Bài 7: MÔ HÌNH PHÂN LỚP NAÏVE BAYES

(*Naïve Bayes classifier*)

1. Giới thiệu mô hình Naïve Bayes

Lấy nền tảng là lý thuyết Bayes, mô hình phân lớp Naïve Bayes hoạt động dựa trên giả định về tính độc lập có điều kiện giữa các thuộc tính khi biết giá trị của nhãn lớp tương ứng. Lý thuyết Bayes trong trường hợp này có thể được phát biểu như sau:

$$P(y|x_1, \dots, x_n) = \frac{P(y)P(x_1,\dots,x_n|y)}{P(x_1,\dots,x_n)} \; với \; x_i \in X, X \in R^{n\times m} \; và \; y \in R^m \quad (1)$$

Giả định độc lập có điều kiện 'ngây thơ – naïve' được phát biểu là:

$$P(x_i|y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i|y)$$

Khi đó phương trình (1) được viết lại là:

$$P(y|x_1, \dots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1,\dots,x_n)} \quad (2)$$

Do $P(x_1, \dots, x_n)$ không đổi, nên có thể suy ra

$$P(y|x_1, \dots, x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y) \quad (3)$$

Từ đó, xác định luật phân lớp là:

$$\hat{y} = \underset{y}{\operatorname{argmax}} P(y)\prod_{i=1}^{n} P(x_i|y) \quad (4)$$

Đặc tính của mô hình phân lớp Naïve Bayes là:

- Tốc độ thực hiện nhanh;
- Rất hữu hiệu với dữ liệu văn bản.

2. Xây dựng mô hình Naïve Bayes với thư viện sklearn

Thư viện sklearn cung cấp nhiều biến thể của mô hình phân lớp Naïve Bayes như: (i) GausianNB; (ii) MultinomialNB (cho dữ liệu định tính như từ, văn bản); (iii) ComplementNB (cho dữ liệu định tính nhưng có nhãn lớp mất cân bằng); và (iv) BernoulliNB (dùng cho dữ liệu tuân theo phân phối Bernoulli).

Ngoài ra, sklearn cung cấp phương thức partial_fit để huấn luyện mô hình trong trường hợp tập *X* quá lớn, vượt quá khả năng lưu trữ của RAM. Một ví dụ mở rộng về việc xây dựng mô hình phân lớp văn bản với tập dữ liệu lớn có thể tham khảo tại đây.

Ví dụ về xây dựng mô hình Naïve Bayes với GaussianNB:

```
import os
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score
from sklearn.model_selection import cross_val_score

#Bước 1: Đọc dữ liệu
D = np.loadtxt(os.path.join('D:/data/hocmay', 'ex2data2.txt'), delimiter=',')
X, y = D[:, :-1], D[:, -1]
#scaler = MinMaxScaler()
#scaler.fit(D)
#X = scaler.transform(X)
#Xác định số lượng mẫu của từng nhãn lớp - balanced dataset?
unique, counts = np.unique(y, return_counts=True)
result = dict(zip(unique, counts))
print(result)

#Bước 2: Phân chia train - test
X_train, X_test, y_train, y_test = train_test_split(X, y, shuffle=True,
                                                    random_state=15,
                                                    test_size=0.30)

#Bước 3: Khởi tạo và huấn luyện mô hình
gnb_model = GaussianNB()
gnb_model.fit(X_train, y_train)

#Bước 4: Dự đoán và đánh giá độ chính xác
y_hat = gnb_model.predict(X_test)
print('Độ chính xác: ', accuracy_score(y_hat, y_test))

#Bước 5: Đánh giá 10-fold CV
scores = cross_val_score(gnb_model, X_train, y_train, cv = 10,
scoring='accuracy')
print('10-fold CV results')
print(scores)
print('ĐỘ chính xác trung bình của 10-fold CV: ', np.mean(scores))
```