Simple Linear Regression

In [ ]:
```
#Name: Leena Rajeshwar Kale
#Roll No.:71
#Sec: C
#Subject:ET - 1
```

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt  #charts(data science) op analysis
import seaborn as sns #ML op analysis
import numpy as np
```

In [2]:
```
import os
```

In [3]:
```
os.getcwd()
```

Out[3]:
```
'C:\\Users\\dishi\\Downloads\\ET'
```

In [4]:
```
os.chdir("C:\\Users\\dishi\\Downloads\\ET")
```

In [5]:
```
dt=pd.read_csv("Salary.csv")
```

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

Out[6]:

	YearsExperience	Salary
0	1.1	39343
1	1.3	46205
2	1.5	37731
3	2.0	43525
4	2.2	39891

In [7]:
```
dt.tail()
```

Out[7]:

	YearsExperience	Salary
30	11.2	127345
31	11.5	126756
32	12.3	128765
33	12.9	135675
34	13.5	139465

In [8]:
```
dt.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35 entries, 0 to 34
Data columns (total 2 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   YearsExperience  35 non-null     float64
 1   Salary           35 non-null     int64  
dtypes: float64(1), int64(1)
memory usage: 692.0 bytes
```

In [9]:
```
dt.describe()
```

Out[9]:

	YearsExperience	Salary
count	35.000000	35.000000
mean	6.308571	83945.600000
std	3.618610	32162.673003
min	1.100000	37731.000000
25%	3.450000	57019.000000
50%	5.300000	81363.000000
75%	9.250000	113223.500000
max	13.500000	139465.000000

In [10]:
```
dt.tail(20)
```

Out[10]:

	YearsExperience	Salary
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
30	11.2	127345
31	11.5	126756
32	12.3	128765
33	12.9	135675
34	13.5	139465

In [11]:
```
dt.shape # 2 attributes & 35 records
```

Out[11]:
```
(35, 2)
```

In [12]:
```
dt.size
```

Out[12]:
```
70
```

In [13]:
```
dt.ndim #dimensions
```

Out[13]:
```
2
```

In [14]:
```
dt.isnull().any()
```

Out[14]:
```
YearsExperience    False
Salary             False
dtype: bool
```

In [15]:
```
dt.isnull().sum()
```

Out[15]:
```
YearsExperience    0
Salary             0
dtype: int64
```

In [16]:
```
A = "Ashish"
```

In [17]:
```
print(A)
```

```
Ashish
```

In [18]:
```
A[-1]
```

Out[18]:
```
'h'
```

In [19]:
```
A[1:4]
```

Out[19]:
```
'shi'
```

In [20]:
```
#Assiging values in X & Y
X = dt.iloc[:, :-1].values
Y = dt.iloc[:, -1].values



#X = dt['YearsExperience']
#y = dt['Salary']
```

In [21]:
```
print(X)
```

```
[[ 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]
 [11.2]
 [11.5]
 [12.3]
 [12.9]
 [13.5]]
```

In [22]:
```
print(Y)
```

```
[ 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
 127345 126756 128765 135675 139465]
```

In [23]:
```
#Splitting testdata into X_train,X_test,y_train,y_test
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=.3,random_state=42)
```

In [24]:
```
print(X_train)
```

```
[[12.9]
 [ 1.1]
 [ 2.2]
 [ 5.3]
 [ 9.6]
 [ 2.9]
 [ 4. ]
 [ 1.3]
 [ 1.5]
 [12.3]
 [ 2. ]
 [11.2]
 [ 8.2]
 [11.5]
 [ 3.9]
 [ 7.9]
 [ 5.9]
 [ 9. ]
 [ 3. ]
 [ 6.8]
 [13.5]
 [ 3.2]
 [ 4.5]
 [10.3]]
```

In [25]:
```
print(X_test)
```

```
[[ 9.5]
 [ 4.1]
 [ 8.7]
 [ 7.1]
 [ 4.9]
 [10.5]
 [ 6. ]
 [ 4. ]
 [ 3.2]
 [ 5.1]
 [ 3.7]]
```

In [26]:
```
print(Y_train)
```

```
[135675  39343  39891  83088 112635  56642  55794  46205  37731 128765
  43525 127345 113812 126756  63218 101302  81363 105582  60150  91738
 139465  54445  61111 122391]
```

In [27]:
```
print(Y_test)
```

```
[116969  57081 109431  98273  67938 121872  93940  56957  64445  66029
  57189]
```

In [28]:
```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(X_train, Y_train)
```

Out[28]:

```
LinearRegression
LinearRegression()
```

In [29]:

```
#Assigning Coefficient (slope) to m
m = lr.coef_
```

In [30]:

```
print("Coefficient  :" , m)
```

```
Coefficient  : [8555.33918938]
```

In [31]:

```
#Assigning Y-intercept to a
c = lr.intercept_
```

In [32]:

```
print("Intercept : ", c)
```

```
Intercept :  29602.07353482097
```

In [33]:

```
lr.score(X_test,Y_test) * 100 #accurecy of model
```

Out[33]:

```
91.71426108885095
```