**1.Naive Bayes**

**1.1 Khái niệm**

Naive Bayes là thuật toán học máy có giám sát sử dụng cho các bài toán phân loại để tìm xác suất.

Ý tưởng chính đằng sau bộ phân loại Naive Bayes là sử dụng định lý Bayes để phân loại dữ liệu dựa trên xác suất của các lớp khác nhau với các đặc trưng của dữ liệu.

* Naive Bayes Classifier là một mô hình phân loại xác suất đơn giản và có rất ít tham số được sử dụng để xây dựng các mô hình Machine Learning có thể dự đoán với tốc độ nhanh hơn với các thuật toán phân loại khác.
* Nó là một bộ phân loại xác suất vì nó giả định rằng một đặc điểm trong mô hình độc lập với sự tồn tại của một đặc điểm khác. Hay nói cách khác mỗi đặc điểm để đưa ra dự đoán không có mối quan hệ nào với nhau
* Thuật toán Naive Bayes được sử dụng trong lọc thư rác, phân tích tình cảm, phân loại bài viết,…

Cho một dataset miêu tả điều kiện thời tiết ảnh hưởng đến quyết định đi chơi golf hay không:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Outlook** | **Temperature** | **Humidity** | **Windy** | **PlayGolf** |
| 0 | Rainy | Hot | High | False | No |
| 1 | Rainy | Hot | High | True | No |
| 2 | Overcast | Hot | High | False | Yes |
| 3 | Sunny | Mild | High | False | Yes |
| 4 | Sunny | Cool | Normal | False | Yes |
| 5 | Sunny | Cool | Normal | True | No |
| 6 | Overcast | Cool | Normal | True | Yes |
| 7 | Rainy | Mild | High | False | No |
| 8 | Rainy | Cool | Normal | False | Yes |
| 9 | Sunny | Mild | Normal | False | Yes |
| 10 | Rainy | Mild | Normal | True | Yes |
| 11 | Overcast | Mild | High | True | Yes |
| 12 | Overcast | Hot | Normal | False | Yes |
| 13 | Sunny | Mild | High | True | No |

Ma trận đặc trưng chứa tất cả các vector (hàng) của tập dữ liệu, trong đó mỗi vector bao gồm giá trị của các đặc trưng độc lập. Trong tập dữ liệu trên, các đặc trưng là ‘Outlook’ (Thời tiết), ‘Temperature’ (Nhiệt độ), ‘Humidity’ (Độ ẩm) và ‘Windy’ (Gió).

Vector phản hồi chứa giá trị của biến lớp (biến mục tiêu hoặc đầu ra) cho mỗi hàng trong ma trận đặc trưng. Trong tập dữ liệu trên, biến lớp có tên là ‘Play golf’ (Chơi golf hay không).

**Giả định Naive Bayes:**

* **Tính độc lập giữa các đặc trưng(Feature independence)**: Khi phân loại một đối tượng, ta giả định rằng mỗi đặc trưng trong dữ liệu không ảnh hưởng đến các đặc trưng khác.
* **Đặc trưng liên tục có phân phối chuẩn(Continuous features are normally distributed):** Nếu một đặc trưng là liên tục, thì nó được giả định là có phân phối chuẩn(Gaussian distribution) trong mỗi lớp.
* **Đặc trưng rời rạc có phân phối đa thức(Discrete features have multinomial distributions):** Nếu một đặc trưng là rời rác, thì nó được giả định có phân phối đa thức(Multinomial distribution) trong mỗi lớp.
* **Tất cả mọi đặc trưng đều có tầm quan trọng như nhau(Features are equally important)**: Mọi đặc trưng đều có đóng góp như nhau trong việc dữ đoán mỗi lớp.
* **Không có dữ liệu bị thiếu(No missing data)**: Dữ liệu không được chứa bất kỳ giá trị bị thiếu nào.

**Định lý Bayes đối với naive bayes**

Định lý bayes tìm xác suất của một sự kiện xảy ra khi biết xác suất của sự kiện khác đã xảy ra:

**Xác suất hậu nghiệm (Posterior probability)**: Xác suất của giả thuyết A khi đã quan sát sự kiện B.

**Xác suất khả năng (Likelihood probability)**: Xác suất của bằng chứng B khi giả thuyết A đúng.

P(A): **Xác suất tiên nghiệm (Prior probability)**: Xác suất ban đầu của giả thuyết A.

P(B): **Xác suất của bằng chứng (Evidence probability):** Xác suất của sự kiện B xảy ra.

Giả sử ta có tập dữ liệu có đặc trưng ta có thể áp dụng định lý Bayes:

Xác suất của lớp y khi đã biết đặc trưng X

Xác suất của dữ liệu X xuất hiện nếu biết lớp y

P(y): Xác suất tiên nghiệm của lớp y.

P(X): Xác suất của dữ liệu X xảy ra trong toàn bộ tập dữ liệu.

Ví dụ minh họa: Ta muốn dự đoán một email có phải là spam hay không dựa trên từ ngữ trong email.

* A: Email là spam
* B: Email có chứa từ “khuyến mãi”
* P(A) = 0.3 (30% email là spam)
* P(B|A) = 80%(Trong 80% email spam có chứa từ khuyến mãi)
* P(B) = 0.4(40% Email có chứa từ “Khuyến mãi)

Có 60% khả năng email chứa từ “Khuyến mãi là spam”.

Quay trở lại với ví dụ về thời tiết: y là biến lớp và X là các vector đặc trưng độc lập:

Nếu ta cần dự đoán cho ngày hôm nay(Sunny, Hot, Normal, False)

Xác suất ngày hôm nay không chơi golf.

Xác suất trời nắng khi không chơi golf.

Xác suất nhiệt độ cao khi không chơi golf.

Xác suất độ ẩm bình thường khi không chơi golf.

Xác suất ngày không có gió khi không chơi golf.

P(No): Xác suất ngày không chơi golf.

Tương tự, ta có:

Trong trường hợp này, mẫu số đều ngang bằng nhau nên ta chỉ cần so sánh tử số.

Nếu A và B là 2 sự kiện độc lập:

Nên ta có:

Hay:

Vì mẫu số không đổi cho một số đầu vào nhất định, nên chúng ta có thể coi:

y là biến lớp, nhãn mà ta muốn dự đoán(yes hay no trong bộ dataset này):

P(y): Xác suất lớp

xác suát có điều kiện

Áp dụng cho bộ today = (Sunny, Hot, Normal, False)

Như vậy dự đoán sẽ là “Yes”

Đối với bên trên, dữ liệu là dữ liệu rời rạc. Đối với dữ liệu liên tục ta dùng phân phối chuẩn Gaussian:

|  |  |
| --- | --- |
| **Giới tính** | **Chiều cao(cm)** |
| Nam | 175 |
| Nam | 180 |
| Nam | 170 |
| Nam | 178 |
| Nam | 177 |
| Nữ | 160 |
| Nữ | 162 |
| Nữ | 158 |
| Nữ | 165 |
| Nữ | 166 |

Trung bình(Kì vọng):

Phương sai:

Đối với Nam

= 11.6

Tương tự, đối với nữ:

Giả sử ta có một người có chiều cao 172 cm, ta cần dự đoán họ thuộc nhóm Nam hay Nữ với công thức phân phối chuẩn:

Tính xác suất: P(172|Nam)

Tính xác suất P(172|Nữ)

Vì P(172|Nam) > P(172|Nữ) nên dự đoán có chiều cao 172cm là Nam.

**1.2 Naive Bayers với NLP**

Công thức của Naive Bayes trong NLP:

Xác suất văn bản thuộc về nhãn y.

Xác suất xuất hiện của nhãn y.

Xác suất một văn bản X xuất hiện khi biết nó thuộc nhãn y, còn gọi lại likelihood.

Xác suất xuất hiện của văn bản X, nhưng nó là hằng số (không phụ thuộc vào nhãn), ta có thể bỏ qua khi so sánh xác suất giữa các nhãn

Giả định Naive

Naive Bayes giả định rằng các từ trong văn bản là độc lập với nhau khi biết nhãn y, tức là:

: .

Ví dụ: Ta có bảng dataset như sau:

|  |  |
| --- | --- |
| **Email** | **Nhãn** |
| "Mua ngay, giảm giá 50%" | Spam |
| "Sản phẩm tốt, miễn phí" | Spam |
| "Gặp bạn lúc 7 giờ tối" | Not spam |
| “Chào bạn, tài liệu đính kèm” | Not spam |

Ta tính xác suất của từ w trong nhãn y theo công thức Laplace:

Kích thước tập từ vựng = 21(21 từ khác nhau, tính cả dấu phẩy):

Để so sánh xác suất của email “Miễn phí sản phẩm tốt” thuộc loại Spam hay Not Spam, ta tính:

Nhãn có xác suất cao hơn sẽ được chọn.

**1.3 Naive bayers cho phân loại spam email**

Khi một tin nhắn mới tới, thuật toán Naive Bayes sẽ phân loại tin nhắn là Spam hoặc Ham dựa trên kết quả dưới đây, với là các từ trong email:

Vì cả 2 đều có chung mẫu số chung nên ta chỉ cần so sánh tử số, có nghĩa là:

* : Thì tin nhắn là spam.
* : Thì tin nhắn là ham.

Xác suất có điều kiện của một từ xuất hiện trong tin nhắn nếu nó thuộc nhóm Spam hoặc Ham được tính:

V: Tổng số từ duy nhất trong tập dữ liệu

Kĩ thuật Laplace Smoothing(Khi +1 ở tử và +V ở mẫu) giúp tránh xác suất bằng 0 nếu một từ chưa xuất hiện ở trong tập dữ liệu.

Do giả định Naive Bayes là các từ xuất hiện độc lập với nhau, nên các xác suất tổng được tính như sau:

**1.4 Các thao tác xử lý dữ liệu trước khi đưa vào mô hình**

Ta cần chuẩn hóa dữ liệu để tránh sai lệch cho mô hình. Các thao tác bao gồm các việc như: Xóa bỏ các khoảng trống dư thừa, loại bỏ những email có dữ liệu trống.

Email thường chứa nhiều dấu câu, kí tự đặc biệt(@,&,!,%,…) nhưng không phải tất cả đều có ý nghĩa, tuy nhiên một số kí tự như $ có thể mang ý nghĩa trong mail spam ví dụ như: “Win $10000”, nên cần kiểm tra trước khi loại bỏ hoàn toàn.

Ví dụ:

**Email gốc:** *"Win $$$ now!!! Click here ->* [*http://spam.com*](http://spam.com)*"*  
**Sau khi xử lý:** *"win now click here http spam com"*

Tiếp theo, ta cần chuyển đổi tất cả các từ thành chữ thường(lowercasing), giúp tránh việc từ viết hoa và viết thường được coi là 2 từ khác nhau.

Các từ dừng(the,is,and,to,in,…) không mang nhiều ý nghĩa trong phân loại spam nên chúng ta có thể loại bỏ chúng.

Ta có thể dùng Stemming hoặc lemmatization để nhóm các dạng khác nhau của cùng một từ với nhau, ví dụ: Running -> run

Ở các email spam thường chứa nhiều liên kết(<https://,http://,www>.), ta nên thay thế các Url bằng một token(Ví dụ: “Url”) để tránh làm tăng số lượng từ không cần thiết. Ví dụ: "Visit our website: https://spam.com" → "visit our website URL"

Các giá trị số trong bài toán phân loại email spam thường không quan trọng, nhưng số thường xuất hiện ở các email spam ("100% free", "win 1000$"). Ta có thể loại bỏ hoàn toàn chúng hoặc thay thế tất cả số bằng một kí tự, chẳng hạn như “digit”.

Phát hiện từ viết tắt và lóng: Một số từ viết tắt phổ biến như: u(you), pls(please), các từ này cần được thay thế để cải thiện độ chính xác của mô hình.

Trong văn bản, có trường hợp những từ xuất hiện quá thường xuyên nhưng không quan trọng, nên ta sử dụng TF-IDF để giảm trọng số của chúng.

Giả sử chúng ta có 3 email:

**Email 1 (Spam)**

*"Win a* ***free*** *iPhone now! Click here to claim your* ***prize****."*

**Email 2 (Spam)**

*"Congratulations! You have won a* ***free*** *vacation trip. Claim your* ***prize*** *now."*

**Email 3 (Ham - Không phải spam)**

*"I got a* ***free*** *coffee at the new café today."*

Đầu tiên, ta tính TF(Term Frequency), tức là số lần một từ xuất hiện trong một văn bản so với tổng số từ trong văn bản đó.

|  |  |  |  |
| --- | --- | --- | --- |
| Từ | Email 1(Spam) | Email 2(Spam) | Email 3(Ham) |
| Free | 0.11 | 0.11 | 0.1 |
| prize | 0.11 | 0.11 | 0 |
| Win | 0.11 | 0 | 0 |
| coffee | 0 | 0 | 0.1 |

Tiếp theo, ta tính DF(Document Frequency), tức là số tài liệu chứa từ đó.

|  |  |
| --- | --- |
| Từ | Xuất hiện trong bao nhiêu email(DF) |
| free | 3 |
| prize | 2 |
| win | 1 |
| coffee | 1 |

IDF được tính bởi công thức:

N: Tổng số email

|  |  |
| --- | --- |
| Từ | IDF |
| free |  |
| prize |  |
| win |  |
| coffee |  |

TF-IDF được tính bằng công thức:

|  |  |  |  |
| --- | --- | --- | --- |
| Từ | Email 1(Spam) | Email 2 (Spam) | Emai3(Ham) |
| free | 0.11 \* 0 = 0 | 0.11 \* 0 = 0 | 0.10 \* 0 = 0 |
| Prize | 0.11 \* 0.18 0.02 | 0.11 \* 0.18 0.02 | 0 |
| win | 0.11 \* 0.48 0.05 | 0 | 0 |
| coffee | 0 | 0 | 0.10 \* 0.48 0.048 |

Từ "free" xuất hiện trong tất cả email nên có IDF = 0, khiến TF-IDF cũng bằng 0. Điều này đúng với nguyên lý của TF-IDF: từ nào có mặt ở mọi văn bản thì không mang giá trị phân loại vì nó không giúp phân biệt giữa spam và ham.

Từ "prize" xuất hiện trong nhiều email spam nhưng không có trong email ham, nên có trọng số TF-IDF nhỏ nhưng không bằng 0 → nó vẫn có ý nghĩa trong việc xác định email spam.

Từ "win" và "coffee" xuất hiện rất ít, có trọng số TF-IDF cao hơn → chúng mang tính đặc trưng và có thể giúp phân biệt email spam (nếu "win" xuất hiện nhiều trong spam) hoặc ham (nếu "coffee" thường thấy trong email hợp lệ).

Ngược lại, những từ chỉ xuất hiện một hoặc hai lần trong toàn bộ tập dữ liệu thường không có giá trị trong việc phân loại.

**1.5 Cân bằng dữ liệu**

Các bộ dataset về email spam thường mất cân bằng, với số lượng email ham thường nhiều hơn spam, ta có thể xử lý để cân bằng dữ liệu. Mục đích của cân bằng dữ liệu là làm cho hai lớp có số lượng mẫu gần bằng nhau

**1.5.1 Undersampling(Giảm dữ liệu của lớp đa số)**

Ta có:

Hoặc:

Ưu điểm: Vì là giảm dữ liệu nên sẽ giúp giảm thời gian huấn luyện mô hình

Nhược điểm: Mất dữ liệu quan trọng, có thể giảm độ chính xác

**1.5.2 Oversampling(Tăng dữ liệu của lớp thiểu số)**

Ta có:

Hoặc:

Phương pháp: Random Oversampling: Sao chép ngẫu nhiên các mẫu trong lớp thiểu số

Ưu điểm: Không gây mất dữ liệu.

Nhược điểm: Dễ gây overfitting vì chỉ lặp lại dữ liệu cũ

**1.5.3 SMOTE(Synthetic Minority Over-Sampling Technique) – Tạo dữ liệu mới cho lớp thiểu số**

Khác với Oversampling, SMOTE tạo dữ liệu giả thay vì sao chép dữ liệu cũ.

Công thức SMOTE:

1. Chọn mẫu thiểu số hiện có

2. Chọn k hàng xóm gần nhất(k-NN)

3. Tạo điểm dữ liệu mới giữa và hàng xóm gần nhất

Với là một số ngẫu nhiên

Ưu điểm: Tạo dữ liệu mới mà không lặp lại dữ liệu.

Nhược điểm: Làm giảm độ đa dạng dữ liệu nếu không chọn đúng số hàng xóm k.

Giả sử chúng ta có 3 email spam với các đặc trưng:

|  |  |  |
| --- | --- | --- |
| ID | Feature 1(Tần suất từ free) | Feature 2(Tần suất từ win) |
| Spam A | 0.2 | 0.7 |
| Spam B | 0.3 | 0.8 |
| Spam C | 0.25 | 0.75 |

Chúng ta chọn Spam A làm điểm gốc.

Ta chọn k = 1. Xét khoảng cách:

Ta thấp spam C có khoảng cách nhỏ nhất đến spam A, nên nếu ta chọn k = 1, Spam C là hàng xóm gần nhất của Spam A.

Ta chọn :

.75-0.7) = 0.725

|  |  |  |
| --- | --- | --- |
| ID | Feature 1 | Feature 2 |
| Synthetic Spam |  | 0.725 |

Khi áp dụng SMOTE vào Naive Bayes, ta phải đảm bảo các giá trị đều hoặc chuẩn hóa lại dữ liệu.