SỞ GIÁO DỤC & ĐÀO TẠO THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG THPT CHUYÊN TRẦN ĐẠI NGHĨA

Báo Cáo Dự Án

NAIVE BAYES

Giáo viên hướng dẫn: Cô Hồ Ngọc Lâm

Học sinh thực hiện 1: Bùi Quốc Minh Nhật

Học sinh thực hiện 2: Nguyễn Khánh Trang

Học sinh thực hiện 3: Nguyễn Xuân Nghĩa

Môn học: Tin chuyên

Tháng 10 năm 2023, Thành phố Hồ Chí Minh

MỤC LỤC

NỘI DUNG

Chương 1: Giới thiệu

Bộ phân lớp Bayes là một giải thuật thuộc lớp giải thuật thống kê, nó có thể dự đoán xác suất của một phần tử dữ liệu thuộc vào một lớp là bao nhiêu. Đây còn là một kỹ thuật quan trọng trong lĩnh vực học máy. Phân lớp Bayes được xây dựng dựa trên định lý Bayes để dự đoán xác suất của một sự kiện dựa trên thông tin trước đó.

Tuy phân lớp Bayes có nhiều dạng và nhiều loại nhưng các giải thuật phân lớp Bayes đều được ứng dụng rộng rãi trong lĩnh vực khoa học máy tính, đặc biệt là trong trí tuệ nhân tạo (AI), nhằm giải quyết các vấn đề liên quan đến xác suất như Phân lớp mail (Mail Classification), Phân lớp văn bản (Document Classification),.... Báo cáo này sẽ lần lượt trình bày về định lý Bayes - định lý cơ bản xây dựng nên phân lớp Bayes - các dạng định lý Bayes, lý thuyết phân lớp Naive Bayes và các ứng dụng thực tiễn của nó.

Chương 2: Định lý Bayes

2.1. Định lý Bayes

Định lý Bayes (được đặt theo tên tác giả của nó là Thomas Bayes). cho phép tính xác suất xảy ra của một sự kiện ngẫu nhiên A khi biết sự kiện liên quan B đã xảy ra. Xác suất này là xác suất có điều kiện[1] và được ký hiệu là P(A|B), và đọc là "xác suất của A nếu có B".

Theo định lý Bayes, xác suất xảy ra A khi biết B sẽ phụ thuộc vào 3 yếu tố:

- Xác suất xảy ra A của riêng nó, không quan tâm đến B. Ký hiệu là **P(A)** và đọc là **"xác suất của A"**. Một số tài liệu còn gọi xác suất này là xác suất "tiên nghiệm" [2], tức nó không quan tâm đến bất kỳ thông tin nào về B.
- Xác suất xảy ra B của riêng nó, không quan tâm đến A. Ký hiệu là **P(B)** và đọc là **"xác suất của B"**. Đại lượng này là một hằng số và thường được gọi là "hằng số chuẩn hóa" (normalizing constant), vì nó luôn giống nhau, không phụ thuộc vào sự kiện A đang muốn biết.
- Xác suất xảy ra B khi biết A xảy ra. Ký hiệu là **P(B|A)** và đọc là **"xác suất của B nếu có A"**. Đại lượng này gọi là khả năng (likelihood [3]) xảy ra B khi biết A đã xảy ra.

Khi biết ba đại lượng trên, ta có định lý Bayes:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)} = \frac{likelihood * prior}{normalizing\ constant}$$

2.2. Các dạng khác của định lý Bayes

Định lý Bayes cũng thường được viết dưới dạng:

$$P(A) = P(AB) + P(A\underline{B}) = P(A|B) * P(B) + P(A|\underline{B}) * P(\underline{B})$$

hoặc dạng:

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B|A) * P(A) + P(B|\underline{A}) * P(\underline{A})}$$

Đối với $\{A_i\}$ tạo thành một phân hoạch [4] của không gian các biến cố, ta có thể tính $P(A_i|B)$ theo công thức xác suất đầy đủ, một dạng khác của định lý Bayes:

$$P(A_i|B) = \frac{P(B|A_i)*P(A_i)}{\sum_j \quad P(B|A_j)*P(A_j)}$$

2.3. Định lý Bayes với hàm mật độ xác suất

Các dạng đẳng thức trên của định lý Bayes được sử dụng để tính xác suất có điều kiện đối với phần tử của một tập các phân hoạch phân bố không liên tục của không gian các biến cố. Tuy nhiên, để tính xác suất có điều kiện của một phân hoạch phân bố liên tục, ta cần áp dụng định lý Bayes cho hàm mật độ xác suất như sau:

$$f(x|y) = \frac{f(y|x) * f(x)}{\int_{-\infty}^{\infty} \quad f(y|x') * f(x')dx'}$$

Trong đó:
- $f(x, y)$ là mật độ phân phối của phân phối đồng thời của các biến ngẫu nhiên X và Y.
- $f(x|y)$ là mật độ phân phối xác suất hậu nghiệm của X với điều kiện Y=y.
- $f(x)$ là phân phối tiên nghiệm của X, là phân phối không quan tâm đến điều kiện Y

Chương 3: Phân lớp Naive Bayes

3.1. Định nghĩa

Phương pháp Naive Bayes là một tập hợp các thuật toán học máy giám sát dựa trên việc áp dụng Định lý Bayes với giả định "ngây thơ" về sự độc lập có điều kiện giữa mỗi cặp đặc trưng khi biết giá trị của biến lớp. Định lý Bayes mô tả mối quan hệ sau, với biến lớp y và vector đặc trưng phụ thuộc $\{x_1, x_2, \ldots, x_n\}$

$$P(y|x_1, x_2, \ldots, x_n) = \frac{P(x_1, x_2, \ldots, x_n|y) * P(y)}{P(x_1, x_2, \ldots, x_n)} \quad (1)$$

Sử dụng giả định "ngây thơ" về sự độc lập có điều kiện rằng:

$$P(x_i|y, x_1, \ldots, x_n) = P(x_i|y)$$

Với mọi i, ta có thể đơn giản hóa đẳng thức (1) như sau:

$$P(y|x_1, \ldots, x_n) = \frac{P(y) * \prod_1^n \quad P(x_i|y)}{P(x_1, \ldots, x_n)}$$

Do $P(x_1,\ldots,x_n)$ là hằng số đối với mỗi bộ input, ta có:

$$P(y|x_1,\ldots,x_n) \sim P(y) * \prod_{i=1}^{n} P(x_i|y)$$

$$\Rightarrow \hat{y} = argmax\, P(y) * \prod_{i=1}^{n} P(x_i|y)$$

Chúng ta có thể sử dụng phương pháp ước lượng Maximum A Posteriori (MAP) [5] để ước lượng $P(y)$ và $P(x_i|y)$. Trong đó, $P(y)$ là tần suất tương đối của lớp y trong bộ dữ liệu training cho mô hình thuật toán.

Điểm khác biệt chính giữa các loại phân lớp Bayes khác nhau nằm ở phương pháp ước lượng giá trị của $P(x_i|y)$.

3.2. Phân loại

Phần này sẽ trình bày một số loại phân lớp Bayes phổ biến

3.2.1. Gaussian Naive Bayes

Thuật toán Gaussian Naive Bayes là một trong những thuật toán phân lớp Naive Bayes đơn giản nhất. Ta có thể cài đặt thuật toán Gaussian Naive Bayes để phân loại. Phương pháp ước lượng khả năng xảy ra về xác suất của các đặc trưng là phân phối Gaussian [6]:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} * exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

Do phương pháp này phức tạp và không hiệu quả khi cài đặt lại từ đầu trong việc giải quyết các bài toán ứng dụng, người ta đã viết sẵn các hàm cần thiết và đưa vào thư viện "sklearn" [7] của ngôn ngữ lập trình Python [8].

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
X, y = load_iris(return_X_y=True)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.5, random_state=0)
gnb = GaussianNB()
y_pred = gnb.fit(X_train, y_train).predict(X_test)
print("Number of mislabeled points out of a total %d points : %d" % (X_test.shape[0], (y_test !=
y_pred).sum()))
```

Hình 1. Hình ảnh mô tả ứng dụng mẫu Gaussian Naive Bayes

3.2.2. Multinomial Naive Bayes

Đây là thuật toán Naive Bayes cho dữ liệu phân phối theo đa thức [9] và là một trong hai biến thể kinh điển của Naive Bayes được sử dụng trong phân loại văn bản [10] (trong đó, dữ liệu thường được biểu diễn dưới dạng vector đặc trưng). Phân phối này được tham số hóa bởi các vector $\theta_y = (\theta_{y1}, \theta_{y2}, \ldots, \theta_{yn})$ với mỗi lớp y, với n là số đặc trưng của dữ liệu và θ_{yi} là xác suất có điều kiện $P(x_i|y)$ của đặc trưng thứ i thuộc lớp y.

Khả năng xảy ra về xác suất của các đặc trưng được ước lượng bằng một phiên bản cải tiến của Maximum Likelihood [11]:

$$\widehat{\theta_y} = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

Trong đó:

- N_{yi} là số lần đặc trưng i xuất hiện trong mẫu dữ liệu huấn luyện của lớp y
- N_y là tổng số đặc trưng khác nhau của lớp y và $\alpha \geq 0$ là thành phần làm mịn được xác định bằng Maximum likelihood để khắc phục vấn đề xác suất bằng 0 nhằm tránh bị ảnh hưởng đến những bước tính toán tiếp theo.

Naive Bayes còn có các dạng khác như Complement Naive Bayes, Binomial Naïve Bayes... nhưng không phổ biến bằng Multinomial Naive Bayes và Gaussian Naive Bayes

3.3. Ưu điểm và nhược điểm

	Ưu điểm	**Nhược điểm**
Giả định độc lập	Dễ dàng phân tích dữ liệu đầu vào. ⇒ Hoạt động tốt trên nhiều miền dữ liệu lớn.	Hầu hết các trường hợp thực tế thường có các thuộc tính trong các đối tượng phụ thuộc và ảnh hưởng lẫn nhau. ⇒ Đưa kết quả dễ sai sót
Mô hình huấn luyện đơn giản	Dễ thực thi và kiểm tra mô hình.	Phương pháp không tối ưu và chặt chẽ.
Yếu tố khác	Cho phép người xây dựng kết hợp thuật toán của mô hình với tri thức tiền nghiệm (prior knowledge) và dữ liệu quan sát được (observed data) từ vị trí chủ quan để việc thực thi được dễ dàng hơn.	Vấn đề xác suất điều kiện bằng 0 cần được khắc phục.

Chương 4: Khắc phục vấn đề xác suất điều kiện bằng 0

Nếu không có đối tượng nào trong thuộc tính F_i của lớp C_k nhận giá trị v^i_j thì xác suất:

$$P(F_i = v_j^i | C_k) = 0$$

Khi phân lớp, nếu xuất hiện v^i_j trong trường hợp nêu trên, xác suất phân vào lớp C_k luôn bằng 0. Để khắc phục tình trạng này, ta sử dụng cách ước lượng theo công thức:

$$P(F_i = v_j^i | C = C_k) = \frac{n_{ijk} + mp}{n_k + m}$$

- Trong đó:
  - n_{ijk} là số đối tượng thuộc lớp C_k mang thuộc tính – giá trị (F_i, v_j^i)
  - n_k là số đối tượng thuộc lớp C_k
  - p là hằng số (có thể p = 1 hoặc $1/P_i$ là số giá trị thuộc tính F_i có thể nhận)
  - m là số lượng đối tượng "ảo" ($m \geq 1$)
- Giải thích:
  - Công thức xác suất: $P(F_i = v^i_j \mid C = C_k) = \frac{n_{ijk}}{n_k}$
  - Nhưng trong trường hợp nijk = 0 nêu trên, thì $P(F_i = v^i_j \mid C = C_k) = 0$
  - Vì vậy, ta cần thêm hằng số p và số lượng đối tượng "ảo" m để biểu thức luôn khác 0

Nhờ việc sử dụng phương pháp này mà vấn đề xác suất điều kiện bằng 0 được khắc phục.

Chương 5: Ứng dụng

5.1. Phân loại loại cây hoa màu

Mặc dù sử dụng giả thiết không thực tế về sự độc lập giữa các đặc trưng, các bộ phân loại Naive Bayes thường được áp dụng trong các bài toán phân loại thực vật. Phần sau đây sẽ minh họa ứng dụng này với bài toán phân loại hoa Iris.

5.1.1. Định nghĩa bài toán

Bài toán phân loại hoa Iris là một bài toán kinh điển trong lĩnh vực học máy dành cho người mới bắt đầu. Trong tự nhiên, hoa Iris có 3 loại: setosa, versicolor và virginica. Mỗi loại có đặc điểm riêng, được mô tả bằng 4 đặc trưng: Chiều dài đài hoa (Sepal length), Chiều rộng đài hoa (Sepal width), Chiều dài cánh hoa (Petal length) và Chiều rộng cánh hoa (Petal width) (tất cả đều tính bằng cm). Trong bài toán này, mô hình nhận vào một vector 4 số tương ứng với 4 đặc trưng trên, và phải dự đoán vector số đầu vào đó thuộc về loài hoa Iris nào.

5.1.2 Minh họa thuật toán

Để dễ hiểu, đầu tiên ta minh họa thuật toán cho một dataset nhỏ.

Để ngắn gọn, ta viết tắt "Chiều dài đài hoa" là SL, "Chiều rộng đài hoa" là SW, "Chiều dài cách hoa" là PL, "Chiều rộng cánh hoa" là PW.

Giả sử ta có training data là một dataset [12] nhỏ gồm 16 sample như sau:

SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
5.4	3.9	1.7	0.4	Iris-setosa
6.1	3.0	4.6	1.4	Iris-versicolor
6.2	2.2	4.5	1.5	Iris-versicolor
7.6	3.0	6.6	2.1	Iris-virginica
7.7	2.8	6.7	2.0	Iris-virginica
6.1	2.6	5.6	1.4	Iris-virginica
5.7	2.8	4.1	1.3	Iris-versicolor
6.4	2.9	4.3	1.3	Iris-versicolor
6.7	3.3	5.7	2.5	Iris-virginica
5.6	3.0	4.1	1.3	Iris-versicolor
6.4	2.8	5.6	2.1	Iris-virginica
4.8	3.4	1.6	0.2	Iris-setosa
5.0	2.0	3.5	1.0	Iris-versicolor
6.4	2.7	5.3	1.9	Iris-virginica
6.4	3.2	4.5	1.5	Iris-versicolor
4.7	3.2	1.6	0.2	Iris-setosa

Hình 2. Training data nhỏ gồm 16 sample

Và dựa vào đó, ta phải dự đoán xem vector đặc trưng sau thuộc loài hoa Iris nào:

$$v = (SL = 6, SW = 2.2, PL = 5, PW = 1.5)$$

Ta phải tìm giá trị cao nhất trong $P(setosa \mid v), P(versicolor \mid v)$, và $P(virginica \mid v)$. Vì các thông số thu được ở đây là các số thực liên tục, ta sẽ sử dụng Gaussian Naive Bayes.

Để làm ví dụ, ta sẽ chỉ tính giá trị $P(verginica \mid v)$, vì hai giá trị kia được tính tượng tự. Thay vì tính chính xác xác suất đó là bao nhiêu, ta chỉ cần giá trị sau:

$$P(v \mid virginica) * P(virginica)$$

Trong dataset có 6 trên 16 sample là loài virginica, nên

$$P(virginica) = 6/16 .$$

Vì ta giả sử các đặc trưng đều độc lập, nên ta có:

$$\begin{aligned} P(v|virginica) &= P(SL = 6, SW = 2.2, PL = 5, PW = 1.5|virginica) \\ &= P(SL = 6|virginica) * P(SW = 2.2|virginica) \\ &* P(PL = 5|virginica) * P(PW = 1.5|virginica) \end{aligned}$$

Khi nhìn vào các thông số SL của các cây virginica trong database, ta thu được giá trị trung bình và độ lệch chuẩn [13] của nó là:

$$\mu_{virginica,SL} = 6.8, \sigma_{virginica,SL} = 0.67$$

Vậy, sử dụng công thức xác suất của phân phối Gauss, ta có:

$$P(SL = 6|virginica) = \frac{1}{\sigma_{virginica,SL}\sqrt{2\pi}} e^{\frac{-1}{2}(\frac{6-\mu_{virginica,SL}}{\sigma_{virginica,SL}})^2}$$

$$= \frac{1}{6.8\sqrt{2\pi}} e^{\frac{-1}{2}(\frac{6-6.8}{0.67})^2} = 0.29$$

Tương tự, ta tính được:

$$P(SW = 2.2|virginica) = 0.044, P(PL = 5|virginica) = 0.21, \text{ và}$$
$$P(PW = 1.5|virginica) = 0.422.$$

Nhân các xác suất trên với nhau, ta có:

$$P(SL = 6, SW = 2.2, PL = 5, PW = 1.5|virginica) * P(virginica)$$
$$= 0.29 * 0.044 * 0.42 * 0.422 * (6/16) = 4.24 * 10^{-4}$$

Làm lại với hai loài hoa còn lại, ta tính được:

$$P(SL = 6, SW = 2.2, PL = 5, PW = 1.5|setosa) * P(setosa) = 0$$
$$P(SL = 6, SW = 2.2, PL = 5, PW = 1.5|versicolor) * P(versicolor) = 0.028$$

Vì 0.028 là giá trị cao nhất, nên model này sẽ dự đoán sample này thuộc về loài versicolor.

Dưới đây là code tham khảo cho bài toán trên:

```
# Train with training data, then give predictions on the testing data
def naive_bayes(X_train, y_train, X_test):
    summarize = summarize_by_class(X_train, y_train)
    predictions = list()
    for row in X_test:
        output = predict(summarize, row)
        predictions.append(output)
    return(predictions)
```

Hình 3: Train model và đưa dự đoán cho testing data

```
# Run the algorithm and print its accuracy with this training and testing data
predictions = naive_bayes(X_train, y_train, X_test)
correctPred = 0
for i in range(len(predictions)):
    correctPred += predictions[i] == y_test[i]

accuracy = correctPred / len(X_test)
print(f'Accuracy: {accuracy}')
```

Hình 4: Tính độ chính xác của các dự đoán

5.2. Phân lớp văn bản

5.2.1. Định nghĩa bài toán

Phân lớp văn bản (Text classification) là một ứng dụng quan trọng của máy học, trong đó mô hình nhận input là một văn bản và trả lời nó thuộc lớp/thể loại nào. Ví dụ: phân loại văn bản thuộc thể loại cổ tích, truyện trinh thám, thơ ca; hay phân loại theo phương thức biểu đạt như tự sự, miêu tả, biểu cảm.

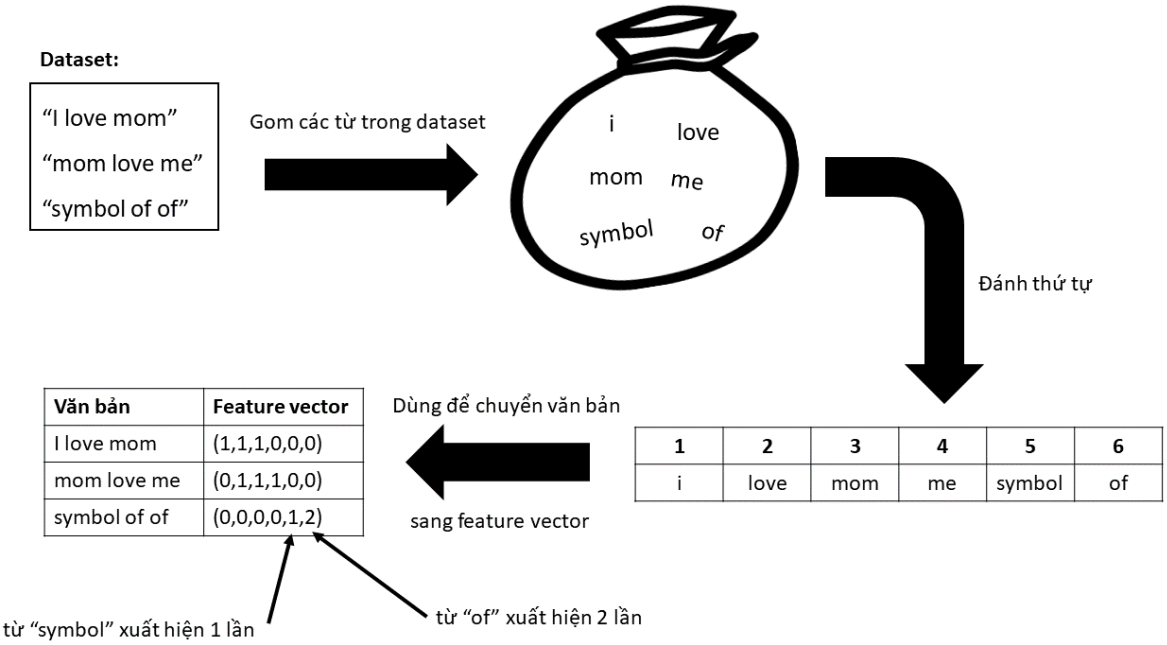
5.2.2. Giải thích thuật toán

Đầu tiên, ta phải biểu diễn văn bản dưới dạng vector đặc trưng. Một cách phổ biến là lưu lại tần số mỗi từ. Ví dự với câu "My parents love me and I love my parents back", thì có tần số các từ là:

and	back	i	love	me	my	parents
1	1	1	2	1	2	2

Suy ra, vector đặc trưng của nó là (1,1,1,2,1,2,2). Ở vị trí "and" có số 1 vì từ "and" xuất hiện 1 lần, ở vị trí "love" có số 2 vì từ "love" xuất hiện 2 lần, các từ còn lại tương tự.

Để tạo vector đặc trưng, trước tiên ta xác định tập từ vựng gồm tất cả các từ xuất hiện trong training data. Sau đó đánh số thứ tự từ 1 đến N với N là số từ trong tập từ vựng. Cuối cùng, mỗi văn bản sẽ được biểu diễn bằng một vector N chiều, trong đó mỗi chiều tương ứng với một từ và mang giá trị là tần suất xuất hiện của từ đó. Ví dụ như với training data có ba văn bản: "I love Mom", "mom love me", "symbol of", thì quá trình đưa về vector đặc trưng của ba văn bản được thể hiện qua sơ đồ sau:



Hình 5. Quá trình đưa ba văn bản ví dụ về vector đặc trưng

Cách này được gọi là "túi từ" vì nó bỏ qua thứ tự các từ trong văn bản. Vì thế, văn bản có cùng tần suất từ nhưng thứ tự từ khác nhau sẽ có cùng một vector đặc trưng. Ví dụ: Văn bản “my mom loves the cat” và “the cat loves my mom” có vector đặc trưng giống nhau vì có tần số các từ giống nhau, nên được sẽ ra output giống nhau khi input vào mô hình.

Dưới đây là code mẫu cho ví dụ trên:

```
from sklearn.feature_extraction.text import CountVectorizer

count_vector = CountVectorizer(stop_words='english', max_df=100)
# default lowercase = True, analyser = True, ngram_range=(1,1)

count_vector.fit(X_train)
X_train_count_vector = count_vector.transform(X_train)
X_test_count_vector = count_vector.transform(X_test)

# Prints [(number of training samples, number of unique words), (number of testing samples, number of unique words)]
print([X_train_count_vector.shape, X_test_count_vector.shape])
```

Hình 6. Khởi tạo các vector đặc trưng

```
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import Pipeline
from sklearn.metrics import accuracy_score

mnb = MultinomialNB()

# Built-in function for the Multinomial Naive Bayes model where the feature vector of a text is its frequency vector
pipeline = Pipeline([('count_vector', count_vector), ('mnb', MultinomialNB())])

# Fit the model to the training data
pipeline.fit(X_train, y_train)

# Give predictions of samples from testing data
pred = pipeline.predict(X_test)

# Accuracy: (data correctly predicted) / (total testing data)
accuracy_score(y_test, pred)
```

Hình 7. Ứng dụng Multinomial Naive Bayes vào giải bài toán ví dụ

Notebook này giải bài toán phân loại các đoạn văn bản thuộc 1 trong 3 lĩnh vực là Hóa học (Chemistry), Vật lý (Physics) và Sinh học (Biology) sử dụng model Multinomial Naive Bayes được tích hợp sẵn vào thư viện sklearn.

5.3. Các ứng dụng khác

Ngoài hai ví dụ trên, Naive Bayes còn có các ứng dụng khác như:

- Lọc spam: cho một email, cho biết nó có phải là spam không
- Phân tích cảm xúc: đánh giá cảm xúc (như vui, giận, hứng thú, không hứng thú) của một đoạn văn bản
- Hệ thống gợi ý (Recommending system)

Chú thích

[1] Xác suất có điều kiện (Conditional probability) là xác suất của một biến cố A nào đó, biết rằng một biến cố B khác xảy ra. Ký hiệu P(A|B), và đọc là "xác suất của A, biết B". Tham khảo thêm: vi.wikipedia.org/wiki/Xác_suất_có_điều_kiện

[2] Xác suất tiên nghiệm (Prior probability), xác suất biết trước, là xác suất khách quan tính được dựa trên kiến thức và suy luận logic, thay vì quan sát và thực nghiệm. Tham khảo thêm: knowledge.sapp.edu.vn/knowledge/tóm-tắt-các-kiến-thức-quan-trong-của-reading-8-probability-concepts

[3] Likelihood là biểu diễn xác suất tổng hợp dưới dạng giá trị một hàm số có tham số là tham số của các hàm thống kê. Tham khảo thêm: https://en.wikipedia.org/wiki/Likelihood_function

[4] Phân hoạch (phân hoạch tập hợp) là một cách nhóm các phần tử của nó thành các tập con không rỗng, theo cách mà mỗi phần tử được chứa trong chính xác một tập hợp con. Tham khảo thêm: vi.wikipedia.org/wiki/Phân_hoạch_tập_hợp

[5] Maximum A Posteriori (MAP) là phương pháp sử dụng những kinh nghiệm trước đây để suy ra các khoảng giá trị và phân bố của tham số. Tham khảo thêm: machinelearningcoban.com/2017/07/17/mlemap/#-maximum-a-posteriori

[6] Phân phối Gaussian: Biến ngẫu nhiên liên tục X được gọi là có phân phối chuẩn hay phân phối Gaussian, ký hiệu $X \sim N(\mu, \sigma^2)$, khi hàm mật độ xác suất có dạng sau:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

[7] Scikit-learn. Tham khảo thêm tại: scikit-learn.org/stable

[8] Ngôn ngữ lập trình Python: python.org

[9] Thí nghiệm đa thức là một thí nghiệm thống kê bao gồm n lần thử. Mỗi lần thử sẽ có một số rời rạc kết cục có thể xảy ra. Trong bất kỳ lần thí nghiệm nào, xác suất xuất hiện một kết cục cụ thể là hằng số. Tham khảo thêm: vimentor.com/vi/lesson/phan-phoi-da-thuc

[10] Bài toán phân loại văn bản là bài toán sử dụng trí tuệ nhân tạo để gán các nhãn lớp hoặc loại lên các văn bản dựa trên các đặc trưng của từng văn bản. Tham khảo thêm: paperswithcode.com/task/document-classification

[11] Maximum likelihood estimation (Ước lượng Xác suất Tối đa) là một phương pháp xác định giá trị cho các tham số của một mô hình xác suất thống kê. Tham khảo thêm: towardsdatascience.com/probability-concepts-explained-maximum-likelihood-estimation-c7b4342fdbb1

[12] Toàn bộ dataset về hoa Iris: https://www.kaggle.com/datasets/jillanisofttech/iris-dataset-uci

[13] Độ lệch chuẩn: https://vi.wikipedia.org/wiki/%C4%90%E1%BB%99_l%E1%BB%87ch_chu%E1%BA%A9n

Nguồn tham khảo

[1] Thuật toán phân lớp Naive Bayes: https://viblo.asia/p/thuat-toan-phan-lop-naive-bayes-924lJWPm5PM
[2] Mô hình và ứng dụng của phân lớp Naive Bayes: https://viblo.asia/p/phan-lop-voi-navie-bayes-classification-mo-hinh-va-ung-dung-WAyK8PRkKxX
[3] Định lý Bayes: vi.wikipedia.org/wiki/Định_lý_Bayes và "Giáo Trình LÝ THUYẾT XÁC SUẤT VÀ THỐNG KÊ ỨNG DỤNG", Trường Đại học Tài chính Thành phố Hồ Chí Minh
[4] Khắc phục vấn đề xác suất điều kiện bằng 0 trong quá trình tính toán xác suất của thuật toán phân lớp Naive Bayes: viblo.asia/p/thuat-toan-phan-lop-naive-bayes-924lJWPm5PM#_3-khac-phuc-van-de-xac-suat-dieu-kien-bang-zero-4
[5] Ưu điểm và nhược điểm của phân lớp Naive Bayes: viblo.asia/p/thuat-toan-phan-lop-naive-bayes-924lJWPm5PM#_4-uu-diem-5 và viblo.asia/p/thuat-toan-phan-lop-naive-bayes-924lJWPm5PM#_5-nhuoc-diem-6
[6] Cài đặt từ đầu model dự đoán hoa Iris: https://machinelearningmastery.com/naive-bayes-classifier-scratch-python/
[7] Mô hình túi từ: https://machinelearningmastery.com/gentle-introduction-bag-words-model/