

## Bài 2: Mô hình Hồi quy tuyến tính

### Hướng dẫn 2: Sử dụng mô hình *LinearRegression*

#### 1. Mô hình hồi quy tuyến tính của thư viện sklearn

Thay vì phải lập trình xây dựng mô hình HQT từ đầu như cách mà chúng ta vẫn làm ở các bài học trước, [\*thư viện sklearn cung cấp nhiều mô hình HQT\*](#) có sẵn để chúng ta sử dụng. Trong đó, lớp [\*LinearRegression\*](#) tương ứng với mô hình HQT đa biến mà chúng ta xây dựng trong các bài trước.

Quy tắc sử dụng chung mô hình hồi quy do sklearn cung cấp là:

- Bước 1: Khởi tạo mô hình
- Bước 2: Huấn luyện mô hình với tập dữ liệu (huấn luyện – sẽ được học ở bài 5) bằng hàm `fit()`
- Bước 3: Tiến hành dự đoán bằng hàm `predict()`

**Lưu ý:** đối với các mô hình do sklearn cung cấp thì vector  $y$  được xét ở không gian  $R^m$  (chứ không phải  $R^{m \times 1}$ ).

#### 2. Hướng dẫn sử dụng

Huấn luyện mô hình [\*LinearRegression\*](#) có kết hợp điều chỉnh dữ liệu bằng [\*MinMaxScaler\*](#)

```
import os
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from sklearn.linear_model import LinearRegression

D = np.loadtxt(os.path.join("D:/data/hocmay", "ex1data2.txt"),
               delimiter=",")
print('Kích thước của tập dữ liệu: ', D.shape)
print('Thực hiện MinMaxScaler')
#Khởi tạo bộ điều chỉnh dữ liệu
scaler = MinMaxScaler()
#Phải thực hiện thao tác fit(data) trước khi điều chỉnh dữ liệu
scaler.fit(D)
#Thực hiện điều chỉnh dữ liệu
D = scaler.transform(D)
print('Lấy ra tập dữ liệu X, y')
X, y = D[:, :-1], D[:, -1]
print('Kích thước tập X: ', X.shape)
print('Kích thước vector y: ', y.shape)
print('Huấn luyện mô hình LinearRegression')
#Khởi tạo mô hình
```

```
model = LinearRegression()  
#Huấn luyện mô hình với tập dữ liệu X, y  
model.fit(X,y)  
#Bộ trọng số tối ưu:  
print('\t\tw optimal: ', model.coef_)
```