BÀI TUẦN 02: SIMPLE LINEAR REGRESSION

1. Thông tin sinh viên

DUONG MINH LUONG-18521071

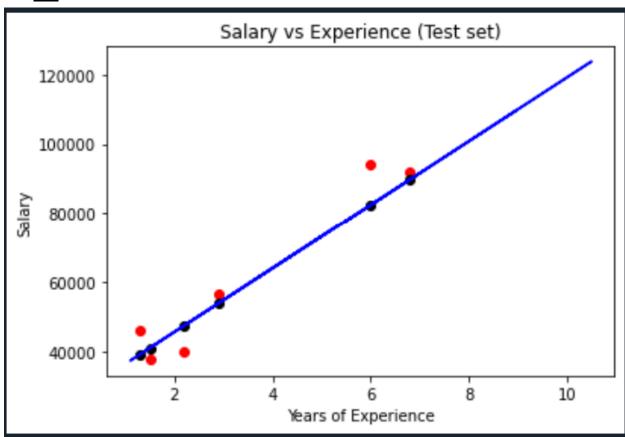
2. Source

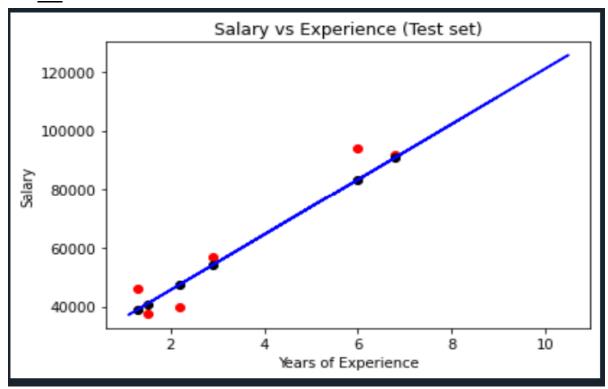
```
1. import pandas as pd #cho du lieu tu file
2. import numpy as np
                       #xu li mang
3. import matplotlib.pyplot as plt #truc quan hóa dữ liệu
4. from sklearn.model selection import train test split
  #phân chia dữ liệu
5. from sklearn.linear model import LinearRegression
6. from sklearn.metrics import r2 score
7. #Tiền xử lí dữ liệu
8. dataset =pd.read csv("Salary Data.csv")
9. dataset1 =pd.read csv("Salary Data Test.csv")
     X=np.array(dataset.iloc[:,:-1].values)
11.
    Y=np.array(dataset.iloc[:,1].values)
     X1 test=np.array(dataset1.iloc[:,:-1].values)
13. Y1 test=np.array(dataset1.iloc[:,1].values)
14. X train, X test, Y train, Y test=train test split(X, Y, t
  rain size=0.5, random state=0)
15. reg=LinearRegression()
    reg.fit(X train, Y train)
16.
17. Y train pred=reg.predict(X train)
18. Y test pred=reg.predict(X1 test)
19.
    #hoàn thiện
20.
    plt.scatter(X1 test, Y1 test, color = 'red')
21.
    plt.scatter(X1 test, Y test pred, color="BLACK")
    plt.plot(X train, Y train pred, color="BLUE")
23. plt.title('Salary vs Experience (Test set)')
24. plt.xlabel('Years of Experience')
25.
    plt.ylabel('Salary')
26.
    plt.show()
27.
    def compare(i example):
28.
            x=X1 test[i example : i example+1]
29.
            y=Y1 test[i example]
30.
            y pred=reg.predict(x)
31.
            print(x,y,y pred)
32. for i in range(len(X1 test)):
33.
         compare(i)
34.
    #Đánh giá mô hình
35.
     from sklearn.metrics import mean squared error
36.
     print(mean squared error(Y1 test, Y test pred))
```

```
37. print('(R)2 test = ',reg.score(X1_test, Y1_test))
38. r2=r2_score(Y1_test,Y_test_pred)
39. print('(R)2 test = ',r2)
40.
41.
42.
43.
```

3. Kết quả

<u>0.5</u>



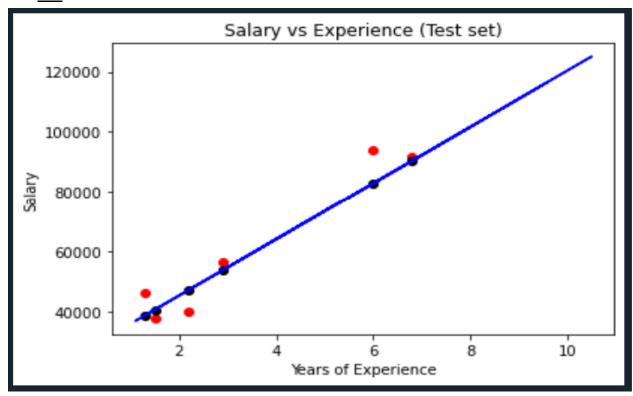


<u>0.7</u>





<u>0.9</u>



	R ²	MSE (Mean Square Error)
0.5	0.9192840479224546	43785512.66922273
0.6	0.9261041740558313	40085838.54617234
0.7	0.9237126514543206	41383153.89056159
0.8	0.9215791609988908	42540496.03869277
0.9	0.9228800977916074	41834784.429609336

Nhận xét:

- Ta thấy với việc chia 60% của 30 điểm dữ liệu cho Train thì dữ liệu test cho kết quả tốt nhất.
- Với việc chia 50% của 30 điểm dữ liệu cho Train thì lỗi sẽ cao nhất với dữ liệu test.
- Ta cần tìm MSE (min) or tối thiểu để tăng độ chính xác.
- MSE ít cost function thì parameter phải tốt nhất.
- Với bộ sữ liệu test cố định này thì kết quả giữa các cách chia train có sự biến đổi nhưng việc biến đổi như thế có thể làm độ chính xác giảm xuống. Nếu ta lấy bộ test khác có thể kết quả cũng sẽ khác.