Lab 3

OBJECTIVE: Implementation linear regression using scikit-learn

Objective: To understand the fundamental concepts of linear regression and implement it using Python. This lab will lay the groundwork for understanding more complex deep learning models

Task: 1

Modify the code in Step 5 to predict the salary for someone with 3 years of experience and someone with 10 years of experience. Run the code and note down the predicted salaries.

```
import pandas as pd
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
```

```
data = pd.read_csv('Salary_dataset.csv')
print("Dataset:\n", data)
```

```
Dataset:
      Unnamed: 0  YearsExperience    Salary
0              0              1.2   39344.0
1              1              1.4   46206.0
2              2              1.6   37732.0
3              3              2.1   43526.0
4              4              2.3   39892.0
5              5              3.0   56643.0
6              6              3.1   60151.0
7              7              3.3   54446.0
8              8              3.3   64446.0
9              9              3.8   57190.0
10            10              4.0   63219.0
11            11              4.1   55795.0
12            12              4.1   56958.0
13            13              4.2   57082.0
14            14              4.6   61112.0
15            15              5.0   67939.0
16            16              5.2   66030.0
17            17              5.4   83089.0
18            18              6.0   81364.0
19            19              6.1   93941.0
20            20              6.9   91739.0
21            21              7.2   98274.0
22            22              8.0  101303.0
23            23              8.3  113813.0
24            24              8.8  109432.0
25            25              9.1  105583.0
26            26              9.6  116970.0
27            27              9.7  112636.0
28            28             10.4  122392.0
29            29             10.6  121873.0
```

```
X = data[['YearsExperience']]
Y = data['Salary']
```

```
model = LinearRegression() model.fit(X,Y)
```

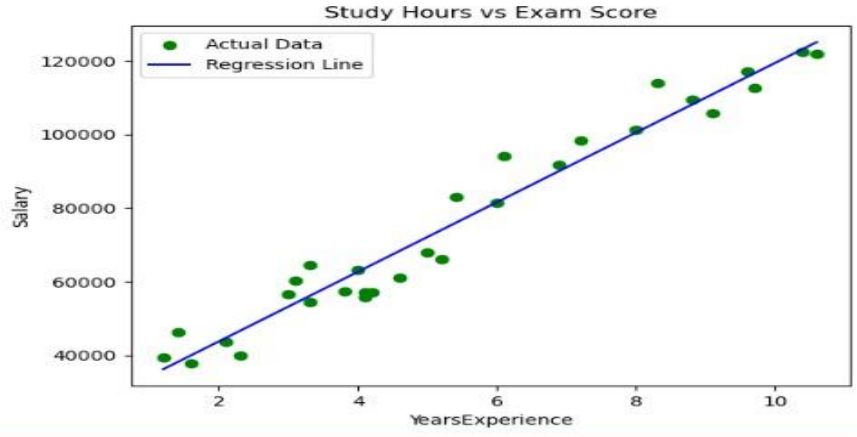
```
model = LinearRegression()
model.fit(X,Y)
predicted_score = model.predict([[3]])
print("Predicted score for  3 years of experience:" ,predicted_score[0])
```

```
Predicted score for  3 years of experience: 53198.09093088844
C:\Users\SED\anaconda3\Lib\site-packages\sklearn\base.py:493: UserWarning:
ture names
  warnings.warn(
```

```
predicted_score = model.predict([[10]])
print("Predicted score for  10 years of experience:" ,predicted_score[0])
```

```
Predicted score for  10 years of experience: 119347.82718107395
```

```
plt.scatter(X, Y, color='green', label='Actual Data')
plt.plot(X, model.predict(X), color='blue', label='Regression Line')
plt.xlabel('YearsExperience')
plt.ylabel('Salary')
plt. title('Study Hours vs Exam Score')
plt.legend()
plt. show()
```



Task: 2

Investigating the Impact of Outliers on Linear Regression

1. **Open the CSV File:**
   Open the Salary_dataset.csv file using a Microsoft Excel.
2. **Add the Outlier Data:**
   (a) Go to the end of the file (add a new row).
   (b) Enter the following values in the appropriate columns:
   (c) **YearsExperience:** 1.5
   (d) **Salary:** 150000
3. **Save the Changes:**
   (a) Save the modified Salary_dataset.csv file.
4. **Run Your Python Script:**
   (a) Now, run your original Python script (from Step 1 onwards). The script will load the modified CSV file, which now includes the outlier data point.

Analyze the Output:

- **Compare the results you obtained in this run (with the outlier) to the results you got when you ran the code with the original dataset (without the outlier).**
  Consider the following questions:
  - How did the regression line on the plot change? Did it tilt more or less? Did it shift up or down?
  - How did the value of the slope (coefficient) change? Did it increase or decrease?
  - How did the value of the intercept (bias) change? Did it increase or decrease?
  - How did the predicted salary for 6 years of experience change? Was it higher or lower?
  - Does the regression line seem to fit the majority of the original data points as well as it did before you added the outlier?
  - What does this experiment demonstrate about the influence of outliers on a linear regression model?

```
import pandas as pd
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
```

```
data = pd.read_csv('Salary_datasetupdated.py')
print("Dataset:\n", data)
```

```
Dataset:
    Unnamed: 0  YearsExperience      Salary
0            0              1.2     39344.0
1            1              1.4     46206.0
2            2              1.6     37732.0
3            3              2.1     43526.0
4            4              2.3     39892.0
5            5              3.0     56643.0
6            6              3.1     60151.0
7            7              3.3     54446.0
8            8              3.3     64446.0
9            9              3.8     57190.0
10          10              4.0     63219.0
11          11              4.1     55795.0
12          12              4.1     56958.0
13          13              4.2     57082.0
14          14              4.6     61112.0
15          15              5.0     67939.0
16          16              5.2     66030.0
17          17              5.4     83089.0
18          18              6.0     81364.0
19          19              6.1     93941.0
20          20              6.9     91739.0
21          21              7.2     98274.0
22          22              8.0    101303.0
23          23              8.3    113813.0
24          24              8.8    109432.0
25          25              9.1    105583.0
26          26              9.6    116970.0
27          27              9.7    112636.0
28          28             10.4    122392.0
29          29             10.6    121873.0
30          30              1.5   1500000.0
```

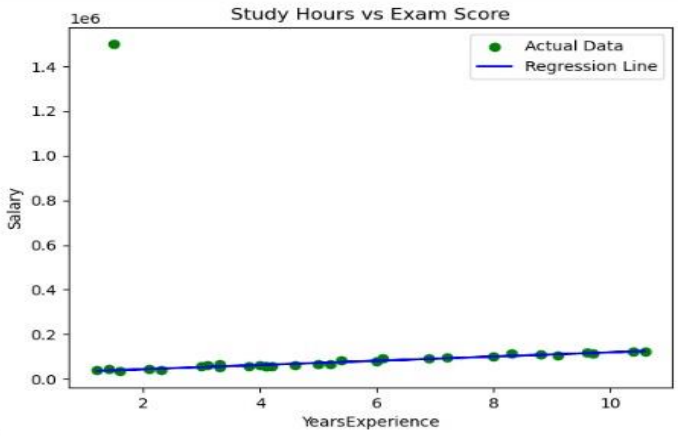
```
X = data[['YearsExperience']]
Y = data['Salary']
```

```
model = LinearRegression() model.fit(X,Y)
```

```
predicted_score = model.predict([[6]])
print("Predicted score for 6 hours of study:" ,predicted_score[0])
```

```
Predicted score for 6 hours of study: 81547.97789525366
```

```
plt.scatter(X, Y, color='green', label='Actual Data')
plt.plot(X, model.predict(X), color='blue', label='Regression Line')
plt.xlabel('YearsExperience')
plt.ylabel('Salary')
plt. title('Study Hours vs Exam Score')
plt.legend()
plt. show()
```




```
25,8.299999999999999,113813.0
24,8.799999999999999,109432.0
25,9.1,105583.0
26,9.6,116970.0
27,9.7,112636.0
28,10.4,122392.0
29,10.6,121873.0
30,1.5,1500000
```

Task 3. Find the Slope and Intercept of the

above code. Hint: Slope→ model.coef_

Intercept→ model.Intercept_

```
[57]: data = pd.read_csv('Salary_dataset.csv')
      print("Dataset:\n", data)
```

```
Dataset:
     Unnamed: 0  YearsExperience    Salary
0             0              1.2   39344.0
1             1              1.4   46206.0
2             2              1.6   37732.0
3             3              2.1   43526.0
4             4              2.3   39892.0
5             5              3.0   56643.0
6             6              3.1   60151.0
7             7              3.3   54446.0
8             8              3.3   64446.0
9             9              3.8   57190.0
10           10              4.0   63219.0
11           11              4.1   55795.0
12           12              4.1   56958.0
13           13              4.2   57082.0
14           14              4.6   61112.0
15           15              5.0   67939.0
16           16              5.2   66030.0
17           17              5.4   83089.0
18           18              6.0   81364.0
19           19              6.1   93941.0
20           20              6.9   91739.0
21           21              7.2   98274.0
22           22              8.0  101303.0
23           23              8.3  113813.0
24           24              8.8  109432.0
25           25              9.1  105583.0
26           26              9.6  116970.0
27           27              9.7  112636.0
28           28             10.4  122392.0
29           29             10.6  121873.0
```

```
[59]: X = data[['YearsExperience']]
      Y = data['Salary']
```

```
[61]: model = LinearRegression()
      model.fit(X,Y)
```

```
[61]: LinearRegression
      LinearRegression()
```

```
[72]: slope = model.coef_
      intercept = model.intercept_
```

```
[74]: print("Slope:", slope)
      print("Intercept:", intercept)
```

```
Slope: [9449.96232146]
Intercept: 24848.203966523222
```

```
model = LinearRegression() model.fit(X,Y)
```

```
[68]: predicted_score = model.predict([[6]])
      print("Predicted score for 6 hours of study:" ,predicted_score[0])
```

Task 4. Calculate the Error using Mean Square

Error Hint:

from sklearn.metrics import

mean_squared_error

mean_squared_error(Actual Y, Predicted Y)

```
from sklearn.metrics import mean_squared_error

actual_y = [3, -0.5, 2, 7]
predicted_y = [2.5, 0.0, 2, 8]

mse = mean_squared_error(actual_y, predicted_y)

print("Mean Squared Error:", mse)
```

```
Mean Squared Error: 0.375
```