TRƯỜNG ĐẠI HỌC SÀI GÒN

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO

TÌM HIỂU VỀ THUẬT TOÁN NAIVE-BAYES CLASSIFIERS

Học phần: Khai phá dữ liệu

Giảng viên hướng dẫn: TS.Lê Minh Nhựt Triều

SINH VIÊN THỰC HIỆN

MSSV	**Họ và tên**
3116410033	Lê Thị Thúy Hằng
3116410044	Nguyễn Đức Huy

Tp.Hồ Chí Minh, ngày 25 tháng 6 năm 2020

MỤC LỤC

LỜI MỞ ĐẦU

- Khai phá dữ liệu (data mining) Là quá trình tính toán để tìm ra các mẫu trong các bộ dữ liệu lớn liên quan đến các phương pháp tại giao điểm của máy học, thống kê và các hệ thống cơ sở dữ liệu. Đây là một lĩnh vực liên ngành của khoa học máy tính.. Mục tiêu tổng thể của quá trình khai thác dữ liệu là trích xuất thông tin từ một bộ dữ liệu và chuyển nó thành một cấu trúc dễ hiểu để sử dụng tiếp. Ngoài bước phân tích thô, nó còn liên quan tới cơ sở dữ liệu và các khía cạnh quản lý dữ liệu, xử lý dữ liệu trước, suy xét mô hình và suy luận thống kê, các thước đo thú vị, các cân nhắc phức tạp, xuất kết quả về các cấu trúc được phát hiện, hiện hình hóa và cập nhật trực tuyến. Khai thác dữ liệu là bước phân tích của quá trình "khám phá kiến thức trong cơ sở dữ liệu" hoặc KDD.
- Khai phá dữ liệu là một bước của quá trình khai thác tri thức (Knowledge Discovery Process), bao gồm:
  - Xác định vấn đề và không gian dữ liệu để giải quyết vấn đề (Problem understanding and data understanding).
  - Chuẩn bị dữ liệu (Data preparation), bao gồm các quá trình làm sạch dữ liệu (data cleaning), tích hợp dữ liệu (data integration), chọn dữ liệu (data selection), biến đổi dữ liệu (data transformation).
  - Khai thác dữ liệu (Data mining): xác định nhiệm vụ khai thác dữ liệu và lựa chọn kỹ thuật khai thác dữ liệu. Kết quả cho ta một nguồn tri thức thô.
  - Đánh giá (Evaluation): dựa trên một số tiêu chí tiến hành kiểm tra và lọc nguồn tri thức thu được.
  - Triển khai (Deployment).
- Quá trình khai thác tri thức không chỉ là một quá trình tuần tự từ bước đầu tiên đến bước cuối cùng mà là một quá trình lặp và có quay trở lại các bước đã qua.
- Các phương pháp:
  - Phân loại (Classification): Là phương pháp dự báo, cho phép phân loại một đối tượng vào một hoặc một số lớp cho trước.
  - Hồi qui (Regression): Khám phá chức năng học dự đoán, ánh xạ một mục dữ liệu thành biến dự đoán giá trị thực.
  - Phân nhóm (Clustering): Một nhiệm vụ mô tả phổ biến trong đó người ta tìm cách xác định một tập hợp hữu hạn các cụm để mô tả dữ liệu.
  - Tổng hợp (Summarization): Một nhiệm vụ mô tả bổ sung liên quan đến phương pháp cho việc tìm kiếm một mô tả nhỏ gọn cho một bộ (hoặc tập hợp con) của dữ liệu.

- Mô hình ràng buộc (Dependency modeling): Tìm mô hình cục bộ mô tả các phụ thuộc đáng kể giữa các biến hoặc giữa các giá trị của một tính năng trong tập dữ liệu hoặc trong một phần của tập dữ liệu.
- Dò tìm biến đổi và độ lệch (Change and Deviation Dectection): Khám phá những thay đổi quan trọng nhất trong bộ dữ liệu.

- Vẫn có các mối lo ngại về tính riêng tư gắn với việc khai thác dữ liệu. Ví dụ, nếu một ông chủ có quyền truy xuất vào các hồ sơ y tế, họ có thể loại những người có bệnh tiểu đường hay bệnh tim. Việc loại ra những nhân viên như vậy sẽ cắt giảm chi phí bảo hiểm, nhưng tạo ra các vấn đề về tính hợp pháp và đạo đức.
- Khai thác dữ liệu các tập dữ liệu thương mại hay chính phủ cho các mục đích áp đặt luật pháp và an ninh quốc gia cũng là những mối lo ngại về tính riêng tư đang tăng cao. 5
- Có nhiều cách sử dụng hợp lý với khai thác dữ liệu. Ví dụ, một CSDL các mô tả về thuốc được thực hiện bởi một nhóm người có thể được dùng để tìm kiếm sự kết hợp của các loại thuốc tạo ra các phản ứng (hóa học) khác nhau. Vì việc kết hợp có thể chỉ xảy ra trong một phần 1000 người, một trường hợp đơn lẻ là rất khó phát hiện. Một dự án liên quan đến y tế như vậy có thể giúp giảm số lượng phản ứng của thuốc và có khả năng cứu sống con người. Không may mắn là, vẫn có khả năng lạm dụng đối với một CSDL như vậy.
- Về cơ bản, khai thác dữ liệu đưa ra các thông tin mà sẽ không có sẵn được. Nó phải được chuyển đổi sang một dạng khác để trở nên có nghĩa. Khi dữ liệu thu thập được liên quan đến các cá nhân, thì có nhiều câu hỏi đặt ra liên quan đến tính riêng tư, tính hợp pháp, và đạo đức.

- Naive – Bayes Classifier là một trong những thuật toán thuộc phương pháp phân loại(Classification).
- Naive Bayes - một trong những thuật toán rất tiêu biểu cho hướng phân loại dựa trên lý thuyết xác suất.

- Theo các bài viết về Machine Learning hoặc các tutorial khác về học máy thì có thể thấy được một điều rằng giữa Machine Learning và lý thuyết xác suất có một sự liên hệ rất khăng khít.
- Các phương pháp phân loại dựa trên lý thuyết xác suất về cơ bản có thể hiểu là việc tính xem xác suất một sự việc của chúng ta sẽ xảy ra theo hướng như thế nào. Xác suất của hướng nào càng cao thì khả năng sự việc xảy ra theo hướng đó càng nhiều. Điều này đặc biết có ý nghĩa trong bài toán dự đoán và phân lớp của lĩnh vực Machine Learning.

- Lý thuyết Bayes thì có lẽ không còn quá xa lạ với chúng ta nữa rồi. Nó chính là sự liên hệ giữa các xác suất có điều kiện. Thuật toán Naive Bayes cũng dựa trên việc tính toán các xác suất có điều kiện đó.
- Dưới đây là phần tìm hiểu về thuật toán Naive-Bayes Classifiers của nhóm em.

- Qua học phần này nhóm em có them cho mình sự hiểu biết về các thuật toán máy học đã và đang phát triển, nắm vững hơn các khái niệm về dữ liệu học, hiểu rõ hơn về lĩnh vực Data đặt biệt là Data Mining.
- Nhóm em xin cảm ơn thầy Lê Minh Nhựt Triều đã hướng dẫn nhóm em hiểu rõ tầm quan trọng và nhu cầu thị trường về vị trí nghề nghiệp mà học phần khai phá dữ liệu hướng đến. Thầy giúp cho nhóm em có cái nhìn tổng quan hơn về các kiến thức đã học nói chung cũng như kiến thức học phần Khai phá dữ liệu nói riêng.

I. Giới thiệu thuật toán Naive-Bayes Classifiers

a. Định lý Bayes

- Định lý Bayes là một kết quả của lý thuyết xác suất. Nó đề cập đến phân bố xác suất có điều kiện của biến ngẫu nhiên A, với giả thiết:
  - Thông tin về một biến khác B: phân bố xác suất có điều kiện của B khi biết A.
  - Phân bố xác suất của một mình A.

Phát biểu định lý:

- 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.

- 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. 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. Ký 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. Ký 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.

- Khi biết ba đại lượng này, xác suất của A khi biết B cho bởi công thức:

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

- Từ đó dẫn tới

$$P(A|\mathrm{B})\mathrm{P}(\mathrm{B}) = \mathrm{P}(\mathrm{A} \cap \mathrm{B}) = P(B|\mathrm{A})P(A)$$

Các dạng khác của định lý Bayes

- Định lý Bayes thường cũng thường được viết dưới dạng
  $P(B) = P(A,B) + P(A^C, B) = P(B|A)P(A) + P(B|A^C)P(A^C)$

- Trong đó AC là biến cố bù của biến cố A (thường được gọi là "không A"). Tổng quát hơn, với {Ai} tạo thành một phân hoạch của không gian các biến cố,

$$P(A_i|\mathrm{B}) = \frac{P(B|A_i)P(A_i)}{\sum j\ P\left(B\middle|A_j\right)P(A_j)'}$$

- Với mọi Ai trong phân hoạch.
- Công thức này còn được biết dưới tên công thức xác suất đầy đủ.

Định lý Bayes với hàm mật độ xác suất

- Cũng có một dạng của định lý Bayes cho các phân bố liên tục. Đối với chúng, thay cho các xác suất trong định lý Bayes ta dùng hàm mật độ xác suất. Như vậy ta có các công thức tương tự định nghĩa xác suất điều kiện:

$$f(x|\mathrm{y}) = \frac{f(y|\mathrm{x})\mathrm{f}(\mathrm{x})}{\int_{-\infty}^{\infty} f\left(y\middle|\mathrm{x}'\right)\mathrm{f}(\mathrm{x}')\mathrm{dx}'}$$

- Ý nghĩa của các thành phần trong các công thức trên là 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(y|x) = L(x|y) là (một hàm của x) hàm khả năng của X với điều kiện Y=y, và f(x) và f(y) là các mật độ phân phối của X và Y tách biệt nhau, với f(x) là mật độ phân phối tiền nghiệm của X.

- Điều kiện mặc định trong các công thức là hàm f khả vi và các tích phân công thức tồn tại.

- Ứng dụng của định lý Bayes thường dựa trên một giả thiết có tính triết học Bayesian probability ngầm định rằng độ bất định và kỳ vọng có thể tính toán được giống như là xác suất. Định lý Bayes được đặt theo tên của Reverend Thomas Bayes (1702—1761), người nghiên cứu cách tính một phân bố với tham số là một phân bố nhị phân. Người bạn của ông, Richard Price, chỉnh sửa và giới thiệu công trình năm 1763, sau khi Bayes mất, với tựa đề An Essay towards solving a Problem in the Doctrine of Chances. Pierre-Simon Laplace mở rộng kết quả trong bài luận năm 1774.

b. Bayes Classifiers

- Phân loại Bayes đại diện cho một phương pháp học tập có giám sát cũng như một phương pháp thống kê để phân loại. Giả sử một mô hình xác suất cơ bản và nó cho phép chúng ta nắm bắt sự không chắc chắn về mô hình theo cách nguyên tắc bằng cách xác định xác suất của kết quả. Nó có thể giải quyết các vấn đề chẩn đoán và dự đoán.

- Phân loại này được đặt theo tên của Thomas Bayes (1702-1761), người đề xuất định lý Bayes.

- Phân loại Bayes cung cấp các thuật toán học tập thực tế và kiến thức trước và dữ liệu quan sát có thể được kết hợp. Phân loại Bayes cung cấp một viễn cảnh hữu ích để hiểu và đánh giá nhiều thuật toán học tập. Nó tính toán xác suất rõ ràng cho giả thuyết và nó rất mạnh đối với nhiễu trong dữ liệu đầu vào.

c. Naive-Bayes Classifiers

- Naive – Bayes Classifier là một trong những thuật toán thuộc phương pháp phân loại(Classification).

- Naive Bayes - một trong những thuật toán rất tiêu biểu cho hướng phân loại dựa trên lý thuyết xác suất.

- Naive Bayes là một kỹ thuật đơn giản để xây dựng các trình phân loại: các mô hình gán nhãn lớp cho các trường hợp vấn đề, được biểu diễn dưới dạng vectơ của các giá trị tính năng , trong đó các nhãn lớp được rút ra từ một số tập hữu hạn.

- Không có một thuật toán duy nhất để đào tạo các trình phân loại như vậy, mà là một nhóm các thuật toán dựa trên một nguyên tắc chung: tất cả các trình phân loại Naive-Bayes đều cho rằng giá trị của một tính năng cụ thể là độc lập với giá trị của bất kỳ tính năng nào khác, được đưa ra biến lớp.

- Ví dụ, một quả có thể được coi là một quả táo nếu nó có màu đỏ, tròn và đường kính khoảng 10 cm. Một bộ phân loại Naive-Bayes xem mỗi tính năng này đóng góp độc lập vào xác suất rằng loại quả này là một quả táo, bởi các mối tương quan giữa các tính năng màu sắc, độ tròn và đường kính.

- Đối với một số loại mô hình xác suất, các trình phân loại Naive-Bayes có thể được đào tạo rất hiệu quả trong môi trường học tập có giám sát . Trong nhiều ứng dụng thực tế, ước lượng tham số cho các mô hình Naive-Bayes sử dụng phương pháp khả năng tối đa ; nói cách khác, người ta có thể làm việc với mô hình Naive-Bayes mà không chấp nhận xác suất Bayes hoặc sử dụng bất kỳ phương pháp Bayes nào.

- Mặc dù có thiết kế ngây thơ và những giả định dường như quá đơn giản, các bộ phân loại Naive-Bayes đã hoạt động khá tốt trong nhiều tình huống thực tế phức tạp. Năm 2004, một phân tích về vấn đề phân loại Bayes cho thấy có những lý do hợp lý cho hiệu quả rõ ràng của các phân loại Naive-Bayes. Tuy nhiên, so sánh toàn diện với các thuật toán phân loại khác trong năm 2006 cho thấy phân loại Bayes vượt trội hơn so với các phương pháp khác, chẳng hạn như Gradient tree boosting hoặc Random forest.

- Một lợi thế của Naive-Bayes là nó chỉ cần một số lượng nhỏ dữ liệu đào tạo để ước tính các tham số cần thiết để phân loại.

d. Các mô hình của thuật toán Naive-Beyes

1. Mô hình Bernoulli

Công thức

- Xác suất $P(x_i \mid y)P(x_i|y)$ được tính bằng:

$$P(x_i|y) = P(i|y) \times x_i + (1 - P(i|y)) \times (1 - x_i)$$

- Với $P(i|y)P(i|y)$ là tỉ lệ số lần từ x_i xuất hiện trong toàn bộ tập training data có nhãn y.
- Nhiều tài liệu biểu diễn công thức dưới dạng khác là:

$$P(x_i|y) = P(i|y)^{x_i} \times (1 - P(i|y))^{1-x_i}$$

Hai công thức trên về giá trị toán học là giống nhau.

2. Mô hình Multinomial

Công thức

- Ở mô hình này, các feature vector là các giá trị số tự nhiên mà giá trị thể hiện số lần từ đó xuất hiện trong văn bản.
- Ta tính xác suất từ xuất hiện trong văn bản $P(x_i \mid y)P(x_i \mid y)$ như sau:

$$P(x_i|y) = \frac{N_i}{N_c}$$

- Trong đó:
- N_i là tổng số lần từ x_i xuất hiện trong văn bản.
- N_c là tổng số lần từ của tất cả các từ x_1, ..., $x_n$xuất hiện trong văn bản.

Laplace Smoothing

- Công thức trên có hạn chế là khi từ xi không xuất hiện lần nào trong văn bản, ta sẽ có $N_i = 0$. Điều này làm cho $P(x_i \mid y) = 0$.
- Vế phải của công thức $P(y|x) \propto P(y)\prod_{i=1}^{n} P(x_i|y)$ bằng 0 nếu bất kì một giá trị nào bằng 0 (mặc dù có thể các giá trị khác rất lớn).

- Để khắc phục vấn đề này, người ta sử dụng kỹ thuật gọi là Laplace Smoothing bằng cách cộng thêm vào cả tử và mẫu để giá trị luôn khác 0.

$$P(x_i|y) = \frac{N_i + \alpha}{N_c + d\alpha}$$

Trong đó:

α thường là số dương, bằng 1.

dα được cộng vào mẫu để đảm bảo $\sum_{i=1}^{d} P(x_i|y) = 1$

II. Thuật toán Naive-Bayes Classifiers

a. Định lý Bayes

Gọi A, B là hai biến cố:

Với P(B)>0:

$$P(A|B) = \frac{P(AB)}{P(B)}$$

Suy ra:

$$P(AB) = P(A|B)P(B) = P(B|A)P(A)$$

Công thức Bayes:

$$P(B|A) = \frac{P(AB)}{P(A)} = \frac{P(A|B)P(B)}{P(A)} = \frac{P(A|B)P(B)}{P(AB)+P(A\overline{B})}$$

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

- ***Công thức Beyes tổng quát:***

  Với P(A) > 0 và $\{B_1, B_2, \dots, B_n\}$ là một hệ đầy đủ các biến cố:

  - Tổng xác suất của hệ bằng 1:

    $$\sum_{k=1}^{n} P(B_k) = 1$$

  - Từng đôi xung khắc:

    $$P(B_i \cap B_j) = 0$$

    Khi đó ta có:

    $$P(B_k|A) = \frac{P(A|B_k)P(B_k)}{P(A)} = \frac{P(A|B_k)P(B_k)}{\sum_{i=1}^{n} P(A|B_i)P(B_i)}$$

- Trong đó ta gọi A là một chứng cứ (evidence) (trong bài toán phân lớp A sẽ là một phần tử dữ liệu), B là một giả thiết nào để cho A thuộc về một lớp C nào đó. Trong bài toán phân lớp chúng ta muốn xác định giá trị P(B/A) là xác suất để giả thiết B là đúng với chứng cứ A thuộc vào lớp C với điều kiện ra đã biết các thông tin mô tả A. P(B|A) là một xác suất hậu nghiệm (posterior probability hay posteriori probability) của B với điều kiện A.

- Giả sử tập dữ liệu liệu khách hàng của chúng ta được mô tả bởi các thuộc tính tuổi và thu nhập, và một khách hàng X có tuổi là 25 và thu nhập là 2000$.

Giả sử H là giả thiết khách hàng đõ sẽ mua máy tính, thì P(H|X) phản ánh xác xuất người dùng X sẽ mua máy tính với điều kiện ta biết tuổi và thu nhập của người đó.

- Ngược lại P(H) là xác suất tiền nghiệm (prior probability hay priori probability) của H. Trong ví dụ trên, nó là xác suất một khách hàng sẽ mua máy tính mà không cần biết các thông tin về tuổi hay thu nhập của họ. Hay nói cách khác, xác suất này không phụ thuộc vào yếu tố X. Tương tự, P(X|H) là xác suất của X với điều kiện H (likelihood), nó là một xác suất hậu nghiệm.

- Ví dụ, nó là xác suất người dùng X (có tuổi là 25 và thu nhập là $200) sẽ mua máy tính với điều kiện ta đã biết người đó sẽ mua máy tính. Cuối cùng P(X) là xác suất tiền nghiệm của X. Trong ví dụ trên, nó se là xác xuất một người trong tập dữ liệu sẽ có tuổi 25 và thu nhập $2000.

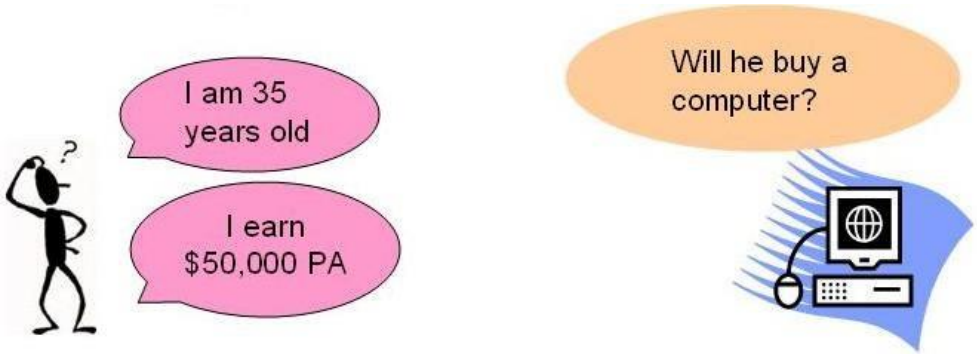
Posterior = Likelihood * Prior / Evidence

Ví dụ:

- Cơ sở dữ liệu khách hàng:

ID	Tuổi	Thu nhập	Sinh viên	Đánh giá tín dụng	Mua máy tính
1	<=30	High	No	Fair	No
2	<=30	High	No	Excellent	No
3	31…40	High	No	Fair	Yes
4	>40	Medium	No	Fair	Yes
5	>40	Low	Yes	Fair	Yes
6	>40	Low	Yes	Excellent	No
7	31…40	Low	Yes	Excellent	Yes
8	<=30	Medum	No	Fair	Yes
9	<=30	Low	Yes	Fair	Yes
10	>40	Medium	Yes	Fair	Yes
11	<=30	Medium	Yes	Excellent	Yes
12	31…40	Medium	No	Excellent	Yes
13	31…40	High	Yes	Fair	Yes
14	>40	Medium	No	Excellent	No

Table 1



B1: Định lý Bayes:

$$P(h/D) = \frac{P(D/h)\,P(h)}{P(D)}$$

- P (h): Xác suất trước của giả thuyết h
- P (D): Xác suất trước của dữ liệu huấn luyện D
- P (h / D): Xác suất của h cho D
- P (D / h): Xác suất của D cho h

B2: Áp dụng lý thuyết:

- D: khách hàng 35 tuổi với thu nhập 50.000 đô la PA
- h: Giả thuyết rằng khách hàng sẽ mua máy tính.
- P (h / D): Xác suất khách hàng D sẽ mua máy tính của chúng tôi khi chúng tôi biết tuổi và thu nhập của anh ấy
- P (h): Xác suất mà bất kỳ khách hàng nào sẽ mua máy tính của chúng tôi bất kể tuổi tác (Xác suất trước) P (D / h): Xác suất khách hàng đó 35 tuổi và kiếm được 50.000 đô la, cho rằng anh ta đã mua máy tính của chúng tôi (Xác suất sau).
- P (D): Xác suất một người trong nhóm khách hàng của chúng tôi là 35 tuổi và kiếm được 50.000 đô la.

B3: Giả thuyết tối đa Posteriori (MAP)

- h1: Khách hàng mua máy tính = Có h2: Khách hàng mua máy tính = Không
- trong đó h1 và h2 là tập hợp con của không gian giả thuyết của 'h'
- P (h / D) (Kết quả cuối cùng) = arg max {P (D / h1) P (h1), P (D / h2) P (h2)} P (D) có thể bị bỏ qua vì nó giống nhau cho cả hai các điều khoản

- Nói chung, chúng tôi muốn giả thuyết có thể xảy ra nhất với dữ liệu huấn luyện hMAP = arg max P (h / D) (trong đó h thuộc về H và H là không gian giả thuyết)

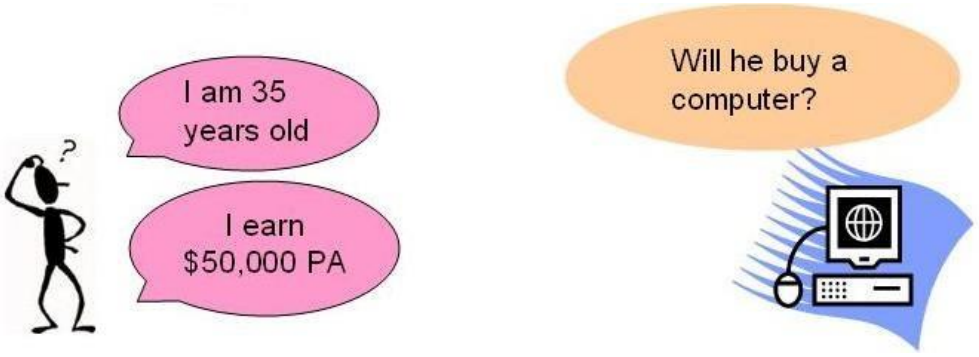
$$hMAP = \arg\max \frac{P(D/h)\,P(h)}{P(D)}$$

$$hMAP = \arg\max P(D/h)\,P(h)$$

B4: Giả thuyết khả năng tối đa (ML)

ID	Tuổi	Thu nhập	Sinh viên	Đánh giá tín dụng	Mua máy tính
1	35	Medium	Yes	Fair	Yes
2	30	Hight	No	Average	No
3	40	Low	Yes	Good	No
4	35	Medium	No	Fair	Yes
5	45	Low	No	Fair	Yes
6	35	Hight	No	Excellent	Yes
7	35	Medium	No	Good	No
8	25	Low	No	Good	No
9	28	Hight	No	Average	No
10	35	Medium	Yes	Average	Yes

Table 2



Giả thuyết:

- Nếu chúng ta giả sử P (hi) = P (hj) trong đó xác suất được tính toán tương đương với việc đơn giản hóa hơn nữa dẫn đến:
- hML = arg max P (D / hi) (trong đó hi thuộc về H)

Áp dụng lý thuyết:

- P (mua máy tính = có) = 5/10 = 0,5 P (mua máy tính = không) = 5/10 = 0,5

- P (khách hàng là 35 tuổi và kiếm được 50.000 đô la) = 4/10 = 0,4
- P (khách hàng là 35 tuổi và kiếm được 50.000 đô la / lần mua máy tính = có) = 3/5 = 0,6 P (khách hàng là 35 tuổi và kiếm được 50.000 đô la / lần mua máy tính = không) = 1/5 = 0,2
-
-
- Khách hàng mua máy tính P (h1 / D) = P (h1) * P (D / h1) / P (D) = 0.5 * 0.6 / 0.4 Khách hàng không mua máy tính P (h2 / D) = P (h2) * P (D / h2) / P (D) = 0,5 * 0,2 / 0,4
-
- Kết quả cuối cùng = arg max {P (h1 / D), P (h2 / D)} = max (0.6, 0.2) => **Khách hàng mua máy tính**

b. Naive-Bayes Classifiers

- Bộ phân lớp Naive bayes hay bộ phân lớp Bayes (simple byes classifier) hoạt động như sau:
  1. Gọi D là tập dữ liệu huấn luyện, trong đó mỗi phần tử dữ liệu X được biểu diễn bằng một vector chứa n giá trị thuộc tính A1, A2,...,An = {x1,x2,...,xn}
  2. Giả sử có m lớp C1, C2,..,Cm. Cho một phần tử dữ liệu X, bộ phân lớp sẽ gán nhãn cho X là lớp có xác suất hậu nghiệm lớn nhất. Cụ thể, bộ phân lớp Bayes sẽ dự đoán X thuộc vào lớp Ci nếu và chỉ nếu:
     - **P(Ci|X) > P(Cj|X) (1<= i, j <=m, i != j)**
     - Giá trị này sẽ tính dựa trên định lý Bayes.
  3. Để tìm xác suất lớn nhất, ta nhận thấy các giá trị P(X) là giống nhau với mọi lớp nên không cần tính. Do đó ta chỉ cần tìm giá trị lớn nhất của P(X|Ci) * P(Ci). Chú ý rằng P(Ci) được ước lượng bằng |Di|/|D|, trong đó Di là tập các phần tử dữ liệu thuộc lớp Ci. Nếu xác suất tiền nghiệm P(Ci) cũng không xác định được thì ta coi chúng bằng nhau P(C1) = P(C2) = ... = P(Cm), khi đó ta chỉ cần tìm giá trị P(X|Ci) lớn nhất.
  4. Khi số lượng các thuộc tính mô tả dữ liệu là lớn thì chi phí tính toàn P(X|Ci) là rất lớn, dó đó có thể giảm độ phức tạp của thuật toán Naive Bayes giả thiết các thuộc tính độc lập nhau. Khi đó ta có thể tính:
     P(X|Ci) = P(x1|Ci)...P(xn|Ci)

- ***Công thức tổng quát:***

$$p(\mathbf{x}|c) = p(x_1, x_2, \ldots, x_d|c) = \prod_{i=1}^{d} p(x_i|c)$$

- ***Ví dụ 1:***

- Phân các bệnh nhân thành 2 lớp ung thư và không ung thư. Giả sử xác suất để một người bị ung thư là 0.008 tức là P(cancer) = 0.008; và P(nocancer) = 0.992. Xác suất để bệnh nhân ung thư có kết quả xét nghiệm dương tính là 0.98 và xác suất để bệnh nhân không ung thư có kết quả dương tính là 0.03 tức là P(+/cancer) = 0.98, P(+/nocancer) = 0.03. Bây giờ giả sử một bệnh nhân có kết quả xét nghiệm dương tính. Ta có:
  - P(+/canncer)P(cancer) = 0.98 * 0.008 = 0.0078
  - P(+/nocancer)P(nocancer) = 0.03 * 0.992 = 0.0298
  - Như vậy, P(+/nocancer)P(nocancer) >> P(+/cancer)P(cancer).
  - Do đó ta xét đoán rằng, bệnh nhân là không ung thư.

- ***Ví dụ 2:***
  - Cơ sở dữ liệu khách hàng:

ID	**Tuổi**	**Thu nhập**	**Sinh viên**	**Đánh giá tín dụng**	**Mua máy tính**
1	<=30	High	No	Fair	No
2	<=30	High	No	Excellent	No
3	31…40	High	No	Fair	Yes
4	>40	Medium	No	Fair	Yes
5	>40	Low	Yes	Fair	Yes
6	>40	Low	Yes	Excellent	No
7	31…40	Low	Yes	Excellent	Yes
8	<=30	Medum	No	Fair	Yes
9	<=30	Low	Yes	Fair	Yes
10	>40	Medium	Yes	Fair	Yes
11	<=30	Medium	Yes	Excellent	Yes
12	31…40	Medium	No	Excellent	Yes
13	31…40	High	Yes	Fair	Yes
14	>40	Medium	No	Excellent	No

Table 1

X = (age= youth, income = medium, student = yes, credit_rating = fair)
Vấn đề đặt ra: Một người X sẽ mua máy tính?
Đạo hàm:
D: Bộ tuple
- Mỗi Tuple là một vectơ thuộc tính chiều thứ n
- X: (x1, x2, x3, Mạnh. Xn)

Hãy để có các lớp học m m: C1, C2, C3 Cm
Trình phân loại Naïve Bayes dự đoán X thuộc về Class Ci iff

- P (Ci / X)> P (Cj / X) cho 1 <= j <= m, j <> i Giả thuyết tối đa Posteriori
- P (Ci / X) = P (X / Ci) P (Ci) / P (X)

Maximum Tối đa hóa P (X / Ci) P (Ci) vì P (X) không đổi

- Với nhiều thuộc tính, sẽ rất tốn kém khi đánh giá P (X / Ci). Giả định Naive Assumption của lớp độc lập có điều kiện

$$P(X/.Ci) = \prod_{k=1}^{n} P(xk / Ci)$$

P(X/Ci) = P(x1/Ci) * P(x2/Ci) *…* P(xn/ Ci)

- P(C1) = P(buys_computer = yes) = 9/14 =0.643
- P(C2) = P(buys_computer = no) = 5/14= 0.357
- P(age=youth /buys_computer = yes) = 2/9 =0.222
- P(age=youth /buys_computer = no) = 3/5 =0.600
- P(income=medium /buys_computer = yes) = 4/9 =0.444
- P(income=medium /buys_computer = no) = 2/5 =0.400
- P(student=yes /buys_computer = yes) = 6/9 =0.667
- P(student=yes/buys_computer = no) = 1/5 =0.200
- P(credit rating=fair /buys_computer = yes) = 6/9 =0.667
- P(credit rating=fair /buys_computer = no) = 2/5 =0.400

- P(X/Buys a computer = yes) = P(age=youth /buys_computer = yes) * P(income=medium/buys_computer = yes) * P(student=yes /buys_computer = yes) * P(credit rating=fair/buys_computer = yes) = 0.222 * 0.444 * 0.667 * 0.667 = 0.044
- P(X/Buys a computer = No) = 0.600 * 0.400 * 0.200 * 0.400 = 0.019

Tìm lớp Ci tối đa hóa
=>P(X/Buys a computer = yes) * P(buys_computer = yes) = 0.028
=>P(X/Buys a computer = No) * P(buys_computer = no) = 0.007
=> **Khách X sẽ mua máy vi tính.**

c. Khắc phục vấn đề xác suất điều kiện bằng zero

- Nếu trong dữ liệu huấn luyện không có đối tượng X nào có thuộc tính lớp Ck có thuộc tính Fi nhận một giá trị cụ thể vij, xác suất điều kiện P(Fi = xij | Ck) sẽ bằng 0.
- Khi phân lớp, nếu có một đối tượng nào mang thuộc tính này thì xác suất phân vào lớp Ck luôn bằng 0.

- Khắc phục bằng cách ước lượng theo công thức sau:

$$P(F_i = v_j^i \mid C = c_k) = \frac{n_{ijk} + mp}{n_{k+m}}$$

Trong đó:

- n_{ijk} là số đối tượng x thuộc lớp c_k mang thuộc tính giá trị (F_i, v^i_j) trong **D**.
- n_k là số lượng đối tượng x thuộc lớp c_k.
- p là hằng số (p có thể = 1 hoặc $1/P_i$ với P_i là số giá trị thuộc tính F_i có thể nhận.
- m là số lượng đối tượng x "ảo", $m \geq 1$.

d. Mô hình Bernoulli

- Ta có training data gồm 10 email, đánh 2 nhãn: Spam (S) và Not Spam (N).
- Ta định nghĩa bảng từ vựng gồm 8 từ như sau:

$$V = [w_1, w_2, w_3, w_4, w_5]$$

Cần phân loại email E11 thuộc loại nào

	Email	w_1	w_2	w_3	w_4	w_5	Label
Training data	E1	1	1	0	1	0	N
	E2	0	1	1	0	0	N
	E3	1	0	1	0	1	S
	E4	1	1	1	1	0	S
	E5	0	1	0	1	0	S
	E6	0	0	0	1	1	N
	E7	0	1	0	0	0	S
	E8	1	1	0	1	0	S
	E9	0	0	1	1	1	N
	E10	1	0	1	0	1	S
Test data	E11	1	0	0	1	1	?

Giải:

Để dễ phân biệt, ra xếp tập training data riêng thành 2 bảng như sau:

class = Spam (S)

Email	w_1	w_2	w_3	w_4	w_5	Label
E3	1	0	1	0	1	S
E4	1	1	1	1	0	S
E5	0	1	0	1	0	S
E7	0	1	0	0	0	S
E8	1	1	0	1	0	S
E10	1	0	1	0	1	S

class = Not Spam (N)

Email	w_1	w_2	w_3	w_4	w_5	Label
E1	1	1	0	1	0	N
E2	0	1	1	0	0	N
E6	0	0	0	1	1	N

Ta có:

$$P(S) = \frac{6}{10}, P(N) = \frac{4}{10}$$

Số lần xuất hiện của từng từ tương ứng với 2 nhãn S và N như sau:

$$n_s(w_1) = 4 \quad n_s(w_2) = 4$$
$$n_s(w_3) = 3 \quad n_s(w_4) = 3$$
$$n_s(w_5) = 2$$

$$n_n(w_1) = 1 \quad n_n(w_2) = 2$$
$$n_n(w_3) = 2 \quad n_n(w_4) = 3$$
$$n_n(w_5) = 2$$

Ta tính được xác xuất của từng từ xuất hiện như sau:

$$P(w_1|S) = \frac{2}{3} \quad P(w_2|S) = \frac{2}{3}$$
$$P(w_3|S) = \frac{1}{2} \quad P(w_4|S) = \frac{1}{2}$$
$$P(w_5|S) = \frac{1}{3}$$

$$P(w_1|N) = \frac{1}{4} \quad P(w_2|N) = \frac{1}{2}$$
$$P(w_3|N) = \frac{1}{2} \quad P(w_4|N) = \frac{3}{4}$$
$$P(w_5|N) = \frac{1}{2}$$

Với test data E11 = { w_1=1, w_4=1, w_5=1 }, a tính xác suất tương ứng của E11 với mỗi loại như sau:

$$\begin{aligned} P(S|E11) \propto & \quad P(S)\prod_{i=1}^{5} P(w_i|S) \times w_i + (1 - P(w_i|S)) \times (1 - w_i) \\ \propto & \quad \frac{6}{10} \times (\frac{2}{3} \times \frac{1}{3} \times \frac{1}{2} \times \frac{1}{2} \times \frac{1}{3}) \\ \propto & \quad 0.0111 \end{aligned}$$

$$\begin{aligned} P(N|E11) \propto & \quad P(N)\prod_{i=1}^{5} P(w_i|N) \times w_i + (1 - P(w_i|N)) \times (1 - w_i) \\ \propto & \quad \frac{4}{10} \times (\frac{1}{4} \times \frac{1}{2} \times \frac{1}{2} \times \frac{3}{4} \times \frac{1}{2}) \\ \propto & \quad 0.0094 \end{aligned}$$

Ta tính được xác suất tương ứng như sau:

$$P(S|E11) = \frac{0.0111}{0.0111 + 0.0094} \approx 0.54$$
$$P(N|E11) = \frac{0.0094}{0.0111 + 0.0094} \approx 0.46$$

Do đó ta phân loại E11 là Spam (S).

e. Mô hình Multinomial

- Vẫn bài toán phân loại mail Spam (S) và Not Spam (N). Ta có bộ training data gồm E1, E2, E3. Cần phân loại E4.
- Bảng từ vựng: [w_1, w_2, w_3, w_4, w_5, w_6, w_7]
- Số lần xuất hiện của từng từ trong từng email tương ứng như bảng dưới.

	Email	w_1	w_2	w_3	w_4	w_5	w_6	w_7	Label
Training data	E1	1	2	1	0	1	0	0	N
	E2	0	2	0	0	1	1	1	N
	E3	1	0	1	1	0	2	0	S
Test data	E4	1	0	0	0	0	0	1	?

Giải

Ta có:

$$P(S) = \frac{1}{3}, P(N) = \frac{2}{3}$$

Sử dụng Laplace Smoothing với α=1 ta tính được xác suất xuất hiện của từng từ trong văn bản như sau:

class = Spam (S)

	Email	w_1	w_2	w_3	w_4	w_5	w_6	w_7
	E3	1	0	1	1	0	2	0
$P(w_i\|S)$	(trước Smoothing)	1/5	0/5	1/5	1/5	0/5	2/5	0/5
$P(w_i\|S)$	(sau Smoothing)	2/12	1/12	2/12	2/12	1/12	3/12	1/12

class = Not Spam (N)

	Email	w_1	w_2	w_3	w_4	w_5	w_6	w_7
	E1	1	2	1	0	1	0	0
	E2	0	2	0	0	1	1	1
Tổng		1	4	1	0	2	1	1
$P(w_i\|N)$	(trước Smoothing)	1/10	4/10	1/10	0/10	2/10	1/10	1/10
$P(w_i\|N)$	(sau Smoothing)	2/17	5/17	2/17	1/17	3/17	2/17	2/17

Vậy ta tính được:

$$
\begin{aligned}
P(S|E4) \propto & \quad P(S)\prod_{i=1}^{7} P(w_i|S) \\
& \propto \quad \frac{1}{3} \times (\frac{2}{12} \times \frac{1}{12}) \\
& \propto \quad 0.0046
\end{aligned}
$$

$$
\begin{aligned}
P(N|E4) \propto & \quad P(N)\prod_{i=1}^{7} P(w_i|N) \\
& \propto \quad \frac{2}{3} \times (\frac{2}{17} \times \frac{2}{17}) \\
& \propto \quad 0.0092
\end{aligned}
$$

Vậy xác suất tương ứng sẽ là:

$$
\begin{aligned}
P(S|E4) = & \quad \frac{0.0046}{0.0046 + 0.0092} \approx 0.334 \\
P(N|E4) = & \quad \frac{0.0092}{0.0046 + 0.0092} \approx 0.666
\end{aligned}
$$

Do đó ta phân loại E4 là Not Spam (N).

f. Ưu điểm

- Giả định độc lập: hoạt động tốt cho nhiều bài toán/miền sữ liệu và ứng dụng.
- Đơn giản nhưng đủ tốt để giải quyết nhiều bài toán như phân lớp văn bản, lọc spam,..
- Cho phép kết hợp tri thức tiền nghiệm (prior knowledge) và dữ liệu quan sát được (obserwed data).
- Tốt khi có sự chệnh lệch số lượng giữa các lớp phân loại.
- Huấn luyện mô hình (ước lượng tham số) dễ và nhanh.

g. Nhược điểm

- Giả định độc lập (ưu điểm cũng chính là nhược điểm)
- hầu hết các trường hợp thực tế trong đó có các thuộc tính trong các đối tượng thường phụ thuộc lẫn nhau.
- Vấn đề zero (đã nêu cách giải quyết ở phía trên)
- Mô hình không được huấn luyện bằng phượng pháp tối ưu mạnh và chặt chẽ.
- Tham số mủa mô hình là các ước lượng xác suất điều kiện đơn lẻ.
- Không tính đến sự tương tác giữa các ước lượng này.

III. Bài toán cụ thể

a. Phân loại chữ viết tay

1. Yêu cầu chính

- Nhận dạng và phân loại được chữ số 1, 2, 3.
- Data ở dạng hình ảnh:

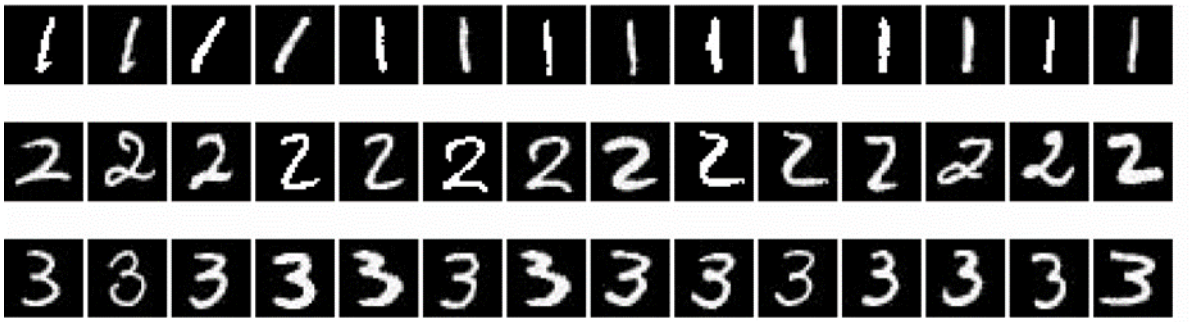


Image 1

- Sử dụng làm thuộc tính pixel trắng (giá trị 255) và vị trí xuất hiện của chúng.

2. Giải thuật cho bài toán

Bước 1: Nó được tải hình ảnh sẽ được phân loại là MỘT, HAI hoặc BA.
Bước 2: Đã tải các hình ảnh được tìm thấy trong các hình ảnh thư mục. Tên của các tập tin thuộc lớp ONE là: hình ảnh1 _ *. Jpg, một trong những tập tin của lớp TWO là: hình ảnh2 _ *. Jpg, và những người cho lớp BA là: hình ảnh3 _ *.
Bước 3: Nó được xác định xác suất tiên nghiệm cho mỗi lớp:
P(UNU) = NrTemplateInClassONE / NumberTotalTemplates P(DOI) = NrTemplateInClassTWO / NumberTotalTemplates P(TREI) = NrTemplateInClassTHREE / NumberTotalTemplates
Bước 4: Xác định xác suất hình ảnh từ Bước 1 sẽ thuộc lớp MỘT, HAI hoặc BA. Đặt (i, j) là vị trí của pixel trắng trong ảnh. Nó được tính xác suất để pixel có tọa độ (i, j) có màu trắng cho lớp ONE, TWO và BA.

count1i,j = 0
for k = 1,n ; n – số lượng hình ảnh trong lớp ONE if image1_k(i,j) = 255 then count1i,j = count1i,j + 1
probability1(i,j) =count1i,j / NrTemplateInClassONE

count2i,j = 0
for k = 1,n ; n- số lượng hình ảnh trong lớp TWO

if image2_k(i,j) = 255 then count2i,j = count2i,j + 1

probability2(i,j) =count2i,j / NrTemplateInClassTWO

count3i,j = 0
for k = 1,n ; n- số lượng hình ảnh trong lớp s THREE if image3_k(i,j) = 255 then
count3i,j = count3i,j + 1
probability 3(i,j) =count3i,j / NrTemplateInClassTHREE

Bước 5: Xác suất hậu sinh mà hình ảnh trong Bước 1 có trong lớp MỘT là:
P(T|ONE) = average (probabilitate1(i,j));
(i, j) - vị trí của các pixel trắng trong ảnh từ Bước 1
Bước 6: Xác suất hậu sinh mà hình ảnh trong Bước 1 có trong lớp HAI là:
P(T|TWO) = average (probabilitate1(i,j));
(i, j) - vị trí của các pixel trắng trong ảnh từ Bước 1
Bước 7: Xác suất hậu sinh mà hình ảnh trong Bước 1 có trong lớp BA là:
P(T|THREE) = average (probabilitate1(i,j));
(i, j) - vị trí của các pixel trắng trong ảnh từ Bước 1
Bước 8: Nó được xác định xác suất P cho mỗi lớp hình ảnh và nó được gán hình ảnh từ Bước 1 cho lớp hình ảnh có xác suất lớn nhất.
P(ONE|T) = P(T| ONE)*P(ONE) P(TWO|T)
= P(T| TWO)*P(TWO) P(THREE|T)
= P(T| THREE)*P(THREE)

3. Code tham khảo

```
import java.awt.*; import java.awt.image.*; import java.io.*;
import javax.swing.*;
import java.util.*;
public class CImagesLoad {
Vector<Image> images1 = new Vector<Image>(); Vector<Image> images2 = new Vector<Image>(); Vector<Image> images3 = new Vector<Image>(); public String getFile(boolean isSaveDialog)
{
String currentDirectoryName = new File("").getAbsolutePath() +File.separator;
try{
JFileChooser fc = new JFileChooser(new File(new File(currentDirectoryName).getParent()));
int result = 0;
if(!isSaveDialog)
result = fc.showOpenDialog(null);
else
result = fc.showSaveDialog(null);
if(result==JFileChooser.CANCEL_OPTION) return null; else { //if(result==JFileChooser.APPROVE_OPTION){
return   fc.getSelectedFile().getAbsolutePath();
}
}
catch(Exception e)
{
return null;
}
}
public void load_images (int template){ String f = getFile(false);
if (f==null)
{
return;
}
int k = 1;
while (true)
{
String curent = new java.io.File (f).getAbsolutePath ();
int pos = curent.lastIndexOf ("\\"); curent = curent.substring (0, pos); if (k < 10)
{
```

```
}
else
{

}

curent += "\\image" + template + "_0" + k + ".jpg";



curent += "\\image" + template + "_" + k + ".jpg";

Image img = null;
img = new javax.swing.ImageIcon(curent).getImage();
if(img==null || img.getWidth(null)<=0 ||img.getHeight(null)<=0)
{
System.out.println("The file \n" + f.toString() + "\nhas an unsupported image format");
break;

}
else
{




k++;
switch (template)
{
case 1:



case 2:


case 3:
```

```
default:


}
}
}
images1.add (img);
break;

images2.add (img);
break;

images3.add (img);
break;

System.out.println("Other class");
break;

}
public void load_pixels (Image image)
{
int width = image.getWidth(null);
int height = image.getHeight(null);
// Allocate buffer to hold the image's pixels
int pixels[] = new int[width * height];
// Grab pixels
PixelGrabber pg = new PixelGrabber (image, 0, 0, width, height, pixels, 0, width);
try
{
pg.grabPixels();
}
catch (InterruptedException e)
{
System.out.println ("Error image loading");
}
}
}
```

b. Chuẩn đoán bệnh nhân bị tiểu đường

1) Bộ dữ liệu:

- Tập dữ liệu này bao gồm dữ liệu của 768 tình nguyện viên bao gồm những người bị tiểu đường và những người không bị tiểu đường. Tập dữ liệu này bao gồm các thuộc tính như sau:
  1. Number of times pregnant
  2. Plasma glucose concentration a 2 hours in an oral glucose tolerance test
  3. Diastolic blood pressure (mm Hg)
  4. Triceps skin fold thickness (mm)
  5. 2-Hour serum insulin (mu U/ml)
  6. Body mass index (weight in kg/(height in m)^2)
  7. Diabetes pedigree function
  8. Age (years)
- Với mỗi tình nguyện viện, dữ liệu bao gồm tập hợp các chỉ số kể trên và tình trạng bị bênh tức class 1 hay không bị bệnh tức class 0.

1	6	148	72	35	0	33.6	0.627	50	1
2	1	85	66	29	0	26.6	0.351	31	0
3	8	183	64	0	0	23.3	0.672	32	1
4	1	89	66	23	94	28.1	0.167	21	0
5	0	137	40	35	168	43.1	2.288	33	1
6	5	116	74	0	0	25.6	0.201	30	0
7	3	78	50	32	88	31.0	0.248	26	1

- Có một điều nhận thấy rằng giá trị của các chỉ số là một biến liên tục chứ không phải một giá trị rời rạc chính vì thế nên khi áp dụng thuật toán Naive Bayes chúng ta cần phải áp dụng một phân phối xác suất cho nó. Một trong những phân phối xác suất phổ biến được sử dụng trong phần này đó chính là phân phối Gaussian. Chúng ta cùng tìm hiểu qua một chút về nó nhé. Phải hiểu được bản chất thì mới có thể thực hành được.

2) Phân phối Gaussian

- Công thức tổng quát:

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

- Với một một dữ liệu x_i thuộc một class c_i chúng ta thấy x_i tuân theo một phân phối chuẩn với kì vọng μ và độ lệch chuẩn σ. Khi đó hàm xác suất của x_i được xác định như sau:

$$P(x_i) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x_i-\mu)^2/2\sigma^2}$$

- Đây chính là cách tính của thư viện sklearn tuy nhiên trong bài viết này mình sẽ hướng dẫn các bạn cài đặt thủ công. Chính việc cài đặt thủ công này giúp cho chúng ta hiểu hơn về bài toán.

3) Cài đặt Code

- ***Load dữ liệu***
  - Dữ liệu của chúng ta được lưu dưới dạng file CSV nên chúng ta sẽ sử dụng thư viện csv của Python để đọc dữ liệu.

```
# Load data tu CSV file
def load_data(filename):
    lines = csv.reader(open(filename, "rb"))
    dataset = list(lines)
    for i in range(len(dataset)):
        dataset[i] = [float(x) for x in dataset[i]]

    return dataset
```

- **Tính độ lệch chuẩn:**

- Chúng ta có thể tham khảo công thức tính của nó như biểu thức sau:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} \left(x_i - \bar{x}\right)^2}{n}}$$

- Trong đó $\bar{x}$ là gía trị trung bình của biến ngẫu nhiên trên toàn tập dữ liệu. Chúng ta sử dụng Python để thực hiện hàm tính giá trị trung bình và độ lệch chuẩn như sau

```
# tinh toan gia tri trung binh cua moi thuoc tinh
def mean(numbers):
    return sum(numbers) / float(len(numbers))

# Tinh toan do lech chuan cho tung thuoc tinh
def standard_deviation(numbers):
    avg = mean(numbers)
    variance = sum([pow(x - avg, 2) for x in numbers]) / float(len(numbers) - 1)

    return math.sqrt(variance)
```

- **Tiền xử lý dữ liệu**
  - Trước khi bắt đầu mỗi bài toán về Machine Learning bước tiền xử lý dữ liệu là rất quan trọng. Nếu như tập dữ của chúng ta chưa chuẩn chúng ta sẽ cần phải làm thêm một số bước khác như lấy mẫu dữ liệu, loại bỏ dữ liệu thiếu và biến đổi dữ liệu về dạng thích hợp để xử lý.. Đối với tập dữ liệu bênh tiểu đường thì dữ liệu đã được chuẩn hóa rồi nên tùy vào từng thuật toán mà chúng ta chọn cách biểu diễn dữ liệu cho phù hợp. Như phần trên đã nói chúng ta sẽ sử dụng Độ lệch chuẩn và giá trị trung bình để tính toán các xác suất cần thiết nên cần có một hàm để chuyển đổi dữ liệu ban đầu về dạng tập hợp của độ lệch chuẩn và trung bình nhằm phục vụ cho các phép tính xác suất sau này.

```
# Chuyen ve cap du lieu  (Gia tri trung binh , do lech chuan)

def summarize(dataset):
    summaries = [(mean(attribute), standard_deviation(attribute)) for attribute in zip(*dataset)]
    del summaries[-1]

    return summaries

def summarize_by_class(dataset):
    separated = separate_data(dataset)
    summaries = {}
    for classValue, instances in separated.iteritems():
        summaries[classValue] = summarize(instances)

    return summaries
```

- **Tính xác suất của từng biến liên tục theo phân phối Gausian**
  - Dựa vào cơ sở lý thuyết ở bên trên. Chúng ta tiến hành tính các xác suất phụ thuộc của biến ngẫu nhiên bao gồm p(x)p(x) của mỗi chỉ số sức khỏe và p(x|c)p(x|c) của mỗi class tương ứng với chỉ số đó.

```
# Tinh toan xac suat theo phan phoi Gause cua bien lien tuc the hien cac chi so suc khoe
def calculate_prob(x, mean, stdev):
    exponent = math.exp(-(math.pow(x - mean, 2) / (2 * math.pow(stdev, 2))))

    return (1 / (math.sqrt(2 * math.pi) * stdev)) * exponent

# Tinh xac suat cho moi chi so suc khoe theo class
def calculate_class_prob(summaries, inputVector):
    probabilities = {}
    for classValue, classSummaries in summaries.iteritems():
        probabilities[classValue] = 1
        for i in range(len(classSummaries)):
            mean, stdev = classSummaries[i]
            x = inputVector[i]
            probabilities[classValue] *= calculate_prob(x, mean, stdev)

    return probabilities
```

- **Dự đoán dựa vào xác suất**
  - Đây là bước áp dụng định lý Bayes đã được giới thiệu bên trên vào dự đoán các class thông qua các chỉ số trong tập dữ liệu

```
# Du doan vector thuoc phan lop nao
def predict(summaries, inputVector):
    probabilities = calculate_class_prob(summaries, inputVector)
    bestLabel, bestProb = None, -1
    for classValue, probability in probabilities.iteritems():
        if bestLabel is None or probability > bestProb:
            bestProb = probability
            bestLabel = classValue

    return bestLabel

# Du doan tap du lieu testing thuoc vao phan lop nao
def get_predictions(summaries, testSet):
    predictions = []
    for i in range(len(testSet)):
        result = predict(summaries, testSet[i])
        predictions.append(result)

    return predictions

# Tinh toan do chinh xac cua phan lop
def get_accuracy(testSet, predictions):
    correct = 0
    for i in range(len(testSet)):
        if testSet[i][-1] == predictions[i]:
            correct += 1

    return (correct / float(len(testSet))) * 100.0
```

- **Learning**
  - Sau bước xử lý dữ liệu ban đầu chúng ta tiến hành learning như sau:

```
def main():
    filename = \'tieu_duong.csv\'
    splitRatio = 0.8
    dataset = load_data(filename)
    trainingSet, testSet = split_data(dataset, splitRatio)

    print(\'Data size {0} \nTraining Size={1} \nTest Size={2}\').format(len(dataset), len(trainingSet), len(testSet))

    # prepare model
    summaries = summarize_by_class(trainingSet)

    # test model
    predictions = get_predictions(summaries, testSet)
    accuracy = get_accuracy(testSet, predictions)
    print(\'Accuracy of my implement: {0}%\').format(accuracy)
```

- **Kết quả cài đặt**
  - Sau khi cài đặt ta nhận thấy thuật toán tính toán rất nhanh và cho độ chính xác khoảng 75% tùy thuộc vào cách phân chia dữ liệu.

```
Connected to pydev debugger (build 172.3544.46)
Data size 768
Training Size=614
Test Size=154
Accuracy of my implement: 75.974025974%
```

4) Sử dụng thuật toán Naive-Bayes bằng thư viện Sklearn

- **Phân chia dữ liệu**
  - Đầu tiên ta cần phân chia tập dữ liệu ban đầu thành hai ma trận, 1 ma trận chứa chỉ số của tình nguyện viên chính là 8 chỉ số đã chỉ ra ở phần đầu tiên và một ma trận chứa các class tương ứng.

```
def get_data_label(dataset):
    data = []
    label = []
    for x in dataset:
        data.append(x[:8])
        label.append(x[-1])

    return data, label
```

- **Training**
  - Chúng ta thêm vào phần hàm main()main() bên trên đoạn code sau để thực hiện training sử dụng thư viện

```
# Compare with sklearn
dataTrain, labelTrain = get_data_label(trainingSet)
dataTest, labelTest = get_data_label(testSet)

from sklearn.naive_bayes import GaussianNB
clf = GaussianNB()
clf.fit(dataTrain, labelTrain)
```

- **Predict**
  - Sau khi training xong chúng ta sẽ sử dụng model để đánh giá trên tập dữ liệu testing.

```
score = clf.score(dataTest, labelTest)
print(\'Accuracy of sklearn: {0}%\').format(score*100)
```

- **Kết quả nhận được**

```
Connected to pydev debugger (build 172.3544.46)
Data size 768
Training Size=614
Test Size=154
Accuracy of my implement: 75.974025974%
Accuracy of sklearn: 75.974025974%

Process finished with exit code 0
```

5) Kết luận thuật toán trong bài toán phân loại bệnh tiểu đường

- Vấn đề của bài toán rõ ràng vẫn là thống kê được những mối liện hệ và tìm được những xác suất của nó với vấn đề mà chúng ta quan tâm (ở đây chính là việc có bị tiểu đường hay không).
- Tuy độ chính xác còn chưa cao do bản chất cuả phương pháp cũng như tập dữ liệu chưa đủ lớn tuy nhiên nó cũng thể hiện được các điểm chính của thuật toán Naive Bayes

IV. Tổng kết

- Naive Bayes Classifiers (NBC) là phương pháp cổ điển nhưng vẫn rất hữu dụng với các bài toán nhất định như phân loại văn bản, email…

- NBC với công thức tính toán đơn giản nên dễ cài đặt (hiện nay nếu dùng thư viện sklearn thì chỉ cần gọi vài dòng lệnh như mình làm bên trên), thời gian training và test nhanh, phù hợp với bài toán data lớn.

- Cần chú ý sử dụng Smoothing để tránh lỗi xác suất tổng được bằng 0 khi xác suất của một feature thành phần bằng 0.

V. Tài liệu tham khảo:

- https://viblo.asia/p/thuat-toan-phan-lop-naive-bayes-924lJWPm5PM
- https://vi.wikipedia.org/wiki/Khai_ph%C3%A1_d%E1%BB%AF_li%E1%BB%87u
- https://en.wikipedia.org/wiki/Naive_Bayes_spam_filtering
- https://techblog.vn/ung-dung-thuat-toan-naive-bayes-trong-giai-quyet-bai-toan-chuan-doan-benh-tieu-duong