A black and white border

Description automatically generated **BỘ NÔNG NGHIỆP VÀ PHÁT TRIỂN NÔNG THÔN**

**PHÂN HIỆU TRƯỜNG ĐẠI HỌC THỦY LỢI**

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----

A blue and white logo

Description automatically generated

**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**Đề tài**

***XÂY DỰNG MÔ HÌNH DỰ ĐOÁN NGUY CƠ ĐAU TIM Ở CON NGƯỜI***

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | ***TS. Hoàng Văn Quý*** |
| **Sinh viên thực hiện:** | **Nguyễn Thị Thanh Mai**  **Nguyễn Thị Kim Ngọc** |
|  | |
|  | |

**BỘ NÔNG NGHIỆP VÀ PHÁT TRIỂN NÔNG THÔN**

**PHÂN HIỆU TRƯỜNG ĐẠI HỌC THỦY LỢI**

Họ và tên

Nguyễn Thị Thanh Mai

Nguyễn Thị Kim Ngọc

**XÂY DỰNG MÔ HÌNH DỰA ĐOÁN NGUY CƠ ĐAU TIM Ở CON NGƯỜI**

Ngành: CÔNG NGHỆ THÔNG TIN

NGƯỜI HƯỚNG DẪN: TS. Hoàng Văn Quý

HỒ CHÍ MINH, NĂM 2024

**MỤC LỤC**

[MỞ ĐẦU 6](#_Toc178084758)

[CHƯƠNG 1 TỔNG QUAN KHAI PHÁ DỮ LIỆU 8](#_Toc178084759)

[1.1 Giới thiệu về Data Minning 8](#_Toc178084760)

[1.2 Các thuật toán sử dụng trong Học Máy. 9](#_Toc178084761)

[1.2.1 Học có giám sát (Supervised Learning) 9](#_Toc178084762)

[1.2.2 Học không giám sát (Unsupervised Learning) 10](#_Toc178084763)

[1.2.3 Học tăng cường (Reinforcement Learning) 11](#_Toc178084764)

[1.3 Bộ dữ liệu cho các thuật toán Học Máy. 11](#_Toc178084765)

[1.3.1 Học có giám sát (Supervised Learning) 11](#_Toc178084766)

[1.3.2 Học không giám sát (Unsupervised Learning) 13](#_Toc178084767)

[1.2.4 Học tăng cường (Reinforcement Learning) 14](#_Toc178084768)

[1.4 Ưu và nhược điểm các thuật toán trong Học Máy . 14](#_Toc178084769)

[1.4.1 Học có giám sát (Supervised Learning) 14](#_Toc178084770)

[1.4.2 Học không giám sát (Unsupervised Learning) 16](#_Toc178084771)

[1.4.3 Học tăng cường (Reinforcement Learning) 17](#_Toc178084772)

[CHƯƠNG 2 MÔ HÌNH HỌC MÁY 18](#_Toc178084773)

[2.1 Mô hình thuật toán 18](#_Toc178084774)

[2.1.1 Mô hình Logistic Regression 18](#_Toc178084775)

[2.1.2 Mô hình thuật toán RandomForestClassifier 20](#_Toc178084776)

[2.2 Tiền xử lý dữ liệu 21](#_Toc178084777)

[2.2.1 Mô tả dữ liệu 21](#_Toc178084778)

[2.2.2 Tiền xử lý dữ liệu 23](#_Toc178084779)

[2.2.3 Trực quan hoá dữ liệu 27](#_Toc178084780)

[2.2.4 Model và kết quả 30](#_Toc178084781)

[CHƯƠNG 3 THỰC NGHIỆM VÀ KẾT QUẢ 33](#_Toc178084782)

[3.1 Tập dữ liệu dự đoán 33](#_Toc178084783)

[3.2 Kết quả dự đoán 33](#_Toc178084784)

[3.3 Đánh giá mô hình 34](#_Toc178084785)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 34](#_Toc178084786)

# MỞ ĐẦU

Hiện nay, các vấn đề liên quan đến tim mạch đang ngày càng trở thành mối đe dọa nghiêm trọng đối với sức khỏe con người trên toàn cầu. Theo Tổ chức Y tế Thế giới (WHO), bệnh tim mạch là nguyên nhân dẫn đến nhiều ca tử vong nhất mỗi năm, vượt qua cả các bệnh lý khác như ung thư hay tiểu đường. Trong bối cảnh đó, việc phát hiện sớm và dự đoán nguy cơ đau tim trở thành một nhiệm vụ cấp bách, giúp ngăn chặn những biến chứng nghiêm trọng và giảm thiểu tỷ lệ tử vong do các bệnh liên quan đến tim mạch.

Với ý nghĩa và tầm quan trọng đó, việc xây dựng một mô hình dự đoán nguy cơ đau tim dựa trên các yếu tố sức khỏe và lối sống của con người là vô cùng thiết yếu. Bài toán này hướng đến việc phát triển các công cụ dự đoán dựa trên những yếu tố nguy cơ như mức cholesterol, huyết áp, chỉ số BMI, tình trạng tiểu đường, tiền sử gia đình, và thói quen vận động. Những thông tin này sẽ giúp dự đoán liệu một người có khả năng cao mắc các bệnh lý về tim mạch hay không.

1. **Mục tiêu đề tài**

Xây dựng một mô hình dự đoán nguy cơ đau tim ở con người dựa trên các yếu tố sức khỏe và lối sống, sử dụng các phương pháp học máy như hồi quy logistic, cây quyết định và rừng ngẫu nhiên. Đề tài sẽ thu thập và xử lý dữ liệu từ các chỉ số như: tuổi, giới tính, mức cholesterol, huyết áp, nhịp tim, tiểu đường, tiền sử gia đình, thói quen hút thuốc, béo phì, và các yếu tố lối sống khác như tập thể dục, chế độ ăn uống, và mức độ căng thẳng.

Mong muốn tạo ra một mô hình có độ chính xác cao, có khả năng dự đoán nguy cơ đau tim, đồng thời cung cấp thông tin chi tiết về các yếu tố nguy cơ để hỗ trợ bác sĩ và bệnh nhân trong việc ra quyết định điều trị và phòng ngừa.

1. **Kết quả đạt được**

Kết quả dự kiến bao gồm việc xây dựng một mô hình dự đoán chính xác nguy cơ đau tim, với các tiêu chí đánh giá hiệu quả như độ chính xác, độ nhạy, và độ đặc hiệu. Mô hình sẽ giúp phân tích các yếu tố nguy cơ chính, đồng thời đưa ra các khuyến nghị cụ thể về việc áp dụng mô hình trong thực tế y tế.

Ngoài ra, mô hình sẽ hỗ trợ các bác sĩ trong việc chẩn đoán và điều trị, đồng thời giúp bệnh nhân nhận biết và điều chỉnh các yếu tố nguy cơ của mình, góp phần giảm thiểu nguy cơ mắc bệnh tim và nâng cao chất lượng cuộc sống.

# TỔNG QUAN KHAI PHÁ DỮ LIỆU

* 1. **Giới thiệu về Data Minning**

Data mining – khai phá dữ liệu là quá trình phân loại, sắp xếp các tập hợp dữ liệu lớn để xác định các mẫu và thiết lập các mối liên hệ nhằm giải quyết các vấn đề nhờ phân tích dữ liệu. Các MCU khai phá dữ liệu cho phép các doanh nghiệp có thể dự đoán được xu hướng tương lai.

* **Các bước quan trọng khi Data Mining bao gồm:**
* Làm sạch dữ liệu – Trong bước này, dữ liệu được làm sạch sao cho không có bất thường trong dữ liệu.
* Tích hợp dữ liệu – Trong quá trình tích hợp dữ liệu, nhiều nguồn dữ liệu sẽ kết hợp lại thành một.
* Lựa chọn dữ liệu – Trong bước này, dữ liệu được trích xuất từ cơ sở dữ liệu.
* Chuyển đổi dữ liệu – Trong bước này, dữ liệu sẽ được chuyển đổi để thực hiện phân tích tóm tắt cũng như các hoạt động tổng hợp.
* Khai phá dữ liệu – Trong bước này, trích xuất dữ liệu hữu ích từ nhóm dữ liệu hiện có.
* Đánh giá mô hình
* Trình bày thông tin – Trong bước cuối cùng, thông tin sẽ được thể hiện dưới dạng cây, bảng, biểu đồ và ma trận.
* **Ứng dụng của Data Mining**
* Phân tích thị trường và chứng khoán
* Phát hiện gian lận
* Quản lý rủi ro và phân tích doanh nghiệp
* Phân tích giá trị trọn đời của khách hàng
* Khám phá thêm 10 ứng dụng khai phá dữ liệu
* **Các công cụ khai phá dữ liệu**
  1. **Các thuật toán sử dụng trong Học Máy.**

Trong Học Máy, các thuật toán phổ biến được phân chia thành ba nhóm chính:

* Học có giám sát (Supervised Learning)
* Học không giám sát (Unsupervised Learning)
* Học tăng cường (Reinforcement Learning).
  + 1. **Học có giám sát (Supervised Learning)**

**Mục đích**: Sử dụng các cặp dữ liệu đầu vào và đầu ra đã gán nhãn để huấn luyện mô hình, sau đó dự đoán kết quả cho dữ liệu mới.

* + - 1. **Hồi quy tuyến tính (Linear Regression)**
* **Mục đích**: Dự đoán các giá trị liên tục dựa trên mối quan hệ tuyến tính giữa các biến.
* **Cách thức hoạt động :** Xây dựng một hàm tuyến tính để dự đoán giá trị đầu ra liên tục. Mô hình tìm các hệ số w sao cho tổng bình phương sai số giữa giá trị dự đoán và giá trị thực tế là nhỏ nhất.
  + - 1. **Hồi quy logistic (Logistic Regression)**
* **Mục đích**: Phân loại nhị phân.
* **Cách thức hoạt động**: Dự đoán xác suất một dữ liệu thuộc về một trong hai lớp bằng cách sử dụng hàm sigmoid. Nếu xác suất lớn hơn ngưỡng (thường là 0.5), mô hình sẽ phân loại dữ liệu thuộc về lớp 1, nếu nhỏ hơn thì thuộc lớp 0.
  + - 1. **Cây quyết định (Decision Tree)**
* **Mục đích**: Phân loại hoặc dự đoán giá trị liên tục.
* **Cách thức hoạt động**: Xây dựng một cây phân cấp, trong đó mỗi nút đại diện cho một câu hỏi về một đặc trưng của dữ liệu. Mỗi nhánh đại diện cho câu trả lời và phân tách dữ liệu theo câu hỏi đó. Cây quyết định sẽ đi xuống các nhánh cho đến khi đạt đến lá, nơi chứa quyết định cuối cùng (dự đoán).
  + - 1. **Random Forest**
* **Mục đích**: Tăng cường độ chính xác của các cây quyết định bằng cách kết hợp nhiều cây.
* **Cách thức hoạt động**: Huấn luyện nhiều cây quyết định trên các tập con ngẫu nhiên của dữ liệu. Mỗi cây sẽ đưa ra một dự đoán và kết quả cuối cùng là trung bình hoặc lấy đa số phiếu từ các cây.
  + - 1. **Máy vector hỗ trợ (SVM - Support Vector Machine)**
* **Mục đích**: Phân loại hoặc hồi quy.
* **Cách thức hoạt động**: SVM tìm ra một siêu phẳng tối ưu để phân tách các lớp trong không gian đặc trưng. Nếu dữ liệu không thể phân tách tuyến tính, nó sử dụng các hạt nhân (kernel) để ánh xạ dữ liệu sang không gian cao hơn, nơi nó có thể được phân tách tuyến tính.
  + 1. **Học không giám sát (Unsupervised Learning)**

Mục đích: Tìm cấu trúc, mẫu hoặc mối quan hệ trong dữ liệu chưa có nhãn.

* + - 1. **K-means**
* Mục đích: Phân cụm dữ liệu thành kkk nhóm dựa trên sự tương đồng.
* Cách thức hoạt động: K-means bắt đầu bằng cách chọn ngẫu nhiên kkk điểm trung tâm (centroids). Sau đó, các điểm dữ liệu được gán vào cụm có điểm trung tâm gần nhất. Trung tâm của mỗi cụm được cập nhật liên tục cho đến khi các cụm ổn định.
  + - 1. **Phân cụm phân cấp (Hierarchical Clustering)**
* Mục đích: Tạo cây phân cấp các cụm dữ liệu.
* Cách thức hoạt động: Thuật toán bắt đầu từ các điểm dữ liệu riêng lẻ hoặc các cụm nhỏ nhất và kết hợp chúng thành các cụm lớn hơn dựa trên khoảng cách giữa các cụm cho đến khi tất cả dữ liệu thuộc về một cụm duy nhất. Kết quả là một cây phân cấp các cụm (dendrogram).
  + - 1. **Phân tích thành phần chính (PCA - Principal Component Analysis)**
* Mục đích: Giảm số chiều của dữ liệu mà vẫn giữ được thông tin quan trọng.
* Cách thức hoạt động: PCA tìm ra các thành phần chính (principal components), là các hướng mà dữ liệu phân tán nhiều nhất, và ánh xạ dữ liệu lên không gian chiều thấp hơn trong khi vẫn giữ được phần lớn phương sai của dữ liệu.
  + - 1. **Thuật toán học luật kết hợp (Apriori)**
* Mục đích: Tìm các mối quan hệ giữa các mục trong dữ liệu.
* Cách thức hoạt động: Thuật toán Apriori hoạt động dựa trên nguyên tắc rằng nếu một tập hợp mục thường xuyên xuất hiện, thì tất cả các tập con của nó cũng thường xuyên xuất hiện. Đầu tiên, thuật toán xác định tất cả các tập con của mục có tần suất xuất hiện lớn hơn ngưỡng tối thiểu (minimum support). Sau đó, nó tạo ra các quy tắc kết hợp dựa trên các tập con đó với chỉ số tin cậy (confidence) lớn hơn một ngưỡng quy định.
  + 1. **Học tăng cường (Reinforcement Learning)**

Mục đích: học cách hành động trong môi trường để tối đa hóa phần thưởng.

* + - 1. **Q-learning**
* Mục đích: Học cách tối ưu hóa các hành động dựa trên phần thưởng nhận được từ môi trường.
* Cách thức hoạt động: Sử dụng bảng Q (Q-table) để lưu trữ giá trị của các hành động trong các trạng thái. Tác nhân cập nhật bảng Q bằng cách học từ kinh nghiệm, tối ưu hóa các hành động dựa trên phần thưởng nhận được.
  + - 1. **Deep Q-Network (DQN)**
* Mục đích: Học cách tối ưu hóa hành động trong môi trường phức tạp sử dụng mạng nơ-ron sâu.
* Cách thức hoạt động: DQN sử dụng mạng nơ-ron để ước tính giá trị QQQ, cho phép tác nhân học từ các trạng thái phức tạp trong môi trường bằng cách sử dụng các kinh nghiệm đã lưu trữ.
  1. **Bộ dữ liệu cho các thuật toán Học Máy.**
     1. **Học có giám sát (Supervised Learning)**
        1. **Hồi quy tuyến tính (Linear Regression)**

**Bộ dữ liệu**: Thường sử dụng các bộ dữ liệu dạng bảng với biến đầu vào và biến đầu ra liên tục. Ví dụ:

* Bộ dữ liệu giá nhà (Boston Housing) để dự đoán giá nhà dựa trên các đặc trưng như diện tích, số phòng, khoảng cách đến trung tâm.
* Dự đoán doanh thu bán hàng dựa trên chi phí quảng cáo.
  + - 1. **Hồi quy logistic (Logistic Regression)**

**Bộ dữ liệu**: Sử dụng dữ liệu phân loại nhị phân, nơi đầu ra thuộc một trong hai lớp (0 hoặc 1). Ví dụ:

* Bộ dữ liệu Titanic để dự đoán hành khách sống sót hoặc không sống sót dựa trên các đặc trưng như độ tuổi, giới tính, hạng vé.
* Phân loại email thành thư rác hoặc không (Spam dataset).
  + - 1. **Cây quyết định (Decision Tree)**

**Bộ dữ liệu**: Áp dụng cho cả dữ liệu phân loại và hồi quy. Ví dụ:

* Bộ dữ liệu về các loài hoa Iris để phân loại các loài hoa dựa trên chiều dài, chiều rộng cánh hoa.
* Dự đoán giá nhà hoặc sản phẩm với dữ liệu bảng.
  + - 1. **Random Forest**

**Bộ dữ liệu**: Sử dụng cho cả phân loại và hồi quy, thường dùng cho các bộ dữ liệu lớn, phức tạp. Ví dụ:

* Bộ dữ liệu về bệnh tiểu đường (Diabetes) để dự đoán bệnh nhân có mắc bệnh hay không.
* Bộ dữ liệu ngân hàng để phân loại khách hàng dựa trên hành vi tín dụng.
  + - 1. **Máy vector hỗ trợ (SVM - Support Vector Machine)**

**Bộ dữ liệu**: Áp dụng cho các bài toán phân loại hoặc hồi quy với bộ dữ liệu dạng bảng có nhiều đặc trưng. Ví dụ:

* Phân loại hình ảnh chữ viết tay từ bộ dữ liệu MNIST.
* Phân loại email thành thư rác hoặc không.
  + 1. **Học không giám sát (Unsupervised Learning)**
       1. **K-means**

**Bộ dữ liệu**: Sử dụng cho dữ liệu không có nhãn với nhiều đặc trưng, thường áp dụng cho bài toán phân cụm. Ví dụ:

* Phân cụm khách hàng từ bộ dữ liệu chi tiêu khách hàng (Customer Segmentation).
* Phân cụm hành vi người dùng từ dữ liệu hành vi trực tuyến.
  + - 1. **Phân cụm phân cấp (Hierarchical Clustering)**

**Bộ dữ liệu**: Áp dụng cho dữ liệu không có nhãn và cần xây dựng cây phân cấp. Ví dụ:

* Phân cụm tài liệu hoặc văn bản.
* Phân tích cây phả hệ các loài sinh vật dựa trên đặc trưng di truyền.
  + - 1. **Phân tích thành phần chính (PCA - Principal Component Analysis)**

**Bộ dữ liệu**: Sử dụng cho các bộ dữ liệu có nhiều đặc trưng cần giảm số chiều. Ví dụ:

* Bộ dữ liệu về hình ảnh hoặc video (ví dụ: bộ dữ liệu hình ảnh khuôn mặt người - LFW).
* Dữ liệu tài chính có nhiều biến như giá cổ phiếu theo thời gian.
  + - 1. **Thuật toán học luật kết hợp (Apriori)**

**Bộ dữ liệu**: Sử dụng dữ liệu giao dịch với các mục mục khác nhau, thường dùng trong bán lẻ. Ví dụ:

* Bộ dữ liệu giỏ hàng siêu thị để tìm quy tắc mua hàng (Market Basket Analysis).
* Phân tích dữ liệu mua sắm trực tuyến.
  + 1. **Học tăng cường (Reinforcement Learning)**
       1. **Q-learning**

**Bộ dữ liệu**: Thường sử dụng các môi trường mô phỏng có thể tương tác. Ví dụ:

* Môi trường chơi game (Atari games) để học cách tối ưu hóa hành động.
* Mô phỏng robot trong môi trường vật lý để học cách di chuyển hoặc thực hiện nhiệm vụ.
  + - 1. **Deep Q-Network (DQN)**

**Bộ dữ liệu**: Sử dụng cho các môi trường phức tạp, cần mạng nơ-ron sâu để dự đoán giá trị. Ví dụ:

* Bộ dữ liệu từ môi trường trò chơi điện tử (Atari) hoặc cờ vây (Go).
* Mô phỏng robot hoặc điều khiển drone.
  1. **Ưu và nhược điểm các thuật toán trong Học Máy .**
     1. **Học có giám sát (Supervised Learning)**
        1. **Hồi quy tuyến tính (Linear Regression)**
* Ưu điểm:
* Dễ hiểu và thực hiện.
* Tính toán nhanh, hiệu quả với các vấn đề tuyến tính.
* Dễ dàng diễn giải trọng số của các biến đầu vào.
* Nhược điểm:
* Không phù hợp với các dữ liệu phi tuyến.
* Nhạy cảm với ngoại lệ (outliers) trong dữ liệu.
  + - 1. **Hồi quy logistic (Logistic Regression)**
* Ưu điểm:
* Đơn giản và dễ triển khai cho các bài toán phân loại nhị phân.
* Kết quả dễ diễn giải, cho biết xác suất thuộc về từng lớp.
* Hoạt động tốt với các dữ liệu tuyến tính.
* Nhược điểm:
* Không phù hợp cho các bài toán phân loại phức tạp hoặc phi tuyến.
* Cần xử lý tốt nếu dữ liệu có nhiều biến không tương quan.

**1.4.1.3 Cây quyết định (Decision Tree)**

* Ưu điểm:
  + Dễ hiểu, trực quan và giải thích kết quả dễ dàng.
  + Có thể xử lý cả dữ liệu phân loại và hồi quy.
  + Không yêu cầu giả định về mối quan hệ giữa các biến.
* Nhược điểm:
  + Dễ bị quá khớp (overfitting) với dữ liệu huấn luyện nhỏ.
  + Có thể không ổn định do sự thay đổi nhỏ trong dữ liệu.
    - 1. **Random Forest**
* Ưu điểm:
  + Giảm hiện tượng quá khớp do sử dụng nhiều cây quyết định.
  + Hoạt động tốt với các dữ liệu phức tạp và có nhiễu.
  + Có khả năng xử lý dữ liệu lớn.
* Nhược điểm:
  + Khó diễn giải hơn so với cây quyết định đơn lẻ.
  + Tốn nhiều tài nguyên tính toán.
    - 1. **Máy vector hỗ trợ (SVM - Support Vector Machine)**
* Ưu điểm:
  + Hiệu quả với dữ liệu có không gian chiều cao và dữ liệu phi tuyến (với kernel).
  + Hoạt động tốt trong các bài toán phân loại và hồi quy.
  + Khả năng xử lý các trường hợp mà dữ liệu không thể phân tách tuyến tính.
* Nhược điểm:
  + Tốn thời gian huấn luyện với các bộ dữ liệu lớn.
  + Cần điều chỉnh tham số kernel cẩn thận, khó khăn khi dữ liệu có nhiều nhiễu.
    1. **Học không giám sát (Unsupervised Learning)**
       1. **K-means**
* Ưu điểm:
  + Đơn giản, dễ hiểu và triển khai.
  + Tính toán nhanh với dữ liệu có kích thước nhỏ đến trung bình.
  + Hoạt động tốt khi các cụm có dạng hình cầu và kích thước tương đồng.
* Nhược điểm:
  + Cần xác định trước số cụm kkk.
  + Dễ bị ảnh hưởng bởi các cụm không có hình dạng cân đối hoặc có nhiều nhiễu.
  + Nhạy cảm với việc chọn điểm trung tâm ban đầu.
    - 1. **Phân cụm phân cấp (Hierarchical Clustering)**
* Ưu điểm:
  + Không cần xác định số lượng cụm từ trước.
  + Xây dựng một cấu trúc phân cấp giúp hiểu rõ mối quan hệ giữa các cụm.
  + Tốt khi có dữ liệu nhỏ và muốn quan sát phân cụm từng bước.
* Nhược điểm:
  + Tốn kém về tính toán với dữ liệu lớn.
  + Nhạy cảm với khoảng cách và thước đo được sử dụng.
    - 1. **Phân tích thành phần chính (PCA - Principal Component Analysis)**
* Ưu điểm:
  + Giảm số chiều của dữ liệu một cách hiệu quả, loại bỏ nhiễu.
  + Dễ dàng tính toán và tăng tốc độ các mô hình học máy.
  + Giữ lại được phần lớn phương sai của dữ liệu, giúp mô hình vẫn có hiệu quả.
* Nhược điểm:
  + Mất khả năng giải thích trực tiếp về các thành phần chính.
  + Không hoạt động tốt nếu dữ liệu không có phương sai lớn giữa các chiều.
    - 1. **Thuật toán học luật kết hợp (Apriori)**
* Ưu điểm:
* Hữu ích trong việc tìm ra các quy tắc kết hợp mạnh mẽ giữa các mục trong các bộ dữ liệu lớn.
* Dễ hiểu và giải thích.
* Nhược điểm:
* Cần tốn nhiều tài nguyên tính toán với các bộ dữ liệu lớn và có nhiều mục.
* Hiệu quả giảm với các bộ dữ liệu có nhiều mục có tần suất xuất hiện nhỏ.
  + 1. **Học tăng cường (Reinforcement Learning)**
       1. **Q-learning**
* Ưu điểm:
  + Đơn giản, dễ hiểu và có thể áp dụng trong nhiều môi trường khác nhau.
  + Có khả năng học từ các tương tác với môi trường mà không cần mô hình hóa đầy đủ.
* Nhược điểm:
  + Tốn thời gian học trong các môi trường lớn với nhiều trạng thái.
  + Khó mở rộng cho các môi trường phức tạp hơn.
    - 1. **Deep Q-Network (DQN)**
* Ưu điểm:
* Khả năng học hành động tối ưu trong các môi trường phức tạp nhờ mạng nơ-ron sâu.
* Có thể áp dụng trong các môi trường không có mô hình đầy đủ.
* Nhược điểm:
* Đòi hỏi tài nguyên tính toán lớn, đặc biệt với mạng nơ-ron phức tạp.
* Khó khăn trong việc lựa chọn kiến trúc mạng và siêu tham số.

**CHƯƠNG 2 MÔ HÌNH HỌC MÁY**

## 2.1 Mô hình thuật toán

### 2.1.1 Mô hình Logistic Regression

Logistic Regression là một mô hình phân loại nhị phân, được áp dụng để phân loại nguy cơ đau tim của bệnh nhân dựa trên các đặc điểm về sức khỏe và lối sống.Cụ thể là dự đoán nguy cơ đau tim (Heart Attack Risk) dựa trên các đặc điểm như tuổi, mức cholesterol, huyết áp, thói quen lối sống, và tiền sử bệnh lý.

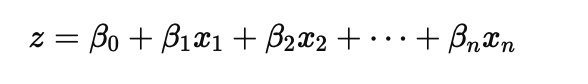
Logistic Regression dự đoán xác suất một đối tượng thuộc vào một trong hai nhóm. Trong dự án này, các nhóm là:

* Nhóm 1: Bệnh nhân có nguy cơ đau tim.
* Nhóm 0: Bệnh nhân không có nguy cơ đau tim

**Tổ hợp tuyến tính (Logit)**

Trong Logistic Regression,đầu tiên em tính toán **logit**, là tổ hợp tuyến tính của các đặc trưng đầu vào X=[x1,x2,…,xn] với các hệ số hồi quy β=[β0,β1,…,βn​].

Giá trị **logit** được tính bằng công thức:

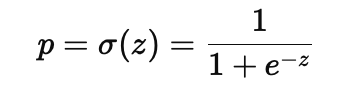


Trong đó:

* z là **logit** (giá trị tổ hợp tuyến tính).
* x1,x2,…,xn là các đặc trưng đầu vào (tuổi, mức cholesterol, huyết áp, v.v.).
* β0​ là hệ số chặn (intercept).
* β1,β2,…,βn là các hệ số hồi quy tương ứng với từng đặc trưng.

**Hàm sigmoid**

Sau khi tính toán được giá trị **logit** (z), Logistic Regression sử dụng **hàm sigmoid** để chuyển đổi giá trị này thành xác suất dự đoán p:

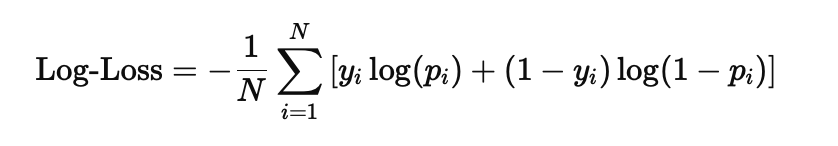


Hàm sigmoid đảm bảo rằng đầu ra p là một giá trị trong khoảng từ 0 đến 1, đại diện cho xác suất đối tượng thuộc về lớp 1 (có nguy cơ đau tim).

* Khi p≥0.5, mẫu được phân loại vào lớp 1.
* Khi p<0.5p, mẫu được phân loại vào lớp 0.

**Hàm Log-Loss**

Để huấn luyện mô hình, Logistic Regression sử dụng **hàm Log-Loss**, hay còn gọi là hàm  **negative log-likelihood**. Hàm này đo lường mức độ khác biệt giữa nhãn thực tế y(i) và xác suất dự đoán p(i)​:



Trong đó:

* y(i)​ là nhãn thực tế của mẫu  (1 hoặc 0).
* p(i)​ là xác suất dự đoán của mẫu.
* N là tổng số mẫu trong tập dữ liệu huấn luyện.

Mục tiêu của Logistic Regression là tối ưu **hàm Log-Loss**, từ đó tìm được các giá trị tham số β.

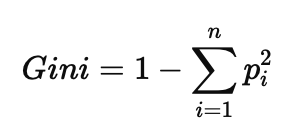
Trong bài của em , mô hình Logistic Regression được huấn luyện bằng cách sử dụng dữ liệu đầu vào. Hàm fit() trong thư viện **scikit-learn** sẽ tự động tối thiểu hóa hàm Log-Loss để điều chỉnh các tham số β0,β1,…,βn​ sao cho mô hình dự đoán chính xác nhất.

### 2.1.2 Mô hình thuật toán RandomForestClassifier

* **Gini Impurity và Entropy trong Decision Trees**

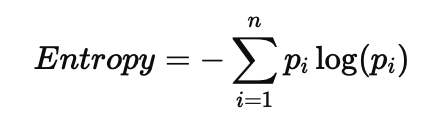
Các cây trong Random Forest được xây dựng dựa trên các phép đo như Gini Impurity hoặc Entropy để xác định cách chia tách dữ liệu tại mỗi node. Công thức cho các phép đo này như sau:

* **Gini Impurity**:



Trong đó, pi là xác suất xảy ra của lớp thứ i. Giá trị Gini càng thấp, dữ liệu tại node càng thuần khiết.

* Entropy:

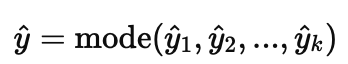


Entropy đo lường mức độ không chắc chắn hoặc hỗn loạn tại một node; giá trị càng thấp càng tốt.

* **Aggregation of Trees (Bagging)**

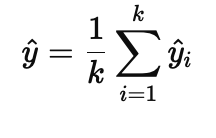
Bagging (Bootstrap Aggregating) là kỹ thuật kết hợp nhiều cây quyết định để cải thiện độ chính xác và giảm độ lệch. Công thức cho kết quả tổng hợp của Random Forest như sau:

* **Đối với phân loại**: Dự đoán là lớp có số phiếu bầu nhiều nhất từ các cây (Majority Voting).



Trong đó, y^i  là dự đoán từ cây thứ i.

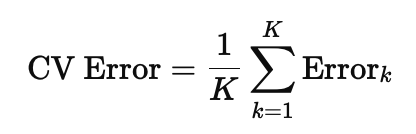
* **Đối với hồi quy**: Dự đoán là trung bình cộng của các dự đoán từ các cây.



Trong đó:

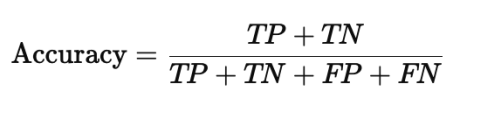
* N là tổng số mẫu trong tập dữ liệu.
* L là hàm mất mát, thường là Cross-Entropy hoặc Gini.
* y^OOB là dự đoán trên mẫu OOB.
* **Cross-Validation (CV) với GridSearch**

Trong GridSearchCV, cross-validation chia dữ liệu thành KK tập con, và mỗi tập con sẽ lần lượt được dùng để kiểm tra trong khi các tập còn lại dùng để huấn luyện. Công thức tính sai số trong K-fold CV là:

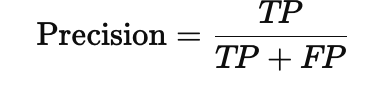


Trong đó Errork là sai số trên tập kiểm tra thứ kk.

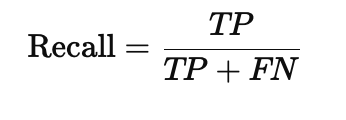
* **Accuracy, Precision, Recall, F1 Score**
* **Accuracy**:



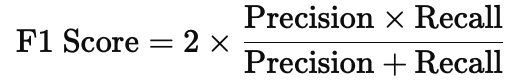
* **Precision**:



* Recall:



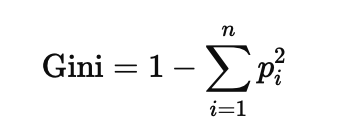
* **F1 Score**:



2.1.3 Mô hình thuật toán DecisionTreeClassifier

* Gini Impurity

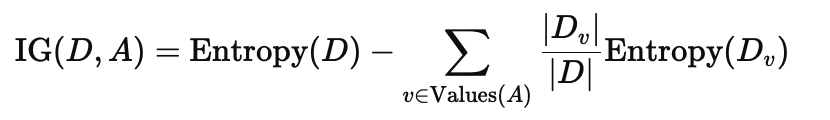
Gini Impurity là công thức mặc định được sử dụng trong DecisionTreeClassifier để đo độ không thuần khiết tại mỗi node khi chia tách dữ liệu. Nó giúp xác định thuộc tính nào chia tách dữ liệu tốt nhất.



Trong đó:

* p\_i là xác suất của từng lớp tại node.
* Công thức này được sử dụng để tính độ không thuần khiết và tìm cách chia tách giảm Gini nhiều nhất.
* **Information Gain**

Information Gain được sử dụng để quyết định thuộc tính nào là tốt nhất để chia tách dữ liệu dựa trên sự thay đổi của Entropy trước và sau khi chia tách.



Công thức này xác định mức độ giảm độ hỗn loạn khi chia tách dữ liệu.

**2.2 Tiền xử lý dữ liệu**

### 2.2.1 Mô tả dữ liệu

Bộ dữ liệu tổng hợp này cung cấp một loạt các tính năng toàn diện liên quan đến sức khỏe tim mạch và lựa chọn lối sống, bao gồm các chi tiết cụ thể của bệnh nhân như tuổi, giới tính, mức cholesterol, huyết áp, nhịp tim và các chỉ số như bệnh tiểu đường, tiền sử gia đình, thói quen hút thuốc, béo phì và uống rượu. Ngoài ra, các yếu tố lối sống như giờ tập thể dục, thói quen ăn uống, mức độ căng thẳng và giờ ít vận động cũng được đưa vào. Các khía cạnh y tế bao gồm các vấn đề về tim trước đây, việc sử dụng thuốc và mức chất béo trung tính được xem xét. Các khía cạnh kinh tế xã hội như thu nhập và các thuộc tính địa lý như quốc gia, châu lục và bán cầu được kết hợp. Bộ dữ liệu, bao gồm 8763 hồ sơ từ bệnh nhân trên toàn cầu, đạt đến đỉnh cao trong một tính năng phân loại nhị phân quan trọng biểu thị sự có hoặc không có nguy cơ đau tim, cung cấp một nguồn tài nguyên toàn diện để phân tích dự đoán và nghiên cứu về sức khỏe tim mạch.

Thông tin cơ bản :

* Ngôn ngữ: Python
* Nguồn dữ liệu: Kaggle
* Link đến Dataset: <https://www.kaggle.com/datasets/iamsouravbanerjee/heart-attack-prediction-dataset>
* Môi trường cài đặt: Google Colab

Cấu trúc dữ liệu :

Bộ dữ liệu này một tập hợp các thuộc tính khác nhau, chủ yếu là các yếu tố ảnh hưởng đến đau tim. Dưới đây là một số thuộc tính chính:

* Patient ID: Mã định danh duy nhất cho mỗi bệnh nhân
* Age: Tuổi của bệnh nhân
* Sex: Giới tính của bệnh nhân (Nam/Nữ)
* Cholesterol: Mức cholesterol của bệnh nhân
* Blood Pressure: Huyết áp của bệnh nhân (systolic/diastolic)
* Heart Rate: Nhịp tim của bệnh nhân
* Diabetes: Bệnh nhân có mắc bệnh tiểu đường không (Có/Không)
* Family History: Tiền sử gia đình có các vấn đề về tim mạch (1: Có, 0: Không)
* Smoking: Trạng thái hút thuốc của bệnh nhân (1: Có hút thuốc, 0: Không hút thuốc)
* Obesity: Tình trạng béo phì của bệnh nhân (1: Béo phì, 0: Không béo phì)
* Alcohol Consumption: Mức tiêu thụ rượu của bệnh nhân (Không/Cao/Trung bình/Nặng)
* Exercise Hours Per Week: Số giờ tập thể dục mỗi tuần
* Diet: Thói quen ăn uống của bệnh nhân (Lành mạnh/Trung bình/Kém lành mạnh)
* Previous Heart Problems: Các vấn đề về tim trước đó của bệnh nhân (1: Có, 0: Không)
* Medication Use: Bệnh nhân có sử dụng thuốc không (1: Có, 0: Không)
* Stress Level: Mức độ căng thẳng của bệnh nhân (1-10)
* Sedentary Hours Per Day: Số giờ ít vận động mỗi ngày của bệnh nhân
* Income: Mức thu nhập của bệnh nhân
* BMI: Chỉ số khối cơ thể (BMI) của bệnh nhân
* Triglycerides: Mức triglyceride của bệnh nhân
* Physical Activity Days Per Week: Số ngày hoạt động thể chất mỗi tuần
* Sleep Hours Per Day: Số giờ ngủ mỗi ngày
* Country: Quốc gia của bệnh nhân
* Continent: Châu lục nơi bệnh nhân sinh sống
* Hemisphere: Bán cầu nơi bệnh nhân sinh sống
* Heart Attack Risk: Sự hiện diện của nguy cơ đau tim (1: Có, 0: Không)

### 2.2.2 Tiền xử lý dữ liệu

* Đọc dữ liệu từ file csv

from google.colab import drive

drive.mount('/content/drive')

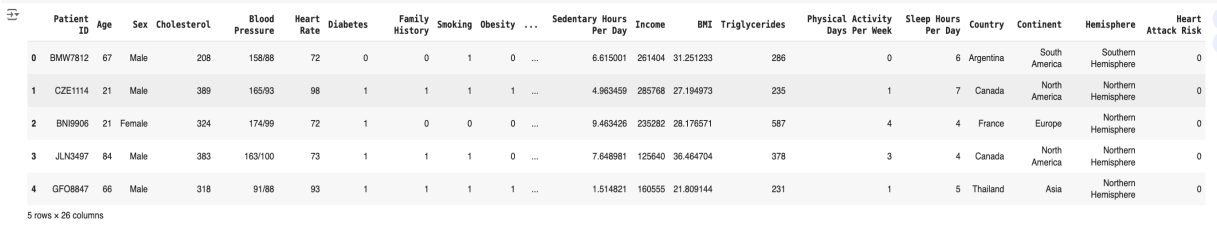
file = '/content/drive/My Drive/Khai phá dữ liệu/heart\_attack\_prediction\_dataset.csv'

Sau khi đã đọc được dữ liệu, em sẽ thực hiện các bước cơ bản đầu tiên của việc tiền xử lý liệu như sau:

* Xem dữ liệu:

data = pd.read\_csv(file)

print(data.head())



*Hình 2. Bảng hiển thị thông tin dữ liệu cột và dòng của dữ liệu*

Dữ liệu có 26 cột như: Patient ID, Age, Sex, Cholesterol, Blood Pressure, Heart Rate, Diabetes, Family History, Smoking, Obesity, Alcohol Consumption, Exercise Hours Per Week, Diet, Previous Heart Problems, Medication Use, Stress Level, Sedentary Hours Per Day, Income, BMI, Triglycerides, Physical Activity Days Per Week, Sleep Hours Per Day, Country, Continent, Hemisphere, Heart Attack Risk.

* Xoá những cột không cần thiết trong phân tích.

Các cột như Hemisphere, Continent, Patient ID, và Income không ảnh hưởng đến biến mục tiêu nên đã được loại bỏ để tối ưu mô hình.

#Xóa những cột không cần thiết

columns\_to\_drop=['Hemisphere','Continent','Patient ID','Income']

data.drop(columns=columns\_to\_drop,inplace=True)

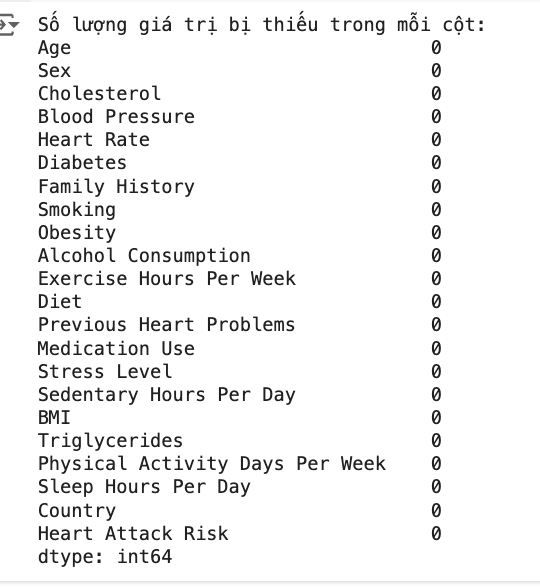
data.head()

4.Kiểm tra dữ liệu thiếu

missing\_values = data.isnull().sum()

print("Số lượng giá trị bị thiếu trong mỗi cột:")

print(missing\_values)



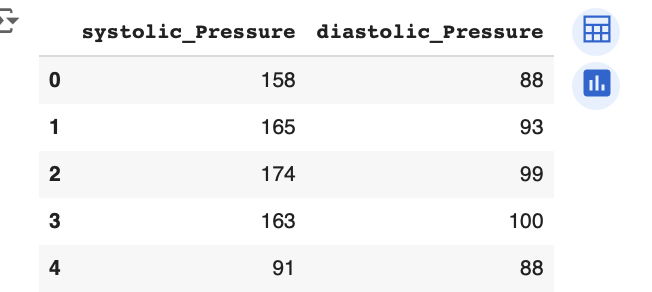
Kết quả cho thấy dữ liệu không có giá trị thiếu.

* Làm sạch dữ liệu.

data[['systolic\_Pressure', 'diastolic\_Pressure']] = data['Blood Pressure'].str.split('/', expand=True).astype(int)

data.drop('Blood Pressure', axis=1, inplace=True)

data[['systolic\_Pressure', 'diastolic\_Pressure']].head()



* Mã Hóa Nhãn Các Biến Phân Loại.

# Tạo đối tượng LabelEncoder

le = LabelEncoder()

# Sao chép tập dữ liệu gốc gán vào biến ds

data = data.copy()

# Chọn các cột phân loại

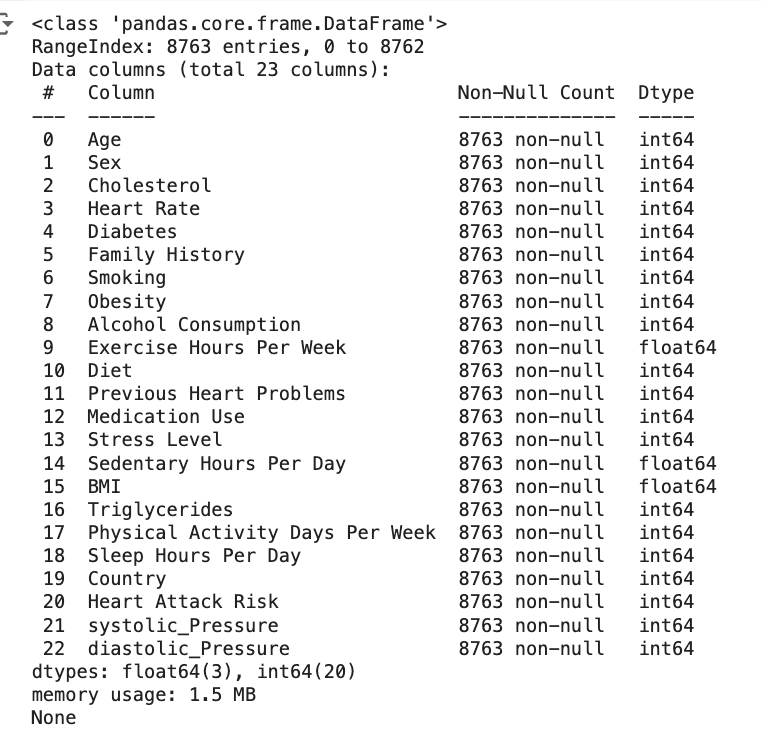
categorical\_cols = data.select\_dtypes(include=['object']).columns.tolist()

for col in categorical\_cols:

data[col] = le.fit\_transform(data[col]) # Mã hóa giá trị trong cột thành giá trị số

# Hiển thị thông tin về DataFrame

print(data.info())



* Chuẩn Hóa Dữ Liệu Sử Dụng Standard Scaler:

from sklearn.preprocessing import StandardScaler

# Sao chép tập dữ liệu và chọn các cột số để chuẩn hóa

numeric\_cols = data.select\_dtypes(include=['int64', 'float64']).columns.tolist() # Chọn các cột số

# Chuẩn hóa dữ liệu chỉ với các cột số

scaler = StandardScaler()

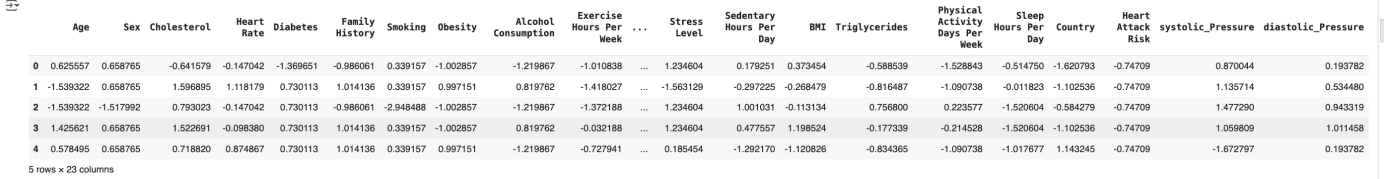
scaled\_data = scaler.fit\_transform(data[numeric\_cols])

# Tạo DataFrame mới với dữ liệu đã được chuẩn hóa

df\_scaled = pd.DataFrame(scaled\_data, columns=numeric\_cols)

# Kiểm tra kết quả

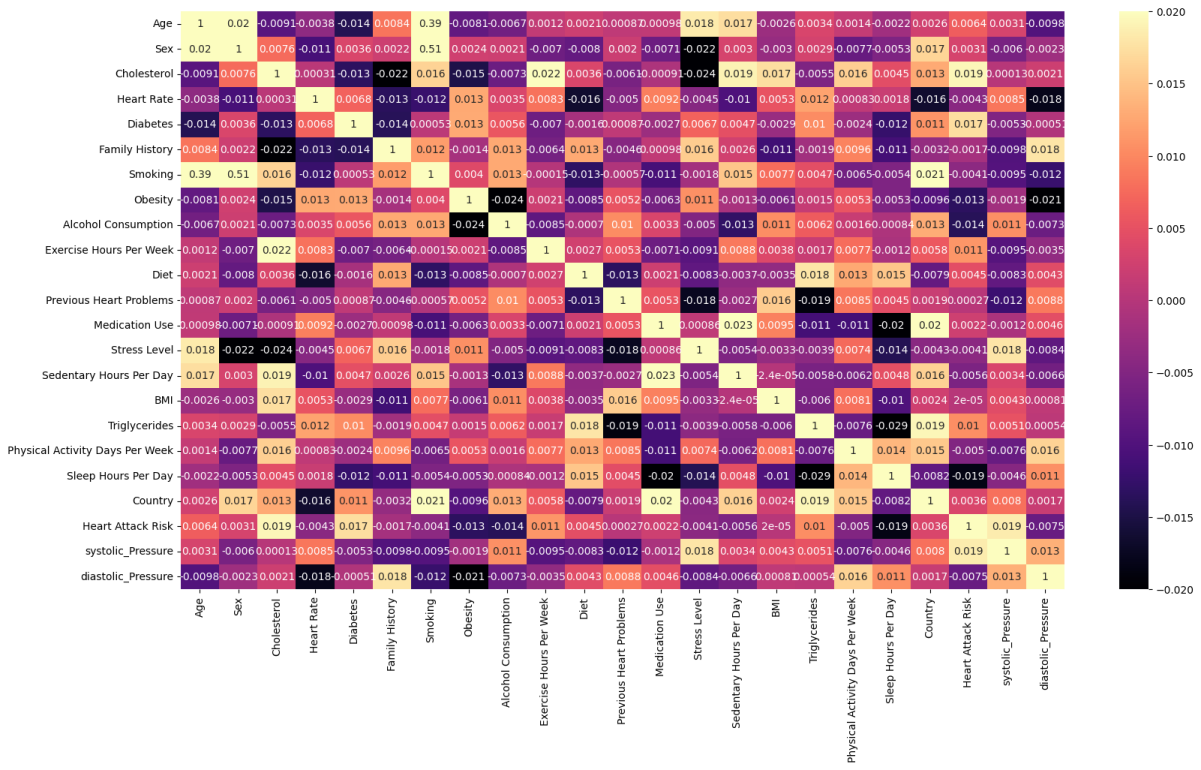
df\_scaled.head()

****

### 2.2.3 Trực quan hoá dữ liệu

- Hệ số tương quan.

Biểu đồ heatmap giúp xác định các biến có mối tương quan mạnh với Heart Attack Risk, như Cholesterol, Diabetes, và Exercise Hours Per Week.



\*Nhận xét:

Từ Heatmap này, chúng ta có thể hiểu rằng Cholesterol, Bệnh tiểu đường, Số giờ tập thể dục mỗi tuần, Triglyceride và huyết áp tâm thu (huyết áp) có liên quan nhiều hơn đến Nguy cơ đau tim so với các đặc điểm khác. BMI và Các vấn đề về tim trước đây có mối tương quan thấp với Nguy cơ đau tim. Do đó, chúng ta có thể loại bỏ chúng.  
- Trực quan hóa phân phối của các biến số.

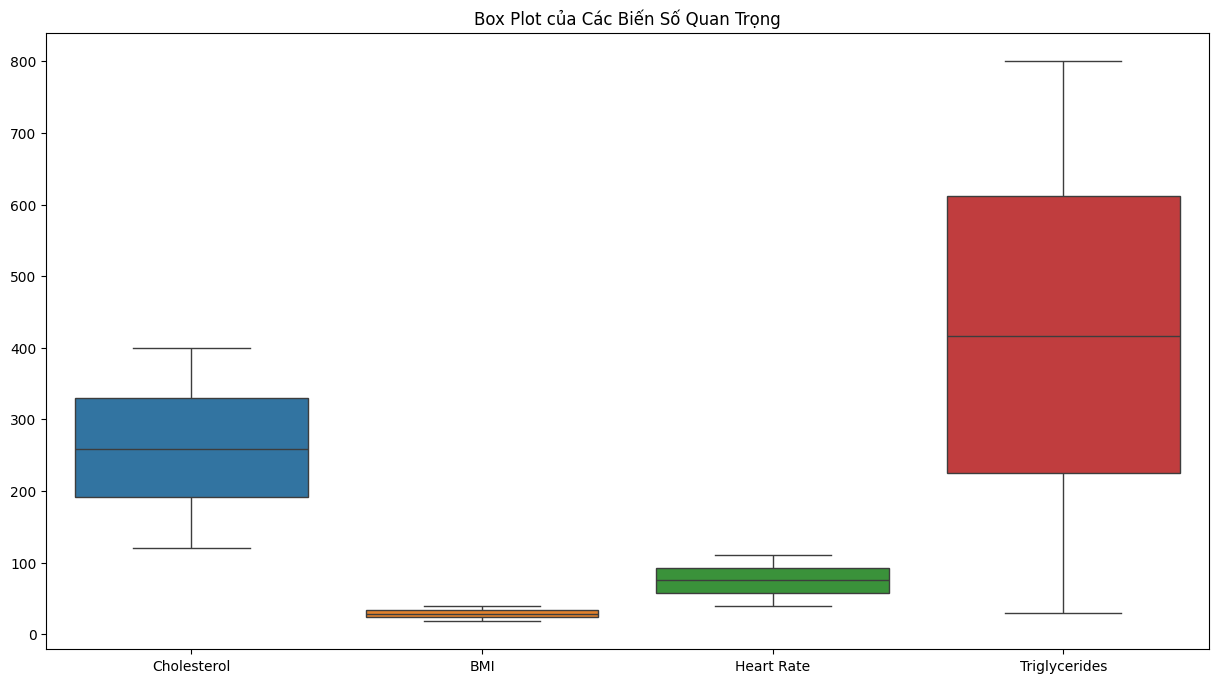
Biểu đồ histogram hiển thị phân phối của các biến số chính như Age, Cholesterol, Heart Rate, Diabetes, Smoking, và nhiều biến khác.



\* Nhận xét :

* Biến số như Cholesterol và Triglycerides có sự phân phối rộng, cho thấy có sự biến động lớn trong các giá trị.
* Các biến như BMI và Heart Rate có phân phối hẹp hơn, cho thấy ít giá trị ngoại lệ và mức độ biến động thấp.
* Hầu hết các biến số có phân phối khá đều hoặc dạng phân bố thông thường, ngoại trừ các biến phân loại nhị phân như Family History, Smoking, Obesity vốn có sự phân chia rõ ràng.
* Xem giá trị ngoại lệ ở các biến số quan trọng.

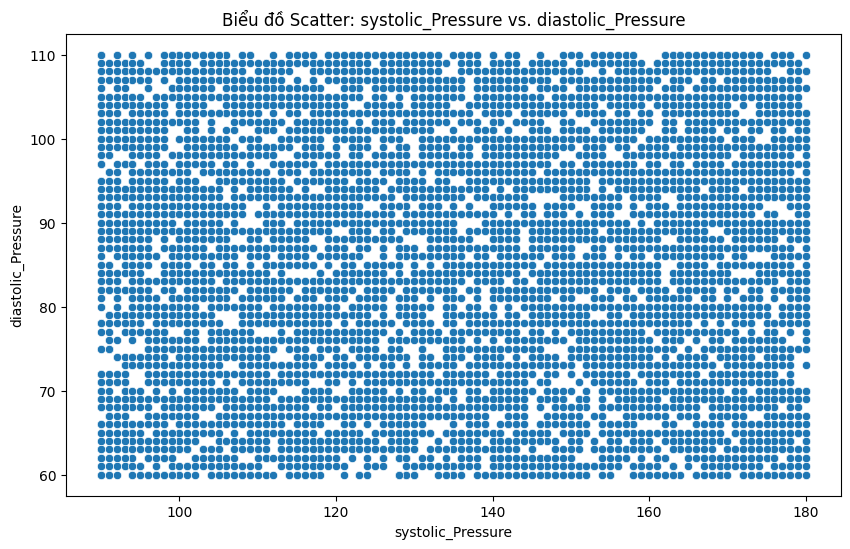
Biểu đồ box plot cho các biến quan trọng như Cholesterol, BMI, Heart Rate, và Triglycerides để nhận diện giá trị ngoại lệ và phân phối dữ liệu.



\*Nhận xét:

* Cholesterol và Triglycerides có phân phối rộng và nhiều giá trị ngoại lệ, cho thấy ảnh hưởng lớn hơn đến nguy cơ đau tim.
* BMI và Heart Rate có phân phối hẹp, ít giá trị ngoại lệ, thể hiện tác động thấp hơn trong phân tích.
* Việc nhận diện giá trị ngoại lệ giúp làm sạch dữ liệu, tránh nhiễu trong quá trình phân tích và mô hình hóa.
* Mối quan hệ giữa systolic\_Pressure và diastolic\_Pressure.

Biểu đồ scatter giữa systolic\_Pressure và diastolic\_Pressure để xác định mối quan hệ giữa hai chỉ số huyết áp.



\*Nhận xét:

Biến systolic\_Pressure và diastolic\_Pressure không có mối quan hệ rõ ràng, nên mỗi chỉ số cần được theo dõi riêng để đánh giá sức khỏe tim mạch chính xác hơn.

### 2.2.4 Model và kết quả

2.2.4.1 Tập dữ liệu dự đoán

Tập dữ liệu dự đoán sử dụng các đặc trưng liên quan đến sức khỏe của bệnh nhân, bao gồm các yếu tố quan trọng liên quan đến nguy cơ đau tim. Các đặc trưng chính trong tập dữ liệu là:

A blurry image of a group of people

Description automatically generated

2.2.4.2 Kết quả dự đoán

Sau khi áp dụng tập dữ liệu dự đoán vào mô hình đã huấn luyện, kết quả dự đoán cho thấy nguy cơ đau tim dưới dạng phần trăm. Mô hình dự đoán xác suất một bệnh nhân có thể mắc bệnh tim dựa trên các đặc trưng của họ.

2.2.4.3 Thuật toán Logistic Regression :

**A screenshot of a computer program

Description automatically generated**

Thuật toán cho ra kết quả dự đoán chỉ 50%.

2.2.4.4 Thuật toán DecisionTreeClassifier

A screenshot of a computer code

Description automatically generated

Thuật toán cho ra kết quả dự đoán 68% cao hơn so với **Logistic Regression**

2.2.4.5 Thuật toán RandomForestClassifier

A screenshot of a computer code

Description automatically generated

Thuật toán cho ra kết quả dự đoán cao lên đến 77% cao hơn rõ rệt so với 2 thuật toán trước đó.

* + - 1. Đánh giá mô hình

**A table with black text

Description automatically generated**

**\*Nhận xét:**

* Logistic Regression:

Mô hình Logistic Regression có chỉ số sai số cao nhất với MAE = 0.5, MSE = 0.5, và RMSE = 0.71. Điều này cho thấy mô hình này kém chính xác nhất trong việc dự đoán kết quả so với các mô hình khác.

* Random Forest:

Random Forest đạt được các chỉ số tốt nhất với MAE = 0.21, MSE = 0.21, và RMSE = 0.46. Điều này chứng tỏ Random Forest là mô hình mạnh nhất, giúp giảm thiểu sai số dự đoán và mang lại kết quả chính xác nhất.

* Decision Tree:

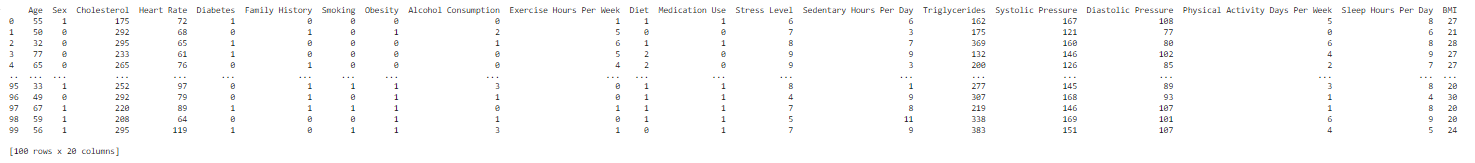
Mô hình Decision Tree có MAE = 0.32, MSE = 0.32, và RMSE = 0.57. Mặc dù Decision Tree cho kết quả tốt hơn Logistic Regression, nhưng vẫn không chính xác bằng Random Forest.

Dựa trên các chỉ số đánh giá, Random Forest là mô hình tốt nhất trong số ba mô hình, với khả năng dự đoán chính xác hơn và ít sai số nhất.

CHƯƠNG 3 THỰC NGHIỆM VÀ KẾT QUẢ

**3.1 Tập dữ liệu dự đoán**

Tập dữ liệu dự đoán sử dụng các đặc trưng liên quan đến sức khỏe của bệnh nhân, bao gồm các yếu tố quan trọng liên quan đến nguy cơ đau tim. Các đặc trưng chính trong tập dữ liệu là:



**3.2 Kết quả dự đoán**

Sau khi áp dụng tập dữ liệu dự đoán vào mô hình đã huấn luyện, kết quả dự đoán cho thấy nguy cơ đau tim dưới dạng phần trăm. Mô hình dự đoán xác suất một bệnh nhân có thể mắc bệnh tim dựa trên các đặc trưng.



Sau khi áp dụng dữ liệu thực tế vào mô hình dự đoán bằng thuật toán Ramdom Forest, ta thấy được kết quả lên đến 87% là con số dự đoán cao.

## 3.3 Đánh giá mô hình

**A table with numbers and letters

Description automatically generated with medium confidence**

Đây là các chỉ số ta nhận được từ việc huấn luyện và kiểm tra thuật toán. Số liệu cho ta thấy mô hình áp dụng trên dữ liệu thực tế chính xác rất cao. Có thể sử dụng vào thực tế để dự đoán nguy cơ đau tim.

**KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

**Kết luận :**

Kết quả của mô hình Random Forest cho thấy mô hình có độ chính xác (Accuracy) là 87%, nghĩa là mô hình dự đoán đúng 87% các trường hợp trong tập kiểm tra. Tuy nhiên, điều này không phải lúc nào cũng đủ để khẳng định mô hình tốt, đặc biệt là trong các bài toán có nhãn không cân bằng.

Chỉ số Precision (Độ chính xác theo nhóm dương tính) đạt 1.00, cho thấy mô hình rất chắc chắn khi dự đoán các mẫu dương tính, nhưng đây có thể là do mô hình đã dự đoán rất ít (hoặc không) các mẫu dương tính. Điều này dẫn đến Recall (Độ nhạy) thấp, chỉ đạt 0.20, cho thấy mô hình bỏ sót nhiều trường hợp dương tính thực tế.

Điểm F1-Score chỉ đạt 0.33, điều này cho thấy sự mất cân bằng giữa Precision và Recall. F1-Score là một thước đo quan trọng trong các bài toán nhãn không cân bằng, và điểm thấp này cho thấy mô hình cần được cải thiện để cân bằng giữa khả năng phát hiện và độ chính xác. Cuối cùng, ROC-AUC của mô hình là 0.60, điều này chỉ ra rằng khả năng phân biệt giữa các lớp (có nguy cơ và không có nguy cơ đau tim) chỉ hơi tốt hơn mức ngẫu nhiên.

**Hướng phát triển :**

Xử lý vấn đề dữ liệu không cân bằng: Để cải thiện mô hình, chúng ta cần xử lý vấn đề dữ liệu không cân bằng. Các mẫu dương tính có thể ít hơn rất nhiều so với mẫu âm tính, khiến mô hình không thể học tốt. Sử dụng các kỹ thuật như SMOTE (Synthetic Minority Over-sampling Technique) hoặc undersampling để cân bằng lại dữ liệu là một giải pháp khả thi. Những kỹ thuật này có thể giúp tăng số lượng các mẫu dương tính hoặc giảm bớt các mẫu âm tính để mô hình không bị thiên lệch.

Tối ưu hóa siêu tham số của mô hình: Ngoài ra, có thể tối ưu hóa các siêu tham số của mô hình Random Forest bằng cách sử dụng các phương pháp như Grid Search hoặc Random Search. Điều này giúp mô hình lựa chọn các tham số tốt nhất để cải thiện hiệu suất dự đoán.

Thử nghiệm các mô hình khác: Mặc dù Random Forest thường mang lại kết quả tốt, nhưng có thể thử nghiệm thêm các mô hình khác như Gradient Boosting, XGBoost hoặc Logistic Regression để xem liệu chúng có thể cải thiện khả năng phát hiện các trường hợp dương tính hay không. Sử dụng mô hình ensemble có thể giúp kết hợp kết quả của nhiều mô hình khác nhau để có dự đoán tốt hơn.

Sử dụng thêm các đặc trưng khác: Cải thiện việc chọn lọc và tạo ra các đặc trưng mới từ dữ liệu ban đầu có thể giúp mô hình học tốt hơn. Những đặc trưng liên quan đến sức khỏe như thói quen ăn uống, chỉ số tập thể dục hoặc tình trạng sức khỏe tổng thể có thể được thêm vào để tăng cường khả năng dự đoán của mô hình.