

**ĐẠI HỌC HUẾ**

# **KHOA KỸ THUẬT VÀ CÔNG NGHỆ**

🙠🙟🕮🙝🙢

****

**BÁO CÁO**

**TIỂU LUẬN NHẬP MÔN KHDL**

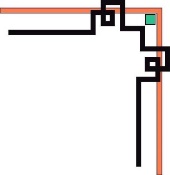
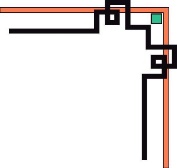
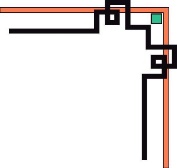
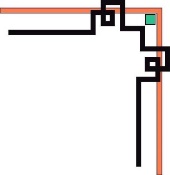
**NĂM HỌC 2020-2021**

**Giảng viên hướng dẫn:**

|  |
| --- |
| Số phách  *(Do hội đồng chấm thi ghi)* |

**Lớp:**

**Thừa Thiên Huế, ngày …tháng…năm.....**



**ĐẠI HỌC HUẾ**

# **KHOA KỸ THUẬT VÀ CÔNG NGHỆ**

🙠🙟🕮🙝🙢

****

(MẪU BÌA PHỤ)

**BÁO CÁO**

**TIỂU LUẬN NHẬP MÔN KHDL**

**NĂM HỌC 2020-2021**

**Giảng viên hướng dẫn: Ts. Trần Văn Hòa**

**Lớp: KHDL & TTNT**

**Sinh viên thực hiện: Lê Thanh Hùng**

*(ký tên và ghi rõ họ tên)*

|  |
| --- |
| Số phách  *(Do hội đồng chấm thi ghi)* |

**Thừa Thiên Huế, ngày …tháng…năm.....**

***Lời cám ơn***

*Trước hết, em xin chân thành gửi lời cám ơn đến các thầy, cô, các cán bộ của khoa Kỹ Thuật Công Nghệ và các cán bộ của Đại Học Huế đã tạo điều kiện thuận lợi cho em trong suốt quá trình học tập. Đặc biệt em xin gửi lời cám ơn đến Ts. Trần Văn Hòa đã giúp đỡ em trong bộ mộn Nhập mô khoa học dữ liệu.*

***Lời mở đầu***

*Định lý Bayes được phát hành vào những năm 1763 nhưng lại phát triễn mạnh mẽ vào thời nay. Đặc biệt với Naive Bayes Classifier thuật toán được suy ra từ định lý Bayes, nó được ứng dụng vào nhiều lĩnh vực và đặc biệt quan trọng trong lĩnh vực trí tuệ nhân tạo. Mô hình phân lớp Naive Bayes là một mô hình giúp giải quyết nhiều vấn đề với thời gian và hiệu quả đáng kể. Bài viết này sẽ giúp bạn hiểu hơn về định lý Bayes và đặc biệt là thuật toán phân lớp Naive Bayes cùng hướng dẫn bạn cách giải một số bài toán với Bayes và lập trình xây dựng mô hình phân lớp Naive Bayes với ngôn ngữ python.*

*Mục Lục*

[Phần 1: Cơ sở lý thuyết Tìm hiểu định luật Bayes. 6](#_Toc77629365)

[1. Định luật Bayes 6](#_Toc77629366)

[2. Suy luận Bayes 7](#_Toc77629367)

[Phần 2: Giới thiệu thuật toán Naive Bayes Classifier Tìm hiểu thuật toán Naive Bayes. 12](#_Toc77629368)

[1. Giới thiệu thuật toán Naive Bayes Classifier 12](#_Toc77629369)

[2. Dạng bài toán Naive Bayes Classifier 13](#_Toc77629370)

[3. Các phân phối thường dùng 16](#_Toc77629371)

[3.1 Gaussian Naive Bayes 16](#_Toc77629372)

[3.2 Multinomial Naive Bayes 16](#_Toc77629373)

[3.3 Bernoilli Naïve Bayes 17](#_Toc77629374)

[Phần 3: Giải quyết bài toán với Naive Bayes Classifier - Coding Sử dụng thuật toán Naive Bayes để giải quyết một số bài toán. 18](#_Toc77629375)

[1. Bài toán Bắc hay Nam 18](#_Toc77629376)

[2. Giới thiệu python và các thư viện 20](#_Toc77629377)

[**2.1** **Thư viện Numpy.** 20](#_Toc77629378)

[**2.2** **Thư viên scikit – learn** 21](#_Toc77629379)

[3. Phân lớp Naive Bayes( Coding) 21](#_Toc77629380)

[**3.1** **Bài toán Bắc hay Nam** 21](#_Toc77629381)

[**3.2** **Bài toán Spam Filtering.** 23](#_Toc77629382)

[Phần 4: Nhận xét về thuật toán Naive Bayes Classifier Các kết luận rút ra được sau khi tìm hiểu về Naive Bayes. 29](#_Toc77629383)

[1. Nhận xét 29](#_Toc77629384)

[2. Tài liệu tham khảo 30](#_Toc77629385)

Phần 1: Cơ sở lý thuyết  
Tìm hiểu định luật Bayes.

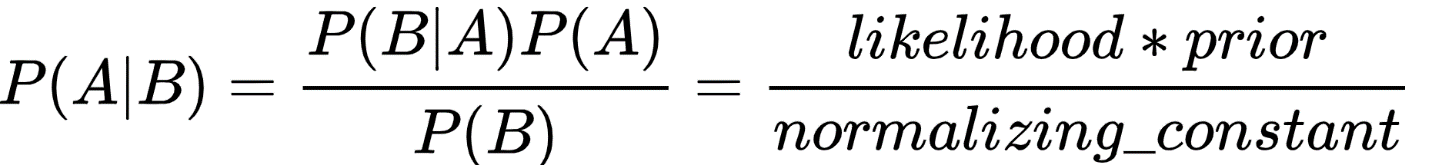
*Tìm hiểu về khái niệm, phát biểu của định luật Bayes và một số ứng dụng của*

## Định luật Bayes

Định luật Bayes được phát biểu như sau:

Định lý 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 được ký hiệu là **P(A|B)**, và đọc là “xác suất của A nếu có B”. Đại lượng này được gọi xác suất có điều kiện hay xác suất hậu nghiệm vì nó được rút ra từ giá trị được cho của B hoặc phụ thuộc vào giá trị đó.

Công thức của định luật Bayes được phát biểu như sau:



Theo định lí 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. Kí hiệu là **P(A)** và đọc là xác suất của A. Đây được gọi là xác suất biên duyên hay xác suất tiên nghiệm, nó là “tiên nghiệm” theo nghĩa rằng 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. Kí hiệu là **P(B)** và đọc là “xác suất của B”. Đại lượng này còn gọi là hằng số chuẩn hóa (normalising 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. Kí hiệu là **P(B|A)** và đọc là “xác suất của B nếu có A”. Đại lượng này gọi là khả năng (likelihood) xảy ra B khi biết A đã xảy ra. Chú ý không nhầm lẫn giữa khả năng xảy ra B khi biết A và xác suất xảy ra A khi biết B

Tóm lại định lý Bayes sẽ giúp ta tính ra xác suất xảy ra của một giả thuyết bằng cách thu thập các bằng chứng nhất quán hoặc không nhất quán với một giả thuyết nào đó. Khi các bằng chứng tích lũy, mức độ tin tưởng vào một giả thuyết thay đổi. Khi có đủ bằng chứng, mức độ tin tưởng này thường trở nên rất cao hoặc rất thấp, tức là xác xuất sảy ra giả thuyết sẽ thay đổi thì các bằng chứng liên quan đến nó thay đổi.

## Suy luận Bayes

* 1. Phát biểu suy luận Bayes

Suy luận Bayes (Bayesian inference) là một kiểu suy luận thống kê mà trong đó các quan sát hay bằng chứng được dùng để cập nhật hoặc suy luận ra xác suất cho việc một giả thuyết có thể là đúng. Cái tên "Bayes" bắt nguồn từ việc sử dụng thường xuyên Định lý Bayes trong quá trình suy luận.

* 1. Ví dụ về suy Bayes

Ví dụ 1: Mặt trời mọc và lặn.



“Mặt trời mọc vào sáng sớm, tối lặn, bao năm qua vẫn thế, tối nay mặt trời lặn. Tôi tin tưởng rằng sáng mai mặt trời sẽ mọc trở lại. Và khó có khả năng nó không mọc trở lại.”

Chúng ta đã chứng kiến việc mặt trời mọc vào hằng sáng và lặn khi đêm về qua nhiều năm. Từ đó, nó tạo cho chúng ta một niềm tin, và chúng ta tin vào nó bởi vì đã quan sát nó qua hàng năm nên. Bởi vậy, xác suất để mọc trời mọc lại vào hôm sau là rất cao và khó có khả năng là nó sẽ không mọc lại.

Ở đây, việc chúng ta quan sát mặt trời mọc và lặn hàng năm theo phát biểu của suy luận Bayes chính là một kiểu suy luận thống kê bằng việc quan sát và được cập nhập hằng ngày.

Từ đó suy ra xác suất mặt trời mọc lại vào hôm sau là đúng và có khả năng rất cao. Ngược lại khả năng mặt trời không mọc lại vào hôm sau là rất thấp.

A picture containing room, gambling house

Description automatically generatedVí dụ 2: Bánh quy từ hộp nào ?

Giả sử có hai hộp đựng đầy bánh quy. Hộp thứ nhất có 10 chiếc bánh quy sô-cô-la và 30 chiếc bánh quy bơ. Hộp thứ hai đựng mỗi loại bánh 20 chiếc. Bé Khoai chọn ngẫu nhiên một hộp, rồi nhặt đại một chiếc bánh. Ta có thể giả thiết rằng bé Khoai còn rất nhỏ nên không phân biệt hộp này hộp kia, và bé thích tất cả các loại bánh kẹo nên bánh loại nào với bé cũng vậy. Và chiếc bánh mà bé Khoai chọn té ra là một chiếc bánh quy bơ. Vậy khả năng Khoai nhặt chiếc bánh đó từ trong hộp thứ nhất là bao nhiêu?

A toy on a black background

Description automatically generated with low confidenceTóm tắt đề:

30 Bánh quy Bơ

10 Bánh Socola

20 Bánh quy Bơ

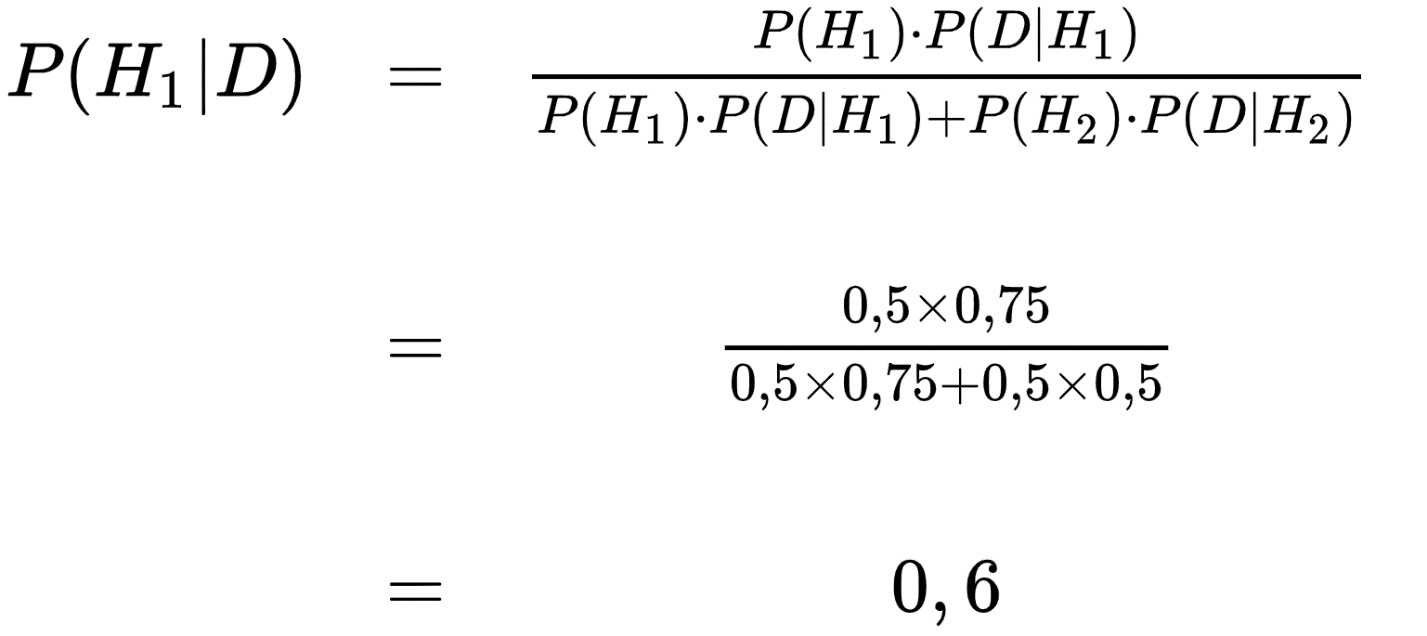
20 bánh Socola

Hộp 1 Hộp 2

Một cách trực quan, có vẻ rõ ràng là câu trả lời phải lớn hơn 1/2, do trong hộp 1 có nhiều bánh quy bơ hơn. Câu trả lời chính xác được tính theo định lý Bayes. Giả sử H1 tương ứng với hộp 1, và H2 tương ứng với hộp 2.

Ta biết rằng đối với bé Khoai, hai hộp là như nhau, do đó, P(H1) = P(H2), và tổng của chúng phải bằng 1, do đó cả hai đều bằng 0,5. Dữ liệu D là quan sát về chiếc bánh quy bơ.

Từ nội dung của hai hộp bánh, ta biết rằng P(D | H1) = 30/40 = 0,75 và P(D | H2) = 20/40 = 0,5. Khi đó, công thức Bayes cho ra kết quả:



* 1. Một số ứng dụng

Một ví dụ ứng dụng trong thế giới thực sẽ là dự báo thời tiết. Naive Bayes là một thuật toán mạnh mẽ để dự báo mô hình dự báo thời tiết. Nhiệt độ của một nơi phụ thuộc vào áp suất tại nơi đó, phần trăm độ ẩm, tốc độ và hướng gió, các ghi chép trước đó về nhiệt độ, nhiễu loạn trên các tầng khí quyển khác nhau và nhiều thứ khác. Vì vậy, khi bạn có một loại dữ liệu nhất định, bạn xử lý chúng một số loại thuật toán nhất định để dự đoán một kết quả cụ thể (hoặc tương lai). Các thuật toán được sử dụng phụ thuộc rất nhiều vào mạng Bayes và định lý.

Chủ nghĩa hình thức mạng Bayes được phát minh để cho phép đại diện hiệu quả và lý luận chặt chẽ với kiến ​​thức không chắc chắn. Cách tiếp cận này phần lớn khắc phục nhiều vấn đề của các hệ thống lý luận xác suất đến những năm 1960 và 70; bây giờ nó thống trị nghiên cứu AI trên các hệ thống lý luận và chuyên gia không chắc chắn. Cách tiếp cận cho phép học hỏi kinh nghiệm và kết hợp tốt nhất giữa AI cổ điển và mạng lưới thần kinh.

Có nhiều ứng dụng khác, đặc biệt là trong khoa học y tế. Giống như dự đoán một bệnh cụ thể dựa trên các triệu chứng và tình trạng thể chất của bệnh nhân. Có nhiều thuật toán hiện đang được sử dụng dựa trên định lý này, như phân loại nhị phân và đa lớp, ví dụ, các bộ lọc thư rác. Có nhiều điều trong chủ đề này.

Trong lĩnh vực, thuật toán Naive Bayes Classifier đang thống trị lĩnh vực này với nhiều ứng dụng như:

* **Real time Prediction:** NBC chạy khá nhanh nên nó thích hợp áp dụng ứng dụng nhiều vào các  ứng dụng chạy thời gian thực, như hệ thống cảnh báo, các hệ thống trading …
* **Multi class Prediction:**Nhờ vào định lý Bayes mở rộng ta có thể ứng dụng vào các loại ứng dụng đa dự đoán, tức là ứng dụng có thể dự đoán nhiều giả thuyết mục tiêu.
* **Text classification/ Spam Filtering/ Sentiment Analysis:**NBC cũng rất thích hợp cho các hệ thống phân loại văn bản hay ngôn ngữ tự nhiên vì tính chính xác của nó lớn hơn các thuật toán khác. Ngoài ra các hệ thống chống thư rác cũng rất ưu chuộng thuật toán này. Và các hệ thống phân tích tâm lý thị trường cũng áp dụng NBC để tiến hành phân tích tâm lý người dùng ưu chuộng hay không ưu chuộng các loại sản phẩm nào từ việc phân tích các thói quen và hành động của khách hàng.
* **Recommendation System:**Naive Bayes Classifier và Collaborative Filtering được sử dụng rất nhiều để xây dựng cả hệ thống gợi ý, ví dụ như xuất hiện các quảng cáo mà người dùng đang quan tâm nhiều nhất từ việc học hỏi thói quen sử dụng internet của người dùng, hoặc như ví dụ đầu bài viết đưa ra gợi ý các bài hát tiếp theo mà có vẻ người dùng sẽ thích trong một ứng dụng nghe nhạc …

Phần 2: Giới thiệu thuật toán Naive Bayes Classifier  
Tìm hiểu thuật toán Naive Bayes.

*Giới thiệu về Naive Bayes Classifier với các dạng toán và những phân phối được dùng trong thuật toán.*

## Giới thiệu thuật toán Naive Bayes Classifier

Naive Bayes là một kỹ thuật đơn giản để xây dựng bộ phân loại: các mô hình gán nhãn lớp cho các cá thể, vấn đề, được biểu diễn dưới dạng vectơ của các giá trị đặc trưng, trong đó nhãn lớp được rút ra từ một số tập hợp hữu hạn. Không có một thuật toán duy nhất để đào tạo các bộ phân loại như vậy, mà là một nhóm các thuật toán dựa trên nguyên tắc chung: tất cả các bộ phân loại Naive Bayes đều giả định rằng giá trị của một đối tượng cụ thể là độc lập với giá trị của bất kỳ đối tượng nào khác, với biến lớp.

Naive Bayes Classifier(NCB) là một giải thuật thuộc lớp thống kê, nó có thể dự đoán xác suất của một phần tử dữ liệu thuộc vào một lớp là bao nhiêu. Naïve Bayes Classifier hay có thể gọi là bộ phân lớp Bayes được dựa trên định lý Bayes.

Trong thống kê, Naive Bayes Classifier là một họ của các "bộ phân loại theo xác suất " đơn giản dựa trên việc áp dụng định lý Bayes với các giả định về tính độc lập(naive) giữa các đối tượng. Chúng là một trong những mô hình mạng Bayes đơn giản nhất, nhưng cùng với ước tính mật độ hạt nhân, chúng có thể đạt được mức độ chính xác cao hơn.

Trong tài liệu thống kê và khoa học máy tính, các mô hình Naive Bayes được biết đến dưới nhiều tên khác nhau, bao gồm Simple Bayes( độc lập) và Independence Bayes( công bằng). Với mối quan hệ với tập dữ liệu của chúng ta, khái niệm này có thể được hiểu là

* Chúng ta giả định rằng không có cặp tính năng nào là phụ thuộc. Ví dụ: nhiệt độ là “Nóng” không liên quan gì đến độ ẩm hoặc triển vọng là “Mưa” không ảnh hưởng đến gió. Do đó, các tính năng được giả định là độc lập.
* Thứ hai, mỗi tính năng có cùng trọng số (hoặc tầm quan trọng). Ví dụ: chỉ biết nhiệt độ và độ ẩm không thể dự đoán chính xác kết quả. Không có thuộc tính nào là không liên quan và được cho là đóng góp như nhau vào kết quả.

Tất cả những cái tên này đề cập đến việc sử dụng định lý Bayes trong quy tắc quyết định của bộ phân loại, nhưng Naive Bayes không (nhất thiết) là một phương pháp Bayes.

Mặc dù giả thiết mà Naive Bayes Classifiers sử dụng là quá phi thực tế, dẫu vậy chúng vẫn hoạt động khá hiệu quả trong nhiều bài toán thực tế, đặc biệt là trong các bài toán phân loại văn bản, ví dụ như lọc tin nhắn rác hay lọc email spam.

Một lợi thế khác của Naive Bayes là nó chỉ yêu cầu một số lượng nhỏ dữ liệu huấn luyện để ước tính các tham số cần thiết cho việc phân loại.

## Dạng bài toán Naive Bayes Classifier

Xét bài toàn phân lớp với C classes 1, 2, …, C. Giả sử có một điểm dữ liệu **x** ∈ Rd. Tính xác suất để điểm dữ liệu này rơi và class c. Ta có:

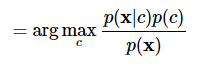
Hay có thể viết:

Nói cách khác là tính xác suất để đầu ra là class c biết đầu vào là x.

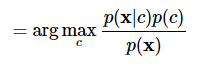
Nếu tính được biểu thức này sẽ giúp chúng ta xác định xác suất để điểm dữ liệu rơi vào mỗi class. Nhờ đó có thể xác định class của điểm dữ liệu đó bằng cách chọn ra class có xác suất cao nhất:



Bởi biểu thức trên khó có thể giải trực tiếp. Thay vào đó, quy tắt Bayes thường đươc sử dụng.

Áp dụng quy tắc Bayes ta có:

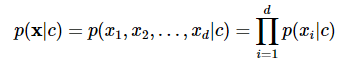


Vì mẫu số  p(x) không phụ thuộc vào c nên ta có :



p(c) có thể được hiểu là xác suất để một điểm rơi vào class c. Giá trị này có thể được tính bằng MLE, nghĩa là tỉ lệ số điểm dữ liệu trong tập training rơi vào class này chi cho tổng số lượng dữ liệu trong tập training, hoặc cũng có thể được đánh giá bằng MAP estimation. Trường hợp thứ nhất thường được sử dụng nhiều hơn.

Thành phần còn lại p(x|c), tức phân phối của các điểm dữ liệu trong class c, thường rất khó tính toán vì x là một biến ngẫu nhiên nhiều chiều, cần rất rất nhiều dữ liệu training để có thể xây dựng được phân phối đó. Để giúp cho việc tính toán được đơn giản, người ta thường giả sử một cách đơn giản nhất rằng các thành phần của biến ngẫu nhiên x là [độc lập với nhau](https://machinelearningcoban.com/2017/07/09/prob/#-independence), nếu biết c (given c). Tức là:

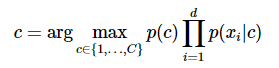


Giả thiết các chiều của dữ liệu độc lập với nhau, nếu biết c, là quá chặt và ít khi tìm được dữ liệu mà các thành phần hoàn toàn độc lập với nhau. Tuy nhiên, giả thiết ngây ngô này lại mang lại những kết quả tốt bất ngờ. Giả thiết về sự độc lập của các chiều dữ liệu này được gọi là Naive Bayes. Cách xác định class của dữ liệu dựa trên giả thiết này có tên là Naive Bayes Classifier.

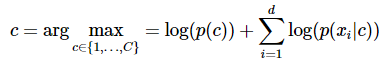
NBC, nhờ vào tính đơn giản một cách ngây thơ, có tốc độ training và test rất nhanh. Việc này giúp nó mang lại hiệu quả cao trong các bài toán large-scale.

Ở bước **training**, các phân phối p(c) và p(xi|c), i =1 ,…, d sẽ được xác định dựa vào training data. Việc xác định các giá trị này có thể dựa vào [Maximum Likelihood Estimation hoặc Maximum A Posteriori](https://machinelearningcoban.com/2017/07/17/mlemap/).

Ở bước **test**, với một điểm dữ liệu mới xx, class của nó sẽ được xác đinh bởi:



Khi d lớn và các xác suất nhỏ, biểu thức ở vế phải của biểu thức trên sẽ là một số rất nhỏ, khi tính toán có thể gặp sai số. Để giải quyết việc này, biểu thức này thường được viết lại dưới dạng tương đương bằng cách lấy log của vế phải:



Việc này không ảnh hưởng tới kết quả vì loglog là một hàm đồng biến trên tập các số dương.

Mặc dù giả thiết mà Naive Bayes Classifiers sử dụng là quá phi thực tế, chúng vẫn hoạt động khá hiệu quả trong nhiều bài toán thực tế, đặc biệt là trong các bài toán phân loại văn bản, ví dụ như lọc tin nhắn rác hay lọc email spam. Trong phần sau của bài viết, chúng ta cùng xây dựng một bộ lọc email spam tiếng Anh đơn giản.

Cả việc training và test của NBC là cực kỳ nhanh khi so với các phương pháp classification phức tạp khác. Việc giả sử các thành phần trong dữ liệu là độc lập với nhau, nếu biết class, khiến cho việc tính toán mỗi phân phối p(xi|c) trở nên cực kỳ nhanh.

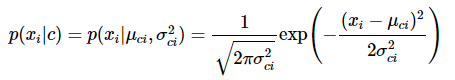
Mỗi giá trị p(c) ,c=1,2,…,C có thể được xác định như là tần suất xuất hiện của class cc trong training data.

## Các phân phối thường dùng

* 1. Gaussian Naive Bayes

Mô hình này được sử dụng chủ yếu trong loại dữ liệu mà các thành phần là các biến liên tục.

Với mỗi chiều dữ liệu **i** và một class **c**, **xi** tuân theo một phân phối chuẩn có kỳ vọng **µci** và phương sai **σ2ci**.



* 1. Multinomial Naive Bayes

Mô hình này chủ yếu được sử dụng trong phân loại văn bản mà feature vectors được tính bằng “Bags of Words”. Lúc này, mỗi văn bản được biểu diễn bởi một vector có độ dài ***d*** chính là số từ trong từ điển. Giá trị của thành phần thứ ***i*** trong mỗi vector chính là số lần từ thứ ***i*** xuất hiện trong văn bản đó.

Khi đó, p(xi|c) tỉ lệ với tần suất thứ i( hay feature thứ i cho trường hợp tổng quát) xuất hiện trong các văn bản của class c. Giá trị này có thể được tính bằng cách.

 (\*)

Trong đó:

* Nci là tổng số lần từ thứ i xuất hiện trong các văn bản của class c, nó được tính là tổng của tất cả các thành phần thứ I của các feature vector ứng với class c.
* Nc là tổng số từ( kể cả lặp) xuất hện trong class c. Nói cách khác, nó bằng tổng độ dài của toàn bộ các văn bản thuộc class c. Có thể suy ra rằng:

 từ đó 

Cách tính này có một số hạn chế là nếu có một từ mới chưa bao giờ xuất hiện trong class c thì biểu thức (\*) sẽ bằng 0. Việc này sẽ dấn dến kết ủa không chính xác.

Để giải quyết vấn đề này, một kỹ thuật được gọi là Laplace smoothing được áp dụng:



Với ***α*** là một số dương, thường bằng 1 , để trành trường hợp tử số bằng 0. Mẫu số được cộng với ***dα*** để đảm bảo tổng của xác suất



Như vậy, mỗi class c sẽ được mô tả bởi bộ các số dương có tổng bằng 1:

* 1. Bernoilli Naïve Bayes

Mô hình này được áp dụng cho các loại dữ liệu mà mỗi thành phần là một giá trị binary – bằng 0 hay 1. Ví dụ: cũng với loại văn bản nhưng thay vì đếm tổng số lần xuất hiện của một từ trong văn bản, ta chỉ cần quan tâm từ đó có xuất hiện hay không.

Khi đó, p(xi|c) được tính bằng:



Với p(i|c) có thể được hiểu là xác suất từ thứ i xuất hiện trong các văn bản class c.

Phần 3: Giải quyết bài toán với Naive Bayes Classifier - Coding  
Sử dụng thuật toán Naive Bayes để giải quyết một số bài toán.

*Giới thiệu về hướng giải quyết các bài toán ứng dụng Naive Bayes Classifier với những phân phối thường dùng và cách code thuật toán với python*

1. Bài toán Bắc hay Nam

|  | **Document** | **Content** | **Class** |
| --- | --- | --- | --- |
| **Training** | d1 | hanoi pho chaolong hanoi | B |
|  | d2 | hanoi buncha pho omai | B |
|  | d3 | pho banhgio omai | B |
|  | d4 | saigon hutiu banhbo pho | N |
| **Test** | d5 | hanoi hanoi buncha hutiu | ? |

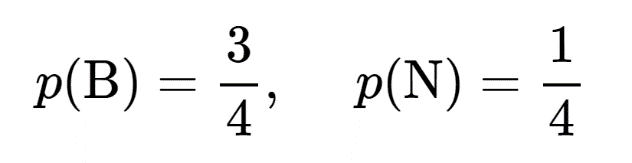
Giả sử trong tập training các văn bản d1, d2, d3, d4 như trong bảng dưới đây.Mỗi văn bản này thuộc vào một trong 2 classes: B(Bắc) hay N(Nam). Hãy xác định class cuẩ văn bản d5.

Ta có thể dự đoán rằng d5 thuộc class B

Bài toán này có thể giải quyết bằng hai mô hình: Multinomial Bayes và Bernouli Naive Bayes. Sau đây sẽ là ví dụ minh họa về mô hình thứ nhất. Việc chọn mô hình phù hợp sẽ tùy thuộc vào mỗi bài toán, chúng ta cũng có thể thử tất cả mô hình để tìm ra mô hình nào là tốt nhất.

Nhận thấy ở đây có hai class là B và N, vậy ta cần tìm p(B) và p(N) dựa trên tần số xuất hiện của mỗi class trong tập dữ liệu training

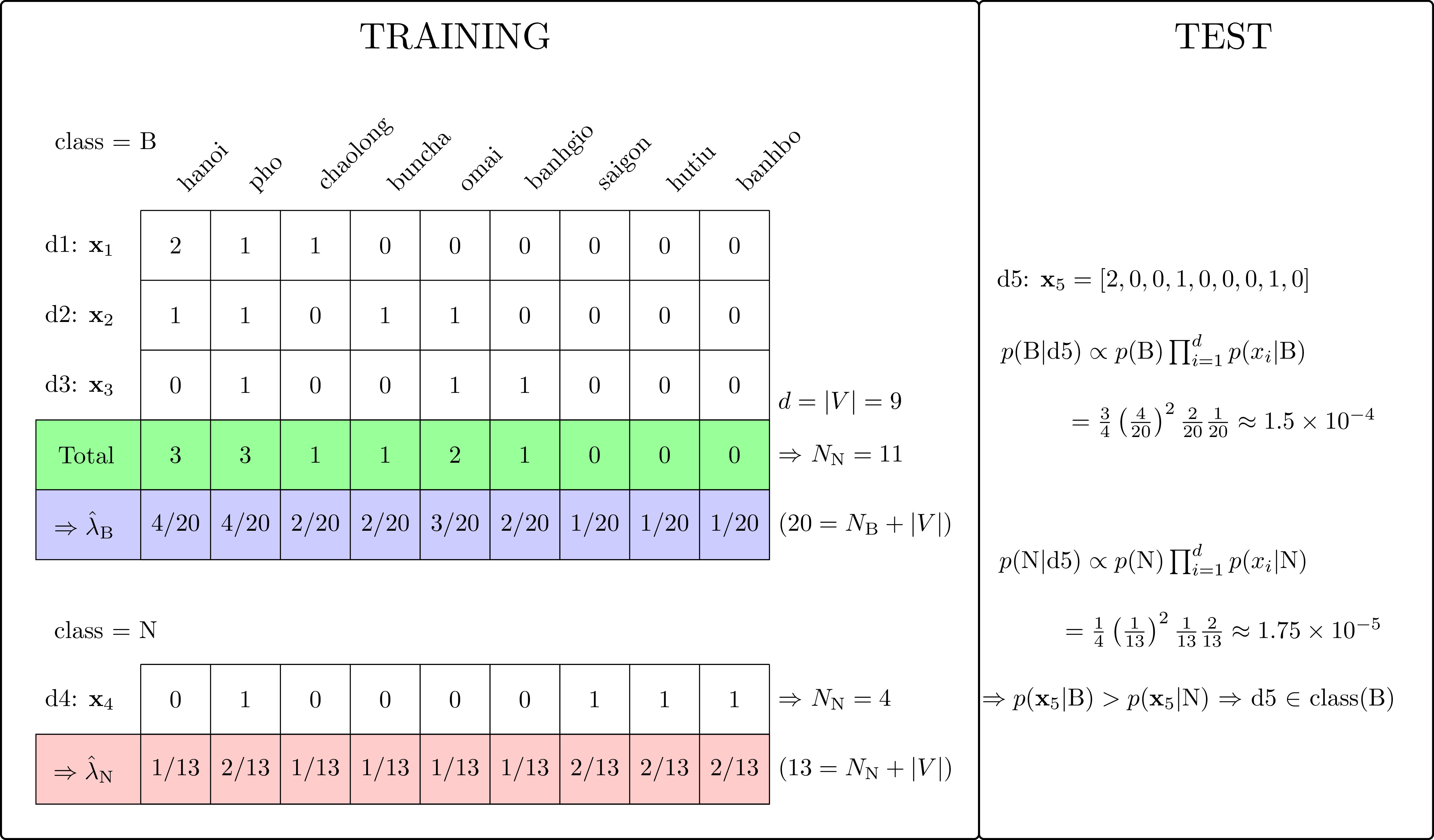
Ta có:



Tập hợp toàn bộ các từ trong văn bản hay còn gọi là từ điển:

V = {Hanoi, pho, chaolong, buncha, omai, banhgio, saigon, hutiu, banhbo} tổng cộng có 9 phần tử |V| = 9.

Dưới đây là hình minh họa quá trình Training và Test cho bài toán này khi sử dụng Multinomial Naive Bayes, trong đó sủ dụng Laplace smoothing với α = 1.



Chú ý : Hai giá trị tìm được 1.5 x 10-4 và 1.75 x 10-5 không phỉa là hai xác suất cần tìm chỉ là hai đại lượng tỉ lệ thuận với hai xác suất đó. Để tính cụ thể, ta có thê làm như sau:



Ta thấy p(B) > p(N) suy ra d5 thuộc class B

Để hiểu hơn về thuật toán, có thể tự tính tiếp với ví dụ khác :

d6 = pho hiutiu banhbo. Kết quả sẽ cho ra d6 thuộc class N.



1. Giới thiệu python và các thư viện

Python là một ngôn ngữ lập trình phổ biến. Nó được tạo ra bởi Guido van Rossum và được phát hành vào năm 1991.

Python là một trong những ngôn ngữ mạnh mẽ trong học máy và được ứng dụng rộng rãi ở nhiều lĩnh vực khác.

Tại sao lại là Python?

* Python hoạt động trên các nền tảng khác nhau (Windows, Mac, Linux, Raspberry Pi, v.v.).
* Python có một cú pháp đơn giản tương tự như ngôn ngữ tiếng Anh.
* Python có cú pháp cho phép các nhà phát triển viết chương trình với ít dòng hơn một số ngôn ngữ lập trình khác.
* Python chạy trên một hệ thống thông dịch, có nghĩa là mã có thể được thực thi ngay sau khi nó được viết. Điều này có nghĩa là việc tạo mẫu có thể rất nhanh chóng.
* Python có thể được xử lý theo một cách thủ tục, một cách hướng đối tượng hoặc một cách chức năng.

Những điểm mạnh của python:

* Phiên bản chính gần đây nhất của Python là Python 3, mà chúng ta sẽ sử dụng trong hướng dẫn này. Tuy nhiên, Python 2, mặc dù không được cập nhật bất kỳ thứ gì khác ngoài các bản cập nhật bảo mật, nhưng vẫn khá phổ biến.
* Trong hướng dẫn này, Python sẽ được viết trong một trình soạn thảo văn bản. Có thể viết Python trong Môi trường phát triển tích hợp, chẳng hạn như Thonny, Pycharm, Netbeans hoặc Eclipse, đặc biệt hữu ích khi quản lý các bộ sưu tập tệp Python lớn hơn.

Trong bài này sẽ tiến hành lập trình với python 3.9.

Sử dụng jupyter để tiến hành viết code.

* 1. **Thư viện Numpy.**

Numpy (Numeric Python): là một thư viện toán học rât phổ biến và mạnh mẽ của Python. NumPy được trang bị các hàm số đã được tối ưu, cho phép làm việc hiệu quả với ma trận và mảng, đặc biệt là dữ liệu ma trận và mảng lớn với tốc độ xử lý nhanh hơn nhiều lần khi chỉ sử dụng Python đơn thuần.

Nếu bạn muốn trở thành một lập trình viên khoa học dữ liệu chuyên sâu, bạn cần phải nắm rõ numpy. Đây là một trong những thư viện hữu ích nhất của python, đặc biệt là nếu bạn đang tìm hiểu về các con số. Vì phần lớn Khoa học Dữ liệu và Máy học xoay quanh Thống kê, nên việc thực hành trở nên quan trọng hơn nhiều.

* 1. **Thư viên scikit – learn**

Scikit-learn ban đầu được đề xuất bởi  David Cournapeau trong một dự án mùa hè của Google vào năm 2007.

Thư viện scikit – learn là một trong những thư viên mạnh mẽ nhất mang các thuật toán học máy trong python. Thư viên scikit – learn tích hợp nhiều thuật toán hiện đại và cổ điển giúp vừa học vừa tiến hành đưa ra các giải pháp hữu ích cho bài toán một cách đơn giản.

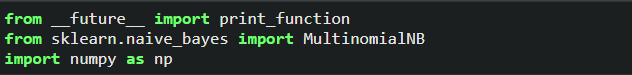
Scikit-learn hỗ trợ mạnh mẽ trong việc xây dựng các sản phẩm. Nghĩa là thư viện này tập trung sâu trong việc xây dựng các yếu tố: dễ sử dụng, dễ code, dễ tham khảo, dễ làm việc, hiệu quả cao.

1. Phân lớp Naive Bayes( Coding)
   1. **Bài toán Bắc hay Nam**

Bài toán Bắc hay Nam đã được giới thiệu ở mục 1. Và bây giờ để hiểu rõ hơn, sau đây sẽ là cách giải quyết bài toán này với thư viện sklearn.

Bước đầu tiên, ta gọi các thư viện cần thiết.

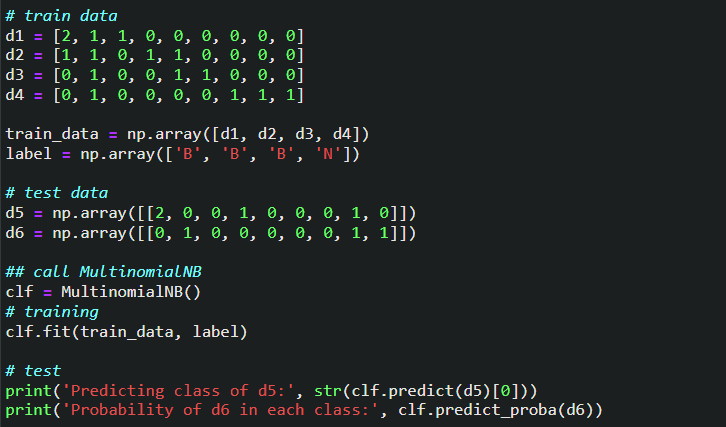
Đầu tiên ta sẽ bắt đầu với phân phối Multinomial



Bởi vì trong bài toán này ta sử dụng phân phối Multinomial nên chỉ cần gọi các thư viên của phân phối này thư sklearn chứ không cần gọi cả thư viện. Việc này giúp chương trình của chúng ta có thể chạy nhanh hơn.

Tiếp theo ta cho các tệp training như bảng ở mục 1 và tiến hành training.

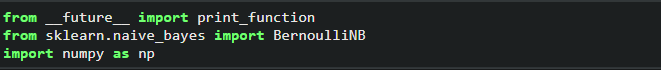
Sau khi train dữ liệu, ta gọi hàm phân phối Multinomial và tiến hành dự đoán xem d5 ở lớp nào và xác suất ở mỗi lớp của d6.



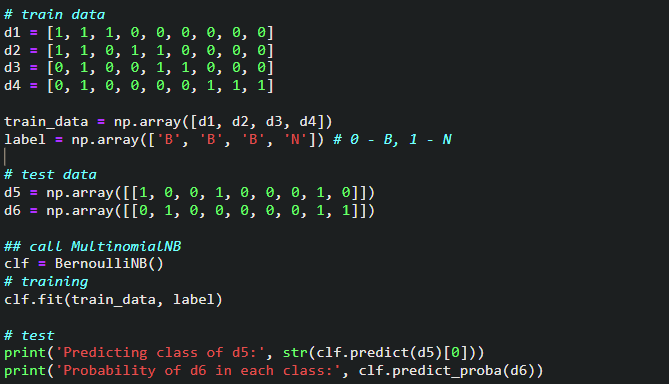
Sau khi chạy, chương trình cho ra kết quả:



Bây giờ, chúng ta sẽ thử sử dụng phân phối Bernoulli cho bài toán này, đối với phân phối Bernoulli ta cần thay đổi một chút về feature vectors. Lúc này, các giá trị khác không đều sẽ đưa về 1 vì ta chỉ quan tâm từ đó có xuất hiện trong văn bản hay không.



Tiếp tục tiếp hành train dữ liệu với phân phối Bernoulli.



Kết quả cho ra:

Ta thấy hai phân phối cho ra kết quả giống nhau( xác suất tìm được không ảnh hưởng đến quyết định cuối cùng).

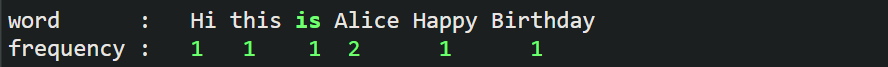
* 1. **Bài toán Spam Filtering.**

Do việc xử lý tiếng việt tương đối khá phức tạp và khó hoàn thiện nên ở phần này sẽ sử dụng email tiếng anh để tiến hành lọc spam.

Các quy tắc xử lý dữ liệu:

* Loại bỏ stop words : những từ xuất hiện thường xuyên như “and”, ”is”, “the”, “this”, “of”….
* Lemmatization: Nhừng từ có gốc được đưa về cùng loại. Ví dụ, “include”, “includes”, “include” đều là những từ chung về “include”. Tất cả các từ cũng được đưaa về dạng ký tự thường(không phải in hoa).
* Loại bỏ non – words: Số dấu câu, ký tự “tabs”, ký tự “xuống dòng.

Sau khi làm sạch mỗi email, ta sẽ xây dựng một ma trận biểu diễn tần suất của mỗi từ. Ví dụ “Hi, this is Alice. Happy Birthday Alice” sau khi được làm sạch sẽ có dạng như sau:



Trong ví dụ này, sẽ sử dụng Multinomail Naïve Bayes.

Để bài toán được đơn giản hơn cho người mới tiếp cận, dữ liệu ở trong bài này là dữ liệu đã được xử lý

Giới thiệu dữ liệu trong bài:

File: “train-features-50.txt” chưa tập dữ liệu thu gọn với 50 training emails

Mỗi file “labels” chứa nhiều dòng, mỗi dòng là một ký tự 0 hoặc 1 thể hiện là email non-spam hoặc spam.



Mỗi file “feature” chưa nhiều dòng, mỗi dòng có 3 số:

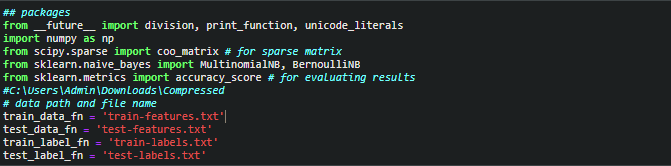


Trong đó số đầu tiên là chỉ số của email, bắt đầu từ số 1, số thứ hai là thứ tự của từ trong từ điển( gồm 2500 từ); số thứ ba là số lường của từ đó trong email đang xét. Dòng đầu tiên nối rằng trong email thứ nhất, từ thứ 8 trong từ điển xuất hiện 1 lần. Cách lưu dữ liệu này giúp tiết kiệm bộ nhớ vì 1 email thường không  chứa hết tất cả các từ trong từ điển mà chỉ chứa một lượng nhỏ, ta chỉ cần lưu các giá trị khác không.

Nếu ta biểu diễn feature vector của mỗi email là một vector hàng có độ dài bằng độ dài từ điển (2500 từ) thì dòng thứ nhất nói rằng thành phần thứ 564 của vector này bằng 1. Tương tự, thành phần thứ 19 của vector này bằng 1. Nếu không xuất hiện, các thành phần khác được mặc định bằng 0.

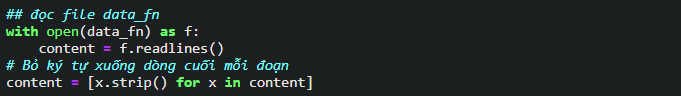
Bây giờ sẽ bắt đầu tiến hành lập trình với Jupyter

Bước đầu tiên luôn là việc gọi tên các thư viện ra. Sau đó, ta gán các biến với là tên của các tệp để tiện khi đọc file.



Hàm số đọc dữ liệu từ file “data\_fn” với labels tương ứng “label\_fn”, số lượng từ trong từ điển là 2 500.

Dữ liệu sẽ được đưa vào một ma trận mà mỗi hàng sẽ thể hiện một email. Ma trận này là một ma trận sparse nên chúng ta sử dụng hàm scipy.sparse.coo\_matrix.



Dùng with open để mở tệp và dùng lệnh readlines( đọc nhiều dòng) để đọc file được truyền vào, tiếp đó ta bỏ đi những ký tự xuống dòng với lệnh strip()

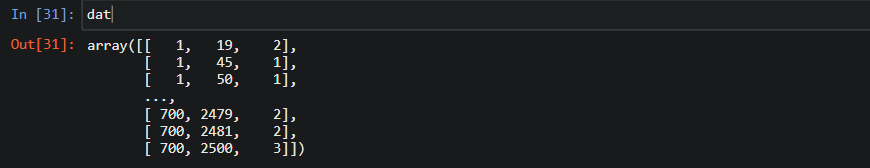
Tiếp theo ta tạo một ma trận rỗng có kích thước tương ứng với file đã đọc, có nghĩa là ma trận gồm 3 cột và có số dòng bằng với số dòng của file.



Tiếp theo ta xóa các ký tự trống( ký tự khoảng cách) và đưa các giá trị trong file vào ma trận. Nói cách khác, ta biểu diễn file dưới dạng ma trận.



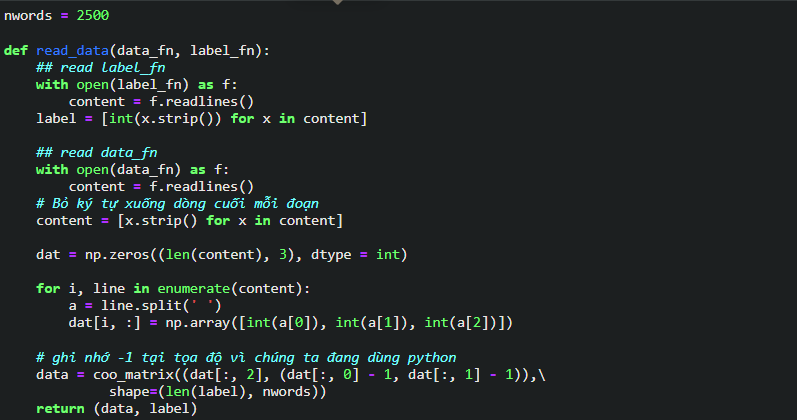
Kết thúc lệnh, dat có dạng:



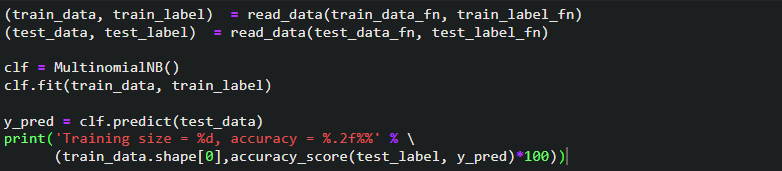
Vì ta sử dụng python để lập trình nên phải tiến hành trừ đi 1.



Tổng hợp tất cả lại, chúng ta có hàm:



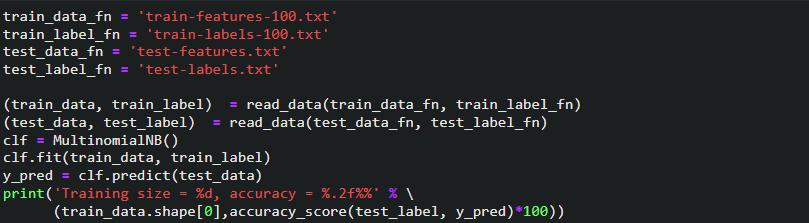
Đọc data và test data, sử dụng class MultinomialMB trong sklearn để tiến hành xây dựng mô hình và dự đoán đầu ra cho test data.



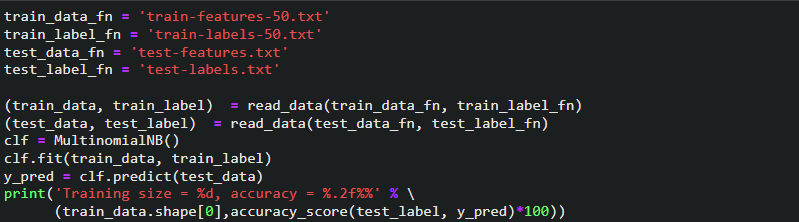
Kết quả cho ra:



Kết quả cho thấy có đến 98.08% các email được phân loại đúng. Chúng ta tiếp tục thử với các bộ dữ liệu training nhỏ hơn:



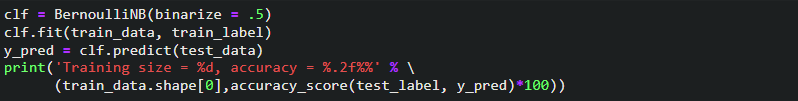






Kết quả cho thấy dù dữ liệu đưa vào training rất nhỏ, chỉ có 50 email nhưng vẫn đạt được kết quả rất ấn tượng.

Bây giờ chúng ta sẽ thử sử dụng mô hình BernoulliNB:





Từ kết quả, ta thấy rằng đối với mô hình phân lớpf Spam Filtering thì MultinomialNB hoạt động hiệu quả hơn.

Phần 4: Nhận xét về thuật toán Naive Bayes Classifier  
Các kết luận rút ra được sau khi tìm hiểu về Naive Bayes.

*Đây là một vài nhận xét mạng tính cả nhận của bản thân về thuật toán Naïve Bayes qua những kiến thức, thông tin mà tôi đã tự tìm hiểu được về Naive Bayes Classifier nói riêng và Bayes nói chung( toàn bộ những gì tôi tìm hiểu được bao gồm cả những điều không được ghi ở trong bài viết này).*

* 1. Nhận xét
* Định lý Bayes được công bố vào những năm 1763 nhưng đến ngày nay nó mới phát triển mạnh mẽ bởi khả năng thu thập thông tin của thời đại công nghệ số.
* Định lý Bayes ra đời tạo trường phái thống kê suy luận giúp giải quyết những vấn đề khó khăn.
* Naive Bayes Classifiers (NBC) thường được sử dụng trong các bài toán Text Classification.
* NBC có thời gian training và test rất nhanh. Điều này có được là do giả sử về tính độc lập giữa các thành phần, nếu biết class.
* Nếu giả sử về tính độc lập được thoả mãn (dựa vào bản chất của dữ liệu), NBC được cho là cho kết quả tốt hơn so với SVM và logistic regression khi có ít dữ liệu training.
* NBC có thể hoạt động với các feature vector mà một phần là liên tục (sử dụng Gaussian Naive Bayes), phần còn lại ở dạng rời rạc (sử dụng Multinomial hoặc Bernoulli).
* Khi sử dụng Multinomial Naive Bayes, Laplace smoothing thường được sử dụng để tránh trường hợp 1 thành phần trong test data chưa xuất hiện ở training data.
  1. Tài liệu tham khảo
* <https://machinelearningcoban.com>
* <https://1upnote.me/post/2018/11/ds-ml-bayes-theorem/>
* <http://hoctructuyen123.net/tong-quan-ve-thuat-toan-phan-lop-naive-bayes-classification-nbc/>
* <https://viblo.asia/p/thuat-toan-phan-lop-naive-bayes-924lJWPm5PM>
* <https://www.python.org/>
* <https://scikit-learn.org/dev/modules/classes.html#module-sklearn.naive_bayes>

Và ngoài ra còn một số trang thông tin khác như Stackover flow, QA stack…

Các trang web được truy cập vào ngày 19.07.21

Dữ liệu được sử dụng:

<http://openclassroom.stanford.edu/MainFolder/courses/MachineLearning/exercises/ex6materials/ex6DataPrepared.zip>

Source Code:

<https://github.com/ThanhHung2112/Naive-Bayes-Classifier>