TRƯỜNG ĐẠI HỌC KHOA HỌC XÃ HỘI VÀ NHÂN VĂN

KHOA THƯ VIỆN – THÔNG TIN HỌC

----- . . . -----

BÁO CÁO ĐỒ ÁN MÔN HỌC

LƯU TRỮ VÀ KHAI THÁC DỮ LIỆU

Tên đề tài: Tìm hiểu và ứng dụng thuật toán Naive Bayes

Giảng viên hướng dẫn: ThS. Nguyễn Tấn Công

Nhóm sinh viên thực hiện:

Họ và tên	**Mã số sinh viên**
Nguyễn Văn Hạnh	2256210013
Nguyễn Thị Minh Anh	2256210004

Thành phố Hồ Chí Minh, ngày 05 tháng 01 năm 2025

LỜI MỞ ĐẦU

Dữ liệu được tạo ra từ mọi hoạt động của con người, từ hoạt động cá nhân, doanh nghiệp đến các tổ chức, cơ quan nhà nước. Khối lượng dữ liệu khổng lồ này có thể mang lại nhiều giá trị nếu được khai thác hiệu quả. Và trong những năm gần đây, sự phát triển của công nghệ thông tin và truyền thông đã dẫn đến hiện tượng bùng nổ dữ liệu.

Khai thác dữ liệu là quá trình thu thập, xử lý và phân tích dữ liệu nhằm tìm ra các thông tin có giá trị, giúp đưa ra các quyết định sáng suốt hơn. Khai thác dữ liệu có thể được áp dụng trong nhiều lĩnh vực khác nhau, như kinh doanh, tài chính, y tế, giáo dục,... Trong số các phương pháp khai phá dữ liệu, phân loại dữ liệu là một phương pháp quan trọng. Phân loại dữ liệu là quá trình gán một nhãn cho mỗi đối tượng trong tập dữ liệu, dựa trên các thuộc tính của đối tượng đó. Phân loại dữ liệu có thể được sử dụng trong nhiều ứng dụng khác nhau, như nhận dạng spam, phân loại email, phân loại bệnh,... Một trong những thuật toán phân loại dữ liệu phổ biến nhất là thuật toán Naive Bayes.

Với đề tài là "Tìm hiểu và ứng dụng thuật toán Naive Bayes" – Nhóm hướng tới mục tiêu hiểu rõ về các khái niệm cơ bản, cách thức hoạt động, ưu điểm và hạn chế của thuật toán, bên cạnh đó là việc cài đặt, thí nghiệm, và demo thuật toán để hiểu rõ hơn về cách thức thuật toán được triển khai. Với việc ứng dụng các kỹ thuật lưu trữ và khai thác dữ liệu như: phân loại dữ liệu; phân lớp dữ liệu, phân tích các yếu tố liên quan,…

Đường link GitHub chứa các thông tin mô tả, dữ liệu và source python về đồ án của Nhóm: *https://github.com/Hanhhh1607/DoAnLuuTru-KhaiThacDuLieu.git*

Nội dung của đồ án được chia thành 3 phần chính:

- ***Chương I.*** *Tổng quan về thuật toán Naive Bayes*
- ***Chương II.*** *Chuẩn bị dữ liệu chạy máy*
- ***Chương III.*** *Cài đặt, thí nghiệm và demo thuật toán*

Nhóm sinh viên thực hiện.

LỜI CẢM ƠN

Lời đầu tiên, Nhóm chúng em xin chân thành cảm ơn thầy Nguyễn Tấn Công, giảng viên khoa Thư viện – Thông tin học, trường Đại học Khoa học Xã hội và Nhân Văn – Đại học Quốc Gia TP. Hồ Chí Minh, đã tận tình hướng dẫn và tạo điều kiện tốt nhất cho Nhóm hoàn thành đồ án môn Lưu trữ và khai thác dữ liệu ở học kỳ 1 năm học 2024 – 2025.

Mặc dù Nhóm đã gặp nhiều khó khăn trong việc hoàn thành đồ án nhưng thông qua những kiến thức mà Thầy truyền về kiến thức tổng quan về lưu trữ và khai thác dữ liệu, phân cụm phân loại,… cùng các kiến thức liên quan đã trở thành nền tảng kiến thức quan trọng để Nhóm có thể hoàn thành đồ án này.

Tuy nhiên, trong quá trình tìm hiểu và thực hiện đồ án Nhóm chúng em vẫn còn nhiều thiếu sót cả về kiến thức lẫn kĩ năng. Chúng em rất mong nhận được sự đánh giá và góp ý của Thầy, từ đó Nhóm có thể rút được kinh nghiệm cho những lần thực hiện đồ án môn học tiếp theo và tích lũy thêm những kỹ năng, kinh nghiệm cho công việc sau này.

Chúng em xin chân thành cảm ơn!

TP. Hồ Chí Minh, ngày 05 tháng 01 năm 2025

Nhóm sinh viên thực hiện.

NHẬN XÉT CỦA GIẢNG VIÊN

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

……

TP. Hồ Chí Minh, ngày... tháng... năm 2025

Giảng viên

BẢNG MÔ TẢ CÁC THUẬT NGỮ

STT	Thuật ngữ tiếng Anh	Chữ viết đầy đủ
1	NBC	Naive Bayes Classifier
2	ML	Machine Learning

DANH MỤC HÌNH ẢNH

MỤC LỤC

CHƯƠNG I. TỔNG QUAN VỀ THUẬT TOÁN NAIVE BAYES

1. Khái niệm, nội dung, thực thi

1. 1. Khái niệm

Naive Bayes Classifier (NBC) là phương pháp phân loại dựa trên xác suất được sử dụng rộng rãi trong lĩnh vực máy học, được sử dụng lần đầu tiên trong lĩnh vực phân loại bởi Maron năm 1961 và ngày càng trở nên phổ biến. Đây là một thuật toán phân loại dựa trên tính toán xác suất áp dụng định lý Bayes. Thuật toán này thuộc nhóm Supervised Learning (Học có giám sát) và có hướng tiếp cận phân lớp theo mô hình xác suất. Dự đoán xác suất một đối tượng mới thuộc về thành viên của lớp đang xét. Naive Bayes Classifier là phương pháp phân lớp dựa trên thống kê.

Naive Bayes = Định lý Bayes + Các giả định độc lập

- **Định lý Bayes:** Cho X, Y là hai tập hợp. Ta gọi tần suất hiện của X trong Y là:

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} \quad (1)$$

Trong đó:

+ P(X|Y): số phần tử tập hợp X trong tập hợp Y
+ P(Y|X): số phần tử tập hợp Y trong tập hợp X
+ P(X): số phần tử của tập hợp X
+ P(Y): số phần tử của tập hợp Y

- **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(Xi|Y)* như sau:

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

Trong đó:

+ $N_{yi} = \sum_{x \in T} x_i$ là tổng số lần xuất hiện trong văn bản training T của từ i.
+ $N_y = \sum_{i=1}^{n} N_{yi}$ là tổng số lần xuất hiện trong văn bản training T của tất cả bộ từ vựng.

+ *n* là kích thước của bộ từ vựng.
+ *alpha* $\alpha \geq 0$ có ý nghĩa giải thích cho các từ không có trong bộ từ vựng học và ngăn chặn xác suất bằng 0 cho các tính toán tiếp theo.
+ $\alpha = 1$ gọi là Laplace Smoothing.
+ $\alpha < 1$ gọi là Lidstone Smoothing.

- Có thể thấy rằng
  + Nếu α nhỏ => phương sai cao.
  + Nếu α lớn => độ lệch cao.

⇒ Vì vậy, xác định đúng alpha, cần sử dụng phương pháp cross-validation. Do đó, alpha là siêu tham số (hyperparameter).

1. 2. Các loại mô hình

Có ba loại Mô hình Naive Bayes, được đưa ra dưới đây:

+ **Gaussian**: Mô hình Gaussian giả định rằng các đối tượng địa lý tuân theo phân phối chuẩn. Điều này có nghĩa là nếu các yếu tố dự đoán nhận các giá trị liên tục thay vì rời rạc, thì mô hình giả định rằng các giá trị này được lấy mẫu từ phân phối Gaussian.
+ **Đa thức**: Bộ phân loại Naïve Bayes đa thức được sử dụng khi dữ liệu được phân phối đa thức. Nó chủ yếu được sử dụng cho các vấn đề phân loại tài liệu, nó có nghĩa là một tài liệu cụ thể thuộc về danh mục nào như Thể thao, Chính trị, giáo dục, v.v. Trình phân loại sử dụng tần suất từ cho các yếu tố dự đoán.
+ **Bernoulli**: Bộ phân loại Bernoulli hoạt động tương tự như bộ phân loại đa thức, nhưng các biến dự báo là các biến Booleans độc lập. Chẳng hạn như nếu một từ cụ thể có trong tài liệu hay không. Mô hình này cũng nổi tiếng với các nhiệm vụ phân loại tài liệu.

1.3. Ứng dụng

Thuật toán Naive Bayes Classification được áp dụng vào các loại ứng dụng sau:

+ **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 phát hiện sự cố…
+ **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 ưa 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 ưa chuộng các loại sản phẩm nào từ việc phân tích các thói quen và hành động của khách hàng.
+ **Recommendation System:** Naive Bayes Classifier được sử dụng rất nhiều để xây dựng hệ thống gợi ý.

1. 4. Ý tưởng cơ bản của thuật toán

Ý tưởng cơ bản của phương pháp xác suất Bayes là dựa vào xác suất có điều kiện của từ hay đặc trưng xuất hiện trong văn bản với chủ đề để dự đoán chủ đề của văn bản đang xét. Điểm quan trọng cơ bản của phương pháp này là các giả định độc lập:

- Các từ hay đặc trưng của văn bản xuất hiện là độc lập với nhau.
- Vị trí của các từ hay các đặc trưng là độc lập và có vai trò như nhau.

Giả sử ta có:

+ n chủ đề (lớp) đã được định nghĩa $□_1$, $□_2$, $□_3$,…$□_□$
+ Tài liệu mới cần được phân loại $□_□$
+ Để tiến hành phân loại tài liệu $□_□$, chúng ta cần phải tính được tần suất xuất hiện của các lớp $□_□$ (i=1,2,..,n) trong tài liệu d

⇒ Sau khi tính được xác suất của văn bản đối với các chủ đề, theo luật Bayes, văn bản sẽ được phân lớp vào chủ đề ci nào có xác suất cao nhất.

1.5. Ưu nhược điểm, đặc trưng

1.5.1. Ưu điểm

- Giả định độc lập: hoạt động tốt cho nhiều bài toán/miền dữ 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 tiên nghiệm (priori knowledge) và dữ liệu quan sát được (observed data).
- Tốt khi có sự chênh lệch số lượng giữa các lớp phân loại.
- Huấn luyện mô hình (ước lượng tham số) dễ và nhanh.

- Naive Bayes Classifiers với công thức tính toán đơn giản nên dễ cài đặt, thời gian training và test nhanh, phù hợp với bài toán data lớn. Thực hiện khá tốt trong Multi class Prediction.
- Khi giả định rằng các feature của dữ liệu là độc lập với nhau thì Naive Bayes chạy tốt hơn so với các thuật toán khác như Logistic Regression và cũng cần ít dữ liệu hơn, thời gian thực thi tương tự như cây quyết định.
- Đạt kết quả tốt trong phần lớn các trường hợp.

1.5.2. 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 phương pháp tối ưu mạnh và chặt chẽ. Tham số của mô hình là các ước lượng xác suất điều kiện đơn lẻ. Không tính đến sự tương tác giữa các ước lượng này.
- Trong thực tế, hầu như là bất khả thi khi giả thiết các feature của dữ liệu test là độc lập với nhau. Điều này làm giảm độ chính xác.
- 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.

1.5.3. Đặc trưng

- Thuật toán Naive Bayes là một thuật toán học có giám sát, dựa trên định lý Bayes và được sử dụng để giải các bài toán phân loại.
- Nó chủ yếu được sử dụng trong phân loại văn bản bao gồm một tập dữ liệu đào tạo chiều cao.
- Naive Bayes Classifier là một trong những thuật toán Phân loại đơn giản và hiệu quả nhất giúp xây dựng các mô hình học máy nhanh có thể đưa ra dự đoán nhanh chóng.
- Thuật toán Naive Bayes có đặc điểm rằng các tham số của mô hình sẽ tăng lên tuyến tính khi số lượng các đặc trưng tăng. Đồng thời, mô hình được huấn luyện thông qua việc giải một phương trình đóng (closed-form equation), tức có thể giải được thông qua một hữu hạn các bước, và chỉ có 1 lời giải đã xác định. Vì các tính chất này, nên thời gian huấn luyện thuật toán Naive Bayes là thời gian tuyến tính thay vì thời gian bậc 2, bậc 3.

2. Cơ sở toán, cơ chế vận hành, nguyên lý thuật toán

2.1. Cơ sở toán

Ý tưởng cơ bản của cách tiếp cận Naive Bayes là sử dụng xác suất có điều kiện giữa từ và chủ đề để dự đoán xác suất chủ đề của một văn bản cần phân loại.

- Điểm quan trọng của phương pháp này chính là:
  + Giả định rằng sự xuất hiện của tất cả các từ trong văn bản đều độc lập với nhau
  + Không khai thác sự phụ thuộc của nhiều từ vào trong một chủ đề cụ thể
- Dữ kiện cần có:
  + D: tập dữ liệu huấn luyện, được vector hóa dưới dạng x = (□1, □2,..., □□)
  + Ci: tập các tài liệu của D thuộc lớp □□ với i = (1, 2, 3, …)
  + Các thuộc tính □1, □2,□3,..., □□ độc lập với xác suất đôi một với nhau.

2.2. Nguyên lý thuật toán

- Thuật toán này dựa trên định lý Bayes với công thức tổng quát là:

$$P(H \mid x) = P(H) * P(x \mid H) / P(x)$$

Trong đó:

  + P(H| x) là xác suất để xảy ra giả thuyết H với đầu vào là tập dữ liệu ngẫu nhiên cần dự đoán x.
  + P(H) là xác suất xảy ra của bản thân giả thuyết H mà không quan tâm đến x.
  + P(x|H) là xác suất xảy ra x khi biết H xảy ra, gọi là "xác suất của x nếu có H".
  + P(x) là xác suất xảy ra của riêng tập dữ liệu dự đoán X.

Tổng quát:

$$P(H \mid x1 \ldots .xn) = P(H)\,P(x1 \mid H) \ldots P(xn \mid H) / P(x1) \ldots .P(xn)$$

- ***Phân tích thuật toán Bayes:***
  + Tương tự như thuật toán K-means, Naive Bayes cũng có vấn đề khi xử lý dữ liệu lớn.
  + Trong quá trình học (training), nếu số lượng dữ liệu quá lớn, dẫn đến các vấn đề về thiếu bộ nhớ, tốc độ xử lý.
  + Với lượng dữ liệu lớn (khoảng vài triệu bản ghi) thì hầu hết thời gian của Naive Bayes là đếm số lần xuất hiện các biến => tính xác suất cần thiết để xây dựng mô hình.
  + Tiêu thụ thời gian chủ yếu do tính: các P(□□) và P(□□|□□).

+ Để tính được các xác suất này, ta cần đếm số lần xuất hiện của các $□_□$ và các $□_□|□_□$
+ Việc tính các xác suất, đếm các số lần xuất hiện của từng biến là độc lập => chia dữ liệu thành nhiều phần nhỏ và thực hiện song song.

2.3. Cơ chế hoạt động

Ta có cách thức hoạt động của bộ phân lớp Naive bayes hay bộ phân lớp Bayes như sau:

- 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 $□_1$, $□_2$,...,$□_□$ = {$□_1$, $□_2$,..., $□_□$}.
- 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 $□_1$, $□_2$,...,$□_□$ = {$□_1$, $□_2$,..., $□_□$}.
- Giả sử có m lớp $□_1$, $□_2$,..,$□_□$. 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 $□_□$ nếu và chỉ nếu: P($□_□$|X) > P($□_□$|X) (1≤ i, j ≤m, i != j). Giá trị này sẽ tính dựa trên định lý Bayes.
- 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($□_□$). Chú ý rằng P($□_□$) được ước lượng bằng |$□_□$|/|D|, trong đó $□_□$ là tập các phần tử dữ liệu thuộc lớp $□_□$. Nếu xác suất tiền nghiệm P($□_□$) cũng không xác định được thì ta coi chúng bằng nhau P($□_1$) = P($□_2$) = ... = P($□_□$), khi đó ta chỉ cần tìm giá trị P(X|$□_□$) lớn nhất.
- Khi số lượng các thuộc tính mô tả dữ liệu là lớn thì chi phí tính toàn P(X|$□_□$) 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|$□_□$) = P($□_1$ |$□_□$)...P($□_□$|$□_□$).

3. Các bước tính toán của thuật toán trên dữ liệu chạy tay

Bước 1: Huấn luyện Naive Bayes dựa vào tập dữ liệu:

- Tính xác suất P($□_□$)
- Tính xác suất P($□_□$ |$□_□$)

Bước 2: Phân lớp X new

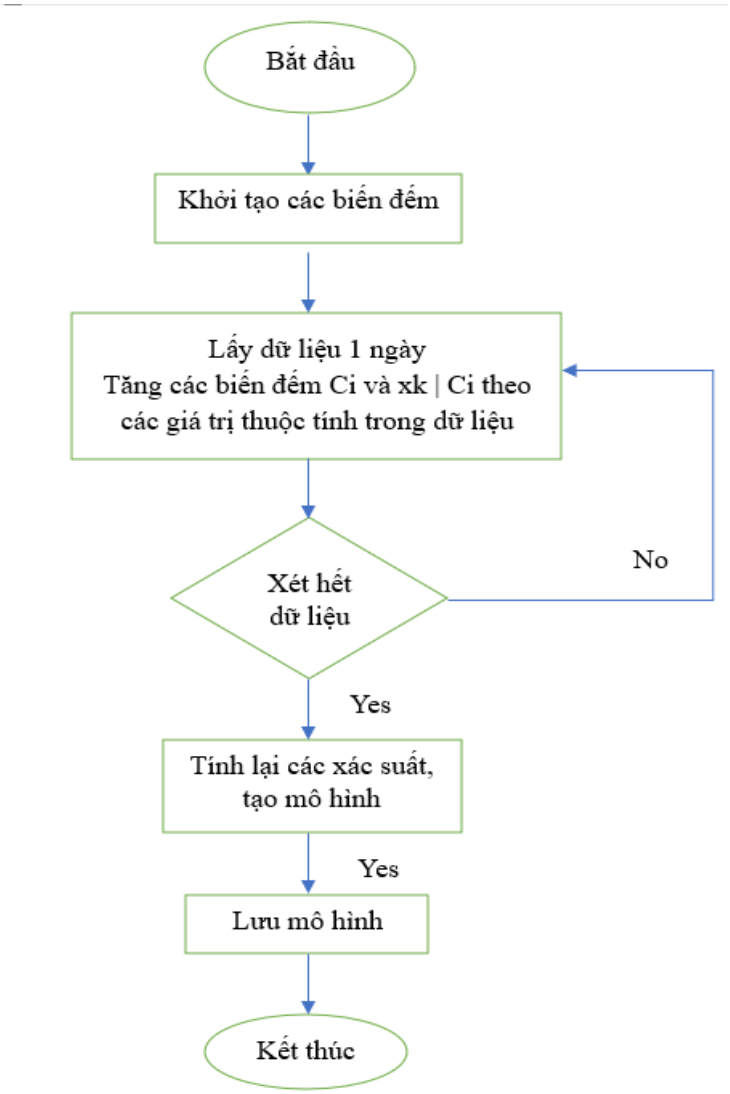
- Tính

$$F(X_{new}, C_i) = P(C_i) \prod_{k=1}^{n} P(x_k|C_i)$$

- X new được gán vào lớp Cq sao cho:

$$F(X_{new}, C_q) = \max(F(X_{new}, C_i))$$

- Lưu đồ thuật toán phân lớp Bayes



Hình 1: Lưu đồ thuật toán phân lớp Bayes

❖ ***Ví dụ minh họa thuật toán Naive Bayes***

Đề bài: dự đoán quyết định của người chơi có đi chơi Tennis hay không với điều kiện về thời tiết đã được biết trước.

Bảng dữ liệu huấn luyện như sau:

Day	Outlook	Temp.	Humidity	Wind	Play Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cold	Normal	Weak	Yes
D6	Rain	Cold	Normal	Strong	No
D7	Overcast	Cold	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Bảng 1: Dữ liệu huấn luyện

- Có 2 lớp dự báo:
  + $□_1$ = "yes" => có đi chơi Tennis
  + $□_2$ = "no" => không đi chơi Tennis

Bước 1: Huấn luyện Bayes

Bước 2: Phân lớp Bayes

❖ ***Các bước làm:***

Tính xác suất: P($□_□$)

- P($□_1$) = P("yes") = 9/14
- P($□_2$) = P("no") = 5/14

Bước 1: Tính các xác suất P($□_□$ |$□_□$)

- Với thuộc tính Outlook
- Với thuộc tính Temp
- Với thuộc tính Humidity
- Với thuộc tính Wind

Với thuộc tính Outlook: Các giá trị: sunny, overcast, rain

+ P(sunny | yes) = 2/9
+ P(sunny | no) = 3/5
+ P(overcast | yes) = 4/9

+ P(overcast | no) = 0/5
+ P(rain | yes) = 3/9
+ P(rain | no) = 2/5

Với thuộc tính Temp: Các giá trị: Hot, Cold, Mild

+ P(hot | yes) = 2/9
+ P(hot| no) = 2/5
+ P(cold | yes) = 3/9
+ P(cold | no) =1/5
+ P(mild | yes) = 4/9
+ P(mild | no) = 2/5

Với thuộc tính Humidity: Các giá trị: Normal, High

+ P(normal | yes) = 6/9
+ P(normal | no) = 1/5
+ P(high | yes) = 3/9
+ P(high | no) = 4/5

Với thuộc tính Wind: Các giá trị: Weak, Strong

+ P(weak | yes) = 6/9
+ P(weak | no) = 2/5
+ P(strong | yes) = 3/9
+ P(strong | no) = 3/5

Bước 2: Phân lớp Bayes

- X new = {sunny, cool, hight, strong}
- Tính các xác suất:
  + F(X new | yes) = P(yes) * P(sunny | yes) * P(cold | yes) * P(high | yes) * * P(strong | yes) = 9/14 * 2/9 * 3/9 * 3/9 * 3/9 = 0.0053
  + F(X new | no) = P(no) * P(sunny | no) * P(cold | no) * P(high | no) * * P(strong | no) = 5/14 * 3/5 * 1/5 * 4/5 * 3/5 = 0.0206

Kết luận: X new thuộc vào lớp No

⇒ Có thể ứng dụng Naive Bayes Classification để tính tỷ lệ xác suất với rất nhiều các dạng bài toán khác nhau, với dữ liệu càng nhiều thì độ chính xác của thuật toán sẽ càng cao, và khi dữ liệu thay đổi thì kết quả cũng thay đổi theo.

CHƯƠNG II. CHUẨN BỊ DỮ LIỆU CHẠY MÁY

- ***Thu thập dữ liệu:*** Dữ liệu được thu thập trên nền tảng Kaggle: https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset/data
- ***Thông tin chi tiết các cột:***
  - age: tuổi của người bệnh
  - sex: giới tính của người bệnh gồm 2 giá trị: 1 và 0. Giá trị 1 được gắn cho nam giới và giá trị 0 là nữ giới.
  - cp: mức độ đau ngực của người bệnh
  - trestbps: huyết áp lúc nghỉ ngơi của người bệnh
  - chol: cholestoral trong huyết thanh tính bằng mg/dl
  - fbs: đường huyết lúc đói của người bệnh (120mg/dl). Bao gồm 2 giá trị: 0 và 1. Giá trị 0 đại diện cho false và 1 là true
  - restecg: kết quả điện tâm đồ khi nghỉ ngơi
  - thalach: nhịp tim tối đa đạt được
  - exang: đau thắt ngực do tập thể dục. Cột này cũng có 2 giá trị là 0 và 1 tương ứng với false và true
  - oldpeak: trầm cảm gây ra do tập thể dục liên quan đến nghỉ ngơi
  - slope: độ dốc của đoạn ST tập thể dục cao điểm
  - ca: số lượng các tàu chính được tô màu bởi flour
  - thal: 1 - bình thường; 2 - khuyết tật cố định; 3 - khiếm khuyết có thể đảo ngược
  - target: kết quả mắc bệnh với 2 giá trị và 0 đại diện cho 2 giá trị tương ứng là có và không. Cột target là thuộc tính nhãn cần dự đoán. Ngoài cột target thì tất cả các cột còn lại là cột thuộc tính mô tả.
  - *Thông tin/ mô tả bài toán*

Bài toán được xây dựng dựa trên một tập dataset với thông tin các chỉ số của người bệnh và một cột mang tính kết luận có mắc bệnh hay không mắc bệnh. Bằng việc sử dụng các mô hình phân loại, Nhóm sẽ tiến hành xây dựng các mô hình và xem, nghiệm thu kết quả.

```
from google.colab import files
uploaded = files.upload()
```

Hình 2: Import dữ liệu

```
[ ] import pandas as pd
    df=pd.read_csv('heart.csv')
    df
```

Hình 3: upload file 'heart.csv'

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

1025 rows × 14 columns

Hình 4: Kết quả dữ liệu

- Tiến hành phân tích các thuộc tính của bộ dữ liệu:

```
[ ] df.columns
```

Hình 5: Các nhãn cột trong dữ liệu

```
Index(['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
       'exang', 'oldpeak', 'slope', 'ca', 'thal', 'target'],
      dtype='object')
```

Hình 6: Kết quả hiển thị các cột

```
df.isna().sum()
```

Hình 7: Kiểm tra dữ liệu thiếu

Kết quả: Kết quả không có dữ liệu thiếu, toàn bộ đều là dữ liệu sạch

```
age         0
sex         0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
target      0
dtype: int64
```

Hình 8: Kết quả kiểm tra dữ liệu thiếu

```
df.info()
```

Hình 9: Kiểm tra thông tin về các cột

Kết quả: Hầu hết các cột đều được mã hóa thành số

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   age       1025 non-null   int64  
 1   sex       1025 non-null   int64  
 2   cp        1025 non-null   int64  
 3   trestbps  1025 non-null   int64  
 4   chol      1025 non-null   int64  
 5   fbs       1025 non-null   int64  
 6   restecg   1025 non-null   int64  
 7   thalach   1025 non-null   int64  
 8   exang     1025 non-null   int64  
 9   oldpeak   1025 non-null   float64
 10  slope     1025 non-null   int64  
 11  ca        1025 non-null   int64  
 12  thal      1025 non-null   int64  
 13  target    1025 non-null   int64  
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

Hình 10: Kết quả thông tin của các cột

```
df.describe()
```

Hình 11: Thống kê mô tả

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.00000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000
mean	54.434146	0.695610	0.942439	131.611707	246.00000	0.149268	0.529756	149.114146	0.336585	1.071512	1.385366	0.754146	2.323902	0.513171
std	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	0.527878	23.005724	0.472772	1.175053	0.617755	1.030798	0.620660	0.500070
min	29.000000	0.000000	0.000000	94.000000	126.00000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	48.000000	0.000000	0.000000	120.000000	211.00000	0.000000	0.000000	132.000000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
50%	56.000000	1.000000	1.000000	130.000000	240.00000	0.000000	1.000000	152.000000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	275.00000	0.000000	1.000000	166.000000	1.000000	1.800000	2.000000	1.000000	3.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.00000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000	1.000000

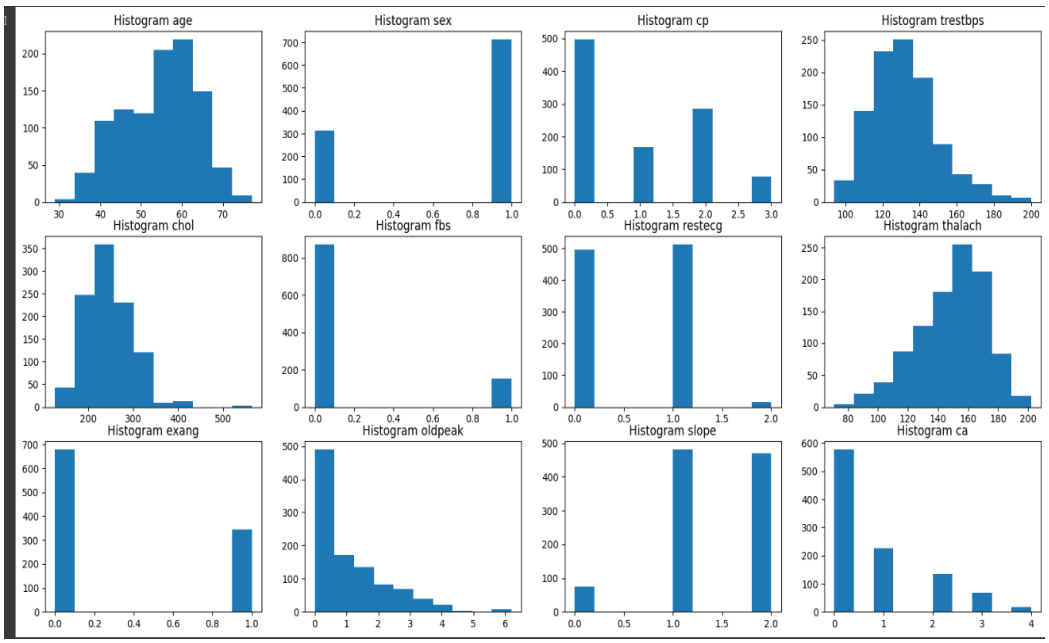
Hình 12: Kết quả thống kê mô tả

```python
import matplotlib.pyplot as plt
import numpy as np
figure, axis = plt.subplots(3, 4, figsize=(20,10))
axis[0, 0].hist(df['age'])
axis[0, 0].set_title("Histogram age")

axis[0, 1].hist(df['sex'])
axis[0, 1].set_title("Histogram sex")

axis[0, 2].hist(df['cp'])
axis[0, 2].set_title("Histogram cp")

axis[0, 3].hist(df['trestbps'])
axis[0, 3].set_title("Histogram trestbps")

axis[1, 0].hist(df['chol'])
axis[1, 0].set_title("Histogram chol")

axis[1, 1].hist(df['fbs'])
axis[1, 1].set_title("Histogram fbs")

axis[1, 2].hist(df['restecg'])
axis[1, 2].set_title("Histogram restecg")

axis[1, 3].hist(df['thalach'])
axis[1, 3].set_title("Histogram thalach")

axis[2, 0].hist(df['exang'])
axis[2, 0].set_title("Histogram exang")


axis[2, 1].hist(df['oldpeak'])
axis[2, 1].set_title("Histogram oldpeak")

axis[2, 2].hist(df['slope'])
axis[2, 2].set_title("Histogram slope")

axis[2, 3].hist(df['ca'])
axis[2, 3].set_title("Histogram ca")
```

```
Text(0.5, 1.0, 'Histogram ca')
```

Hình 13: Khai báo thư viện

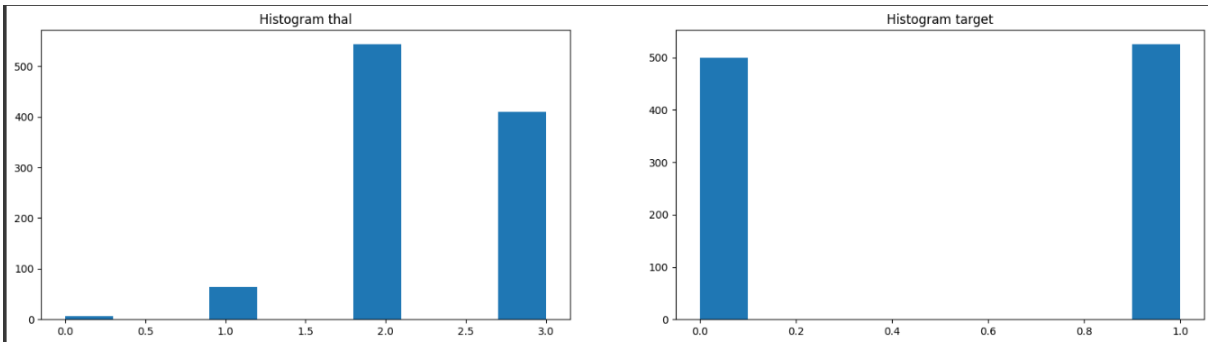
- Mỗi chỉ số đều được hiển thị biểu đồ Histogram tương ứng



Hình 14: Biểu đồ Histogram

```
figure, axis = plt.subplots (1 , 2, figsize=(20, 5))
axis[0].hist (df['thal'])
axis[0].set_title("Histogram thal")

axis[1].hist(df['target'])
axis[1].set_title("Histogram target")


Text(0.5, 1.0, 'Histogram target')
```

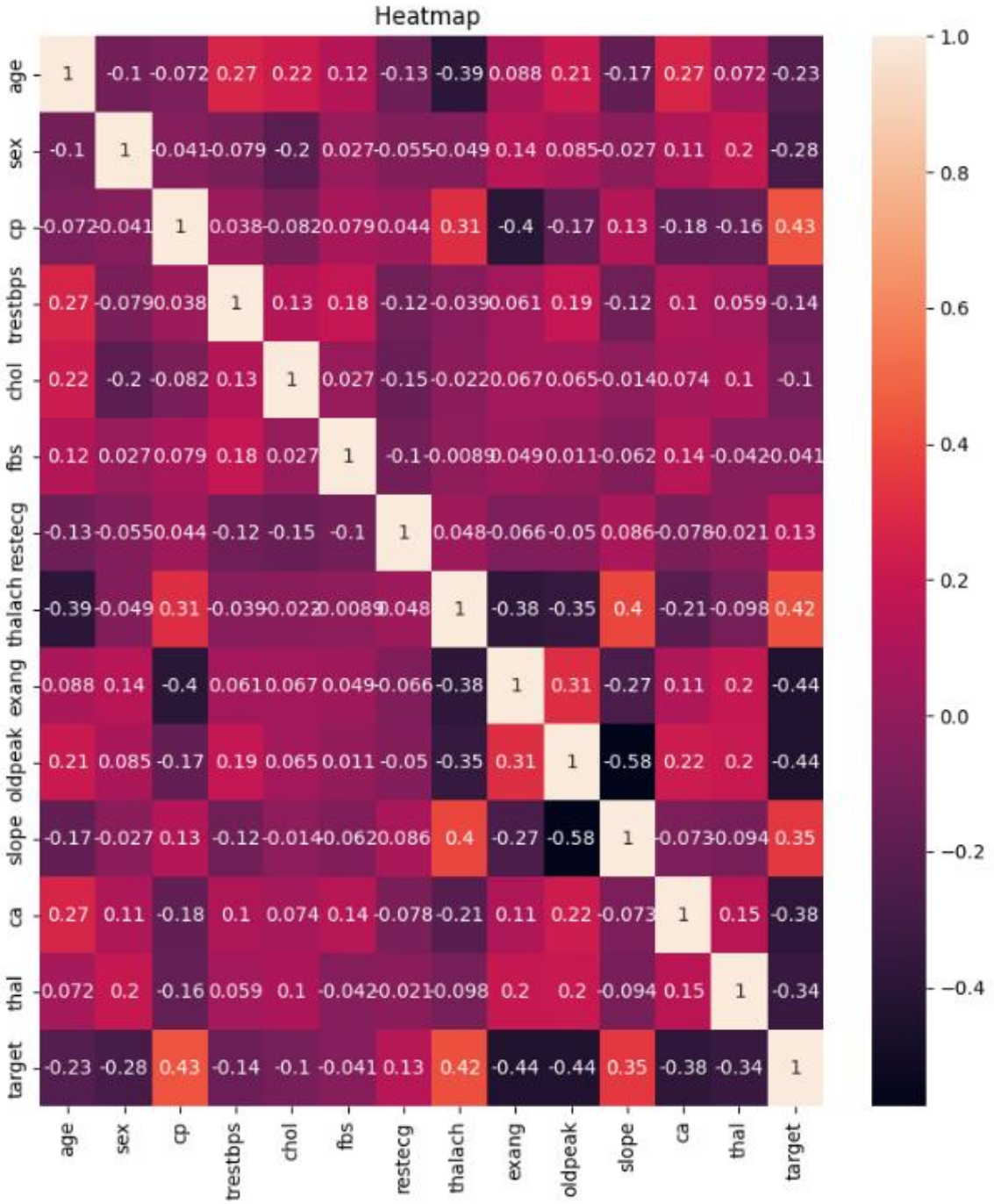
Hình 15: Khai báo dữ liệu target và thal



Hình 16: Hiển thị biểu đồ histogram cột target và thal

```
#trực quan độ tương quan giữa các cột
import seaborn as sns
figure, axis = plt.subplots (1, 2, figsize=(20, 10))
axis[0] = sns.heatmap(df.corr (), annot = True)
axis[0].set_title("Heatmap ")
# sns,heatmap (df . corr (), annot - True)

Text(0.5, 1.0, 'Heatmap ')
```

Hình 17: Trực quan hóa độ tương quan giữa các cột



Hình 18: Biểu đồ thể hiện trực quan hóa

- ***Tiền xử lý dữ liệu:***
  + Với cột target là cột cần dự đoán (thuộc tính nhãn) và các cột còn lại sẽ là các đặc trưng đầu vào (thuộc tính mô tả)
  + Đối với phần tiền xử lí dữ liệu: đây là bộ dữ liệu sạch, ở phần tìm hiểu dữ liệu, có thể thấy bộ dữ liệu đã được mã hóa kèm theo đó là không có giá trị thiếu (missing value)
  + Để xây dựng được mô hình cho kết quả tốt cần phải có các bước chuẩn bị tốt. Đối với bài toán dự đoán khả năng mắc bệnh của người dân (không hoặc là có) thì việc cân

bằng số nhãn sẽ giúp việc huấn luyện mô hình chính xác hơn, tránh trường hợp overfitting. Như biểu đồ histogram vẽ phần bên trên có thể thấy 2 nhãn 0 và 1 chênh lệch không nhiều nhưng để đảm bảo tính chính xác cho mô hình, Nhóm quyết định sử dụng kỹ thuật updersampling data - có nghĩa là lấy số lượng 2 nhãn bằng nhau.

+ Việc cho hết các thuộc tính mô tả vào mô hình không phải là cách hay. Cách đơn giản nhất để lựa chọn đặc trưng đầu vào có hiệu quả cho mô hình là sử dụng độ tương quan, ở đồ thị heatmap bên trên cho thấy độ tương quan giữa các cột không thực sự cao nên cách này không khả thi. Nhóm quyết định sử dụng kỹ thuật feature selection để chọn lọc các đặc trưng tốt với mục đích tăng độ chính xác của mô hình lên cao nhất có thể.

```
[ ]  df['target'].value_counts()
```

Hình 19: Cân bằng dữ liệu

```
1    526
0    499
Name: target, dtype: int64
```

Hình 20: Kết quả cân bằng

```
df_class_0 = df[df['target']==0]
df_class_1 = df[df[ 'target' ]==1]
```

Hình 21: Lọc kết quả hai cột target

```
[ ]  df_class_0_under = df_class_0.sample (499)
     # sau khi lấy mẫu bằng nhau xong, tiến hành hợp nhất dữ liệu
     concat_df = pd.concat ( [df_class_1, df_class_0_under]) # hiện thị vài dòng dữ liệu để xem kết quả sau khi hợp nhất
     concat_df.head ()
```

Hình 22: Lấy mẫu và hợp nhất dữ liệu

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
5	58	0	0	100	248	0	0	122	0	1.0	1	0	2	1
10	71	0	0	112	149	0	1	125	0	1.6	1	0	2	1
12	34	0	1	118	210	0	1	192	0	0.7	2	0	2	1
15	34	0	1	118	210	0	1	192	0	0.7	2	0	2	1
16	51	0	2	140	308	0	0	142	0	1.5	2	1	2	1

Hình 23: Kết quả lấy mẫu và hợp nhất

```
#Tiến hành sử dụng kỹ thuật feature selection để chọn lọc, trích xuất đặc trưng
#Cần lấy các cột target (thuộc tính nhãn)
labels = concat_df['target']
value = concat_df[['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach', 'exang',
#Khai báo thư viện
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import chi2
#Tiến hành sử dụng kỹ thuật feature selection để chọn lọc, trích xuất đặc trưng
bestfeatures = SelectKBest(score_func=chi2, k = 'all')
fit = bestfeatures.fit(value, labels)
dfscores = np.array(fit.scores_)
dfcolumns = np.array(value.columns)
#Tạo data frame để xem dữ liệu dễ hơn
featureScores = pd.DataFrame({"Feature": dfcolumns, "Score": dfscores})
featureScores['Score']=featureScores['Score'].apply(lambda x : round(x,2))
featureScores = featureScores.sort_values(['Score'], ascending = False)
featureScores
```

Hình 24: Sử dụng kỹ thuật feature selection

	Feature	Score
7	thalach	650.01
9	oldpeak	253.65
2	cp	217.82
11	ca	210.63
8	exang	130.47
4	chol	110.72
0	age	81.43
3	trestbps	45.97
10	slope	33.67
1	sex	24.37
12	thal	19.37
6	restecg	9.74
5	fbs	1.48

Hình 25: Kết quả dữ liệu

- **Nhận xét:** Sau khi dùng kĩ thuật feature selection có thể thấy các cột: thalach, cp, ca có kết quả tương đối cao hơn so với phần còn lại nên quyết định chọn cột đó làm dữ liệu đầu vào cho bài toán.

CHƯƠNG III. CÀI ĐẶT, THÍ NGHIỆM VÀ DEMO THUẬT TOÁN

1. Vấn đề tồn tại

Viện tim mạch Quốc gia bệnh viện Bạch Mai có tiền thân là khoa tim mạch bệnh viện Bạch Mai, thành lập ngày 11 tháng 11 năm 1989 với chức năng số 1 là khám và điều trị các bệnh tim mạch, với ứng dụng kỹ thuật cao bao gồm 3 lĩnh vực tim mạch: nội khoa, can thiệp và phẫu thuật ở cả hai đối tượng người lớn và trẻ em. Viện đã đào tạo và chỉ đạo tuyến với trách nhiệm cao nhất của viện tim mạch đầu ngành, nghiên cứu khoa học với lượng lớn đề tài cấp nhà nước cấp bộ và công tác dự phòng phòng các bệnh lý tim mạch.

Mặc dù Viện đã chủ động có sự hợp tác quốc tế chặt chẽ và đầy hiệu quả với nhiều viện tim mạch và các giáo sư bác sĩ chuyên ngành trong lĩnh vực, nhờ đó mà nhiều bác sĩ trẻ của Viện đã được đào tạo với những kỹ thuật tiên tiến về tim mạch. Bên cạnh đó, Viện cũng được trang bị nhiều máy móc hiện đại, trang thiết bị hiện đại. Tuy nhiên với số lượng bệnh nhân ngày càng đông đúc và quá tải như hiện nay, những máy móc hiện đại như trên cũng không thể đáp ứng được hết nhu cầu khám chữa bệnh của người dân. Từ đó có thể làm giảm uy tín và doanh thu Viện.

2. Lý do chọn đề tài

Bệnh tim mạch là một trong những nguyên nhân gây tử vong hàng đầu trên thế giới, với tỷ lệ tử vong cao hơn cả bệnh lý ung thư. Theo thống kê của Tổ chức Y tế Thế giới (WHO), vào năm 2022 có 19.5 triệu ca tử vong vì bệnh tim mạch trên toàn cầu, chiếm khoảng 1/3 ca tử vong do mọi nguyên nhân. Tại Việt Nam, mỗi năm có khoảng 200.000 người Việt tử vong vì bệnh tim mạch, chiếm 33% ca tử vong.

Trước đây, bệnh tim được cho là chỉ xuất hiện ở người lớn tuổi, tuy nhiên các vấn đề tim mạch có thể xảy ra ở bất kỳ ai và bất kỳ lứa tuổi nào. Trên thực tế, tần suất mắc bệnh ở người trẻ và trung niên cao hơn chúng ta nghĩ và đang ngày càng gia tăng. Nguyên nhân của bệnh lý tim mạch thường là do hút thuốc, béo phì, ít vận động, căng thẳng (stress), tăng huyết áp, chế độ ăn nhiều muối, chất béo, rượu bia,.. Đây là các chỉ số mà AI có thể phát hiện sự bất thường và giúp ích trong việc chẩn đoán các bệnh lý về tim mạch.

Việc phát hiện sớm bệnh tim là chìa khóa để có thể điều trị kịp thời và hiệu quả. Tuy nhiên, điều này không phải lúc nào cũng dễ dàng, đặc biệt là ở những giai đoạn đầu khi bệnh chưa có biểu hiện rõ ràng. Ngày nay, dưới sự phát triển của y học cùng với sự giúp sức của trí tuệ nhân tạo đã mở ra một kỷ nguyên mới trong việc chẩn đoán và điều trị bệnh. Trong đó, việc áp dụng các mô hình trí tuệ nhân tạo trong dự đoán khả năng mắc bệnh nhận được

rất nhiều sự quan tâm. Việc ứng dụng trí tuệ nhân tuệ trong khám chữ bệnh của Viện tim mạch quốc gia bệnh viện Bạch Mai sẽ giúp giúp nâng cao hiệu quả và chất lượng khám chữa bệnh.

Nhận thức được tầm quan trọng của việc này, Nhóm đã quyết định chọn đề tài "Xây dựng mô hình máy học dự đoán khả năng mắc bệnh tim mạch của người dân" để nghiên cứu. Mô hình này sử dụng dữ liệu về các yếu tố nguy cơ tim mạch, chẳng hạn như tuổi tác, các chỉ số huyết áp, đường huyết... để dự đoán khả năng mắc bệnh tim mạch của một người.

3. Tiến hành xây dựng mô hình

3.1. Các bước xây dựng

```
[ ]  #khai báo thư viện
     from sklearn.model_selection import train_test_split
     from sklearn.naive_bayes import GaussianNB
     import pandas as pd
```

Hình 26: Khai báo thư viện Naive Bayes

```
[ ]  #chia tập dữ liệu thành 2 tập train và test với tỷ lệ 7:3
     x = concat_df[['thalach', 'oldpeak', 'cp', 'ca']]
     y = concat_df['target']
     x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.3, random_state=42)
```

Hình 27: Chia tập dữ liệu

```
[ ]  #train mô hình Naive Bayes
     gaussNB_model = GaussianNB().fit(x_train, y_train)
```

Hình 28: Train mô hình

```
[ ]  #nghiệm thu kết quả mô hình trên tập train
     pre_y = gaussNB_model.predict(x_train)
     target_names = ['class 0', 'class 1']
     print (classification_report(y_train, pre_y, target_names=target_names))
```

Hình 29: Nghiệm thu kết quả tập train

	precision	recall	f1-score	support
class 0	0.82	0.74	0.78	355
class 1	0.77	0.84	0.80	362
accuracy			0.79	717
macro avg	0.79	0.79	0.79	717
weighted avg	0.79	0.79	0.79	717

Hình 30: Kết quả nghiệm thu tập train

```
#model gaussNB thông số kỹ thuật trên tập test
pre_test1 = gaussNB_model.predict(x_test)
target_names = ['class 0', 'class 1']
print (classification_report(y_test, pre_test1, target_names=target_names))
```

Hình 31: Model gaussNB

	precision	recall	f1-score	support
class 0	0.85	0.72	0.78	144
class 1	0.78	0.89	0.83	164
accuracy			0.81	308
macro avg	0.82	0.80	0.80	308
weighted avg	0.81	0.81	0.81	308

Hình 32: Kết quả thông số kỹ thuật trên tập test

```
#khai báo thư viện
from sklearn.metrics import classification_report
from sklearn.metrics import ConfusionMatrixDisplay
import matplotlib.pyplot as plt
```

Hình 33: Khai báo thư viện

```
y_true = np.array([0, 1, 2, 2, 1])
y_pred = np.array([0, 2, 1, 1, 2])
```

Hình 34: Đánh giá hiệu xuất mô hình

```
# Print the values of y_true and y_pred
print(y_true)
print(y_pred)

# Check their data types
print(type(y_true))
print(type(y_pred))
```

Hình 35: In và kiểm tra dữ liệu

```
[0 1 2 2 1]
[0 2 1 1 2]
<class 'numpy.ndarray'>
<class 'numpy.ndarray'>
```

Hình 36: Kết quả in

```
import matplotlib.pyplot as plt
```

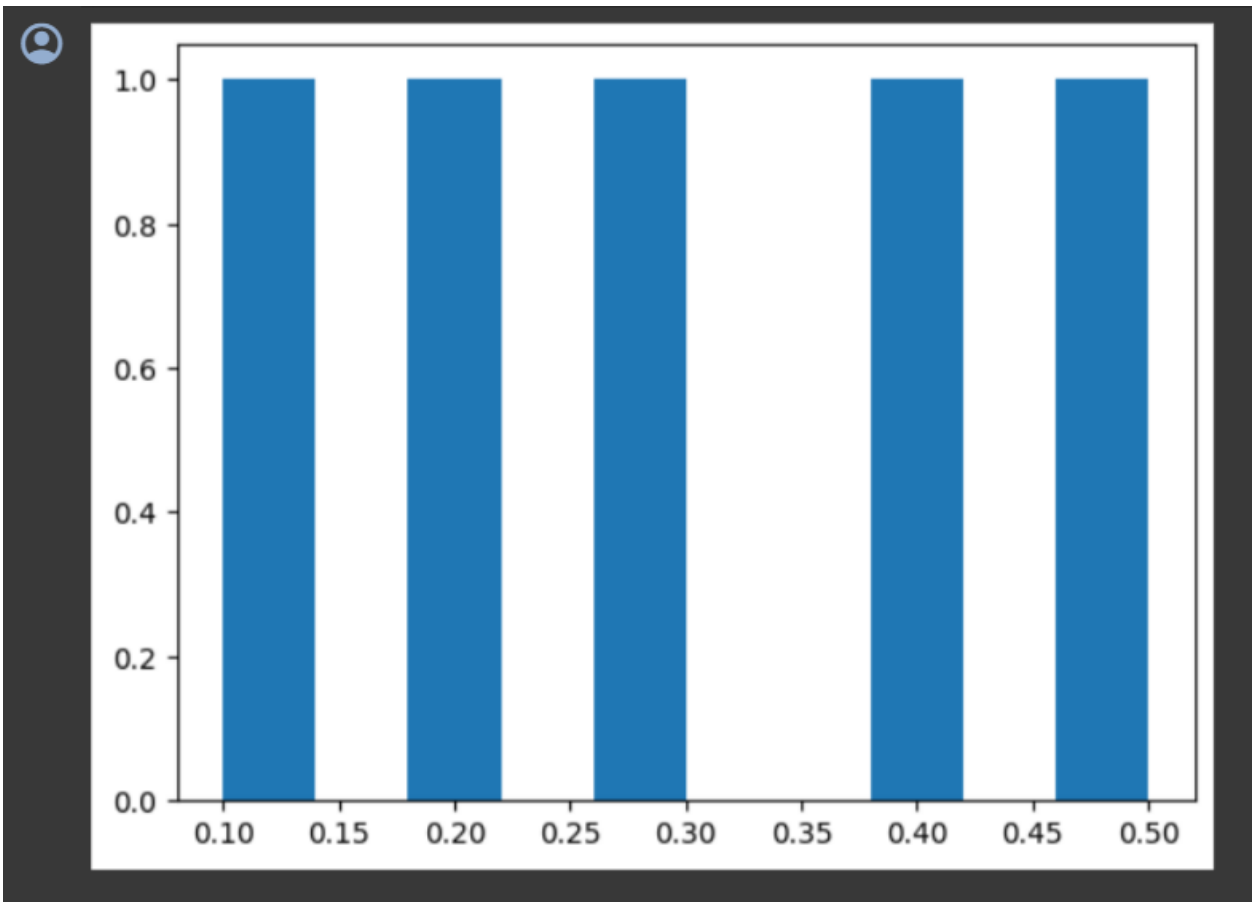
Hình 37: Khai báo thư viện

```
# Replace this line with the code that defines the `probability` variable
# based on your specific data and calculations.
probability = [0.1, 0.2, 0.3, 0.4, 0.5]
```

Hình 38: Gán danh sách các giá trị cho biến

```
plt.hist(probability)
plt.show()
```

Hình 39: Vẽ biểu đồ histogram



Hình 40: Hiển thị biểu đồ histogram

3.2. **Kết luận**

Dựa vào kết quả mô hình dự đoán bệnh tim của người dân để đánh giá độ chính xác dựa trên bộ dữ liệu huấn luyện với 70% bộ dữ liệu gốc và bộ dữ liệu kiểm tra là 30% bộ dữ liệu gốc. Khả năng dự đoán của mô hình Naive Bayes thông qua tập train và tập test, có thể thấy đối với bài toán dự đoán người dân mắc bệnh tim, mô hình Naive Bayes cho kết quả dự đoán khá cao với độ chính xác trên tập train 79% và tập test của mô hình là: 81%.

Bên cạnh đó, người dùng có thể tùy biến cấu hình thuật toán để mang lại kết quả dự đoán với tỉ lệ chính xác cao hơn thông qua việc thay đổi thông số/ tỉ lệ ở tập train và tập test, thông qua các bước sau:

Bước 1. Tìm hiểu tập dữ liệu: Bước đầu tiên là tìm hiểu tập dữ liệu, bao gồm các thuộc tính và nhãn của dữ liệu. Điều này sẽ giúp người dùng hiểu được các thuộc tính nào có liên quan đến nhãn và cách các thuộc tính này phân bố trong tập dữ liệu.

Bước 2. Chọn phân phối xác suất: Naive Bayes giả định rằng các thuộc tính phân phối độc lập với nhau. Điều này có nghĩa là xác suất của một thuộc tính không phụ thuộc vào giá trị của các thuộc tính khác. Có nhiều loại phân phối xác suất khác nhau có thể được sử dụng trong naive bayes. Một số loại phân phối phổ biến bao gồm:

- Phân phối chuẩn (Gaussian)
- Phân phối nhị phân (Bernoulli)
- Phân phối mũ (Poisson)

Bước 3. Tính toán các tham số: Sau khi chọn phân phối xác suất, cần tính toán các tham số của phân phối cho từng thuộc tính. Các tham số này có thể được tính toán bằng cách sử dụng tập dữ liệu train.

Bước 4. Xây dựng mô hình: Khi đã tính toán các tham số, có thể xây dựng mô hình naive bayes. Mô hình này sẽ sử dụng các tham số để tính toán xác suất của một điểm dữ liệu thuộc về mỗi nhãn.

Bước 5. Đánh giá mô hình: Sau khi xây dựng mô hình, cần tiến hành đánh giá hiệu suất của mô hình trên tập dữ liệu test. Điều này sẽ xác định xem mô hình có thể dự đoán chính xác nhãn của các điểm dữ liệu mới hay không.

Dưới đây là một số phương pháp cụ thể để tùy biến cấu hình thuật toán Naive Bayes trên tập train và tập test:

1. **Chọn phân phối xác suất phù hợp:** với tập dữ liệu sẽ giúp mô hình đạt được hiệu suất tốt hơn. Bạn có thể sử dụng các kỹ thuật như phân tích phân phối xác suất hoặc kiểm định giả thuyết để chọn phân phối xác suất phù hợp.
2. **Chọn giá trị tối ưu cho các tham số:** Các tham số của phân phối xác suất có thể ảnh hưởng đáng kể đến hiệu suất của mô hình, vì vậy có thể sử dụng các kỹ thuật như tối ưu hóa hoặc học máy để tìm giá trị tối ưu cho các tham số.
3. **Tăng kích thước tập dữ liệu train:** Tập dữ liệu train càng lớn, mô hình càng có thể học được các mối quan hệ phức tạp giữa các thuộc tính và nhãn. Điều này có thể giúp mô hình đạt được hiệu suất tốt hơn.
4. **Sử dụng các kỹ thuật điều chỉnh:** Các kỹ thuật điều chỉnh có thể giúp cải thiện hiệu suất của mô hình trên tập dữ liệu test.

⇒ Tóm lại, việc tùy biến cấu hình thuật toán Naïve Bayes trên tập train và tập test có thể giúp bạn cải thiện hiệu suất của mô hình.

3.3. **Nhận xét**

Độ chính xác của mô hình Naive Bayes trên tập train là 79%, nghĩa là trong 100 mẫu dữ liệu train, mô hình dự đoán chính xác 79 mẫu. Độ chính xác của mô hình trên tập test là 81%, nghĩa là trong 100 mẫu dữ liệu test, mô hình dự đoán chính xác 81 mẫu.

Trên tập test mô hình có độ chính xác (accuracy) là 0.79, cho thấy mô hình khá chính xác nhưng không hoàn hảo. Giá trị trung bình macro và trọng số cho precision, recall và f1-score đều xấp xỉ 0.78, cho thấy mô hình có sự cân đối giữa các lớp. Tuy nhiên, cần lưu ý rằng mặc dù các chỉ số này cho thấy mô hình hoạt động tốt, nhưng chúng ta cũng cần xem xét các yếu tố khác như độ tin cậy của dữ liệu, sự cân đối giữa các lớp và tầm quan trọng của việc dự đoán chính xác từng lớp trong bối cảnh cụ thể. Trong trường hợp của bệnh tim, việc dự đoán chính xác lớp 1 (mắc bệnh) có thể quan trọng hơn nhiều so với việc dự đoán chính xác lớp 0 (không mắc bệnh).

Việc độ chính xác trên tập train thấp hơn độ chính xác trên tập test là điều bình thường, bởi vì mô hình được huấn luyện trên tập train, nên nó sẽ có xu hướng dự đoán chính xác các mẫu trong tập train. Tuy nhiên, nếu độ chính xác trên tập test thấp hơn đáng kể so với độ chính xác trên tập train, thì có thể mô hình đang bị overfitting. Overfitting là tình trạng mô hình quá khớp với tập dữ liệu huấn luyện, dẫn đến việc mô hình không thể dự đoán chính xác các mẫu dữ liệu mới.

Trong trường hợp này, độ chính xác trên tập test chỉ cao hơn độ chính xác trên tập train 2%, nên có thể nói rằng mô hình không bị overfitting nghiêm trọng. Tuy nhiên, để cải thiện độ chính xác của mô hình, có thể thực hiện một số biện pháp sau:

- Tăng kích thước của tập dữ liệu huấn luyện.
- Sử dụng các kỹ thuật giảm thiểu overfitting, chẳng hạn như regularization hoặc data augmentation.
- Chọn tham số mô hình phù hợp hơn.

Ngoài ra, cũng cần lưu ý rằng độ chính xác của mô hình phụ thuộc vào nhiều yếu tố khác, chẳng hạn như chất lượng của tập dữ liệu, độ phức tạp của mô hình, và các đặc trưng được sử dụng.

3.4. **Đánh giá**

+ Ưu điểm:

    - Đơn giản và hiệu quả: Thuật toán Naive Bayes đơn giản và dễ hiểu. Nó hoạt động nhanh và hiệu quả với các tập dữ liệu có số chiều lớn.

- Hiệu suất tốt: Dựa trên các chỉ số hiệu suất từ hình ảnh, mô hình Naive Bayes cho thấy hiệu suất khá tốt trong việc phân loại bệnh tim. Độ chính xác cao, theo biểu đồ trong hình, thuật toán Naive Bayes có thể chẩn đoán bệnh tim với độ chính xác trung bình là 0,79. Đây là một con số khá cao, cao hơn đáng kể so với tỷ lệ chẩn đoán bệnh tim theo chẩn đoán lâm sàng (khoảng 0,5).
- Tốc độ nhanh: Thuật toán Naive Bayes có thể hoạt động nhanh chóng, giúp tiết kiệm thời gian và chi phí.

+ Nhược điểm:

- Giả định về độc lập: Naive Bayes hoạt động dựa trên giả định rằng tất cả các đặc trưng đều độc lập với nhau. Trong thực tế, điều này không phải lúc nào cũng đúng và có thể dẫn đến những sai lệch trong dự đoán.
- Độ chính xác và độ phủ không cân đối: Dựa trên dữ liệu, có thể thấy rằng độ chính xác và độ phủ giữa hai lớp không cân đối. Điều này có thể dẫn đến việc mô hình dự đoán chính xác hơn cho một lớp so với lớp kia.
- Khả năng giải thích thấp: Thuật toán Naive Bayes có khả năng giải thích thấp. Điều này có thể gây khó khăn trong việc hiểu lý do tại sao thuật toán đưa ra một kết quả nhất định.

4. Tổng kết

- Bệnh tim mạch là một trong những nhóm bệnh lý nguy hiểm được ví như sát thủ thầm lặng đe dọa đến sự phát cá nhân và kinh tế xã hội. Việc cải thiện chất lượng cuộc sống từ bữa ăn cho đến thói quen sinh hoạt hàng ngày và đặc biệt là trang bị kiến thức về vấn đề dinh dưỡng, lối sống lành mạnh vô cùng cần thiết để phòng ngừa bệnh.
- Bệnh tim mạch mặc dù là bệnh lý nguy hiểm và có nguy cơ tử vong cao nhưng có thể sàng lọc và phát hiện được, vì vậy mỗi người cần phải khám sức khỏe định kỳ, sàng lọc các yếu tố nguy cơ về bệnh tim mạch, nhất là những người lớn tuổi. Do các bệnh lý tim mạch đa số không có biểu hiện và thường diễn biến âm thầm, nên khi nhận thấy các dấu hiệu bất thường như khó thở, đau ngực, hồi hộp, choáng, ngất, thay đổi huyết áp, đánh trống ngực,... thì cần đến ngay các cơ sở y tế để được khám và điều trị kịp thời.
- Naive Bayes là một thuật toán dựa trên định lý Bayes về lý thuyết xác suất để đưa ra các phán đoán cũng như phân loại dữ liệu dựa trên các dữ liệu được quan sát và thống kê, được ứng dụng rất nhiều trong các lĩnh vực Machine Learning (ML) dùng để đưa

các dự đoán chính xác nhất dựa trên một tập dữ liệu đã được thu thập, vì nó khá dễ hiểu và độ chính xác cao. Việc sử dụng phần mềm dự đoán sẽ giúp hỗ trợ chẩn đoán nhanh chóng và chính xác hơn giảm thiểu sai sót của con người cũng như chi phí mua máy móc thiết bị dự đoán.

TÀI LIỆU THAM KHẢO

[1] David Lapp. (2018). *Heart Disease Dataset*. Truy xuất từ: https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset/data

[2] Nguyen Thi Hop. (2019). *Thuật toán phân lớp Naïve Bayes. VIBLO*. Truy xuất từ: https://viblo.asia/p/thuat-toan-phan-lop-naive-bayes-924lJWPm5PM

[3] Pham Van Toan. (2017). *Ứng dụng thuật toán Naïve Bayes trong giải quyết bài toán chuẩn đoán bệnh tiểu đường*. Truy xuất từ: https://viblo.asia/p/ung-dung-thuat-toan-naive-bayes-trong-giai-quyet-bai-toan-chuan-doan-benh-tieu-duong-eW65GYejZDO

[4] Science Alert. (2011). *Nested Circles Boundary Algorithm for Rotated Texture Classification.* Truy xuất từ: https://scialert.net/fulltext/?doi=jas.2011.3351.3361

[5] Whitehat. (2020). *Thuật toán phân loại Naive Bayes và ứng dụng.* Truy xuất từ: https://whitehat.vn/threads/thuat-toan-phan-loai-naive-bayes-va-ung-dung.13775/

[6] Huynh Minh Tan. (2018). *Thuật toán Naive Bayes dùng trong Text Classification với thư viện Scikit-learn.* Truy xuất từ: https://github.com/huynhminhtan/thuattoanthongminh/blob/master/NaiveBayes_Sklearn.md

[7] ScienceDirect. (2019). *Stencil selection algorithms for WENO schemes on unstructured meshes.* Truy xuất từ: https://www.sciencedirect.com/science/article/pii/S2590055219300538

[8] Huynh Chi Trung. (2020). *Phân lớp với Navie Bayes Classification - Mô hình và ứng dụng.* Truy xuất từ: https://viblo.asia/p/phan-lop-voi-navie-bayes-classification-mo-hinh-va-ung-dung-WAyK8PRkKxX

[9] machinelearningcoban. (2017). *Naive Bayes Classifier*. Truy xuất từ: https://machinelearningcoban.com/2017/08/08/nbc/

-- Hết --