16
3	9	8
4	10	9

In [5]:
```
x = ctc_new[['LastCTC']]
y = ctc_new[['CTCoffered']]
```

In [6]:
```
x.head()
```

Out[6]:

	LastCTC
0	18
1	16
2	16
3	8
4	9

In [7]:
```
y.head()
```

Out[7]:

	CTCoffered
0	19
1	17
2	17
3	9
4	10

Splitting the data into train and test

In [8]:
```
from sklearn.model_selection import train_test_split
```

In [9]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,train_size = 0.8,random_state = 23)
```

In [10]:
```
len(x_train),len(x_test),len(y_train),len(y_test)
```

Out[10]:
```
(152, 39, 152, 39)
```

In [11]:
```
x_train.head()
```

Out[11]:

	LastCTC
86	7
97	17
55	11
17	13
63	13

Building a linear regression

In [12]:
```
from sklearn.linear_model import LinearRegression
```

In [13]:
```
lr = LinearRegression()
model = lr.fit(x_train,y_train)
```

CTCoffered = (m*lastctc)+C

In [14]:
```
 # To find the slope, we would use the coef_function

model.coef_
```

Out[14]:
```
array([[0.95671194]])
```

In [15]:
```
# To find the constant, we would use the intercept_ function

model.intercept_
```

Out[15]:
```
array([1.5547131])
```

In [16]:
```
# TO find R-squared value

model.score(x_train,y_train)
```

Out[16]:
```
0.9706408914740792
```

In [17]:
```
# To predict the test data , using model
```

CTC = (0.94*lastCTC) + 1.8

In [18]:
```
CTC = (0.94*10)+1.8
```

In [19]:
```
CTC
```

Out[19]:
```
11.2
```

Predicting on test data

In [20]:
```
y_test.head()
```

Out[20]:

	CTCoffered
14	8
176	8
189	10
170	14
102	16

In [21]:
```
y_test['Predicted CTC'] = model.predict(x_test)
y_test.head()
```

```
C:\Users\Admin\AppData\Local\Temp/ipykernel_17808/4254980658.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  y_test['Predicted CTC'] = model.predict(x_test)
```

Out[21]:

	CTCoffered	Predicted CTC
14	8	6.338273
176	8	7.294985
189	10	9.208409
170	14	15.905392
102	16	15.905392