Group members:

Mohd. Umair Sayed-6118028

Mirajkar Shamim-6118027

Khan Needa-6118019

Experiment No : 9

AIM: To implement Simple Linear Regression using Python

Theory:

Simple linear regression is used to estimate the relationship between **two quantitative variables**. You can use simple linear regression when you want to know:

1. How strong the relationship is between two variables (e.g. the relationship between rainfall and soil erosion).
2. The value of the dependent variable at a certain value of the independent variable (e.g. the amount of soil erosion at a certain level of rainfall).

Simple linear regression is a **parametric test**, meaning that it makes certain assumptions about the data. These assumptions are:

1. **Homogeneity of variance (homoscedasticity)**: the size of the error in our prediction doesn't change significantly across the values of the independent variable.
2. **Independence of observations**: the observations in the dataset were collected using statistically valid sampling methods, and there are no hidden relationships among observations.
3. **Normality**: The data follows a normal distribution.

Linear regression makes one additional assumption:

4. The relationship between the independent and dependent variable is **linear**: the line of best fit through the data points is a straight line (rather than a curve or some sort of grouping factor).

If your data do not meet the assumptions of homoscedasticity or normality, you may be able to use a nonparametric test instead, such as the Spearman rank test.

IMPLEMENTATION OF SIMPLE LINEAR REGRESSION

Step 1: Import required Libraries

```
#Mohammad Umair
import pandas as pd
import numpy as np
import math
import matplotlib.pyplot as plt
import plotly.express as px
```

Step 2: Import dataset

```
#Shamim Mirajkar
from google.colab import files
uploaded = files.upload()
```

Choose Files Salary_Data.csv

- **Salary_Data.csv**(application/vnd.ms-excel) - 352 bytes, last modified: 4/24/2021 - 100% done

```
Saving Salary_Data.csv to Salary_Data (2).csv
```

```
#Khan Needa
import io
data=pd.read_csv(io.BytesIO(uploaded['Salary_Data.csv']))
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30 entries, 0 to 29
Data columns (total 2 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   YearsExperience  30 non-null     float64
 1   Salary           30 non-null     int64  
dtypes: float64(1), int64(1)
memory usage: 608.0 bytes
```

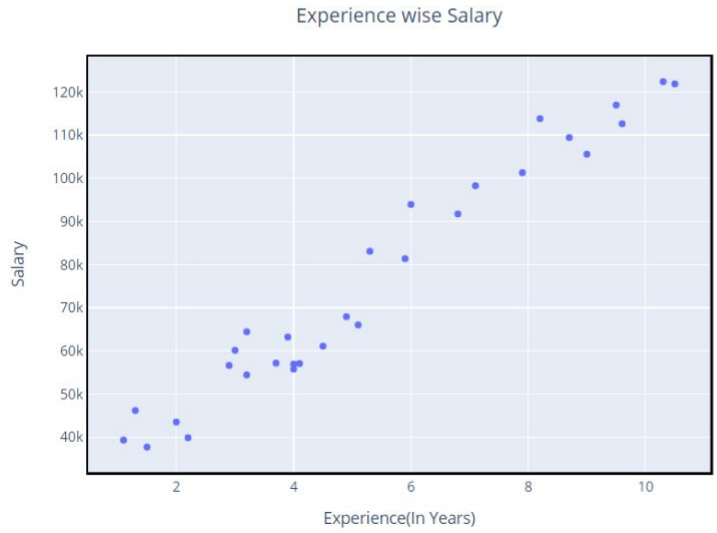
Step 3: Display dataset

```
print (data.columns)
data.head(5)
```

```
Index(['YearsExperience', 'Salary'], dtype='object')
```

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

```
fig = px.scatter(x=data['YearsExperience'],y=data['Salary'])
fig.update_layout(title='Experience wise Salary',title_x=0.5,xaxis_title='Experience(In Years)',yaxis_title='Salary',height=500,width=700)
fig.update_xaxes(showline=True,linewidth=2,linecolor='black',mirror=True)
fig.update_yaxes(showline=True,linewidth=2,linecolor='black',mirror=True)
fig.show()
```



Step 4: Calculate Mean & Variance

```
data['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
Name: YearsExperience, dtype: float64
```

```
mean_x=np.mean(data['YearsExperience'])
mean_y=np.mean(data['Salary'])
var_x=np.var(data['YearsExperience'])
var_y=np.var(data['Salary'])
print('Experience  stats: mean=%.3f variance=%.3f' %(mean_x,var_x))
print('y Salary stats: mean=%.3f variance=%.3f' %(mean_x,var_x))
```

```
x stats: mean=5.313 variance=7.785
y stats: mean=5.313 variance=7.785
```

Step 5: Calculate Covariance

```
def covariance(YearsExperience,Salary):
  mean_x=np.mean(YearsExperience)
  mean_y =np.mean(Salary)
  covar=0.0
  for i in range(len(YearsExperience)):
    covar += (YearsExperience[i]-mean_x)*(Salary[i]-mean_y)
    return covar/len(YearsExperience)

covar_xy=covariance(data['YearsExperience'],data['Salary'])
print(f'Cov(YearsExperience,Salary):{covar_xy}')
```

```
Cov(YearsExperience,Salary):5148.693333333333
```

Step 6: Define Coefficients

```
b1=covar_xy/var_x
b0=mean_y-b1*mean_x
print(f'coefficients:\n b0:{b0} b1:{b1}')
```

```
coefficients:
 b0:72489.04016761243 b1:661.3475217793408
```

Step 7: Predicting

```
x=data['YearsExperience'].values.copy()
x
```

```
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])
```

```
x = data['YearsExperience'].values.copy()

print(f'x: {x}')

y_hat = b0 + b1 *x
print(f'\n\ny_hat: {y_hat}')
y = data['Salary'].values
print(f'\n\ny: {y}')
```

```
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]


y_hat: [73216.52244157 73348.79194593 73481.06145028 73811.73521117
 73944.00471553 74406.94798077 74473.08273295 74605.35223731
 74605.35223731 74936.0259982  75068.29550255 75134.43025473
 75134.43025473 75200.56500691 75465.10401562 75729.64302433
 75861.91252869 75994.18203304 76390.99054611 76457.12529829
 76986.20331571 77184.60757225 77713.68558967 77912.0898462
 78242.76360709 78441.16786363 78771.84162452 78837.97637669
 79300.91964194 79433.1891463 ]


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]
```

Step 8: Visual Comparision

```
import plotly.graph_objects as go
fig = go.Figure()
fig.add_trace(go.Scatter(x=data['YearsExperience'], y=data['Salary'], name='train', mode='markers', marker_color='rgba(152, 0, 0, .8)'))
fig.add_trace(go.Scatter(x=data['YearsExperience'], y=y_hat, name='prediction', mode='lines+markers', marker_color='rgba(0, 152, 0, .8)'))

fig.update_layout(title = f'Salary Based on Experience)', title_x=0.5, xaxis_title= "Experience IN years", yaxis_title="Salary")
fig.update_xaxes (showline=True, linewidth=2, linecolor='black', mirror=True)
fig.update_yaxes (showline=True, linewidth=2, linecolor='black', mirror=True)
fig.show()
print("OUTPUT BY =====>> MOHD.UMAIR, SHAMIM MIRAJKAR & KHAN NEEDA")
```

OUTPUT:


```
OUTPUT BY =====>> MOHD.UMAIR, SHAMIM MIRAJKAR & KHAN NEEDA
```

Conclusion:

Hence, we learned how to implement simple linear regression using python.