Homework_answer

September 2, 2021

[17]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import math

from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score
```

1 Đọc dữ liệu

Dữ liệu về giá nhà ở Boston được hỗ trợ bởi sklearn, đọc dữ liệu thông qua hàm `datasets.load_boston()`

Xem thêm các bộ dữ liệu khác tại https://scikit-learn.org/stable/datasets/index.html#toy-datasets.

Dữ liệu được chia thành các thành phần data và target như tập diabetes. Dữ liệu cũng đã được chuẩn hóa, chỉ cần gọi ra và huấn luyện

[32]:
```
# lay du lieu dataset - du lieu ve giá nhà
dataset = datasets.load_boston()
print("Số chiều dữ liệu input: ", dataset.data.shape)
print("Số chiều dữ liệu target: ", dataset.target.shape)
print()

print("5 mẫu dữ liệu đầu tiên:")
print("input: ", dataset.data[:5])
print("target: ",dataset.target[:5])
```

```
Số chiều dữ liệu input:  (506, 13)
Số chiều dữ liệu target:  (506,)

5 mẫu dữ liệu đầu tiên:
input:  [[6.3200e-03 1.8000e+01 2.3100e+00 0.0000e+00 5.3800e-01 6.5750e+00
  6.5200e+01 4.0900e+00 1.0000e+00 2.9600e+02 1.5300e+01 3.9690e+02
  4.9800e+00]
 [2.7310e-02 0.0000e+00 7.0700e+00 0.0000e+00 4.6900e-01 6.4210e+00
  7.8900e+01 4.9671e+00 2.0000e+00 2.4200e+02 1.7800e+01 3.9690e+02
```

```
  9.1400e+00]
 [2.7290e-02 0.0000e+00 7.0700e+00 0.0000e+00 4.6900e-01 7.1850e+00
  6.1100e+01 4.9671e+00 2.0000e+00 2.4200e+02 1.7800e+01 3.9283e+02
  4.0300e+00]
 [3.2370e-02 0.0000e+00 2.1800e+00 0.0000e+00 4.5800e-01 6.9980e+00
  4.5800e+01 6.0622e+00 3.0000e+00 2.2200e+02 1.8700e+01 3.9463e+02
  2.9400e+00]
 [6.9050e-02 0.0000e+00 2.1800e+00 0.0000e+00 4.5800e-01 7.1470e+00
  5.4200e+01 6.0622e+00 3.0000e+00 2.2200e+02 1.8700e+01 3.9690e+02
  5.3300e+00]]
target:  [24.  21.6 34.7 33.4 36.2]
```

Chia dữ liệu làm 2 phần training 362 mẫu và testing 80 mẫu

[19]:
```
# cat nho du lieu, lay 1 phan cho qua trinh thu nghiem,
# chia train test cac mau du lieu
# dataset_X = dataset.data[:, np.newaxis, 2]
dataset_X = dataset.data

dataset_X_train = dataset_X[:404]
dataset_y_train = dataset.target[:404]

dataset_X_test = dataset_X[405:]
dataset_y_test = dataset.target[405:]
```

2 Xây dựng mô hình

2.1 Xây dựng mô hình bằng thư viện

[20]:
```
regr = linear_model.LinearRegression()
```

2.2 Xây dựng mô hình Linear Regression tự viết

[21]:
```
def linear_regression(dataset_X_train,dataset_y_train):
    one = np.ones((dataset_X_train.shape[0],1))
    Xbar = np.concatenate((one, dataset_X_train), axis = 1)

    A = np.dot(Xbar.T, Xbar)
    b = np.dot(Xbar.T, dataset_y_train)
    w_lr = np.dot(np.linalg.pinv(A), b)
    coef = w_lr[1:]
    intercept = w_lr[0]
    return coef, intercept
```

2.3 Hàm test mô hình tự viết

[22]:
```
def predict(intercept, coef, dataset_X_test):
    y_pred_ = [intercept] + coef.dot(dataset_X_test[0].T)
    for i in range(1, len(dataset_X_test)):
        y_pred = intercept + coef.dot(dataset_X_test[i].T)
        y_pred_= np.append(y_pred_,y_pred)
    return y_pred_
```

3 Huấn luyện mô hình

3.1 Huấn luyện mô hình của thư viện

[33]:
```
regr.fit(dataset_X_train, dataset_y_train)
print("[w1, ... w_n] = ", regr.coef_)
print("w0 = ", regr.intercept_)
```

```
[w1, … w_n] =  [-2.02135297e-01  4.41276341e-02  5.26739364e-02
1.88474315e+00
 -1.49281487e+01  4.76038673e+00  2.88734527e-03 -1.30025278e+00
  4.61661953e-01 -1.55434673e-02 -8.11632369e-01 -1.97174433e-03
 -5.32273431e-01]
w0 =  30.077166922901856
```

3.2 Training mô hình bằng Linear regression tự viết

[24]:
```
coef, intercept = linear_regression(dataset_X_train,dataset_y_train)
print("[w1, ... w_n] = ", coef)
print("w0 = ", intercept)
```

```
[w1, … w_n] =  [-2.02135297e-01  4.41276341e-02  5.26739364e-02
1.88474315e+00
 -1.49281487e+01  4.76038673e+00  2.88734527e-03 -1.30025278e+00
  4.61661953e-01 -1.55434673e-02 -8.11632369e-01 -1.97174433e-03
 -5.32273431e-01]
w0 =  30.07716691924543
```

4 Dự đoán các mẫu dữ liệu

4.1 Dự đoán các mẫu dữ liệu theo mô hình của thư viện

[25]:
```
dataset_y_pred_lib = regr.predict(dataset_X_test)
pd.DataFrame(data=np.array([dataset_y_test, dataset_y_pred_lib,
                            abs(dataset_y_test - dataset_y_pred_lib)]).T,
             columns=["Thực tế", "Dự đoán", "Lệch"])
```

[25]:

	Thực tế	Dự đoán	Lệch
0	5.0	3.787057	1.212943
1	11.9	6.640550	5.259450
2	27.9	21.312765	6.587235
3	17.2	15.412714	1.787286
4	27.5	23.652298	3.847702
..	…	…	…
96	22.4	23.755044	1.355044
97	20.6	22.081673	1.481673
98	23.9	28.181773	4.281773
99	22.0	26.572420	4.572420
100	11.9	22.020566	10.120566

[101 rows x 3 columns]

4.2 Dự đoán các mẫu dữ liệu tính theo linear regression tự viết

[26]:
```
dataset_y_pred = predict(intercept, coef, dataset_X_test)
pd.DataFrame(data=np.array([dataset_y_test, dataset_y_pred,
                            abs(dataset_y_test - dataset_y_pred)]).T,
             columns=["Thực tế", "Dự đoán", "Lệch"])
```

[26]:

	Thực tế	Dự đoán	Lệch
0	5.0	3.787057	1.212943
1	11.9	6.640550	5.259450
2	27.9	21.312765	6.587235
3	17.2	15.412714	1.787286
4	27.5	23.652298	3.847702
..	…	…	…
96	22.4	23.755044	1.355044
97	20.6	22.081673	1.481673
98	23.9	28.181773	4.281773
99	22.0	26.572420	4.572420
100	11.9	22.020566	10.120566

[101 rows x 3 columns]

4.3 Đánh giá mô hình linear regression của thư viện

[27]:
```
loss = math.sqrt(mean_squared_error(dataset_y_test, dataset_y_pred_lib))
print("lỗi :",loss)
```

```
lỗi : 5.749521870254025
```

4.4 Đánh giá mô hình linear regression tự viết

[28]:
```
loss = math.sqrt(mean_squared_error(dataset_y_test, dataset_y_pred))
print("lỗi :",loss)
```

```
lỗi : 5.749521870168214
```

[ ]:
```

```