BÀI TUẦN 03: MULTIPLE LINEAR REGRESSION

1.Thông tin sinh viên

DƯƠNG MINH LƯỢNG-18521071

2.Source

```
1.  import numpy as np
2.  import pandas as pd
3.  from sklearn.preprocessing import OneHotEncoder
4.  #from sklearn.preprocessing import LabelEncoder
5.  from sklearn.compose import ColumnTransformer
6.  from sklearn.linear_model import LinearRegression
7.  # dataset= pd.read_csv("50_Startups.csv")
8.  # X = dataset.iloc[:,0:4].values
9.  # Y = dataset.iloc[:,-1].values
10. dataset= pd.read_csv("50_Startups_Train.csv")
11. dataset1= pd.read_csv("50_Startups_Test.csv")
12. X_train = dataset.iloc[:,1:5].values
13. Y_train= dataset.iloc[:,-1].values
14.
15. X_test = dataset1.iloc[:,1:5].values
16. Y_test = dataset1.iloc[:,-1].values
17.
18. # le = LabelEncoder()
19. # le.fit(X[:,3])
20. # X[:,3]= le.fit_transform(X[:,3])
21. # ohe = OneHotEncoder(handle_unknown = 'ignore')
22. # df = ohe.fit_transform(dataset[['State']]).toarray()
23. # X = np.concatenate((df, X), axis = 1)
24.
25. ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [3])],
    remainder='passthrough')
26. X_train = np.array(ct.fit_transform(X_train))
27. X_test = np.array(ct.fit_transform(X_test))
28.
29. lin_reg = LinearRegression()
30. lin_reg.fit(X_train, Y_train)
31. print("R(2)_Train= ",lin_reg.score(X_train, Y_train))
32. print("R(2)_Test= ",lin_reg.score(X_test, Y_test))
```

```
33. def compare(i_example):
34.     x = X_test[i_example : i_example + 1]
35.     y = Y_test[i_example]
36.     y_pred = lin_reg.predict(x)[0]
37.     print(x[:,3:6],x[:, 0:3],y, y_pred)
38. for i in range(len(X_test)):
39.     compare(i)
```

3. Kết quả

Source:1

```
R(2)_Train=  0.9501847627493607
R(2)_Test=  0.9347068473283919
[[66051.52 182645.56 118148.2]] [[0.0 1.0 0.0]] 103282.38 103015.201597961
[[100671.96 91790.61 249744.55]] [[1.0 0.0 0.0]] 144259.4 132582.27760817177
[[101913.08 110594.11 229160.95]] [[0.0 1.0 0.0]] 146121.95 132447.73845177446
[[27892.92 84710.77 164470.71]] [[0.0 1.0 0.0]] 77798.83 71976.09851257701
[[153441.51 101145.55 407934.54]] [[0.0 1.0 0.0]] 191050.39 178537.48221060817
[[72107.6 127864.55 353183.81]] [[0.0 0.0 1.0]] 105008.31 116161.24230164687
[[20229.59 65947.93 185265.1]] [[0.0 0.0 1.0]] 81229.06 67851.69209673627
[[61136.38 152701.92 88218.23]] [[0.0 0.0 1.0]] 97483.56 98791.73374685843
[[73994.56 122782.75 303319.26]] [[0.0 1.0 0.0]] 110352.25 113969.43533013933
[[142107.34 91391.77 366168.42]] [[0.0 1.0 0.0]] 166187.94 167921.06569555763
```

Source:2

```
R(2)_Train=  0.9489030839764381
R(2)_Test=  0.6167464436795949
[[46014.02 85047.44 205517.64]] [[0.0 0.0 1.0]] 96479.51 88939.52457907514
[[28663.76 127056.21 201126.82]] [[0.0 1.0 0.0]] 90708.19 74658.12163536082
[[44069.95 51283.14 197029.42]] [[1.0 0.0 0.0]] 89949.14 94684.25565065976
[[20229.59 65947.93 185265.1]] [[0.0 0.0 1.0]] 81229.06 68559.27702236232
[[38558.51 82982.09 174999.3]] [[1.0 0.0 0.0]] 81005.76 88267.63122737015
[[28754.33 118546.05 172795.67]] [[1.0 0.0 0.0]] 78239.91 78646.46057988092
[[27892.92 84710.77 164470.71]] [[0.0 1.0 0.0]] 77798.83 75191.42658789334
[[23640.93 96189.63 148001.11]] [[1.0 0.0 0.0]] 71498.49 75006.81667685887
[[15505.73 127382.3 35534.17]] [[0.0 0.0 1.0]] 69758.98 58550.551879867715
[[22177.74 154806.14 28334.72]] [[1.0 0.0 0.0]] 65200.33 68431.42420258804
[[1000.23 124153.04 1903.93]] [[0.0 0.0 1.0]] 64926.08 46240.18776936808
[[1315.46 115816.21 297114.46]] [[0.0 1.0 0.0]] 49490.75 55237.35882966414
[[0.0 135426.92 0.0]] [[1.0 0.0 0.0]] 42559.73 50796.337698024465
[[542.05 51743.15 0.0]] [[0.0 0.0 1.0]] 35673.41 49199.70336513858
[[0.0 116983.8 45173.06]] [[1.0 0.0 0.0]] 14681.4 52660.96951077474
```

Nhận xét:
- Với source 1 thì mô hình cho ra kết quả tốt với Score R Test :
  0.9347068473283919

- Với source 2 thì mô hình cho ra kết quả không tốt R(2)_Test= 0.6167464436795949 (<0.8)
- Cho thấy với cùng cách huấn luyện thì cho ra 2 mô hình khi test thì kết quả khác nhau .
- Cần xem lại việc huấn luyện