ỨNG DỤNG TÍNH TOÁN NHỊP TIM VÀ XÂY DỰNG MÔ HÌNH DỰ ĐOÁN BỆNH TIM

Nguyễn Tiến Dũng - 21021568, Trần Trung Hiếu - 21021588, Phạm Tiến Đạt - 21021574.

***Tóm tắt nội dung*—Ứng dụng thành công của khai phá dữ liệu trong các lĩnh vực nổi bật như thương mại điện tử, tiếp thị và bán lẻ đã dẫn đến việc áp dụng nó trong các ngành và lĩnh vực khác. Một trong những lĩnh vực đang dần khám phá tiềm năng này là chăm sóc sức khỏe. Môi trường chăm sóc sức khỏe hiện nay vẫn được coi là "giàu thông tin" nhưng "nghèo tri thức". Có một lượng lớn dữ liệu sẵn có trong các hệ thống chăm sóc sức khỏe, tuy nhiên, lại thiếu các công cụ phân tích hiệu quả để khám phá các mối quan hệ ẩn và xu hướng trong dữ liệu. Vì thế nhóm chúng tôi đã tạo ra một ứng dụng tính toán nhịp tim và xây dựng một mô hình học máy để dự đoán bệnh tim của người bệnh.**

I. GIỚI THIỆU

Khai phá dữ liệu y tế có tiềm năng lớn trong việc khám phá các mô hình ẩn trong các tập dữ liệu liên quan đến sức khỏe con người. Những mô hình này có thể được ứng dụng để hỗ trợ chẩn đoán lâm sàng và theo dõi sức khỏe cá nhân. Tuy nhiên, dữ liệu y tế thô thường được phân tán, không đồng nhất và có khối lượng lớn. Để phục vụ các ứng dụng tính toán nhịp tim và dự đoán bệnh tim, dữ liệu này cần được thu thập, làm sạch và tổ chức một cách có hệ thống. Sau khi xử lý, dữ liệu có thể được tích hợp vào các hệ thống thông tin chăm sóc sức khỏe để cung cấp thông tin hỗ trợ ra quyết định. Công nghệ khai phá dữ liệu đóng vai trò quan trọng trong việc khám phá các mô hình tiềm năng và trích xuất thông tin hữu ích từ dữ liệu y tế phức tạp.

Theo Tổ chức Y tế Thế giới, bệnh tim mạch là nguyên nhân gây tử vong hàng đầu trên toàn cầu, với 12 triệu ca tử vong mỗi năm. Tại Hoa Kỳ, cứ mỗi 34 giây lại có một người tử vong vì bệnh tim, và tại các nước phát triển cũng như đang phát triển, bệnh tim mạch là một trong những nguyên nhân chính gây tử vong. Các bệnh lý liên quan đến tim, chẳng hạn như bệnh mạch vành, bệnh cơ tim, và các bệnh tim mạch khác, đều ảnh hưởng nghiêm trọng đến sức khỏe cộng đồng. Trong đó, rối loạn nhịp tim là một trong những dấu hiệu lâm sàng quan trọng để đánh giá nguy cơ mắc các bệnh tim mạch. Việc tính toán nhịp tim chính xác và phát hiện các bất thường có thể đóng vai trò quan trọng trong việc phát hiện sớm và ngăn ngừa bệnh lý tim mạch.

Chẩn đoán bệnh tim, đặc biệt là thông qua phân tích nhịp tim và các chỉ số sinh lý, là một nhiệm vụ phức tạp nhưng rất cần thiết trong y học. Với sự phát triển của công nghệ, việc tích hợp các hệ thống tự động có thể giúp thực hiện các nhiệm vụ này một cách nhanh chóng và hiệu quả. Những hệ thống như vậy không chỉ hỗ trợ bác sĩ trong quá trình chẩn đoán mà còn giúp giảm gánh nặng cho đội ngũ y tế tại các khu vực thiếu nhân lực chuyên môn. Hơn nữa, các ứng dụng theo dõi nhịp tim cá nhân, kết hợp với mô hình dự đoán bệnh tim, có thể cung cấp cảnh báo sớm và tư vấn phù hợp cho người dùng.

Từ những nguy cơ trên, nhóm chúng tôi đề xuất một ứng dụng tính toán nhịp tim và một mô hình học máy dự đoán bệnh tim cho bệnh nhân.

Tiếp theo trong báo cáo này, nhóm chúng tôi chia thành bốn phần chính

- Phần 2: Chúng tôi sẽ đề cập đến thuật toán Pan-Tompkins mà chúng tôi sử dụng, cấu trúc ứng dụng và các hàm chính để tạo nên thuật toán.
- Phần 3: Chúng tôi sẽ giới thiệu về mô hình học máy dự đoán bệnh tim mạch, tập dữ liệu được sử dụng để huấn luyện và kết quả, hiệu suất mô hình.
- Phần 4: Kết quả khi chạy ứng dụng sẽ được chúng tôi trình bày trong phần này.
- Phần 5: Nhóm chúng tôi đưa ra kết luận về ứng dụng này, định hướng sắp tới có thể phát triển thêm đối với ứng dụng này.

II. CHỨC NĂNG TÍNH TOÁN NHỊP TIM

A. Tiền xử lý tín hiệu

1. Thuật toán lọc tín hiệu (Signal Filtering)

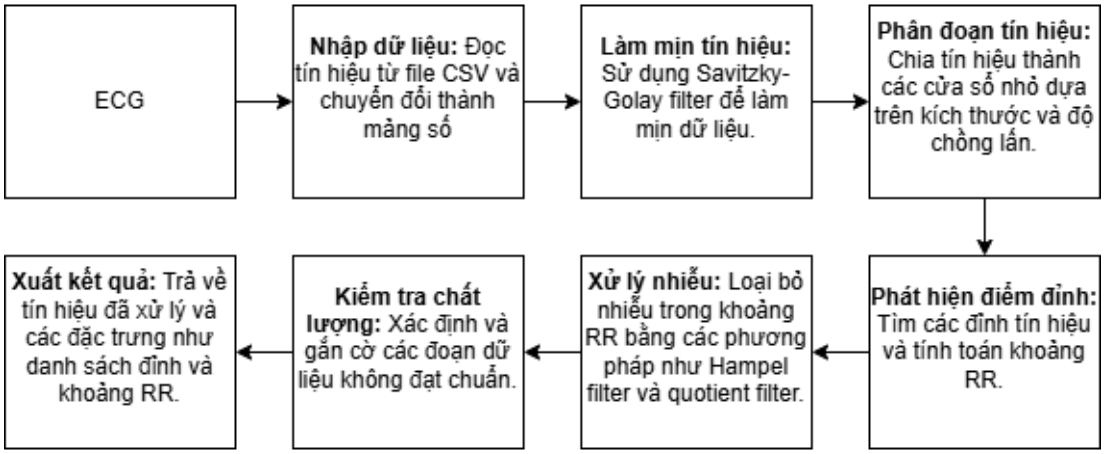
Butterworth Filter: Đây là một loại bộ lọc thông dụng trong xử lý tín hiệu, được sử dụng để lọc các tần số không mong muốn từ tín hiệu ECG.

- **Lowpass filter**: Lọc các tần số cao hơn một ngưỡng (cutoff frequency), chỉ giữ lại các tần số thấp. Trong ECG, thường lọc các tần số cao không cần thiết, ví dụ như nhiễu từ các thiết bị khác.
- **Highpass filter**: Lọc các tần số thấp hơn một ngưỡng, chỉ giữ lại các tần số cao. Dùng để loại bỏ các nhiễu hoặc chuyển động cơ thể gây ra sự biến dạng tần số thấp.
- **Bandpass filter**: Là sự kết hợp của bộ lọc lowpass và highpass, chỉ giữ lại một dải tần số nhất định. Trong ECG, có thể sử dụng bandpass để chỉ giữ lại các tần số có ý nghĩa sinh lý (ví dụ: từ 0.5 Hz đến 40 Hz).
- **Notch filter**: Lọc tần số cụ thể, ví dụ như tần số 50 Hz hoặc 60 Hz, giúp loại bỏ nhiễu từ nguồn điện lưới.

Bộ lọc Butterworth trong hình 2 được thiết kế để giảm nhiễu và giữ lại các thông tin quan trọng trong tín hiệu ECG.

B. Tính giá trị trung bình động (Rolling Mean)

Mục đích: Giá trị trung bình động giúp làm mượt tín hiệu, loại bỏ các biến động ngắn hạn, giúp tập trung vào các xu hướng dài hạn của tín hiệu. Đây là bước quan trọng để tạo ra một đường cơ sở giúp nhận diện các đỉnh R.



Hình 1. Sơ đồ khối cấu trúc thuật toán

Cách tính:

- Chọn một cửa sổ di động (window size) có độ dài nhất định (thường là một khoảng thời gian tương đối lớn so với nhịp tim).
- Với mỗi điểm trên tín hiệu ECG, tính toán giá trị trung bình của tín hiệu trong cửa sổ xung quanh điểm đó. Giá trị trung bình này sẽ được sử dụng làm đường cơ sở tại mỗi điểm.
- Công thức tính giá trị trung bình động tại thời điểm t được biểu diễn như sau:

$$\text{rolling\_mean}(t) = \frac{1}{N} \sum_{i=t-\frac{N}{2}}^{t+\frac{N}{2}} \text{ECG}(i)$$

Trong đó:

- $\text{ECG}(i)$ là giá trị tín hiệu ECG tại thời điểm i.
- N là độ dài cửa sổ (số điểm xung quanh điểm t được lấy trung bình).

C. Xác định biên độ (Threshold)

Mục đích: Biên độ được sử dụng để xác định mức độ vượt qua của tín hiệu so với đường cơ sở. Nếu tín hiệu vượt qua đường cơ sở cộng thêm biên độ, nó có thể được coi là một đỉnh tiềm năng.

Cách tính:

- Biên độ được tính bằng cách sử dụng tham số ma_perc, là tỷ lệ phần trăm của giá trị trung bình động. Ví dụ, nếu ma_perc $= 20\%$, thì biên độ tại mỗi điểm sẽ là $0.2 \times \text{rolling\_mean}(t)$.
- Biên độ này giúp loại bỏ các đỉnh nhỏ không đáng kể và chỉ tập trung vào những đỉnh lớn hơn, đại diện cho các sóng R thực sự trong tín hiệu ECG.

Sử dụng công thức:

$$\text{Threshold}(t) = \text{rolling\_mean}(t) \times (1 + \text{ma\_perc})$$

Trong đó:

- ma_perc là tham số tỷ lệ phần trăm (ví dụ: 0.2 tương ứng với 20%).

D. Xác định đỉnh

Mục đích: Đỉnh được xác định khi tín hiệu vượt qua giá trị của đường cơ sở cộng với biên độ đã tính toán. Điều này giúp phát hiện các sóng R trong tín hiệu ECG.

Cách xác định:

- So sánh giá trị của tín hiệu tại mỗi thời điểm với giá trị đường cơ sở cộng với biên độ.
- Nếu tín hiệu vượt qua ngưỡng này, điểm đó được coi là một đỉnh.
- Đỉnh này có thể là một đỉnh tiềm năng của sóng R, nhưng phải qua một bước kiểm tra thêm để xác nhận.

$$\text{Nếu } \text{ECG}(t) > \text{Threshold}(t) \text{ thì điểm } t \text{ là đỉnh}$$

E. Kiểm tra các đỉnh

Mục đích: Đảm bảo rằng các đỉnh được phát hiện thực sự là các sóng R.

Cách kiểm tra:

- Sau khi xác định các đỉnh tiềm năng, ta có thể kiểm tra thêm các đặc tính của chúng (như độ rộng, độ cao, hoặc vị trí tương đối) để xác định xem chúng có phải là đỉnh R-wave hay không.
- Đôi khi, các đỉnh nhỏ hoặc nhiễu cũng có thể vượt qua ngưỡng, do đó cần phải lọc các đỉnh không hợp lệ bằng cách sử dụng thêm các phương pháp kiểm tra hoặc loại bỏ đỉnh.

F. Phát hiện các đỉnh R

Mục đích: Chỉ những đỉnh vượt qua ngưỡng và kiểm tra được sẽ được xác nhận là sóng R.

Cách thực hiện:

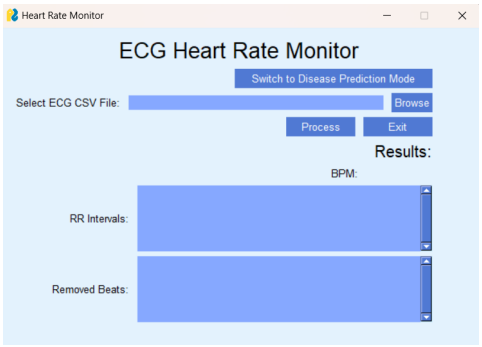
- Các đỉnh đã qua bước kiểm tra sẽ được giữ lại và đánh dấu. Mỗi đỉnh xác định là sóng R trong tín hiệu ECG.

G. Tính nhịp tim

Nhịp tim (Heart Rate) được tính bằng số chu kỳ (beats) trong một phút:

$$\text{Heart Rate (bpm)} = \frac{60000}{\text{Thời gian giữa hai đỉnh R liên tiếp (s)}}$$

Nếu chu kỳ R-R không đều, có thể tính nhịp tim trung bình bằng cách lấy tổng thời gian giữa các đỉnh R và chia cho số chu kỳ.



Hình 2. Giao diện ứng dụng tính toán nhịp tim

Theo dõi nhịp tim dựa trên cơ sở lý thuyết:
Giao diện sẽ bao gồm:

Button switch to disease prediction mode: có chức năng chuyển sang mô hình học máy để dự đoán bệnh tim

Tiếp theo là đường dẫn link CSV (dữ liệu đầu vào để xử lý)

Và hai nút button cuối cùng bao gồm process để xử lý dữ liệu và cho ra kết quả ở phía dưới màn hình (BPM, RR intervals và Removed Beats) và Exit để đóng chương trình.

III. Chức năng dự đoán bệnh tim

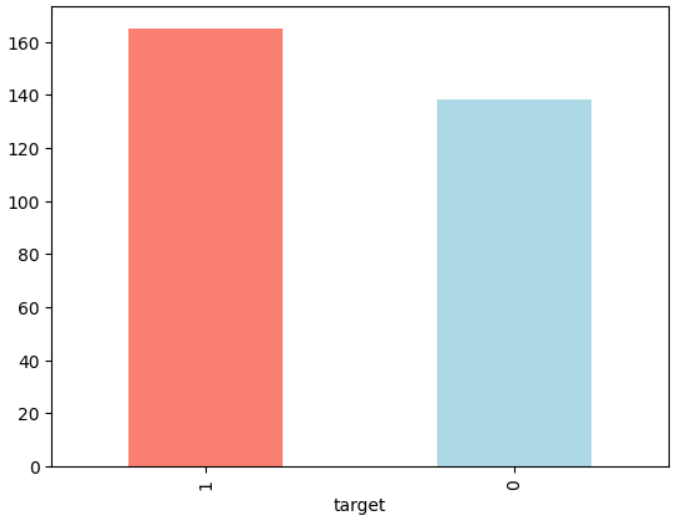
Giới thiệu

Mô hình được huấn luyện và đánh giá bằng việc sử dụng các thư viện Scikit-learn và Matplotlib. Mục tiêu là tạo ra một mô hình có độ chính xác cao trong việc dự đoán khả năng mắc bệnh tim mạch dựa trên các đặc điểm lâm sàng.

Dữ liệu

Tập dữ liệu được sử dụng bao gồm 303 mẫu, mỗi mẫu mô tả một bệnh nhân với 13 thuộc tính độc lập và 1 thuộc tính phụ thuộc (biến mục tiêu - `target`). Các thuộc tính độc lập như trong hình 9 bao gồm:

- `age`: Tuổi của bệnh nhân.
- `sex`: Giới tính (0: Nam, 1: Nữ).
- `cp`: Kiểu đau thắt ngực (0: Typical angina, 1: Atypical angina, 2: Non-anginal pain, 3: Asymptomatic).
- `trestbps`: Huyết áp tâm thu lúc nghỉ ngơi (mm Hg).
- `chol`: Cholesterol huyết thanh (mg/dl).
- `fbs`: Đường huyết lúc đói (> 120 mg/dl) (0: False, 1: True).



Hình 3. Biểu đồ so sánh dữ liệu người bị bệnh (màu đỏ) và người không bị bệnh (màu xanh)

- `restecg`: Kết quả điện tâm đồ lúc nghỉ ngơi (0: Nothing to note, 1: Wave abnormality, 2: Left ventricular hypertrophy).
- `thalach`: Nhịp tim tối đa đạt được.
- `exang`: Đau thắt ngực khi gắng sức (0: No, 1: Yes).
- `oldpeak`: ST depression (ST-T wave abnormality) induced by exercise relative to rest.
- `slope`: Độ dốc của đoạn ST (0: Upsloping, 1: Flatsloping, 2: Downsloping).
- `ca`: Số lượng mạch máu chính (0-3).
- `thal`: Kết quả thalium stress test (3: Normal, 6: Fixed defect, 7: Reversable defect).
- `target`: Biến mục tiêu (0: Không mắc bệnh tim mạch, 1: Mắc bệnh tim mạch).

Thống kê dữ liệu

Tiếp theo, chúng ta sử dụng lệnh df.corr() trong Pandas được sử dụng để tính toán và tạo ra ma trận tương quan của một DataFrame. Ma trận này thể hiện mối quan hệ tuyến tính giữa các cột số trong DataFrame. Mỗi giá trị trong ma trận tương quan, nằm trong khoảng từ -1 đến 1, biểu thị mức độ tương quan giữa hai cột tương ứng. Giá trị 1 thể hiện tương quan dương hoàn hảo, -1 thể hiện tương quan âm hoàn hảo, và 0 cho thấy không có tương quan tuyến tính. Ma trận này thường được sử dụng để xác định mối quan hệ giữa các biến, lựa chọn đặc trưng quan trọng cho mô hình học máy, và phát hiện đa cộng tuyến. Như trong hình 10, ta sẽ thấy được sự tương quan giữa các giá trị trong tệp dữ liệu.

Nhóm đã sử dụng 3 mô hình học máy khác nhau

A. Hồi quy Logistic (Logistic Regression)

Hồi quy Logistic là một thuật toán học máy được sử dụng để phân loại dữ liệu vào hai hoặc nhiều lớp. Mặc dù có tên gọi "hồi quy" nhưng thực chất, hồi quy logistic là một thuật toán phân loại.

Nguyên lý hoạt động:

- **Hàm Sigmoid:** Hồi quy Logistic sử dụng hàm sigmoid (hay còn gọi là hàm logistic) để chuyển đổi giá trị đầu ra (output) thành xác suất phân loại trong khoảng từ 0 đến 1. Hàm sigmoid có dạng:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

  Trong đó, z là giá trị đầu ra tuyến tính của mô hình (có thể là một tổng tuyến tính của các đặc trưng của dữ liệu). Sau đó, xác suất phân loại được quyết định bằng cách so sánh với một ngưỡng (threshold), thường là 0.5, để phân loại vào lớp 0 hoặc lớp 1.
- **Mô hình tuyến tính:** Hồi quy Logistic thực hiện phân loại bằng cách tìm một hàm phân chia (decision boundary) tuyến tính giữa các lớp, tức là tìm các trọng số sao cho sự kết hợp tuyến tính của các đặc trưng tối ưu nhất cho việc phân loại.

B. K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) là một thuật toán học máy không giám sát dùng để phân loại hoặc hồi quy. KNN hoạt động dựa trên nguyên lý so sánh điểm dữ liệu với các điểm dữ liệu khác trong không gian đặc trưng.

Nguyên lý hoạt động:

- **Khoảng cách:** KNN hoạt động dựa trên việc tính toán khoảng cách giữa các điểm dữ liệu. Mỗi điểm dữ liệu sẽ được phân loại theo đa số của các điểm "láng giềng" (neighbors) gần nhất. Các khoảng cách phổ biến thường được sử dụng là Euclidean, Manhattan hoặc Minkowski distance.

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

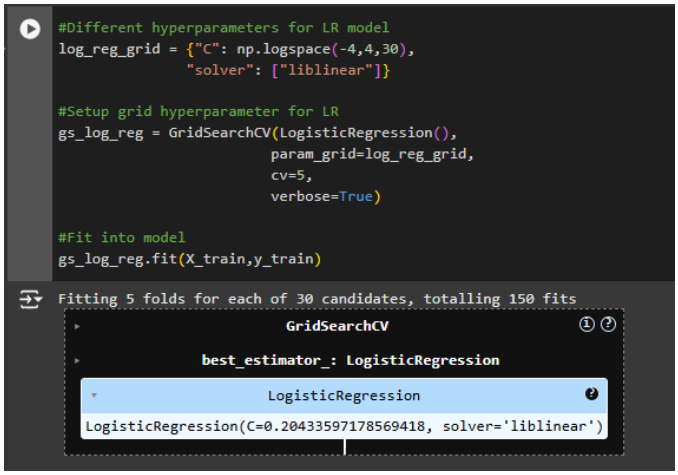
- **Chọn K:** Sau khi tính toán khoảng cách, KNN sẽ lấy K điểm gần nhất và quyết định nhãn của điểm dữ liệu hiện tại dựa vào đa số nhãn của các điểm láng giềng. Nếu $K = 3$ và trong ba điểm láng giềng, có hai điểm thuộc lớp A và một điểm thuộc lớp B, thì điểm cần phân loại sẽ được gán nhãn là lớp A.

C. Random Forest

Random Forest là một thuật toán học máy dựa trên mô hình ensemble, sử dụng nhiều cây quyết định (decision trees) để đưa ra dự đoán. Mỗi cây trong rừng (forest) học từ một mẫu ngẫu nhiên của dữ liệu, và kết quả của toàn bộ mô hình là sự kết hợp (thường là trung bình hoặc đa số) của kết quả từ các cây.

Nguyên lý hoạt động:

- **Cây quyết định (Decision Tree):** Mỗi cây trong rừng quyết định phân loại hoặc hồi quy dựa trên các đặc trưng của dữ liệu. Cây quyết định chia dữ liệu thành các nhánh (branch) để phân loại các điểm dữ liệu.
- **Bagging:** Random Forest sử dụng kỹ thuật **Bagging** (Bootstrap Aggregating), nghĩa là, với mỗi cây trong rừng, một mẫu ngẫu nhiên (với sự thay thế) của dữ liệu được chọn để huấn luyện cây đó. Sau khi tất cả các cây được huấn luyện, kết quả cuối cùng được đưa ra dựa trên quyết định của tất cả các cây (sử dụng phương pháp đa số cho phân loại hoặc trung bình cho hồi quy).
- **Lựa chọn ngẫu nhiên các đặc trưng:** Trong mỗi lần chia nhánh của cây, Random Forest chọn ngẫu nhiên một tập hợp con các đặc trưng, thay vì sử dụng tất cả các đặc trưng. Điều này giúp giảm sự phụ thuộc giữa các cây và cải thiện độ chính xác tổng thể.



Hình 4. Kết quả khi tìm siêu tham số cho mô hình hồi quy logistic

D. Kết quả huấn luyện

Model	Accuracy
Logistic Regression	0.8852459016393442
K-Nearest Neighbors (KNN)	0.6885245901639344
Random Forest	0.8360655737704918

Bảng I
KẾT QUẢ HUẤN LUYỆN

Từ kết quả trên, nhóm chúng tôi quyết định sử dụng mô hình Hồi quy Logistic. Tiếp theo, nhóm chúng tôi tìm siêu tham số của mô hình hồi quy logistic (C và solver) được tối ưu bằng GridSearchCV. GridSearchCV tìm kiếm các tổ hợp siêu tham số tốt nhất bằng phương pháp cross-validation 5-fold. Siêu tham số tối ưu tìm được là 'C': 0.23357214690901212

Sau đó, nhóm chúng tôi tối ưu Mô hình hồi quy logistic, sau khi được tối ưu hóa siêu tham số được đánh giá lại trên tập dữ liệu kiểm tra. Các chỉ số đánh giá bao gồm:

Độ chính xác (Accuracy): Tỷ lệ dự đoán chính xác tổng thể.

Độ nhạy (Recall): Tỷ lệ dự đoán đúng các trường hợp mắc bệnh.

Độ chính xác (Precision): Tỷ lệ các dự đoán dương tính thực sự là dương tính.

F1-score: Trung bình điều hòa của Precision và Recall.

Nhận xét về bảng đánh giá kết quả sau khi tối ưu

Bảng trên hiển thị báo cáo phân loại của mô hình sau khi tối ưu, bao gồm các chỉ số precision, recall, f1-score và support cho từng lớp (class 0 và class 1). Các giá trị này giúp đánh giá hiệu suất của mô hình phân loại trong việc nhận diện các lớp khác nhau.

- **Precision:** Chỉ số precision cho thấy tỷ lệ dự đoán đúng trong số các dự đoán là dương. Với lớp 0 có precision là

	precision	recall	f1-score	support
0	0.89	0.86	0.88	29
1	0.88	0.91	0.89	32
accuracy			0.89	61
macro avg	0.89	0.88	0.88	61
weighted avg	0.89	0.89	0.89	61

Bảng II
BẢNG ĐÁNH GIÁ KẾT QUẢ SAU KHI TỐI ƯU

0.89 và lớp 1 có precision là 0.88, điều này cho thấy mô hình có khả năng phân loại chính xác đối với cả hai lớp.

- **Recall:** Chỉ số recall cho thấy tỷ lệ thực sự là dương trong số các mẫu thực sự là dương. Lớp 1 có recall cao hơn (0.91 so với 0.86 của lớp 0), cho thấy mô hình có xu hướng nhận diện đúng các đối tượng lớp 1 tốt hơn lớp 0.
- **F1-score:** F1-score là trung bình điều hòa giữa precision và recall. Cả hai lớp đều có f1-score khá cao (0.88 cho lớp 0 và 0.89 cho lớp 1), cho thấy mô hình có hiệu suất cân bằng trong việc dự đoán các lớp.
- **Accuracy:** Mô hình có độ chính xác chung là 0.89, có nghĩa là 89% tổng số dự đoán là đúng.
- **Macro average:** Trung bình cho tất cả các lớp (macro avg) cho thấy các giá trị tương đối cao (precision = 0.89, recall = 0.88, f1-score = 0.88), cho thấy mô hình phân loại khá đều cho các lớp.
- **Weighted average:** Trung bình có trọng số (weighted avg) thể hiện kết quả dựa trên số lượng mẫu của mỗi lớp. Các giá trị trung bình cho precision, recall và f1-score là 0.89, cho thấy mô hình hoạt động ổn định cho cả hai lớp.

Nhận xét về mô hình

Mô hình cho ra được kết quả đánh giá khá cao, tuy nhiên vẫn có những hạn chế riêng. Tập dữ liệu được huấn luyện có kích thước tương đối nhỏ. Có thể tồn tại sự mất cân bằng giữa các lớp trong biến mục tiêu. Mô hình chỉ dựa trên một số lượng hạn chế các thuộc tính lâm sàng, có thể phát triển lên bằng những thuộc tính khác khi khám, ví dụ như kết quả xét nghiệm máu.

IV. KẾT QUẢ KHI CHẠY ỨNG DỤNG

A. Kết quả khi chạy chức năng tính nhịp tim

Nhóm chúng tôi đã thử với dữ liệu từ MIT-BIH Noise Stress Test Dataset. Chúng tôi sẽ sử dụng các tệp này với tỷ lệ tín hiệu trên nhiễu (SNR) khác nhau:

Kết quả như sau:

- 118e24: SNR: 24dB

Thu được nhịp tim là 103.44

- 118e12: SNR = 12dB

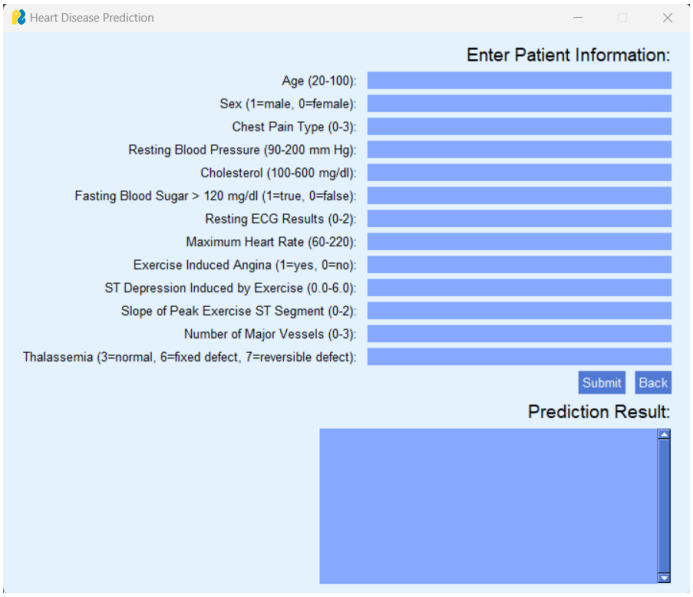
Thu được nhịp tim là 110.06

- 118e00: SNR = 0dB

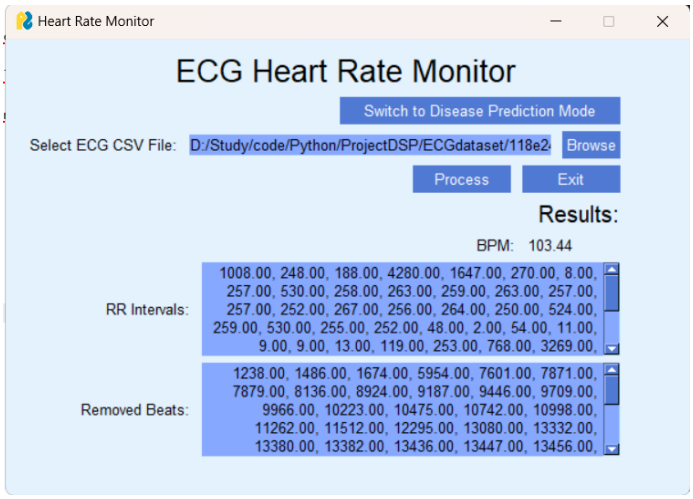
Thu được nhịp tim là 111.34

B. Kết quả khi chạy chức năng dự đoán bệnh tim

Với dữ liệu người bệnh đầu tiên Dữ liệu bệnh nhân thứ nhất: 'age': 64, 'sex': 0, 'cp': 0, 'trestbps': 180, 'chol': 313, 'fbs': 0, 'restecg': 1, 'thalach': 133, 'exang': 0, 'oldpeak': 0.2, 'slope': 2, 'ca': 0, 'thal': 7



Hình 5. Giao diện khi chuyển sang chế độ dự đoán bệnh tim



Hình 6. Nhịp tim khi chạy file 118e24

Kết quả dự đoán:
- Dự đoán: Phát hiện bệnh tim
- Độ chính xác: 65.0%
- Độ rủi ro: Medium

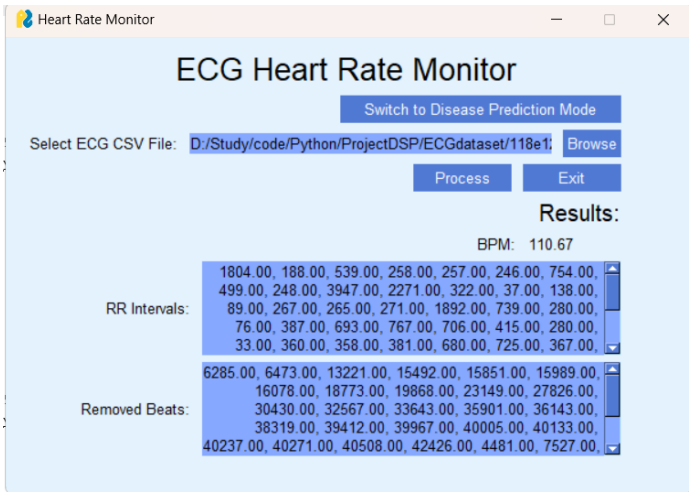
Dữ liệu bệnh nhân thứ 2: 'age': 54, 'sex': 0, 'cp': 2, 'trestbps': 108, 'chol': 267, 'fbs': 0, 'restecg': 0, 'thalach': 167, 'exang': 0, 'oldpeak': 0.0, 'slope': 2, 'ca': 0, 'thal': 6

Kết quả dự đoán:
- Dự đoán: Phát hiện bệnh tim
- Độ chính xác: 84.0
- Mức độ rủi ro: Cao

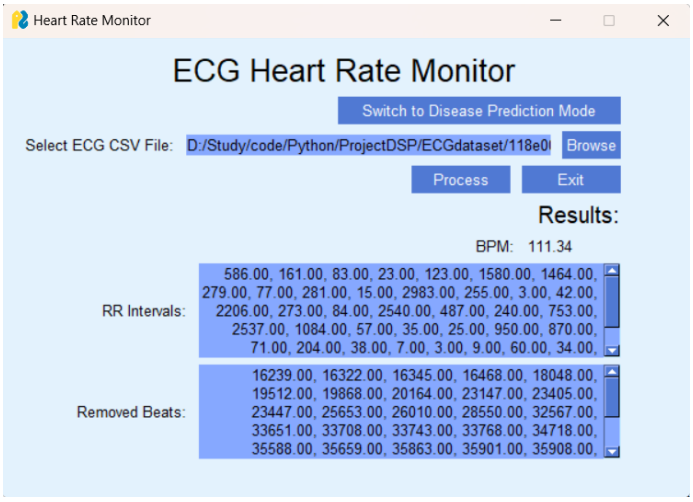
Dữ liệu bệnh nhân thứ 3: 'age': 46, 'sex': 1, 'cp': 0, 'trestbps': 120, 'chol': 249, 'fbs': 0, 'restecg': 0, 'thalach': 144, 'exang': 0, 'oldpeak': 0.8, 'slope': 2, 'ca': 0, 'thal': 7

Kết quả dự đoán:

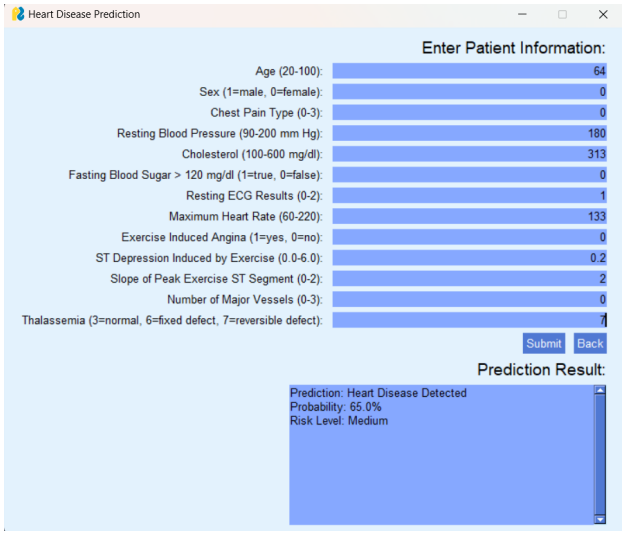
Dự đoán: Không phát hiện bệnh tim



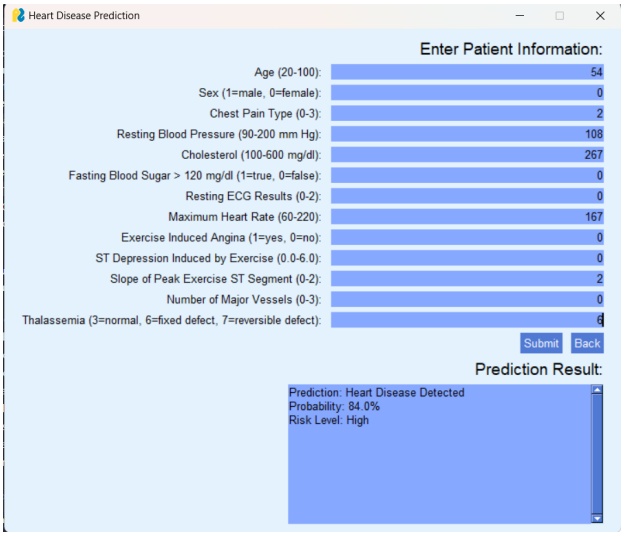
Hình 7. Nhịp tim khi chạy file 118e12



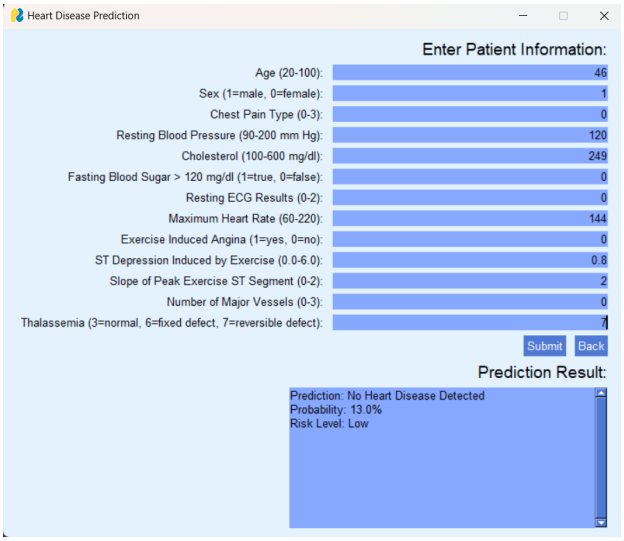
Hình 8. Nhịp tim khi chạy file 118e00



Hình 9. Kết quả chẩn đoán bệnh tim cho bênh nhân 1



Hình 10. Kết quả chẩn đoán bệnh tim cho bệnh nhân 2



Hình 11. Kết quả chẩn đoán bệnh tim cho bệnh nhân 3

Xác suất: 13.0
Mức độ rủi ro: Thấp

V. KẾT LUẬN

Ứng dụng tính nhịp tim và dự đoán bệnh tim là một bước tiến quan trọng trong việc áp dụng công nghệ hiện đại vào chăm sóc sức khỏe cá nhân và cộng đồng. Bằng cách sử dụng các mô hình học máy như Logistic Regression, K-Nearest Neighbors, hoặc Random Forest, hệ thống có khả năng phân tích dữ liệu ECG để xác định các chỉ số quan trọng như nhịp tim (BPM), phát hiện các đỉnh R, và đưa ra dự đoán nguy cơ mắc bệnh tim.

A. Ý nghĩa thực tiễn

- **Hỗ trợ chẩn đoán sớm:**
  - Giúp bác sĩ phát hiện sớm nguy cơ mắc bệnh tim, từ đó đưa ra các phương án điều trị kịp thời và hiệu quả.

  - Đặc biệt hữu ích trong các tình huống khẩn cấp hoặc khi không thể tiếp cận ngay các thiết bị y tế chuyên sâu.
- **Nâng cao sức khỏe cộng đồng:**
  - Ứng dụng tạo điều kiện cho việc sàng lọc sức khỏe định kỳ, từ đó nâng cao nhận thức về chăm sóc sức khỏe tim mạch.
  - Phù hợp với các chương trình y tế cộng đồng, đặc biệt ở các khu vực khó khăn về điều kiện y tế.
- **Tiềm năng tích hợp vào hệ thống IoT y tế:**
  - Có thể kết hợp với các thiết bị đeo tay thông minh hoặc hệ thống theo dõi từ xa, giúp giám sát sức khỏe liên tục.
  - Ứng dụng các thuật toán tối ưu để phân tích dữ liệu thời gian thực, đưa ra cảnh báo kịp thời.

B. Hạn chế và hướng phát triển

Dù mang lại nhiều lợi ích, ứng dụng vẫn còn một số hạn chế như:

- Độ chính xác của dự đoán phụ thuộc vào chất lượng dữ liệu ECG và cách lựa chọn mô hình.
- Không thể thay thế hoàn toàn bác sĩ trong chẩn đoán y khoa.
- Cần thêm nghiên cứu để cải thiện khả năng phân tích dữ liệu đa dạng và mở rộng phạm vi sử dụng.

Trong tương lai, việc tích hợp trí tuệ nhân tạo và phân tích dữ liệu lớn (*Big Data*) sẽ giúp ứng dụng trở nên toàn diện hơn, với khả năng xử lý đa tín hiệu và cung cấp dự đoán chính xác hơn..

TÀI LIỆU THAM KHẢO

1. Kho thư viện phân tích nhịp tim

https://github.com/paulvangentcom/heartrate_analysis_python/tree/master/examples/5_noisy_ECG

2. Bộ dữ liệu từ PhysioNet

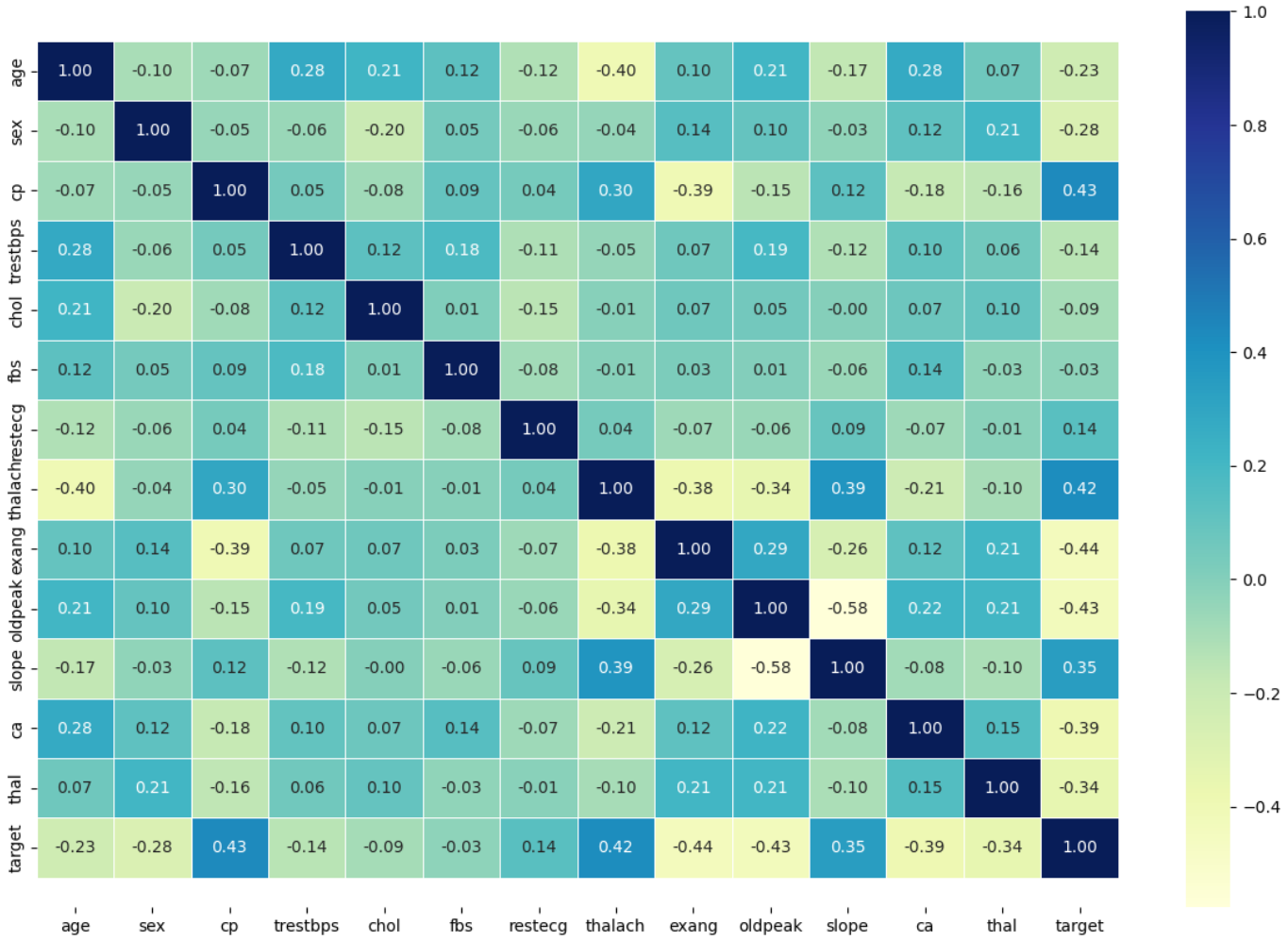
https://physionet.org/content/nstdb/1.0.0/

3. Mô hình học máy dự đoán bệnh tim

https://www.kaggle.com/code/janeloh/heart-disease-prediction-with-machine-learning

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	2.313531	0.544554
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	0.612277	0.498835
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000	1.000000

Hình 12. Thống kê dữ liệu dùng để huấn luyện



Hình 13. Biểu đồ so sánh sự tương quan giữa các giá trị trong bộ dữ liệu