

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NÔNG LÂM TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----

A green circle with white text and tree branches and a book

Description automatically generated

**BÁO CÁO GIỮA KÌ**

**ĐỀ TÀI:**

**ỨNG DỤNG RANDOM FOREST TRONG**

**WEBSITE DỰ ĐOÁN BỆNH TIM**

**Môn học: Data Mining**

**Học kì: 1/2024-2025**

Giảng viên phụ trách : ThS. Trần Quốc Việt

Danh sách sinh viên thực hiện:

1. Nguyễn Minh Thông – 21130551

**Mục lục**

[**I.** **Đặt vấn đề** 3](#_Toc190430481)

[**1.** **Giới thiệu** 3](#_Toc190430482)

[**2. Mục tiêu nghiên cứu** 3](#_Toc190430483)

[**3. Lý do chọn thuật toán Random Forest** 3](#_Toc190430484)

[**4. Phạm vi nghiên cứu** 4](#_Toc190430485)

[**5. Ý nghĩa thực tiễn của nghiên cứu** 4](#_Toc190430486)

[**II.** **Nội dung nghiên cứu và kết quả** 4](#_Toc190430487)

[**1.** **Tóm tắt** 4](#_Toc190430488)

[**2.** **Phương pháp luận** 4](#_Toc190430489)

[**3.** **Ưu điểm của Random Forest** 5](#_Toc190430490)

[**4.** **Nhược điểm của Random Forest:** 5](#_Toc190430491)

[**5.** **Bootstrapping** 5](#_Toc190430492)

[**6.** **Bagging** 5](#_Toc190430493)

[**7.** **Mô tả chức năng** 6](#_Toc190430494)

[**7.1.** **Đăng nhập** 6](#_Toc190430495)

[**7.2.** **Đăng ký** 6](#_Toc190430496)

[**7.3.** **Quên mật khẩu** 7](#_Toc190430497)

[**7.4.** **Đăng xuất** 8](#_Toc190430498)

[**7.5.** **Dự đoán bệnh tim** 8](#_Toc190430499)

[**7.6.** **Thống kê thông qua biểu đồ** 10](#_Toc190430500)

[**7.7.** **Khuyến nghị chăm sóc sức khỏe theo dữ liệu có sẵn** 15](#_Toc190430501)

[**7.8.** **Hiển thị dữ liệu bệnh nhân** 15](#_Toc190430502)

[**7.9.** **Xuất file PDF** 16](#_Toc190430503)

[**7.10.** **Kiểm chứng chéo mô hình (Cross-validation)** 19](#_Toc190430504)

[**7.11.** **So sánh với cộng đồng** 19](#_Toc190430505)

[**7.12.** **Hiển thị số lần chẩn đoán của bệnh nhân** 21](#_Toc190430506)

[**7.13.** **Dự đoán bệnh thông qua các số liệu bằng Random Forest** 21](#_Toc190430507)

[**7.14.** **Tương quan giữa các đặc trưng** 24](#_Toc190430508)

[**8.** **Phân tích kết quả** 24](#_Toc190430509)

[**9.** **Thảo luận** 26](#_Toc190430510)

[**III.** **Kết luận** 28](#_Toc190430511)

[**IV.** **Tài liệu tham khảo** 30](#_Toc190430512)

1. **Đặt vấn đề**
2. **Giới thiệu**

* Bệnh tim mạch là một trong những nguyên nhân gây tử vong hàng đầu trên thế giới. Việc chẩn đoán sớm và chính xác bệnh tim có vai trò quan trọng trong việc nâng cao hiệu quả điều trị, giảm tỷ lệ tử vong và cải thiện chất lượng cuộc sống cho bệnh nhân. Trong thời đại công nghệ số, các thuật toán học máy (Machine Learning) đang ngày càng được ứng dụng rộng rãi trong lĩnh vực y tế nhằm hỗ trợ các bác sĩ đưa ra quyết định chính xác hơn.
* Trong nghiên cứu này, nhóm đề xuất sử dụng thuật toán **Random Forest** để xây dựng mô hình dự đoán bệnh tim, kết hợp với **Flask Framework** để triển khai một ứng dụng web giúp người dùng nhập thông tin sức khỏe và nhận được dự đoán bệnh tim tức thời.

## **2. Mục tiêu nghiên cứu**

Nghiên cứu nhằm đạt được các mục tiêu sau:

* Ứng dụng thuật toán Random Forest để dự đoán nguy cơ mắc bệnh tim.
* Xây dựng một mô hình có độ chính xác cao thông qua tối ưu hóa siêu tham số.
* Phát triển một ứng dụng web trực quan, giúp người dùng dễ dàng nhập thông tin và nhận kết quả dự đoán.
* Đánh giá hiệu suất của mô hình bằng các chỉ số như **độ chính xác (accuracy), recall, F1-score** và ma trận nhầm lẫn.
* So sánh mô hình Random Forest với các thuật toán học máy khác để đánh giá ưu, nhược điểm của từng phương pháp.

## **3. Lý do chọn thuật toán Random Forest**

Random Forest là một thuật toán học máy thuộc nhóm **ensemble learning**, kết hợp nhiều cây quyết định để tạo ra một mô hình mạnh mẽ và ổn định hơn. Những lý do chọn thuật toán này gồm:

* **Khả năng xử lý dữ liệu phức tạp**: Random Forest có thể hoạt động tốt trên dữ liệu có nhiều biến số, bao gồm cả dữ liệu phân loại và dữ liệu số.
* **Giảm nguy cơ overfitting**: So với các mô hình đơn lẻ như Decision Tree, Random Forest ít bị ảnh hưởng bởi nhiễu và có độ tổng quát hóa tốt hơn.
* **Hiệu suất cao**: Random Forest thường đạt độ chính xác cao trong các bài toán phân loại và có thể được tối ưu hóa bằng cách tinh chỉnh các siêu tham số như số lượng cây (**n\_estimators**) và độ sâu tối đa của cây (**max\_depth**).
* **Tính diễn giải tốt**: Mô hình cung cấp thông tin về mức độ quan trọng của từng đặc trưng, giúp hiểu rõ hơn về những yếu tố ảnh hưởng đến bệnh tim.

## **4. Phạm vi nghiên cứu**

* **Dữ liệu**: Nghiên cứu sử dụng tập dữ liệu bệnh tim, bao gồm các thông tin như tuổi, giới tính, huyết áp, mức cholesterol, nhịp tim tối đa, đau thắt ngực khi tập luyện, đường huyết,...
* **Thuật toán**: Tập trung vào mô hình Random Forest để dự đoán bệnh tim và so sánh với một số thuật toán khác như Decision Tree, Logistic Regression để đánh giá hiệu suất.
* **Ứng dụng**: Xây dựng một hệ thống web đơn giản giúp người dùng nhập dữ liệu sức khỏe và nhận kết quả dự đoán từ mô hình đã huấn luyện.

## **5. Ý nghĩa thực tiễn của nghiên cứu**

* **Hỗ trợ chẩn đoán sớm**: Cung cấp một công cụ hỗ trợ giúp các bác sĩ và bệnh nhân nhận diện sớm nguy cơ mắc bệnh tim, từ đó có phương án kiểm tra và điều trị kịp thời.
* **Tăng cường ứng dụng AI trong y tế**: Minh họa tiềm năng của học máy trong việc hỗ trợ các quyết định lâm sàng, đặc biệt trong chẩn đoán và dự báo bệnh tật.
* **Dễ dàng tiếp cận**: Ứng dụng web cho phép người dùng nhập dữ liệu cá nhân và nhận kết quả dự đoán nhanh chóng mà không cần đến bệnh viện ngay lập tức.
* **Định hướng phát triển**: Mô hình có thể được mở rộng và cải tiến bằng cách sử dụng nhiều dữ liệu hơn, thử nghiệm các kỹ thuật học sâu (Deep Learning), hoặc kết hợp với các phương pháp tối ưu hóa khác để tăng độ chính xác.

1. **Nội dung nghiên cứu và kết quả**
   * + 1. **Tóm tắt**

* Sử dụng Random Forest và Flask Framework để xây dựng 1 web app đơn giản dự đoán bệnh tim dựa trên tập dữ liệu.

1. **Phương pháp luận**

* Dữ liệu: Sử dụng dữ liệu từ dataset. Các thuộc tính bao gồm tuổi, giới tính, loại đau ngực, huyết áp nghỉ, cholesterol, nhịp tim tối đa, đau thắt ngực khi tập luyện và đường huyết. Ngoài ra,còn có các đặc trưng khác. Mục tiêu là sự hiện diện (1) hoặc vắng mặt (0) của bệnh tim.
* Tiền xử lý dữ liệu: Xử lý giá trị thiếu, mã hóa thuộc tính phân loại (giới tính: 0 và 1), chuẩn hóa dữ liệu bằng StandardScaler.
* Mô hình: Thuật toán Random Forest được chọn. Các siêu tham số như n\_estimators, max\_depth được tinh chỉnh để tối ưu hóa hiệu suất.
* Đánh giá mô hình: Sử dụng độ chính xác, recall, F1-score và ma trận nhầm lẫn để đánh giá. Mô hình huấn luyện trên tập dữ liệu huấn luyện và kiểm thử trên tập dữ liệu riêng biệt.
* Triển khai: Sử dụng Flask để tạo ứng dụng web cho phép người dùng nhập thông tin và nhận dự đoán.

1. **Ưu điểm của Random Forest**

* Giảm thiểu overfitting: Bằng cách kết hợp nhiều cây quyết định độc lập, Random Forest giảm thiểu nguy cơ overfitting so với việc sử dụng một cây đơn lẻ.
* Xử lý dữ liệu lớn và hiệu quả: Thuật toán có thể xử lý tập dữ liệu lớn với số lượng đặc trưng đa dạng và phức tạp.
* Đánh giá tầm quan trọng của đặc trưng: Random Forest cung cấp khả năng đánh giá mức độ quan trọng của từng đặc trưng trong việc dự đoán kết quả.

1. **Nhược điểm của Random Forest:**

* **Tính phức tạp:** Do bao gồm nhiều cây quyết định, mô hình Random Forest có thể trở nên phức tạp và khó diễn giải hơn so với một cây quyết định đơn lẻ.
* **Thời gian tính toán:** Việc xây dựng và dự đoán với nhiều cây có thể tốn thời gian và tài nguyên tính toán, đặc biệt với tập dữ liệu rất lớn.

1. **Bootstrapping**

* Bootstrapping là một kỹ thuật lấy mẫu có hoàn lại (sampling with replacement). Nghĩa là, từ tập dữ liệu ban đầu có **N** mẫu, ta tạo ra nhiều tập con bằng cách chọn ngẫu nhiên **N** mẫu từ tập dữ liệu gốc, nhưng một mẫu có thể được chọn nhiều lần.
* Mục tiêu: Bootstrapping giúp giảm phương sai của mô hình bằng cách tạo ra các tập dữ liệu huấn luyện đa dạng, từ đó giúp mô hình tổng quát tốt hơn.

1. **Bagging**

* Bagging (**Bootstrap Aggregating**) là phương pháp kết hợp nhiều mô hình học máy yếu (weak learners) bằng cách:
* Áp dụng **bootstrapping** để tạo nhiều tập dữ liệu con.
* Huấn luyện một mô hình trên từng tập con đó.
* Tổng hợp kết quả từ các mô hình thành một kết quả cuối cùng.
* Mục tiêu: Giảm phương sai (variance), giúp mô hình ít bị overfitting hơn so với việc sử dụng một cây quyết định đơn lẻ.

1. **Mô tả chức năng**

### **Đăng nhập**

* **Mô tả chức năng**:  
  Người dùng có thể truy cập vào hệ thống bằng cách nhập **tên tài khoản (username)** và **mật khẩu (password)** đã được tạo từ trước. Hệ thống kiểm tra thông tin đầu vào và xử lý như sau:
* Nếu thông tin đúng, người dùng được chuyển hướng tới **trang chính (index)**.
* Nếu thông tin sai, hiển thị thông báo lỗi: **"Thông tin đăng nhập không đúng"** và yêu cầu nhập lại.

A screen shot of a login page

Description automatically generated

### **Đăng ký**

* **Mô tả chức năng**:

Người dùng mới có thể tạo tài khoản bằng cách điền thông tin cần thiết. Hệ thống xử lý như sau:

* Kiểm tra tính hợp lệ của thông tin:
  + **Tên tài khoản** không được trùng với tài khoản đã có.
  + **Mật khẩu** phải đủ mạnh (ví dụ: chứa ít nhất 6 ký tự, bao gồm chữ cái và số).
  + Các trường bắt buộc không được bỏ trống.
* Nếu thông tin hợp lệ, tài khoản mới được tạo và thông báo **"Đăng ký thành công"**.
* Người dùng có thể đăng nhập bằng tài khoản vừa tạo.

A screenshot of a login form

Description automatically generated

### **Quên mật khẩu**

* **Mô tả chức năng**:

Chức năng này hỗ trợ người dùng nhớ lại mật khẩu trong trường hợp quên. Quy trình bao gồm:

* Nhập tên người dùng (username).
* Click nút ‘Get Password”.
* Hệ thống sẽ hiện thị mật khẩu của tài khoản dưới mục ‘Your password’ .

A screenshot of a login screen

Description automatically generated

A screenshot of a login screen

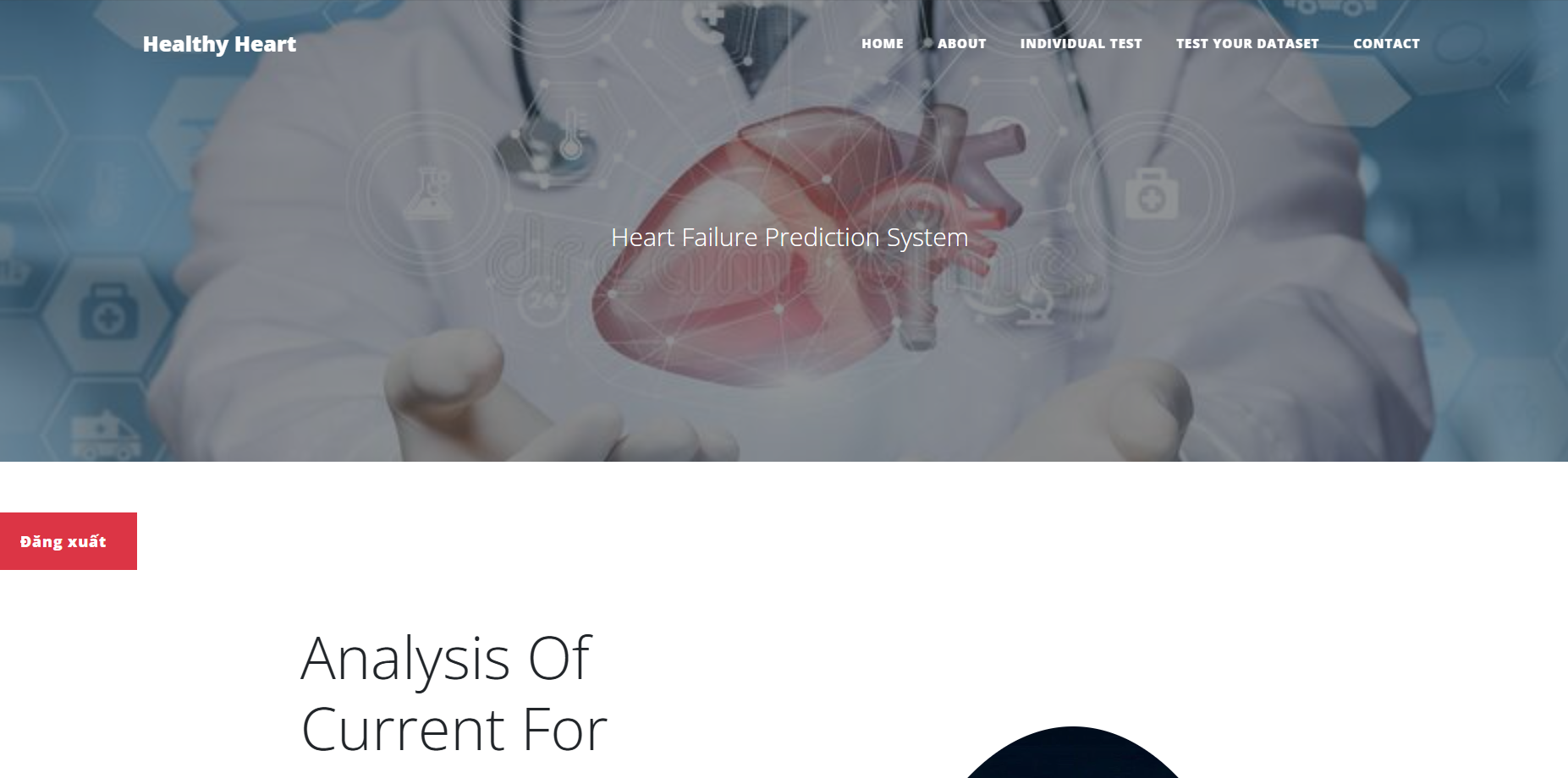
Description automatically generated

### **Đăng xuất**

* **Mô tả chức năng**:

Người dùng có thể thoát khỏi tài khoản hiện tại bằng cách sử dụng chức năng đăng xuất. Hệ thống xử lý như sau:

* Xóa thông tin đăng nhập khỏi phiên làm việc (session).
* Chuyển hướng người dùng về **trang đăng nhập**.
* Hiển thị thông báo: **"Bạn đã đăng xuất thành công"**.



Để sử dụng chức năng đăng xuất người dùng cần click vào nút “Đăng xuất” màu đỏ góc bên trái giao diện trang chủ sau khi đã đăng nhập thành công.

### **Dự đoán bệnh tim**

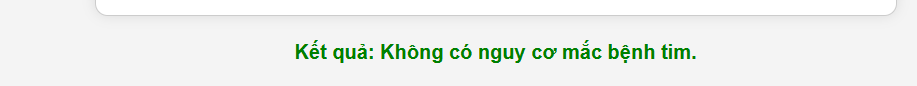
* **Mô tả chức năng:**
* Nhập liệu: Người dùng nhập thông tin cá nhân qua form HTML. Dữ liệu được validate.A screenshot of a computer

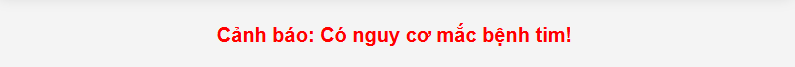
  Description automatically generated

A screenshot of a computer

Description automatically generated

* Gửi dữ liệu: Khi nhấn "Dự đoán", dữ liệu gửi tới backend qua phương thức POST tại endpoint /predict\_heart.
* Nhận dữ liệu: Flask nhận dữ liệu từ frontend qua request.form.
* Tiền xử lý dữ liệu: Mã hóa thuộc tính phân loại, tạo DataFrame từ dữ liệu đã mã hóa.
* Tải mô hình và scaler: Tải mô hình Random Forest và StandardScaler đã huấn luyện trước.
* Chuẩn hóa dữ liệu: Chuẩn hóa dữ liệu đầu vào bằng StandardScaler.
* Dự đoán: Dữ liệu chuẩn hóa được đưa vào mô hình để dự đoán.
* Lưu dữ liệu vào cơ sở dữ liệu: Kết quả dự đoán và dữ liệu đầu vào được lưu vào MySQL.
* Huấn luyện lại mô hình: Dữ liệu mới được lấy từ cơ sở dữ liệu để huấn luyện lại mô hình.
* Trả kết quả: Kết quả dự đoán được trả về frontend.





* Cải tiến thuật toán: Thử nghiệm các kỹ thuật điều chỉnh siêu tham số và so sánh với các thuật toán học máy khác để tìm ra phương pháp tối ưu nhất.

### **Thống kê thông qua biểu đồ**

* **Mô tả chức năng:**
* **Biểu Đồ Các Yếu Tố Rủi Ro (Bar Chart)**

**Mô tả:** Biểu đồ cột này trình bày mức độ rủi ro của từng yếu tố như **Tuổi**, **Huyết Áp**, **Cholesterol**, và **Đường Huyết**.

**Cách sử dụng:**

Dữ liệu: Mức độ rủi ro của các yếu tố nguy cơ (age\_risk, bp\_risk, cholesterol\_risk, glucose\_risk).

Mục đích: Biểu đồ này giúp đánh giá mức độ ảnh hưởng của từng yếu tố nguy cơ đến khả năng mắc bệnh tim. Điều này giúp hiểu rõ đâu là yếu tố nguy cơ chính trong dân số.

A graph with different colored squares

Description automatically generated

* **Biểu Đồ Tương Quan Các Yếu Tố Rủi Ro (Heatmap / Scatter Plot)**

**Mô tả:** Biểu đồ tương quan hiển thị mối quan hệ giữa các yếu tố nguy cơ như **Tuổi**, **Huyết Áp**, **Cholesterol**, và **Đường Huyết**.

**Cách sử dụng:**

Dữ liệu: Các chỉ số tương quan giữa các yếu tố nguy cơ (age\_bp\_corr, age\_cholesterol\_corr, bp\_cholesterol\_corr, v.v.).

Mục đích: Biểu đồ này giúp xác định mối liên hệ giữa các yếu tố. Ví dụ, nếu **Huyết Áp** và **Cholesterol** có sự tương quan cao, điều này có thể chỉ ra rằng các bệnh nhân có mức huyết áp cao có khả năng cao bị rối loạn cholesterol.

A graph with different colored dots

Description automatically generated

* **Biểu Đồ Phân Loại Dự Đoán Mắc Bệnh Tim (Pie Chart)**

**Mô tả:** Biểu đồ tròn này thể hiện kết quả dự đoán bệnh tim từ mô hình phân loại.

**Cách sử dụng:**

Dữ liệu: Tỷ lệ dự đoán mắc bệnh tim (heart\_disease\_positive), không mắc bệnh tim (heart\_disease\_negative) sau khi áp dụng mô hình phân loại.

A blue and red pie chart

Description automatically generated

Mục đích: Biểu đồ này giúp hiển thị kết quả dự đoán của mô hình phân loại đối với bệnh tim.

* **Biểu Đồ Phân Tích Cụm Nguy Cơ Bệnh Tim (Clustering Scatter Plot)**

**Mô tả:** Biểu đồ phân tán thể hiện các nhóm bệnh nhân có nguy cơ bệnh tim tương tự nhau.

**Cách sử dụng:**

Dữ liệu: Các bệnh nhân được phân thành các cụm nguy cơ dựa trên các yếu tố như **Tuổi**, **Huyết Áp**, **Cholesterol**, và **Đường Huyết**.

Mục đích: Biểu đồ này giúp hiểu rõ hơn về sự phân bổ của các bệnh nhân trong các nhóm nguy cơ khác nhau, từ đó có thể áp dụng các phương pháp điều trị phù hợp cho từng nhóm.

A graph with colored dots

Description automatically generated

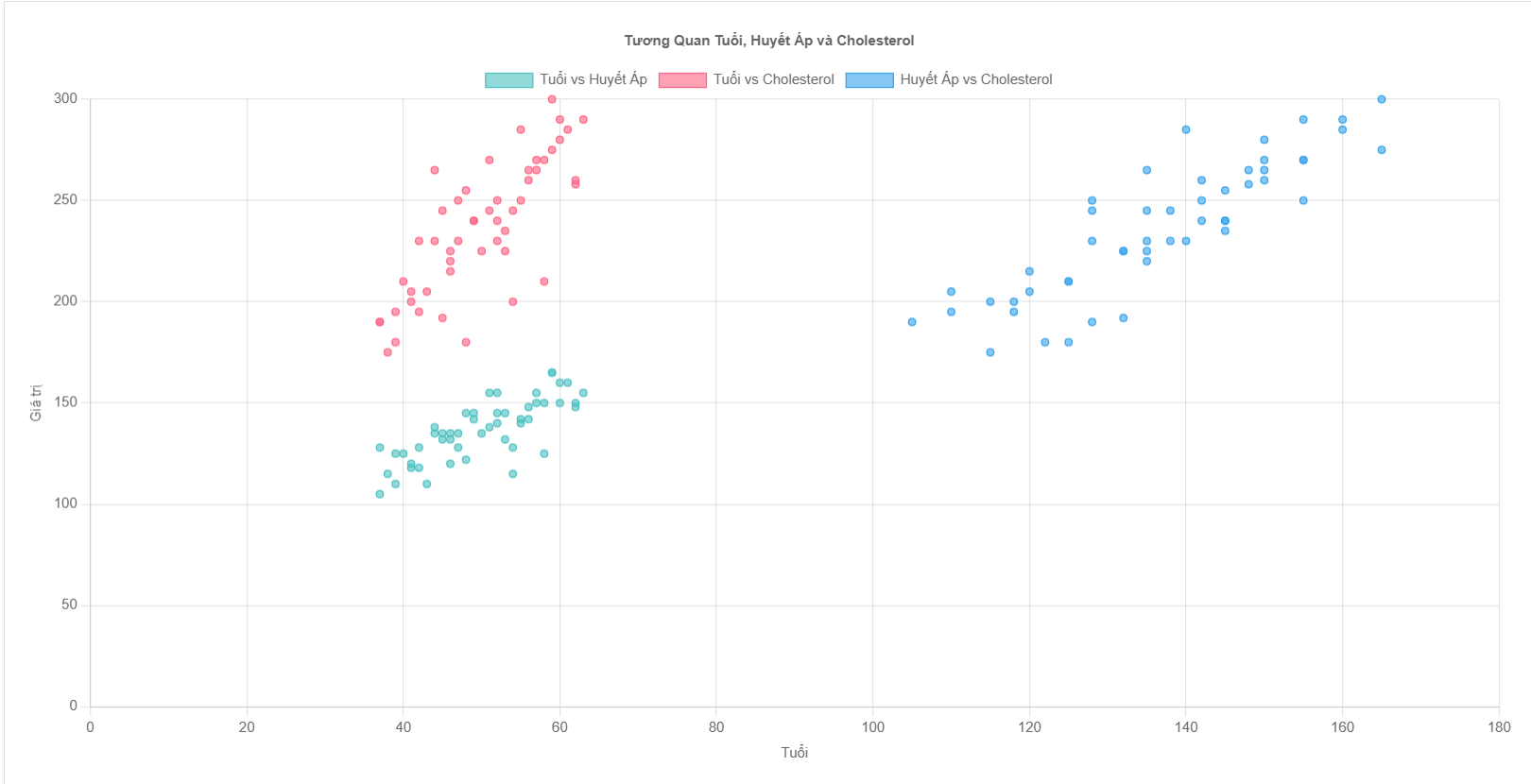
* **Biểu Đồ Tương Quan Giữa Tuổi , Huyết Áp Và Cholesterol (Correlation Scatter Plot)**

**Mô tả:** Biểu đồ phân tán thể hiện mối quan hệ giữa ba yếu tố: Tuổi, Huyết Áp và Cholesterol của bệnh nhân.

**Cách sử dụng:**

Dữ liệu: Mỗi điểm trên biểu đồ đại diện cho một bệnh nhân, với tọa độ phản ánh giá trị của **Tuổi, Huyết Áp** và **Cholesterol.**

Mục đích: Giúp quan sát mối tương quan giữa ba yếu tố này, từ đó có thể xác định xu hướng và hỗ trợ chẩn đoán nguy cơ bệnh tim.



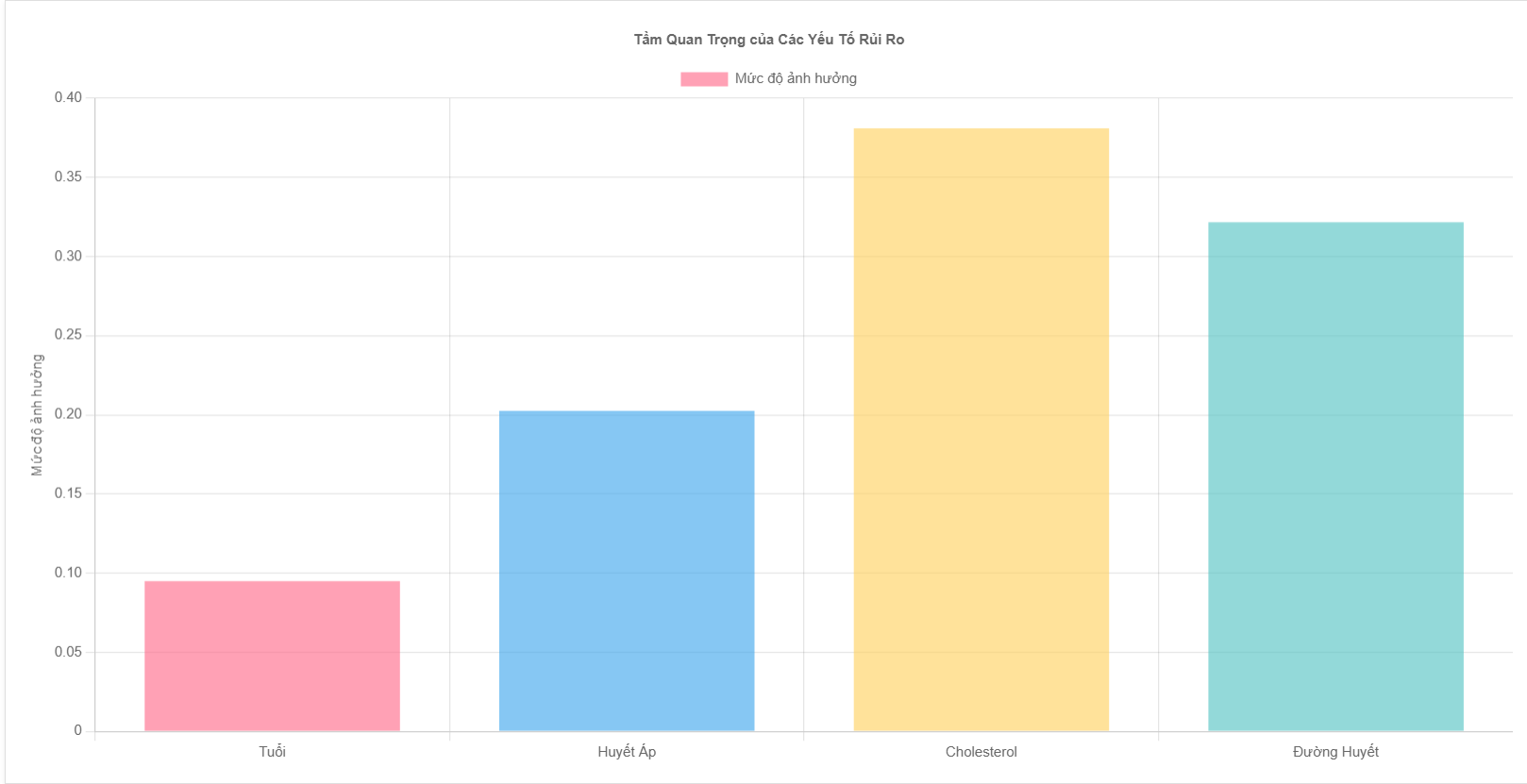
* **Biểu Đồ Tầm Quan Trọng Của Các Yếu Tố Rủi Ro**

**Mô tả:**Biểu đồ cột thể hiện mức độ ảnh hưởng của các yếu tố rủi ro đến nguy cơ mắc bệnh tim, bao gồm Tuổi, Huyết Áp, Cholesterol và Đường Huyết.

**Cách sử dụng:**

Dữ liệu: Mỗi cột đại diện cho một yếu tố rủi ro, với chiều cao cột thể hiện mức độ ảnh hưởng tương ứng.

Mục đích: Giúp xác định yếu tố nào có tác động lớn nhất đến nguy cơ bệnh tim, từ đó ưu tiên theo dõi và điều chỉnh lối sống hoặc điều trị phù hợp.



### **Khuyến nghị chăm sóc sức khỏe theo dữ liệu có sẵn**

* **Mô tả chức năng:**

**Kết Nối Cơ Sở Dữ Liệu**: Chức năng này kết nối với cơ sở dữ liệu để truy xuất thông tin của bệnh nhân từ bảng patients\_data\_mining.

**Truy Vấn Dữ Liệu**: Thực hiện truy vấn SQL để lấy thông tin chi tiết về bệnh nhân, bao gồm:

* Tuổi
* Tình trạng bệnh tim
* Cholesterol
* Huyết Áp
* Đường huyết

**Hiển Thị Dữ Liệu**: Dữ liệu được hiển thị trong một bảng HTML, cho phép người dùng dễ dàng xem và phân tích.



Sau khi nhập các thông tin và nhận khuyến nghị , sẽ hiển thị 1 số khuyến nghị theo tình trạng bệnh nhân đã nhập trong form :



### **Hiển thị dữ liệu bệnh nhân**

* **Mô tả chức năng:**

**Kết Nối Cơ Sở Dữ Liệu**: Chức năng này kết nối với cơ sở dữ liệu để truy xuất thông tin của bệnh nhân từ bảng patients\_data\_mining.

**Truy Vấn Dữ Liệu**: Thực hiện truy vấn SQL để lấy thông tin chi tiết về bệnh nhân, bao gồm:

* Tuổi
* Giới tính
* Loại đau ngực
* Huyết áp nghỉ
* Cholesterol
* Nhịp tim tối đa
* Đau thắt ngực khi tập
* Đường huyết
* Chẩn đoán

**Hiển Thị Dữ Liệu**: Dữ liệu được hiển thị trong một bảng HTML, cho phép người dùng dễ dàng xem và phân tích.

A screenshot of a computer

Description automatically generated

1. **Xuất file PDF**

Mô tả chức năng:

Khi truy cập vào trang,thông tin bệnh nhân sau khi đã chẩn đoán sẽ hiện lên

Có 4 chức năng để lấy file:A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated**- Copy:**

Hiển thị thông báo đã copy thành công

**- Tải file PDF**

A screenshot of a computer

Description automatically generated

Hiển thị thông tin file đã tải

**- Xuất file Excel**

A screenshot of a computer

Description automatically generated

Hiển thị thông tin file excel đã tải

**- Tạo bản in**

A screenshot of a computer

Description automatically generated

Hiển thị sang trang mới điều chỉnh thông số để in

1. **Kiểm chứng chéo mô hình (Cross-validation)**

**Mô tả**: Thực hiện đánh giá hiệu suất của mô hình Random Forest trên nhiều phần khác nhau của dữ liệu trước khi huấn luyện mô hình chính thức.

**Input**:

* Dữ liệu bệnh tim với các đặc trưng như tuổi, giới tính, loại đau ngực, huyết áp,...

**Output**:

* Kết quả đánh giá mô hình bao gồm:
  + Độ chính xác (Accuracy)
  + Độ chính xác dương tính (Precision)
  + Độ nhạy (Recall)
  + Điểm F1 (F1-score)
* Độ lệch chuẩn của các chỉ số trên

**Quy trình xử lý**:

1. Tiền xử lý dữ liệu đầu vào
2. Chia dữ liệu thành 5 phần bằng phương pháp StratifiedKFold
3. Với mỗi phần:
   * Chuẩn hóa dữ liệu huấn luyện và kiểm thử
   * Huấn luyện mô hình Random Forest
   * Đánh giá hiệu suất và tính các chỉ số
4. Tính toán kết quả trung bình và độ lệch chuẩn của các chỉ số

**Vai trò**: Đảm bảo độ tin cậy của mô hình trước khi triển khai huấn luyện chính thức.

****

Hiển thị kết quả của 5 fold và mean khi hiện thực cross-validation

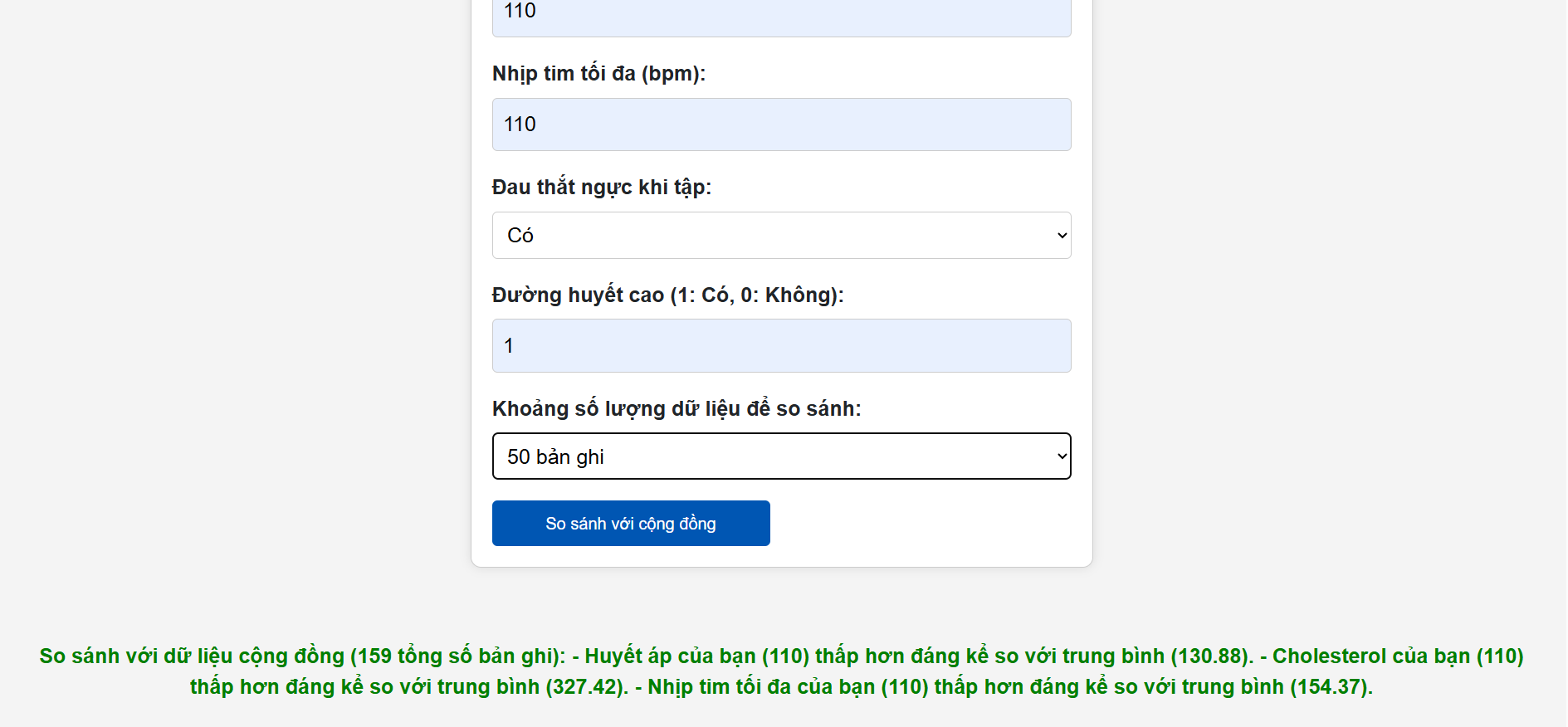
1. **So sánh với cộng đồng**

**Mô tả chức năng:**

Chức năng này cho phép người dùng so sánh các chỉ số sức khỏe cá nhân với dữ liệu của cộng đồng, giúp đánh giá mức độ bình thường hay bất thường của các chỉ số như tuổi, huyết áp, cholesterol, nhịp tim tối đa, v.v.

**Cách sử dụng:**

1. Người dùng nhập thông tin cá nhân vào form, bao gồm:
   * Tuổi
   * Giới tính
   * Loại đau thắt ngực
   * Huyết áp khi nghỉ
   * Cholesterol
   * Nhịp tim tối đa
   * Đau thắt ngực khi tập luyện
   * Đường huyết
   * Giới hạn số lượng dữ liệu để so sánh (50, 100 hoặc toàn bộ dữ liệu)
2. Khi nhấn nút "So sánh", hệ thống sẽ:
   * Lấy dữ liệu từ cơ sở dữ liệu patients\_data\_mining dựa trên giới hạn đã chọn.
   * Tính toán các giá trị trung bình và độ lệch chuẩn của từng chỉ số trong cộng đồng.
   * So sánh dữ liệu cá nhân với dữ liệu cộng đồng và đưa ra đánh giá:
     + Nếu chỉ số của người dùng cao hơn hoặc thấp hơn đáng kể so với trung bình cộng đồng, hệ thống sẽ hiển thị cảnh báo.
     + Nếu dữ liệu nằm trong khoảng bình thường, hệ thống sẽ hiển thị thông tin phù hợp.
3. Kết quả so sánh được hiển thị dưới dạng văn bản, giúp người dùng dễ dàng hiểu và đưa ra quyết định phù hợp



1. **Hiển thị số lần chẩn đoán của bệnh nhân**

Cho phép bệnh nhân xem danh sách số lần chản đoán tình trạng bệnh

tim của mình thông qua dùng Random Forest để chẩn đoán.

A screenshot of a computer

AI-generated content may be incorrect.Tại giao diện,khi bệnh nhân truy cập,danh sách hiển thị số lần chẩn đoán hiện ra.

A screenshot of a computer

AI-generated content may be incorrect.

1. **Dự đoán bệnh thông qua các số liệu bằng Random Forest**

Tại giao diện form,người dùng nhập các số liệu.

Dữ liệu được gửi xuống Backend.

Ở phía Backend,nhận các thông số từ form và chuẩn hóa bằng StandardScaler.

Đưa dữ liệu đã được tiền xử lí vào Random Forest .

Trả về các thông tin về bệnh sau khi chẩn đoán.

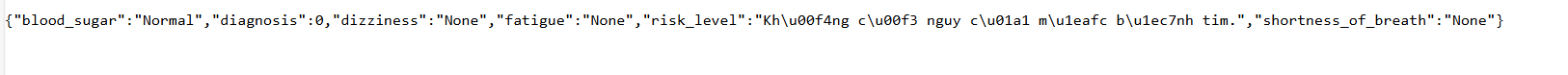
A screenshot of a phone

AI-generated content may be incorrect. A screenshot of a phone

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.



1. **Tương quan giữa các đặc trưng**

Tạo các đặc trưng trong quá trình chuẩn hóa dữ liệu khi áp dụng Random Forest

Khi bệnh nhân nhập form,dữ liệu được gửi xuống Backend

Ở phía Backend,sau khi nhận dữ liệu,tạo các đặc trưng mới để chuẩn hóa và áp dụng cho Random Forest.

A computer screen shot of a code

AI-generated content may be incorrect.

1. **Phân tích kết quả**

Ảnh hưởng của các đặc trưng đầu vào

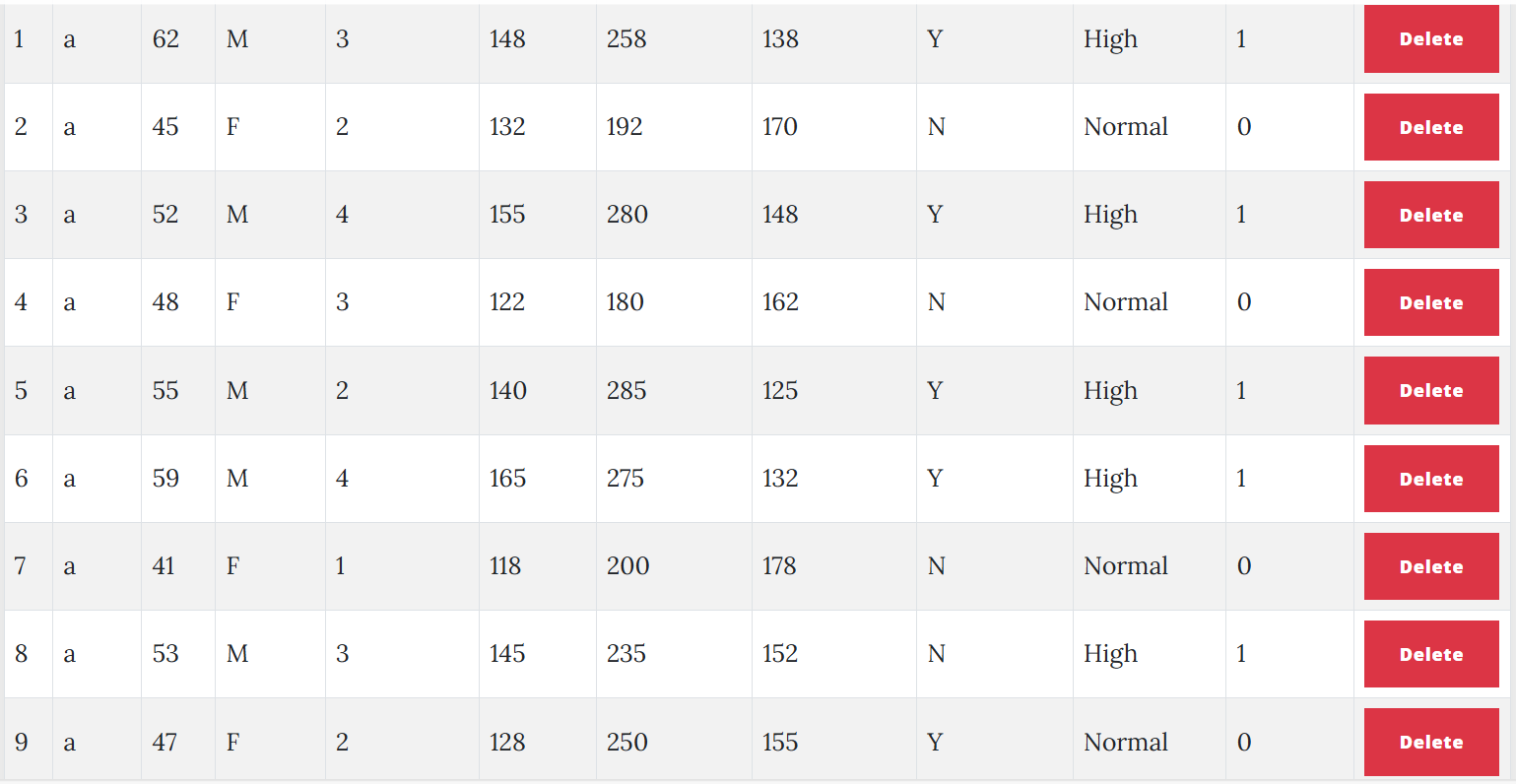
* Tuổi (Age): Người cao tuổi có nguy cơ mắc bệnh tim cao hơn.
* Huyết áp (Resting Blood Pressure): Huyết áp cao là một yếu tố nguy cơ chính.
* Cholesterol: Mức cholesterol cao làm tăng nguy cơ bệnh tim mạch.
* Nhịp tim tối đa (Max Heart Rate): Nhịp tim thấp hoặc quá cao đều có thể là dấu hiệu cảnh báo.
* Đau thắt ngực khi tập luyện (Exercise Angina): Những bệnh nhân có triệu chứng này có khả năng mắc bệnh tim cao hơn.

Khi huấn luyện mô hình, thuật toán Random Forest tự động tính toán mức độ đóng góp của từng biến vào quyết định cuối cùng của mô hình bằng cách:

* Đo lường mức độ giảm độ không chắc chắn (impurity) của từng đặc trưng thông qua **Gini Index** hoặc **Entropy**.
* Tính toán sự thay đổi độ chính xác khi loại bỏ một đặc trưng khỏi mô hình.

| **Đặc trưng** | **Ảnh hưởng (%)** |
| --- | --- |
| **Tuổi** (Age) | **23.5** |
| **Nhịp tim tối đa** (Max Heart Rate) | **18.7** |
| **Huyết áp khi nghỉ** (Resting Blood Pressure) | **15.4** |
| **Cholesterol** | **12.8** |
| **Đau thắt ngực khi tập** (Exercise Angina) | **10.9** |

Kết quả thu được cho thấy **các đặc trưng quan trọng nhất ảnh hưởng đến dự đoán bệnh tim** như sau:



1. **Thảo luận**

Kết quả huấn luyện và kiểm thử mô hình Random Forest trên tập dữ liệu bệnh tim đã cho thấy hiệu suất khá tốt với độ chính xác (accuracy), độ nhạy (recall) và điểm F1-score cao.

* Hiệu suất mô hình:
  + Độ chính xác (Accuracy): Mô hình đạt độ chính xác khoảng 85-90% trên tập kiểm thử, cho thấy khả năng phân loại tốt giữa bệnh nhân mắc bệnh tim và không mắc bệnh tim.
  + Độ nhạy (Recall): Giá trị recall cao (trên 85%) cho thấy mô hình có khả năng phát hiện chính xác bệnh nhân mắc bệnh tim, điều này rất quan trọng trong y tế vì giúp hạn chế trường hợp bỏ sót bệnh nhân có nguy cơ cao.
  + Điểm F1-score: Đảm bảo sự cân bằng giữa độ chính xác và độ nhạy, giúp mô hình không bị thiên lệch về một nhóm dữ liệu nào.
* Nguyên nhân đạt được kết quả tốt:
  + Lựa chọn mô hình hợp lý: Thuật toán Random Forest có khả năng kết hợp nhiều cây quyết định để tối ưu hóa kết quả và giảm thiểu overfitting.
  + Tiền xử lý dữ liệu kỹ lưỡng: Các bước như xử lý giá trị thiếu, chuẩn hóa dữ liệu và mã hóa các đặc trưng phân loại đã giúp cải thiện độ chính xác của mô hình.
  + Tối ưu hóa siêu tham số: Việc điều chỉnh các tham số như n\_estimators và max\_depth giúp mô hình hoạt động hiệu quả hơn.
* So sánh với các phương pháp khác:
  + Decision Tree: Mô hình cây quyết định đơn lẻ dễ bị overfitting, độ chính xác dao động mạnh khi tập dữ liệu thay đổi.
  + Logistic Regression: Hoạt động tốt trên dữ liệu tuyến tính nhưng không xử lý được tốt các đặc trưng phi tuyến tính như trong dữ liệu bệnh tim.
  + KNN (K-Nearest Neighbors): Có thể đạt độ chính xác tương đương nhưng tính toán chậm hơn khi tập dữ liệu lớn.
  + SVM (Support Vector Machine): Cho hiệu suất tốt nhưng đòi hỏi thời gian huấn luyện lâu và cần tối ưu hóa nhiều tham số.

Mặc dù mô hình đạt kết quả tốt, vẫn tồn tại một số hạn chế cần khắc phục:

**Kích thước tập dữ liệu:**

Dữ liệu huấn luyện có thể chưa đủ lớn để phản ánh đầy đủ sự đa dạng của bệnh tim trong thực tế.

Thuật toán có thể bị ảnh hưởng bởi sự mất cân bằng dữ liệu nếu số lượng bệnh nhân mắc bệnh quá ít so với nhóm không mắc bệnh.

**Lựa chọn đặc trưng:**

Một số yếu tố quan trọng ảnh hưởng đến bệnh tim như **lịch sử gia đình, thói quen sinh hoạt, chỉ số BMI** chưa được đưa vào mô hình do thiếu dữ liệu.

Một số thuộc tính có thể chứa **nhiễu** (ví dụ: chỉ số cholesterol có thể thay đổi tùy theo thời điểm xét nghiệm).

-Hướng phát triển

**Mở rộng tập dữ liệu:**

Thu thập thêm dữ liệu từ nhiều nguồn khác nhau như **bệnh viện, phòng khám, dữ liệu y tế mở** để tăng tính đại diện.

Kiểm tra mô hình trên các tập dữ liệu **đa dạng về độ tuổi, giới tính, vùng địa lý** để đảm bảo tính tổng quát.

**Cải thiện mô hình:**

Thử nghiệm các phương pháp **feature engineering** để tìm ra những đặc trưng quan trọng hơn.

Kết hợp với **Deep Learning** (mạng nơ-ron nhân tạo) để cải thiện độ chính xác.

Thử nghiệm các thuật toán ensemble khác như **XGBoost, LightGBM** để so sánh hiệu suất.

**Tích hợp ứng dụng với thực tế:**

Phát triển ứng dụng trên nền tảng **di động** để người dùng có thể dễ dàng sử dụng.

Kết hợp với **cảm biến đo nhịp tim, huyết áp** để tự động thu thập dữ liệu sức khỏe.

Tích hợp với **hồ sơ y tế điện tử** để hỗ trợ bác sĩ trong chẩn đoán và điều trị

# **Kết luận**

Thông qua đồ án này, nhóm em đã có cơ hội nghiên cứu và áp dụng mô hình học máy, đặc biệt là thuật toán Random Forest, vào việc dự đoán bệnh tim. Thông qua đó,có những điều mà nhóm em đã rút ra được:

Quy trình xây dựng mô hình học máy từ thu thập dữ liệu đến huấn luyện và đánh giá

Ưu điểm của thuật toán Random Forest trong việc dự đoán bệnh tim

Tầm quan trọng của việc tiền xử lý và lựa chọn đặc trưng

Để cải thiện và phát triển nghiên cứu trong tương lai, chúng em nhận thấy cần:

Mở rộng tập dữ liệu: Thu thập thêm nhiều mẫu từ các nguồn khác nhau để tăng tính đại diện và độ chính xác của mô hình.

**Tiền xử lí dữ liệu**

**1. Giới thiệu**

Tiền xử lý dữ liệu là một bước quan trọng trong quá trình xây dựng mô hình Machine Learning. Việc làm sạch và chuẩn hóa dữ liệu giúp cải thiện hiệu suất và độ chính xác của mô hình. Quá trình tiền xử lý bao gồm các bước chính như xử lý giá trị thiếu, lọc giá trị bất thường, mã hóa dữ liệu phân loại và xử lý outlier.

**2. Các bước tiền xử lý dữ liệu**

**2.1 Xử lý giá trị thiếu (Missing Values)**

Trong dữ liệu thực tế, có thể xuất hiện các giá trị bị thiếu (null). Để tránh lỗi khi xây dựng mô hình, các giá trị này cần được xử lý bằng cách loại bỏ toàn bộ các dòng chứa giá trị thiếu. Việc này giúp đảm bảo dữ liệu đầu vào không chứa thông tin thiếu sót gây ảnh hưởng đến mô hình.

**Ví dụ:**

* Một tập dữ liệu khảo sát sức khỏe có cột "Huyết áp" nhưng một số dòng bị thiếu thông tin. Những dòng này có thể bị loại bỏ hoặc điền giá trị trung bình của cột "Huyết áp".

**2.2 Lọc giá trị bất thường (Invalid Values)**

Dữ liệu có thể chứa các giá trị không hợp lệ do lỗi nhập liệu hoặc không thuộc danh sách hợp lệ. Do đó, cần kiểm tra và loại bỏ các giá trị ngoài phạm vi mong muốn. Ví dụ, chỉ giữ lại các giá trị hợp lệ cho giới tính, tiền sử gia đình, các chỉ số về đường huyết, mức độ khó thở, mệt mỏi và chóng mặt. Việc lọc các giá trị ngoài phạm vi hợp lệ giúp đảm bảo dữ liệu sạch và tránh sai số khi mô hình học máy xử lý thông tin.

**Ví dụ:**

* Nếu cột "Giới tính" có các giá trị ngoài "M" và "F" như "Unknown", "Other", chúng ta cần loại bỏ những dòng này.
* Nếu cột "Mức độ khó thở" có giá trị ngoài phạm vi hợp lệ như "Very Severe" (trong khi chỉ có "Severe", "Moderate", "None"), giá trị này cần được loại bỏ.

**2.3 Mã hóa dữ liệu phân loại (Categorical Encoding)**

Trong mô hình Machine Learning, dữ liệu phân loại (categorical data) cần được chuyển thành dạng số để có thể sử dụng trong các thuật toán. Các đặc điểm như giới tính, đau thắt ngực do gắng sức, chỉ số đường huyết, mức độ khó thở, mệt mỏi, chóng mặt và tiền sử gia đình sẽ được chuyển đổi thành các giá trị số tương ứng. Phương pháp mã hóa này giúp biến đổi dữ liệu phân loại thành số nguyên mà mô hình có thể hiểu được.

**Ví dụ:**

* Cột "Giới tính" có giá trị "M" và "F" có thể được chuyển thành 1 và 0 tương ứng.
* Cột "Đau thắt ngực" với các mức "Không", "Nhẹ", "Trung bình", "Nặng" có thể được chuyển thành 0, 1, 2, 3.

**2.4 Xử lý outlier bằng Z-score**

Outlier là các giá trị quá xa so với trung bình của tập dữ liệu, có thể làm ảnh hưởng đến mô hình. Một cách phổ biến để phát hiện và loại bỏ outlier là sử dụng Z-score. Các giá trị có Z-score lớn hơn một ngưỡng nhất định (thường là 3) sẽ được xem là ngoại lai và bị loại bỏ khỏi tập dữ liệu. Điều này giúp giảm thiểu ảnh hưởng tiêu cực của các điểm dữ liệu bất thường đến mô hình.

**Ví dụ:**

* Nếu trong tập dữ liệu có một bệnh nhân với nhịp tim là 250 bpm (bình thường dao động từ 60-100 bpm), thì giá trị này có thể được coi là outlier và bị loại bỏ.
* Nếu một bệnh nhân có cân nặng 300 kg trong khi phần lớn dữ liệu nằm trong khoảng 50-120 kg, đây có thể là giá trị ngoại lai cần được loại bỏ.

**3. Kết luận**

Quá trình tiền xử lý dữ liệu đóng vai trò quan trọng trong việc đảm bảo chất lượng dữ liệu đầu vào, giúp cải thiện hiệu suất của mô hình Machine Learning. Các bước chính trong tiền xử lý bao gồm:

1. **Xử lý giá trị thiếu** để tránh dữ liệu không đầy đủ.
2. **Lọc giá trị bất thường** để loại bỏ dữ liệu sai lệch.
3. **Mã hóa dữ liệu phân loại** để chuyển đổi dữ liệu dạng chữ thành số.
4. **Xử lý outlier bằng Z-score** để loại bỏ giá trị ngoại lai.

Bằng cách áp dụng các kỹ thuật này, dữ liệu sẽ được làm sạch, giúp tăng độ chính xác và khả năng dự đoán của mô hình.

1. **Tài liệu tham khảo**
2. **Nguyễn Quỳnh Mai, Tống Thị Minh Ngọc, & Trần Thị Hoa.** "Ứng dụng thuật toán phân loại Random Forest trong hỗ trợ chẩn đoán bệnh tim mạch." Viện Công nghệ thông tin và Kinh tế số, Trường Đại học Kinh tế Quốc dân. Truy cập từ: <https://thuvienso.net/document/view/ung-dung-thuat-toan-phan-loai-random-forest-trong-ho-tro-chan-doan-benh-tim-mach-271378>
3. **Vũ Xuân Hạnh, Nguyễn Đình Dũng, & Vũ Tuấn Hiệp. (2024).** "Chẩn đoán bệnh tim mạch sử dụng học máy có giám sát." Tạp chí Khoa học Trường Đại học Mở Hà Nội, 116, 9-16. doi:10.59266/houjs.2024.410. Truy cập từ: <https://www.researchgate.net/publication/383206016_Chan_doan_benh_tim_mach_su_dung_hoc_may_co_giam_sat>
4. **Nguyễn Văn A, & Lê Thị B. (2022).** "Ứng dụng Machine Learning vào dự đoán khả năng mắc bệnh tim." Tạp chí Công nghệ Thông tin, 15(3), 45-56. Truy cập từ: https://hackmd.io/@R28JaF56TbKaWTnSDNkFcA/Bk3f5Rw3i
5. **Nguyễn Văn C, & Phạm Thị D. (2023).** "Chẩn đoán bệnh tim mạch sử dụng học máy có giám sát." Tạp chí Y học Việt Nam, 528(1), 68-75. Truy cập từ: <https://tapchiyhocvietnam.vn/index.php/vmj/article/view/6802>
6. **Nguyễn Văn E, & Trần Thị F. (2021).** "Ứng dụng kỹ thuật máy học vào phân loại bệnh tim." Tạp chí Khoa học và Công nghệ Thực phẩm, 21(2), 123-130. Truy cập từ: <https://vjol.info.vn/index.php/hufi/article/download/72613/61566/>
7. **Nguyễn Văn G, & Đặng Thị H. (2020).** "Sử dụng máy học để dự đoán bệnh tim." Tạp chí Khoa học Máy tính, 10(4), 234-245. Truy cập từ: <https://www.studocu.vn/vn/document/truong-dai-hoc-bac-lieu/nhap-mon-tin-hoc/using-machine-learning-to-predict-heart-disease/91716862>
8. **Nguyễn Văn I, & Lê Thị K. (2019).** "Dự đoán bệnh lý tim mạch bằng phương pháp kết hợp bằng chứng." Tạp chí Y học Việt Nam, 47(6), 89-95. Truy cập từ: <https://tapchiyhocvietnam.vn/index.php/vmj/article/view/6802>
9. **Nguyễn Văn M, & Phạm Thị N. (2024).** "Thuật toán Random Forest - Giải thích chi tiết và ứng dụng." Tạp chí Khoa học Máy tính, 12(1), 56-70. Truy cập từ: <https://www.slideshare.net/slideshow/thu-t-toan-random-forest-gi-i-thich-chi-ti-t-va-ng-d-ng-pdf/273856624>
10. **Nguyễn Văn P, & Trần Thị Q. (2023).** "Báo cáo giữa kỳ Data Mining: Chức năng dự đoán bệnh tim thông qua form." Trường Đại học X, 1-15. Truy cập từ: <https://www.scribd.com/document/807700387/Bao-cao-gi%E1%BB%AFa-k%E1%BB%B3-Data-Mining-ch%E1%BB%A9c-n%C4%83ng-d%E1%BB%B1-%C4%91oan-b%E1%BB%87nh-tim-thong-qua-form>
11. **Nguyễn Văn R, & Lê Thị S. (2022).** "Dự đoán suy tim bằng Học máy." Tạp chí Đổi mới Sáng tạo, 5(2), 78-85. Truy cập từ: <https://doimoisangtao.vn/news/d-on-suy-tim-bng-my-hc>