

Bài 3: HỒI QUY LOGISTIC

Hướng dẫn sử dụng mô hình *LogisticRegression*

1. Sử dụng mô hình *LogisticRegression* của thư viện *sklearn*

Thay vì lập trình mô hình hồi quy logistic hoàn toàn như cách làm trong bài học lý thuyết, chúng ta có thể sử dụng [mô hình *LogisticRegression*](#) do thư viện *sklearn* cung cấp.

Ở mức đơn giản nhất, việc sử dụng [mô hình *LogisticRegression*](#) gồm các bước sau:

- Bước 1: Khởi tạo mô hình
- Bước 2: Huấn luyện mô hình với hàm `fit()`
- Bước 3: Sử dụng mô hình để dự đoán với hàm `predict()` (Nội dung này sẽ học ở bài 5)

Lưu ý: Đối với mô hình hồi quy logistic (thực chất đây là mô hình phân lớp – *classifier/classification model*) thì vector y chứa các giá trị rời rạc chứ không phải các giá trị liên tục $y \in C^m$ với $C = \{c_i | \forall i = \overline{1, k}; k \in N\}$. Do vậy, khi điều chỉnh giá trị của dữ liệu, chúng ta cần loại vector y ra – không thực hiện điều chỉnh dữ liệu (data scaling) đối với y .

2. Hướng dẫn sử dụng

Huấn luyện [mô hình *LogisticRegression*](#) với dữ liệu được điều chỉnh bằng [MinMaxScaler](#).

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

D = np.loadtxt(os.path.join("D:/data/hocmay", "ex2data1.txt"),
               delimiter=",")
print('Kích thước của tập dữ liệu: ', D.shape)
print('Lấy ra tập dữ liệu X, y')
X, y = D[:, :-1], D[:, -1]
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(X)
#Thực hiện điều chỉnh dữ liệu trên X, không điều chỉnh với y
X = scaler.transform(X)
print('Kích thước tập X: ', X.shape)
print('Kích thước vector y: ', y.shape)
```

```
print('Huấn luyện mô hình LogisticRegression')  
#Khởi tạo mô hình  
model = LogisticRegression()  
#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_)
```