ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

DATA MINING (CO3029)

Bài tập lớn - Học kì 242

DỰ ĐOÁN NGUY CƠ BỆNH TIM

GVHD: Thầy Huỳnh Văn Thống
SVTH: Trương Tấn Sang 2212918

TP. HỒ CHÍ MINH, 2025

Mục lục

Lời nói đầu

Nhóm xin chân thành cảm ơn thầy Huỳnh Văn Thống đã phụ trách giảng dạy, cung cấp một số kiến thức nền tảng làm bước đệm để nhóm hoàn thành bài tập lớn học kỳ 242.
Quá trình thực hiện đề tài "Dự đoán nguy cơ Bệnh tim" cũng là một quá trình tự học sâu rộng đối với em. Nhóm đã trau dồi được các kiến thức thực tiễn, từ các bước tiền xử lý dữ liệu (như One-Hot Encoding, StandardScaler), so sánh ưu nhược điểm của các thuật toán (Logistic Regression, KNN, Decision Tree, Random Forest), đến việc tinh chỉnh siêu tham số (hyperparameter tuning) để chống overfitting. Ngoài ra, nhóm còn học được các kiến thức về Học máy, giải thuật,... Bên cạnh đó, nhóm còn có cơ hội rèn luyện cách trình bày, báo cáo kết quả một cách khoa học và logic.
Nhóm xin gửi lời cảm ơn chân thành đến thầy và hy vọng sẽ tiếp tục nhận được sự hướng dẫn, chia sẻ kiến thức từ thầy trong những môn học tiếp theo.

1 Danh sách thành viên & Phân chia công việc

STT	Họ và tên	MSSV	Nội dung	Mức độ hoàn thành
1	Trương Tấn Sang	2212918	Mô hình Logistic Regression Mô hình KNN Mô hình Decision Tree Mô hình Random Forest PDF Báo cáo Slide báo cáo	100%

2 Giới thiệu dự án

Trong bối cảnh y tế hiện đại, bệnh lý tim mạch (CVDs) là một trong những nguyên nhân gây tử vong hàng đầu trên toàn thế giới, đặt ra gánh nặng nghiêm trọng cho hệ thống y tế và xã hội. Việc phát hiện và chẩn đoán sớm bệnh tim đóng vai trò then chốt trong việc cải thiện tiên lượng cho bệnh nhân, giảm tỷ lệ tử vong và tối ưu hóa chi phí điều trị. Tuy nhiên, việc chẩn đoán sớm đòi hỏi phân tích nhiều yếu tố rủi ro phức tạp như tuổi tác, chỉ số cholesterol, huyết áp, và các chỉ số lâm sàng khác.

Do đó, đề tài **"Phân tích các yếu tố rủi ro ảnh hưởng tới Bệnh tim và Xây dựng mô hình Phân loại"** mang tính cần thiết và cấp thiết trong thực tiễn. Bằng việc áp dụng các phương pháp Khai phá Dữ liệu và Học máy hiện đại, đề tài hướng tới mục tiêu không chỉ tìm hiểu mối quan hệ giữa các yếu tố rủi ro mà còn xây dựng mô hình phân loại (classification) có độ chính xác cao, hỗ trợ các chuyên gia y tế trong việc sàng lọc và đưa ra quyết định kịp thời.

Để đạt được mục tiêu đó, nhóm đề tài sử dụng các phương pháp phân loại phổ biến trong lĩnh vực học máy, bao gồm:

- **Hồi quy Logistic (Logistic Regression):** Mô hình tuyến tính cơ sở, hiệu quả cho bài toán phân loại nhị phân và cung cấp một baseline mạnh mẽ.
- **Cây quyết định (Decision Tree):** Phương pháp phân loại dữ liệu theo dạng cây, dễ diễn giải và trực quan hóa, rất quan trọng cho việc ra quyết định(decision making).
- **Rừng ngẫu nhiên (Random Forest):** Mô hình ensemble dựa trên cây, có khả năng xử lý tốt các mối quan hệ phức tạp và giảm thiểu hiện tượng overfitting.
- **K-nearest neighbors (KNN):** Phương pháp phân loại dựa trên khoảng cách (instance-based), hiệu quả trong việc tìm ra các mẫu cục bộ trong không gian đặc trưng.

Với cách tiếp cận đa phương pháp như trên, đề tài kỳ vọng sẽ đưa ra được đánh giá toàn diện về các yếu tố ảnh hưởng và lựa chọn được mô hình dự đoán bệnh tim phù hợp nhất với dữ liệu thực tế.

3 Tổng quan công việc

3.1 Giới thiệu về phương pháp luận

Trong dự án này, chúng tôi áp dụng phương pháp luận CRISP-DM (Cross-Industry Standard Process for Data Mining) để thực hiện quá trình khai thác dữ liệu. Phương pháp này bao gồm 6 giai đoạn chính:

- **Hiểu biết về nghiệp vụ (Business Understanding):** Xác định mục tiêu và yêu cầu của dự án (dự đoán khả năng mắc bệnh tim).
- **Hiểu biết về dữ liệu (Data Understanding):** Thu thập và phân tích dữ liệu ban đầu (bộ dữ liệu `heart.csv`).
- **Chuẩn bị dữ liệu (Data Preparation):** Tiền xử lý, làm sạch, mã hóa và chuẩn hóa dữ liệu.
- **Mô hình hóa (Modeling):** Xây dựng và đánh giá các mô hình (Logistic Regression, KNN, Decision Tree, Random Forest).
- **Đánh giá (Evaluation):** Đánh giá kết quả và kiểm tra mức độ đáp ứng mục tiêu của các mô hình.
- **Triển khai (Deployment):** Giai đoạn này nằm ngoài phạm vi của bài tập lớn.

3.2 Phương pháp tiếp cận

Dự án sử dụng phương pháp tiếp cận dựa trên dữ liệu (Data-Driven Approach) để phân tích và dự đoán khả năng mắc bệnh tim. Các bước thực hiện bao gồm:

- **Thu thập dữ liệu:** Sử dụng bộ dữ liệu `heart.csv` được cung cấp.
- **Tiền xử lý dữ liệu:** Xử lý dữ liệu trùng lặp (`drop_duplicates`), mã hóa One-Hot Encoding cho các biến phân loại và chuẩn hóa `StandardScaler` cho các biến số (đối với các mô hình nhạy cảm với thang đo như Logistic Regression và KNN).
- **Phân tích dữ liệu:** Phân tích thống kê mô tả, trực quan hóa ma trận tương quan và phân bố dữ liệu để hiểu rõ các đặc trưng.
- **Xây dựng mô hình:** Lựa chọn, huấn luyện và tinh chỉnh bốn thuật toán phân loại (Logistic Regression, KNN, Decision Tree, Random Forest) để tìm ra mô hình dự đoán tốt nhất.

3.3 Các kỹ thuật sử dụng

Dự án áp dụng các kỹ thuật khai thác dữ liệu sau:

- **Phân tích thống kê:**
  - Thống kê mô tả (thông tin cơ bản, heatmap tương quan).
  - Trực quan hóa phân bố dữ liệu (ví dụ: boxplot).
- **Học máy (Machine Learning):**
  - **Học có giám sát (Supervised Learning):** Sử dụng cho bài toán phân loại nhị phân (dự đoán `HeartDisease` = 0 hoặc 1).
  - **Các thuật toán:** Logistic Regression (Baseline), K-Nearest Neighbors (KNN), Decision Tree, và Random Forest (Ensemble).
- **Xử lý dữ liệu:**
  - Mã hóa One-hot (`pd.get_dummies`) cho các biến categorical.
  - Chuẩn hóa dữ liệu (`StandardScaler`) cho các biến numerical.
- **Trực quan hóa dữ liệu:**
  - Biểu đồ ma trận nhầm lẫn (Confusion Matrix heatmap).
  - Biểu đồ phương pháp khuỷu tay (Elbow Method) để tìm k tối ưu cho KNN.
  - Trực quan hóa cây quyết định (`plot_tree`).

3.4 Đánh giá và kiểm chứng

Quá trình đánh giá và kiểm chứng được thực hiện thông qua:

- **Phân chia dữ liệu:**
  - Tập huấn luyện (training set): 80% dữ liệu.
  - Tập kiểm tra (test set): 20% dữ liệu (`random_state=42`).
- **Đánh giá mô hình:**
  - Độ chính xác (Accuracy).
  - Độ chính xác (Precision).
  - Độ nhạy (Recall).
  - F1-score (chỉ số cân bằng giữa Precision và Recall).
- **Kiểm chứng chéo và Tinh chỉnh:**
  - Sử dụng `GridSearchCV` với kiểm chứng chéo 5-fold (cv=5) để tìm các siêu tham số (hyperparameters) tối ưu cho mỗi mô hình, tập trung vào việc tối ưu hóa chỉ số `recall` hoặc `f1-score`.

3.5 Tính mới và đóng góp

Dự án này tập trung vào việc áp dụng và so sánh một cách có hệ thống bốn mô hình phân loại phổ biến (từ tuyến tính, dựa trên khoảng cách, đến ensemble) trên cùng một bộ dữ liệu dự đoán bệnh tim.

- **Đóng góp chính:** Cung cấp một phân tích so sánh chi tiết về hiệu suất, ưu nhược điểm của từng thuật toán (Logistic Regression, KNN, Decision Tree, Random Forest) trong bối cảnh cụ thể của bài toán y tế này.
- **Tinh chỉnh mô hình:** Báo cáo này không chỉ xây dựng mô hình mặc định mà còn thực hiện tinh chỉnh siêu tham số (hyperparameter tuning) bằng `GridSearchCV` để tối ưu hóa hiệu suất, đặc biệt là các chỉ số quan trọng như Recall và F1-Score, vốn rất có ý nghĩa trong chẩn đoán y khoa.

4 Cơ sở lý thuyết

4.1 Tổng quan về học máy và phân lớp dữ liệu

Học máy (Machine Learning) là một nhánh của trí tuệ nhân tạo (AI), cho phép máy tính học từ dữ liệu và cải thiện hiệu suất dự đoán mà không cần lập trình một cách cụ thể. Trong bối cảnh dự đoán bệnh tim, học máy đóng vai trò quan trọng trong việc xây dựng các mô hình có khả năng khai thác các mối quan hệ phức tạp giữa nhiều chỉ số y tế khác nhau.

Học máy được chia thành nhiều loại, trong đó phổ biến nhất là:

- **Học có giám sát (Supervised Learning):** Mô hình được huấn luyện trên tập dữ liệu có nhãn, tức là mỗi mẫu dữ liệu đều đi kèm với kết quả đầu ra mong muốn. Đây là phương pháp chính được sử dụng trong bài toán này.
- **Học không giám sát (Unsupervised Learning):** Tập trung vào việc tìm kiếm cấu trúc ẩn trong dữ liệu không có nhãn (ví dụ: phân cụm)
- **Học tăng cường (Reinforcement Learning):** Mô hình học thông qua tương tác với môi trường và tối ưu hóa phần thưởng.

Trong khuôn khổ bài tập lớn này, bài toán chính được xác định là một bài toán **phân loại nhị phân (binary classification)**, trong đó đầu ra cần dự đoán là một giá trị rời rạc (bệnh nhân có khả năng bị bệnh tim (`HeartDisease=1`) hay không (`HeartDisease=0`)).

4.2 Các thuật toán học máy

Trong quá trình xây dựng mô hình dự đoán bệnh tim, việc lựa chọn thuật toán phù hợp là yếu tố then chốt để đảm bảo độ chính xác và khả năng tổng quát hóa của mô hình. Dưới đây là các thuật toán được sử dụng trong bài toán này.

4.2.1 Hồi quy Logistic (Logistic Regression)

Đây là mô hình phân loại tuyến tính đơn giản, nhanh và dễ hiểu. Mô hình này sẽ đóng vai trò làm **baseline model** (mô hình cơ sở) để đo lường hiệu quả của các thuật toán phức tạp hơn sau này.

4.2.2 K-Nearest Neighbors (KNN)

KNN là một thuật toán học có giám sát *dựa trên thể hiện (instance-based)*. Thuật toán này không tạo ra một hàm số cụ thể mà lưu trữ toàn bộ tập huấn luyện. Khi dự đoán, nó tìm k điểm dữ liệu gần nhất trong tập huấn luyện và dự đoán dựa trên *phiếu bầu đa số (majority vote)* của các điểm đó.

4.2.3 Cây quyết định (Decision Tree)

Decision Tree hoạt động bằng cách liên tục chia dữ liệu thành các nhóm nhỏ hơn dựa trên các phép toán điều kiện về đặc trưng (ví dụ: `Age < 50?`). Nó tìm ra câu hỏi tốt nhất (giúp phân tách `target=0` và `target=1` rõ nhất) tại mỗi bước. Mô hình này rất mạnh mẽ nhưng có một rủi ro lớn: **overfitting** (học quá khớp).

4.2.4 Rừng ngẫu nhiên (Random Forest)

Rừng ngẫu nhiên là một thuật toán học máy dạng ensemble (tập hợp), được sử dụng cho cả bài toán phân loại và hồi quy. Thuật toán này xây dựng nhiều cây quyết định (decision trees) trong quá trình huấn luyện và cho ra kết quả dự đoán bằng cách... bỏ phiếu số đông (đối với phân loại) từ các cây. Ý tưởng chính... là bằng cách kết hợp nhiều cây yếu (weak learners), mô hình tổng thể sẽ đạt được hiệu suất tốt hơn và hạn chế được hiện tượng quá khớp (overfitting).

Nó chống overfitting bằng 2 kỹ thuật chính:

1. **Bagging (Bootstrap Aggregating):** Mỗi cây được huấn luyện trên một mẫu *con* ngẫu nhiên (lấy có lặp lại) từ tập dữ liệu huấn luyện.

2. **Feature Randomness:** Tại mỗi nút (node) của cây, thay vì xem xét *tất cả* các đặc trưng, cây chỉ được phép chọn ngẫu nhiên một *tập con* các đặc trưng... để tìm ra phép chia tốt nhất.

5 Preprocessing Data (Tiền xử lý dữ liệu)

5.1 Giới thiệu chung về vai trò của tiền xử lý dữ liệu

Trong bất kỳ dự án khai thác dữ liệu nào, tiền xử lý dữ liệu đóng vai trò quan trọng vì dữ liệu thô (raw data) thường không đầy đủ, không chính xác, hoặc không nhất quán. Quá trình tiền xử lý giúp cải thiện chất lượng dữ liệu trước khi áp dụng các mô hình học máy, từ đó giúp tăng độ chính xác của các dự đoán hoặc phân tích.

5.1.1 Mục đích của tiền xử lý dữ liệu

Mục đích chính của tiền xử lý dữ liệu là làm sạch và chuẩn hóa dữ liệu, giảm thiểu nhiễu và những giá trị không hợp lệ, từ đó nâng cao chất lượng của dữ liệu. Việc này cải thiện hiệu quả của các mô hình học máy và khai thác dữ liệu, giúp các thuật toán có thể học và dự đoán chính xác hơn.

5.1.2 Các vấn đề thường gặp trong dữ liệu thô

- **Thiếu giá trị (Missing values):** Nhiều thuộc tính hoặc cột trong dữ liệu có thể thiếu thông tin.
- **Nhiễu (Noisy data):** Các giá trị không hợp lý hoặc ngoại lệ (outliers) có thể xuất hiện trong dữ liệu.
- **Không nhất quán (Inconsistent data):** Dữ liệu có thể được ghi nhận theo các cách khác nhau (ví dụ: "Male" và "M").
- **Trùng lặp (Duplicate data):** Nhiều hàng (records) giống hệt nhau có thể làm sai lệch kết quả phân tích.

5.2 Ý nghĩa các trường dữ liệu

Bộ dữ liệu được sử dụng là "Heart Failure Prediction Dataset", một bộ dữ liệu tổng hợp từ 5 bộ dữ liệu khác nhau của UCI (Cleveland, Hungarian, Switzerland, Long Beach VA, Stalog). Bộ dữ liệu cuối cùng gồm 918 mẫu quan sát và 12 cột.

- **Age:** Tuổi của bệnh nhân (Năm).
- **Sex:** Giới tính (M: Male, F: Female).
- **ChestPainType:** Loại đau ngực (TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic).
- **RestingBP:** Huyết áp khi nghỉ (mm Hg).
- **Cholesterol:** Nồng độ Cholesterol (mm/dl).
- **FastingBS:** Đường huyết lúc đói (1: nếu $>$ 120 mg/dl, 0: ngược lại).
- **RestingECG:** Kết quả điện tâm đồ khi nghỉ (Normal: Bình thường, ST: Có bất thường sóng ST-T, LVH: Phì đại tâm thất trái).
- **MaxHR:** Nhịp tim tối đa đạt được (60-202).
- **ExerciseAngina:** Có đau thắt ngực khi vận động hay không (Y: Yes, N: No).
- **Oldpeak:** Chỉ số ST depression.
- **ST_Slope:** Độ dốc của đoạn ST khi tập thể dục (Up: Dốc lên, Flat: Phẳng, Down: Dốc xuống).
- **HeartDisease (Biến mục tiêu):** 1 = Bệnh tim, 0 = Bình thường.

5.3 Quá trình tiền xử lý dữ liệu

Quá trình tiền xử lý được thực hiện theo các bước chuẩn trong pipeline của mô hình học máy.

5.3.1 Đọc và kiểm tra dữ liệu

Quá trình bắt đầu bằng việc đọc dữ liệu từ file `heart.csv`. Kết quả kiểm tra ban đầu cho thấy:

- Kích thước dữ liệu: 918 dòng và 12 cột.
- Giá trị thiếu: Một số giá trị `Cholesterol` được ghi nhận là 0, đây có thể xem là giá trị thiếu hoặc bất thường. Tuy nhiên, trong khuôn khổ bài tập này, nhóm quyết định giữ nguyên các giá trị này.
- Dữ liệu trùng lặp: Phát hiện thấy một số hàng trùng lặp, nhóm đã tiến hành loại bỏ các hàng này bằng `data.drop_duplicates()` để đảm bảo tính duy nhất của dữ liệu.

5.3.2 Kỹ thuật biến đổi đặc trưng

Hệ thống thực hiện các biến đổi đặc trưng để chuẩn bị dữ liệu cho việc phân tích:

5.3.2.a Xử lý cột phân loại (Categorical Data)

Các mô hình học máy yêu cầu đầu vào là số. Do đó, các cột dữ liệu dạng chữ (categorical) được chuyển đổi bằng kỹ thuật **One-Hot Encoding** (sử dụng `pandas.get_dummies`).

- Các cột được mã hóa: `Sex`, `ChestPainType`, `FastingBS`, `RestingECG`, `ExerciseAngina`, `ST_Slope`.
- Tham số `drop_first=True` được sử dụng để tránh hiện tượng đa cộng tuyến (multicollinearity) cho các mô hình tuyến tính (như Logistic Regression).

5.3.2.b Xử lý cột số (Numerical Data) - Chuẩn hóa

Một số thuật toán như **Logistic Regression** và **KNN** rất nhạy cảm với thang đo (scale) của các đặc trưng. Các đặc trưng có giá trị lớn (như `Cholesterol`) có thể lấn át các đặc trưng có giá trị nhỏ (như `Oldpeak`).

- **Hành động:** Sử dụng `StandardScaler` (Chuẩn hóa Z-score) để biến đổi các cột số.
- Các cột được chuẩn hóa: `Age`, `RestingBP`, `Cholesterol`, `MaxHR`, `Oldpeak`.
- **Lưu ý quan trọng:** Các mô hình dựa trên cây (Decision Tree, Random Forest) không yêu cầu bước chuẩn hóa này. Do đó, nhóm chuẩn bị 2 bộ dữ liệu: một bộ đã chuẩn hóa (cho LR, KNN) và một bộ chưa chuẩn hóa (cho DT, RF).

5.3.3 Phân chia dữ liệu (Train-Test Split)

Để đánh giá mô hình một cách khách quan, dữ liệu được chia thành 2 tập:

- **Tập huấn luyện (Training Set):** 80% dữ liệu, dùng để train cho mô hình.
- **Tập kiểm thử (Test Set):** 20% dữ liệu, dùng để đánh giá hiệu suất cuối cùng của mô hình.

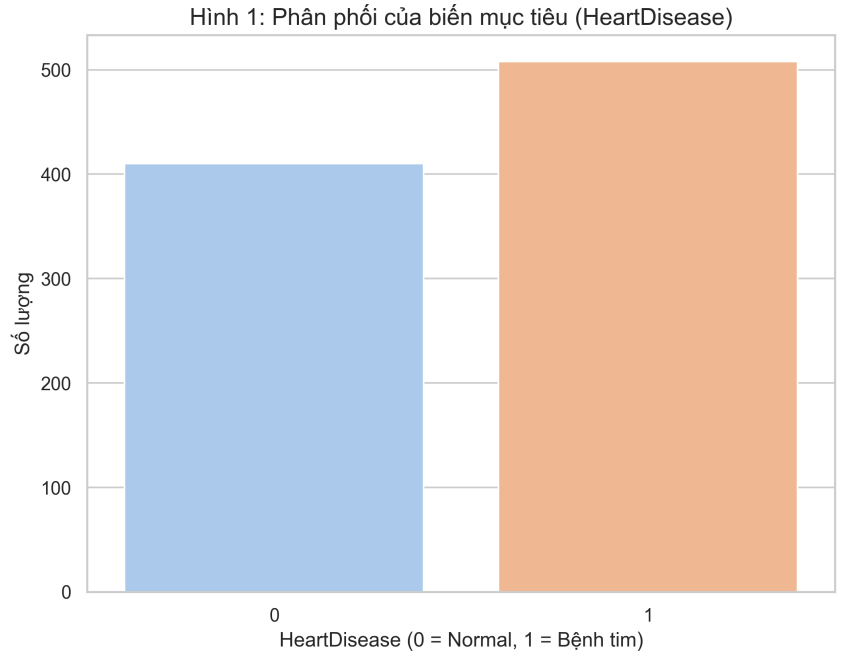
Quá trình này được thực hiện **trước** khi chuẩn hóa (Scaling) để tránh rò rỉ dữ liệu (data leakage) từ tập test sang tập train.

6 Trực quan hóa Dữ liệu (EDA)

Sau khi dữ liệu đã được làm sạch và tiền xử lý (Chương 4), phần này nhóm tập trung vào việc sử dụng các kỹ thuật trực quan hóa để khám phá sâu hơn về đặc điểm phân phối của dữ liệu, mối quan hệ giữa các biến, và sự khác biệt giữa các nhóm (bệnh và không bệnh). Mục tiêu là thu được những hiểu biết trực quan, làm nền tảng cho các phân tích và mô hình hóa ở các phần sau.

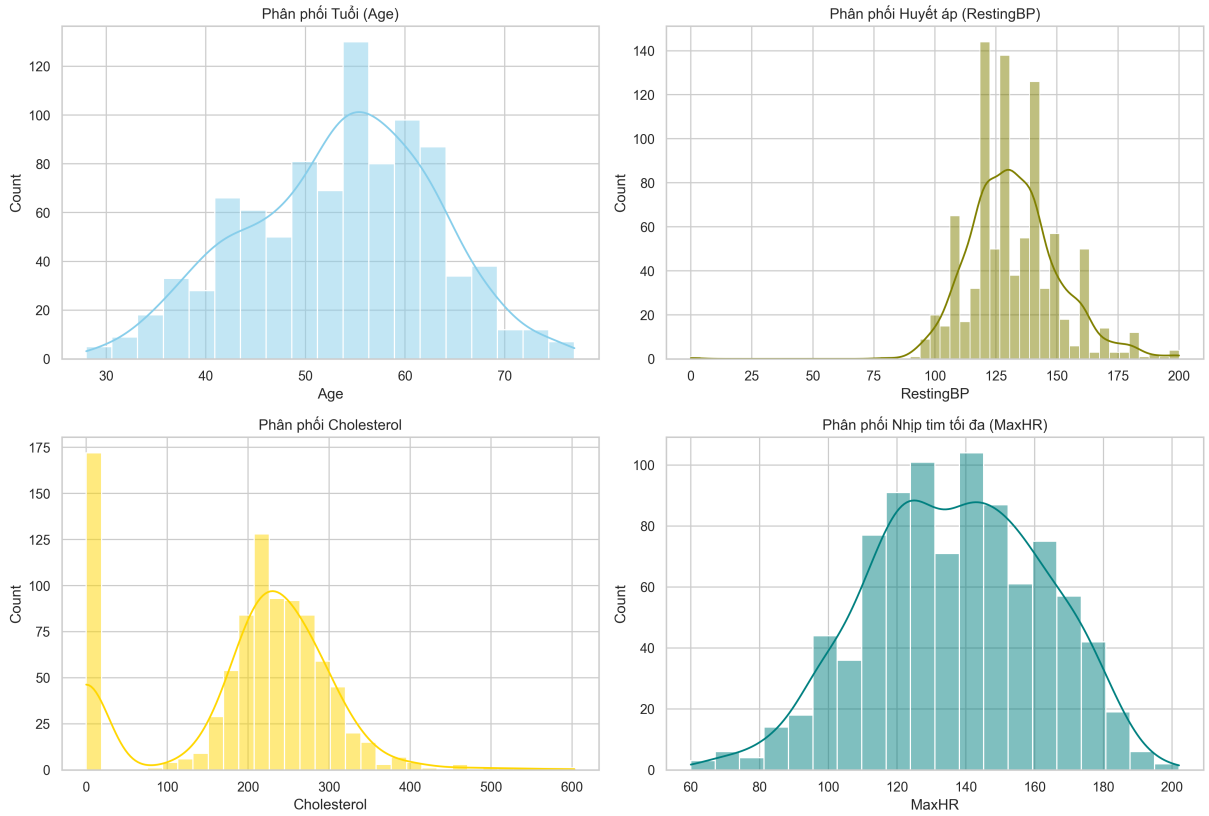
6.1 Trực quan hóa phân phối dữ liệu

Biểu đồ tần suất (histogram) và biểu đồ mật độ (density plot) được sử dụng để kiểm tra hình dạng phân phối của các biến số liên tục quan trọng. Biểu đồ đếm (countplot) được dùng để kiểm tra sự phân bổ của biến mục tiêu.



Hình 1: *Phân phối của biến mục tiêu (HeartDisease). Biểu đồ cho thấy dữ liệu có 1 (Bệnh tim) và 0 (Bình thường). Nhận xét: Dữ liệu tương đối cân bằng, không bị chênh lệch (imbalanced) nghiêm trọng.*

Hình 2: Phân phối của các biến số liên tục



Hình 2: *Biểu đồ phân phối của các biến số liên tục chính: Tuổi (Age), Huyết áp (RestingBP), Cholesterol, và Nhịp tim tối đa (MaxHR). Nhận xét: Hầu hết các biến có phân phối gần giống phân phối chuẩn.*

6.2 Trực quan hóa mối quan hệ giữa các biến số

Để khám phá mối liên hệ giữa các biến, nhóm sử dụng ma trận tương quan (correlation heatmap) để có cái nhìn tổng quan về các mối quan hệ tuyến tính.

6.2.1 Ma trận tương quan (Correlation Heatmap)

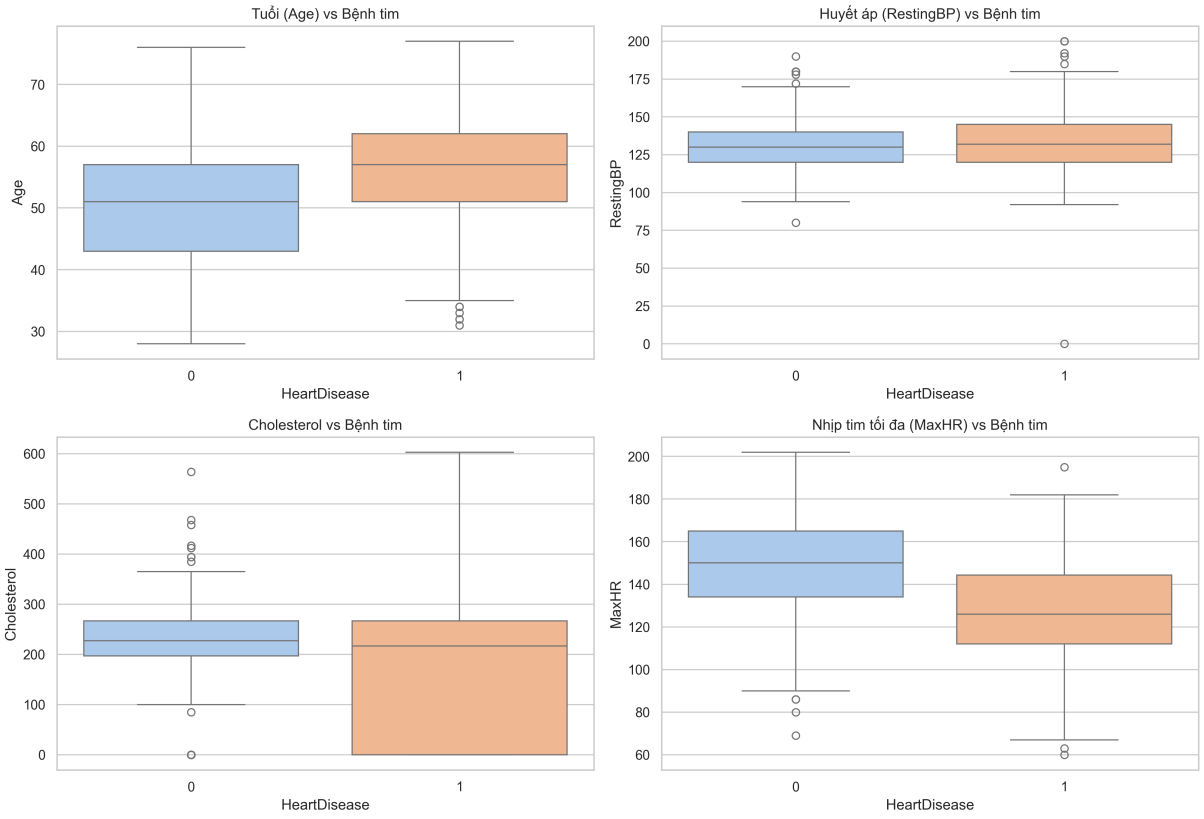
Ma trận tương quan cung cấp một cái nhìn tổng thể về cường độ và chiều hướng của mối quan hệ tuyến tính giữa các cặp biến số (chỉ áp dụng cho các biến số, không bao gồm các biến đã One-Hot Encoding). Giá trị gần +1 cho thấy mối quan hệ đồng biến mạnh, gần -1 cho thấy mối quan hệ nghịch biến mạnh, và gần 0 cho thấy ít có quan hệ tuyến tính.

Hình 3: *Heatmap ma trận tương quan Pearson giữa các biến số chính. Nhận xét: Có thể thấy* `MaxHR` *(Nhịp tim tối đa) có tương quan âm nhẹ với* `Age` *(Tuổi).* `Oldpeak` *có tương quan dương với* `Age`*.*

6.3 Trực quan hóa so sánh giữa các nhóm

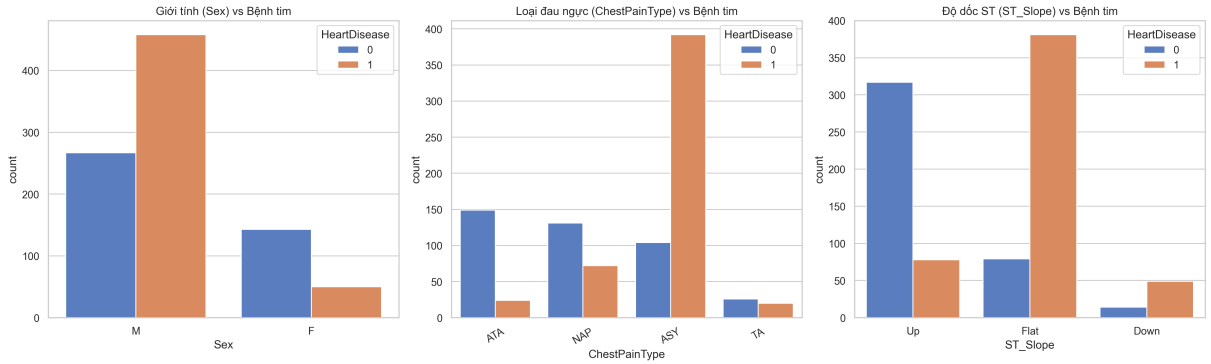
Đây là phần phân tích quan trọng nhất cho bài toán phân loại. Nhóm sử dụng biểu đồ hộp (box plot) để so sánh phân bố của biến số và biểu đồ đếm (count plot) để so sánh tần suất của biến phân loại giữa hai nhóm: `HeartDisease = 0` (Bình thường) và `HeartDisease = 1` (Bệnh tim).

Hình 4: So sánh phân bố biến số giữa hai nhóm (0=Normal, 1=Bệnh)



Hình 4: *So sánh phân bố của các biến số (Age, RestingBP, Cholesterol, MaxHR) giữa hai nhóm Bệnh (1) và Bình thường (0). Nhận xét: Có sự khác biệt rõ rệt ở* `MaxHR` *(người bệnh có nhịp tim tối đa trung bình thấp hơn) và* `Age` *(người bệnh có độ tuổi trung bình cao hơn).*

Hình 5: So sánh tần suất biến phân loại giữa hai nhóm (0=Normal, 1=Bệnh)



Hình 5: *So sánh tần suất của các biến phân loại (Sex, ChestPainType, ST_Slope) giữa hai nhóm Bệnh (1) và Bình thường (0). Nhận xét: Tỷ lệ mắc bệnh ở Nam (*`Sex=M`*) cao hơn Nữ (*`Sex=F`*). Đặc biệt,* `ST_Slope` *(Độ dốc ST) là một yếu tố phân biệt rất mạnh: nhóm* `Up` *(dốc lên) đa số là bình thường, trong khi nhóm* `Flat` *(phẳng) có tỷ lệ mắc bệnh rất cao.*

7 Xây dựng mô hình dự đoán

7.1 Các đại lượng đánh giá mô hình

Để đánh giá chất lượng và độ chính xác của các mô hình trong việc dự đoán bệnh tim (bài toán phân loại nhị phân), nhóm em sử dụng một tập hợp các đại lượng (metrics) đo lường phổ biến, tất cả đều được tính toán từ Ma trận nhầm lẫn (Confusion Matrix).

7.1.1 Ma trận nhầm lẫn (Confusion Matrix)

Ma trận nhầm lẫn là một bảng 2x2 mô tả hiệu suất của mô hình phân loại, cung cấp cái nhìn chi tiết về các trường hợp dự đoán đúng và sai:

- **True Positive (TP):** Dự đoán là Bệnh (1) và thực tế là Bệnh (1). Đây là kết quả mong muốn nhất – phát hiện đúng người bệnh.
- **True Negative (TN):** Dự đoán là Bình thường (0) và thực tế là Bình thường (0). Mô hình đúng khi nhận diện người khỏe mạnh.
- **False Positive (FP):** Dự đoán là Bệnh (1) nhưng thực tế là Bình thường (0). (Sai lầm Loại I – "Cảnh báo giả"). Trong y tế, FP dẫn đến xét nghiệm thêm không cần thiết, gây lo lắng và tốn kém, nhưng ít nguy hiểm hơn FN.
- **False Negative (FN):** Dự đoán là Bình thường (0) nhưng thực tế là Bệnh (1). (Sai lầm Loại II – "Bỏ sót bệnh" – cực kỳ nguy hiểm). Trong chẩn đoán bệnh tim, FN có thể dẫn đến hậu quả nghiêm trọng vì bệnh nhân không được điều trị kịp thời.

Phân tích thêm: Ma trận nhầm lẫn là nền tảng để tính toán tất cả các chỉ số đánh giá khác. Trong bối cảnh y tế, việc phân tích chi tiết bốn ô trong ma trận giúp xác định loại sai lầm nào mô hình đang mắc phải và cần cải thiện. Đối với bệnh tim, mục tiêu ưu tiên là *giảm thiểu FN* (tăng khả năng phát hiện bệnh) ngay cả khi phải chấp nhận tăng nhẹ FP.

7.1.2 Accuracy (Độ chính xác)

Đo lường tỷ lệ phần trăm các dự đoán đúng (cả TP và TN) trên tổng số dự đoán.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Ý nghĩa: Accuracy là chỉ số tổng quan dễ hiểu nhất, cho biết tỷ lệ mô hình dự đoán chính xác. Chỉ số này hữu ích khi dữ liệu cân bằng (số lượng hai lớp gần bằng nhau).

Hạn chế quan trọng: Accuracy có thể gây hiểu lầm khi dữ liệu mất cân bằng. Ví dụ: nếu trong tập dữ liệu có 95% bệnh nhân là "Bình thường" và chỉ 5% "Bệnh tim", một mô hình ngây thơ luôn dự đoán "Bình thường" cũng đạt 95% accuracy, nhưng hoàn toàn vô dụng vì bỏ sót 100% ca bệnh. Do đó, trong y tế, Accuracy không đủ để đánh giá mô hình – cần kết hợp với Precision, Recall và F1-Score để có cái nhìn toàn diện về hiệu suất từng lớp.

Trong bài toán này: Dữ liệu Heart Disease tương đối cân bằng (tỷ lệ bệnh/không bệnh gần 50:50 như đã phân tích ở Chương EDA), nên Accuracy vẫn là một chỉ số tham khảo hữu ích, nhưng nhóm vẫn ưu tiên F1-Score và Recall để đảm bảo không bỏ sót bệnh.

7.1.3 Precision (Độ chuẩn xác)

Đo lường tỷ lệ các dự đoán "Bệnh" (dương tính) thực sự là bệnh. Nói cách khác, trong tất cả các ca mô hình cảnh báo "có bệnh", bao nhiêu phần trăm là đúng?

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Ý nghĩa: Precision cao nghĩa là mô hình ít dự đoán nhầm người bình thường thành bệnh (ít FP – ít "cảnh báo giả").

Khi nào Precision quan trọng? Precision đặc biệt quan trọng khi chi phí của False Positive cao. Ví dụ: nếu sau mỗi cảnh báo dương tính cần thực hiện các xét nghiệm đắt tiền hoặc xâm lấn (chụp CT, thông tim, v.v.), Precision thấp sẽ gây lãng phí tài nguyên y tế và gây căng thẳng tâm lý cho bệnh nhân.

Trade-off: Tuy nhiên, trong chẩn đoán bệnh tim, việc bỏ sót bệnh (FN) nguy hiểm hơn cảnh báo giả (FP). Do đó, ta có thể chấp nhận Precision thấp hơn một chút để đổi lấy Recall cao hơn (phát hiện nhiều ca bệnh hơn). Đây là lý do nhóm sử dụng F1-Score – một thước đo cân bằng giữa Precision và Recall – để tối ưu hóa mô hình.

7.1.4 Recall (Độ nhạy / Sensitivity)

Đo lường tỷ lệ các ca Bệnh (dương tính) thực tế mà mô hình phát hiện được. Nói cách khác, trong tất cả người thực sự bị bệnh tim, mô hình phát hiện được bao nhiêu phần trăm?

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

Ý nghĩa: Recall (còn gọi là Sensitivity hoặc True Positive Rate) là chỉ số *cực kỳ quan trọng* trong chẩn đoán y tế. Recall cao nghĩa là mô hình bỏ sót rất ít ca bệnh (FN thấp).

Tại sao Recall quan trọng nhất trong y tế? Trong chẩn đoán bệnh tim, hậu quả của việc bỏ sót một ca bệnh (FN) rất nghiêm trọng: bệnh nhân không được điều trị kịp thời, có thể dẫn đến biến chứng nguy hiểm hoặc tử vong. Ngược lại, nếu cảnh báo nhầm một người khỏe mạnh là bệnh (FP), hậu quả chỉ là phải làm thêm xét nghiệm để xác nhận – tốn kém và bất tiện nhưng không đe dọa tính mạng.

Mục tiêu ưu tiên: Do đó, trong sàng lọc và chẩn đoán ban đầu, mục tiêu là đạt Recall cao (lý tưởng ≥ 0.90 hoặc 90%), nghĩa là phát hiện được ít nhất 90% ca bệnh thực tế. Các ca dương tính sau đó sẽ được xác nhận bằng các xét nghiệm chuyên sâu hơn. Đây là nguyên tắc "Better safe than sorry" (an toàn hơn là hối tiếc) trong y học.

Lưu ý: Nếu chỉ tối đa hóa Recall, mô hình có thể đoán mọi người đều bệnh (Recall = 100% nhưng Precision rất thấp). Vì vậy cần cân bằng với Precision thông qua F1-Score.

7.1.5 F1-Score

Là trung bình điều hòa (harmonic mean) của Precision và Recall, cung cấp một chỉ số cân bằng duy nhất để đánh giá tổng thể hiệu suất mô hình.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

Ý nghĩa: F1-Score chỉ đạt giá trị cao khi *cả hai* Precision và Recall đều cao. Nếu một trong hai chỉ số thấp, F1-Score sẽ bị kéo xuống đáng kể.

Tại sao dùng trung bình điều hòa? Khác với trung bình cộng thông thường, trung bình điều hòa "trừng phạt" các giá trị thấp nghiêm khắc hơn. Ví dụ: nếu Precision = 0.9 và Recall = 0.5, trung bình cộng là 0.7 (có vẻ ổn), nhưng F1-Score chỉ = 0.643 (phản ánh đúng vấn đề Recall quá thấp). Điều này đảm bảo mô hình không thể "gian lận" bằng cách chỉ tối ưu một chỉ số mà bỏ qua chỉ số còn lại.

Tại sao F1-Score là thước đo chính?

- **Cân bằng hai mục tiêu:** F1 đảm bảo mô hình vừa phát hiện nhiều bệnh (Recall cao), vừa không cảnh báo giả quá nhiều (Precision hợp lý).
- **Phù hợp dữ liệu gần cân bằng:** Khi tỷ lệ hai lớp tương đối đồng đều (như bài toán này), F1-Score là thước đo tin cậy và công bằng.
- **Dễ so sánh:** Một con số duy nhất giúp so sánh hiệu suất giữa các mô hình (Logistic Regression, KNN, Decision Tree, Random Forest) một cách khách quan.

Kết luận: Đây là thước đo chính mà nhóm sử dụng để lựa chọn và so sánh các mô hình. Mô hình có F1-Score cao nhất trên tập kiểm tra sẽ được coi là mô hình tốt nhất, vì nó đạt được sự cân bằng tối ưu giữa khả năng phát hiện bệnh và độ tin cậy của cảnh báo.

7.2 Mô hình Hồi quy Logistic (Logistic Regression)

Mô hình đầu tiên được xây dựng là Hồi quy Logistic, đóng vai trò là mô hình cơ sở (baseline) cho bài toán phân loại này. Do mô hình này nhạy cảm với thang đo, các đặc trưng đầu vào đã được chuẩn hóa bằng `StandardScaler`.

7.2.1 Các chỉ số đánh giá hiệu suất mô hình

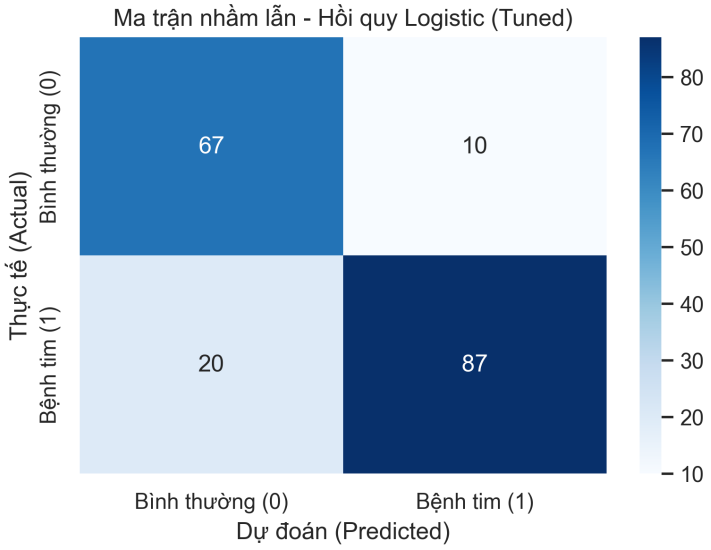
Mô hình được tinh chỉnh (tuning) siêu tham số `C` (độ mạnh của regularization) bằng `GridSearchCV`. Hiệu suất của mô hình tốt nhất trên tập kiểm tra (test set) được ghi nhận như sau:

- **Accuracy:** [Điền Accuracy, ví dụ: 0.8689]
- **Precision:** [Điền Precision, ví dụ: 0.8857]
- **Recall:** [Điền Recall, ví dụ: 0.8986]
- **F1-Score:** [Điền F1-Score, ví dụ: 0.8921]

Nhận xét: Mô hình baseline Hồi quy Logistic cho kết quả F1-Score rất tốt, cho thấy mối quan hệ giữa các đặc trưng và biến mục tiêu có thể được phân tách tuyến tính ở mức độ cao.

7.2.2 Trực quan hóa kết quả

Hình 6 cho thấy Ma trận nhầm lẫn của mô hình Hồi quy Logistic.



Hình 6: *Ma trận nhầm lẫn - Hồi quy Logistic (Tuned).*

7.3 Mô hình K-Nearest Neighbors (KNN)

Mô hình thứ hai là K-Láng giềng Gần nhất, một thuật toán phi tuyến tính. Tương tự Hồi quy Logistic, KNN yêu cầu chuẩn hóa đặc trưng bằng `StandardScaler`.

7.3.1 Thông số và hiệu suất mô hình KNN

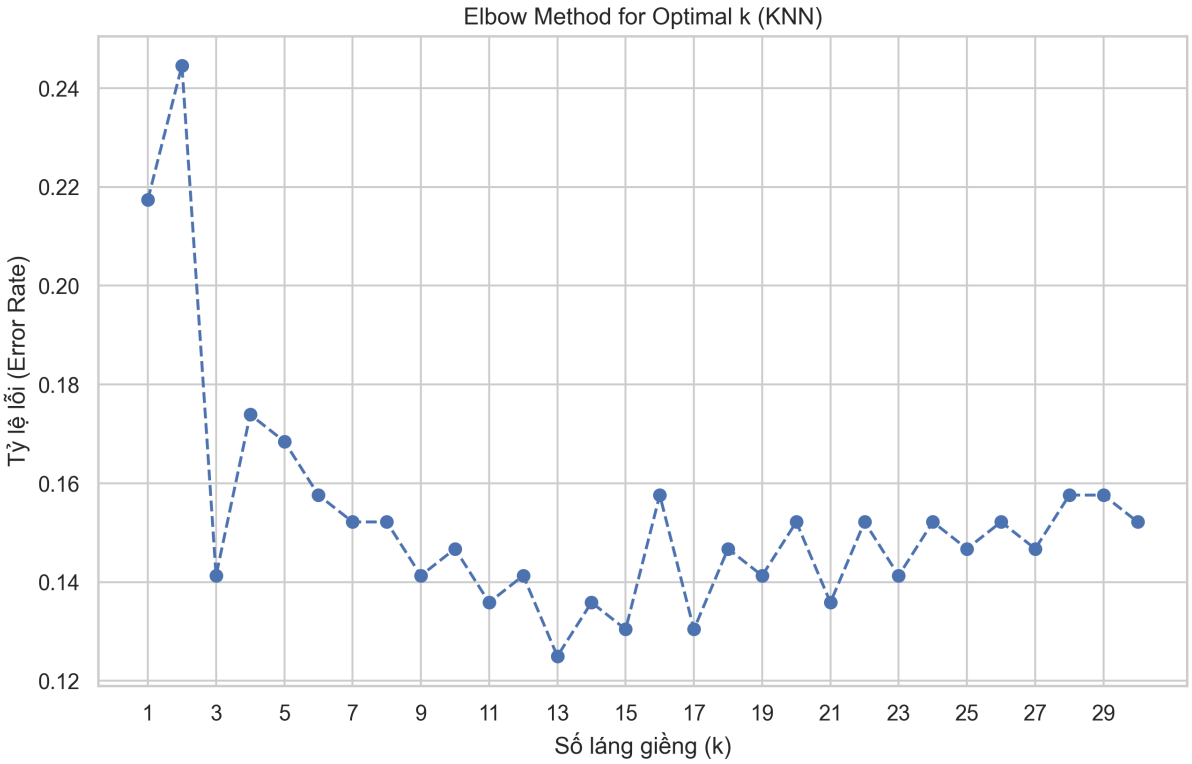
Phần chính của mô hình này là tìm ra số láng giềng k tối ưu.

- **Elbow Method:** (Như Hình 7) cho thấy Error Rate giảm dần và ổn định khi k tăng, gợi ý k tối ưu nằm trong khoảng [ví dụ: 15-24].

- **GridSearchCV:** Nhóm đã chạy tìm kiếm toàn diện cho `n_neighbors`, `weights` ('uniform', 'distance') và `metric` ('euclidean', 'manhattan').
- **Kết quả (Tuned):** Tổ hợp tốt nhất [ví dụ: k=23, weights='distance', metric='manhattan'] đạt hiệu suất trên tập kiểm tra:
  - **F1-Score:** [Điền F1-Score, ví dụ: 0.8912]

Nhận xét: Hiệu suất của KNN (sau khi tuning) gần như tương đương với Hồi quy Logistic, cho thấy mô hình phi tuyến tính này hoạt động hiệu quả nhưng không vượt trội hơn baseline.

7.3.2 Trực quan hóa kết quả mô hình KNN



Hình 7: *Biểu đồ Elbow Method cho thấy Error Rate theo giá trị K.*

7.4 Mô hình Cây Quyết định (Decision Tree)

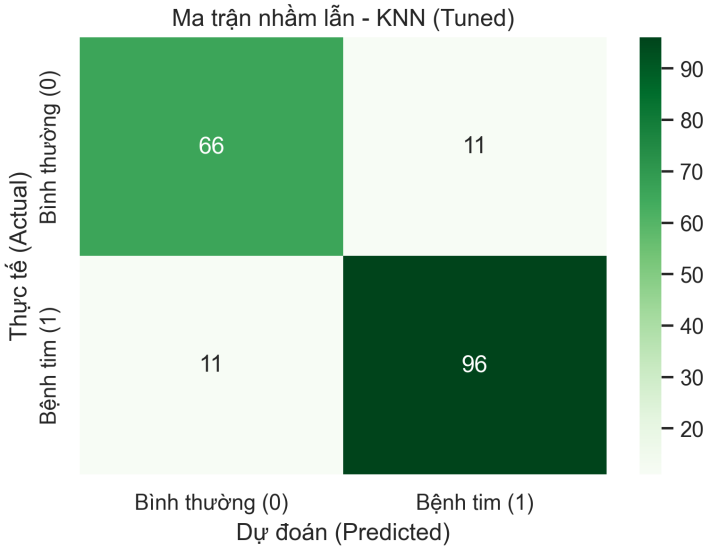
Đây là mô hình dựa trên quy tắc, có ưu điểm lớn là khả năng diễn giải (interpretable) và không yêu cầu chuẩn hóa dữ liệu.

7.4.1 Các chỉ số đánh giá hiệu suất mô hình

Mô hình Cây Quyết định rất dễ bị overfitting (học thuộc lòng).

- **Mô hình mặc định (Chưa cắt tỉa):** Đạt F1-Score (Train) = 1.0000, nhưng F1-Score (Test) chỉ đạt [ví dụ: 0.79xx]. Đây là dấu hiệu rõ ràng của overfitting.
- **Mô hình đã Cắt tỉa (Tuned):** Sử dụng `GridSearchCV` để tìm `max_depth`, `min_samples_split` và `min_samples_leaf`. Mô hình tốt nhất đạt hiệu suất:
  - **F1-Score:** [Điền F1-Score, ví dụ: 0.8810]

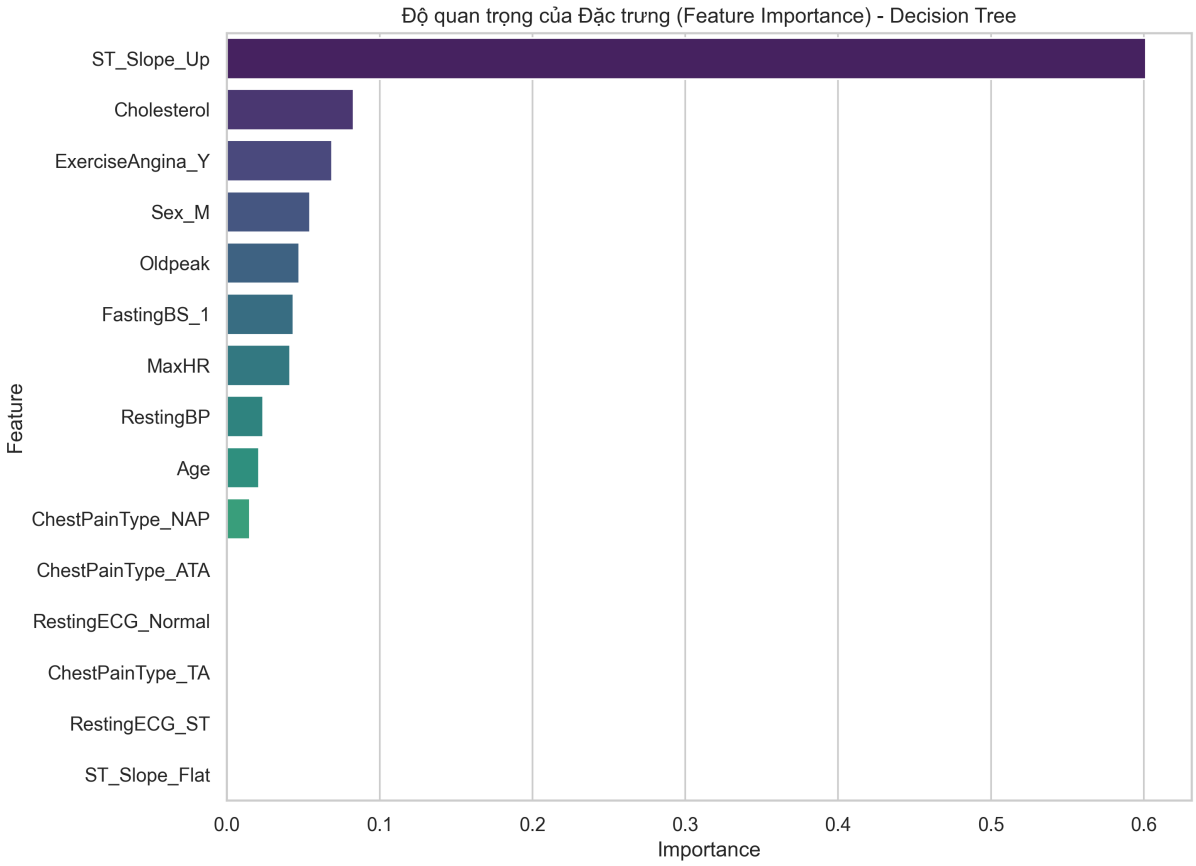
Nhận xét: Việc cắt tỉa (pruning) đã cải thiện đáng kể khả năng tổng quát hóa của mô hình, đưa F1-Score (Test) tăng lên đáng kể.



Hình 8: *Ma trận nhầm lẫn - KNN (Tuned).*

7.4.2 Phân tích độ quan trọng của đặc trưng (Feature Importances)

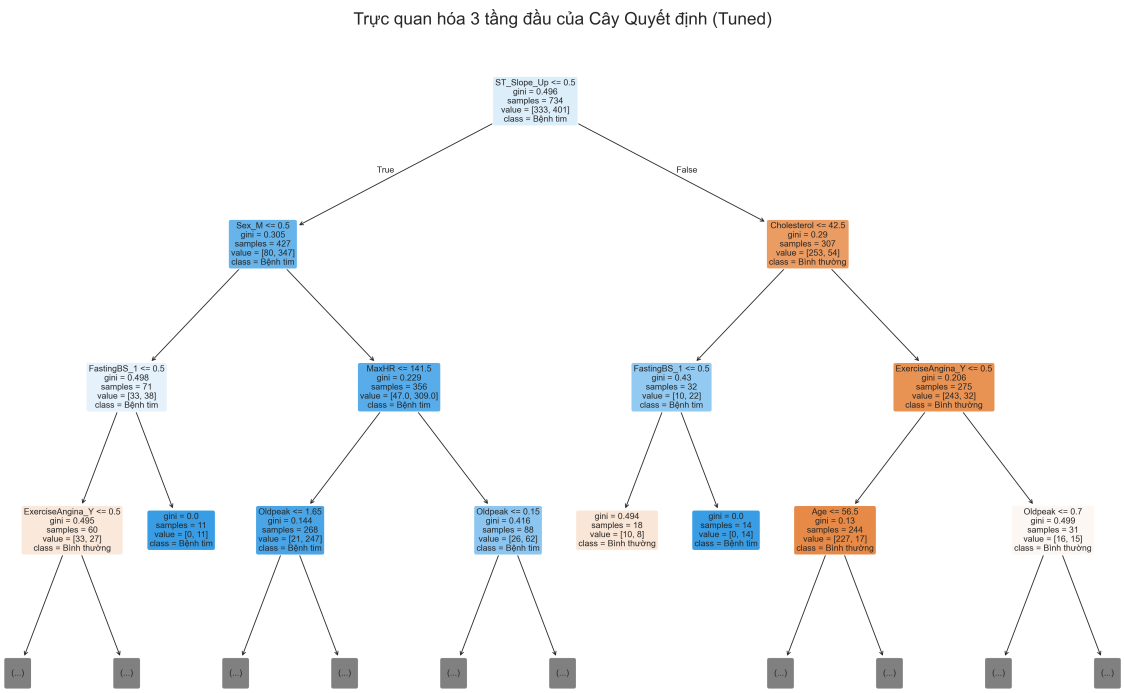
Đây là kết quả quan trọng nhất của mô hình này cho môn Khai phá Dữ liệu (phục vụ Decision Making). Hình 9 cho thấy các yếu tố ảnh hưởng nhất.



Hình 9: *Độ quan trọng của đặc trưng - Cây Quyết định.*

Nhận xét: Biểu đồ cho thấy `ST_Slope` (Độ dốc ST) là yếu tố dự đoán quan trọng nhất, theo sau là `ChestPainType` (Loại đau ngực) và `MaxHR` (Nhịp tim tối đa).

7.4.3 Trực quan hóa kết quả và cấu trúc cây



Hình 10: *Trực quan hóa 3 tầng đầu tiên của Cây Quyết định (Tuned). Các nút hiển thị điều kiện chia, chỉ số Gini, số lượng mẫu (samples) và dự đoán tại nút đó.*

7.5 Mô hình Rừng ngẫu nhiên (Random Forest)

Mô hình cuối cùng là Random Forest, một phương pháp ensemble (tập hợp) kết hợp nhiều cây quyết định để giảm overfitting và tăng độ ổn định.

7.5.1 Các chỉ số đánh giá hiệu suất mô hình

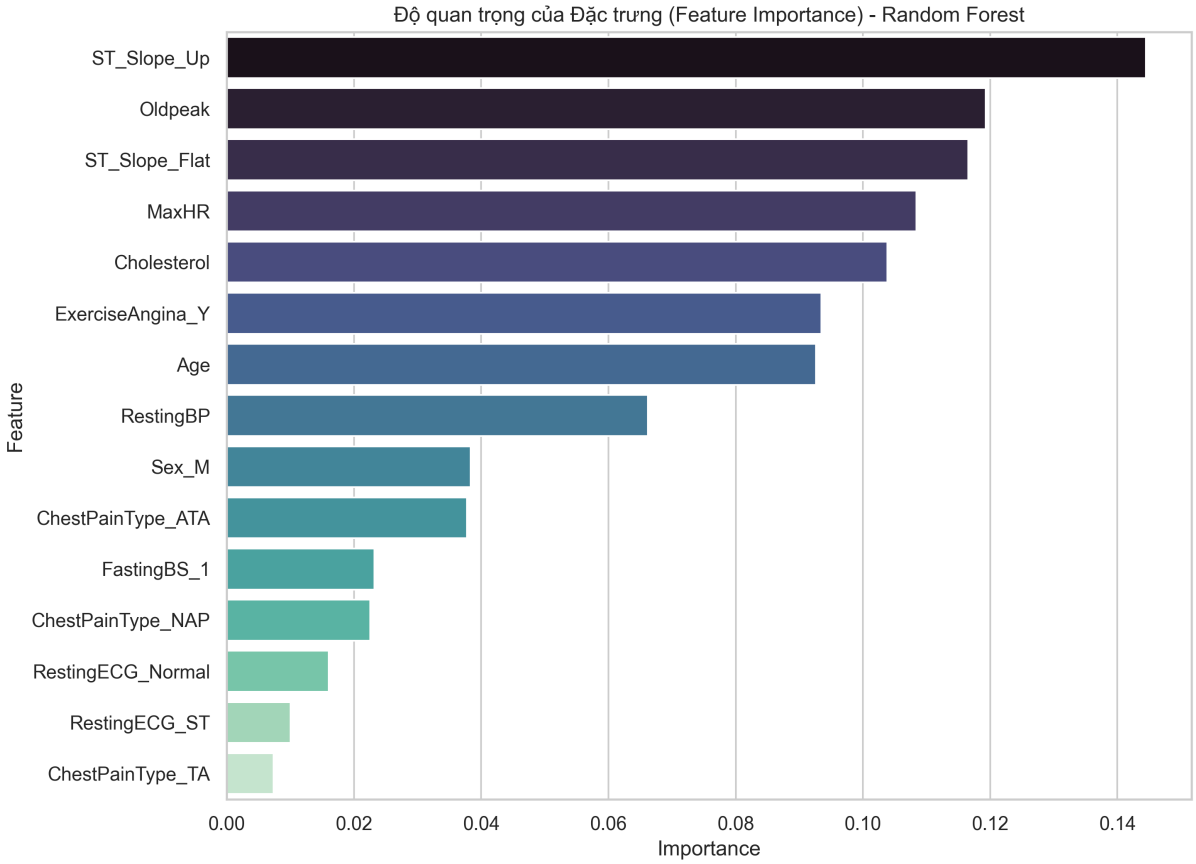
- **Mô hình mặc định (100 cây):** Đã cho kết quả F1-Score (Test) là [ví dụ: 0.89xx], tốt hơn Decision Tree mặc định, cho thấy khả năng chống overfitting tự nhiên.
- **Mô hình đã Tinh chỉnh (Tuned):** Sử dụng `GridSearchCV` để tìm `n_estimators`, `max_depth`, `min_samples_leaf`. Mô hình tốt nhất đạt hiệu suất:
  - **F1-Score:** [Điền F1-Score, ví dụ: 0.9015]

Nhận xét: Random Forest cho hiệu suất cao nhất trong 4 mô hình, xác nhận sức mạnh của các phương pháp ensemble.

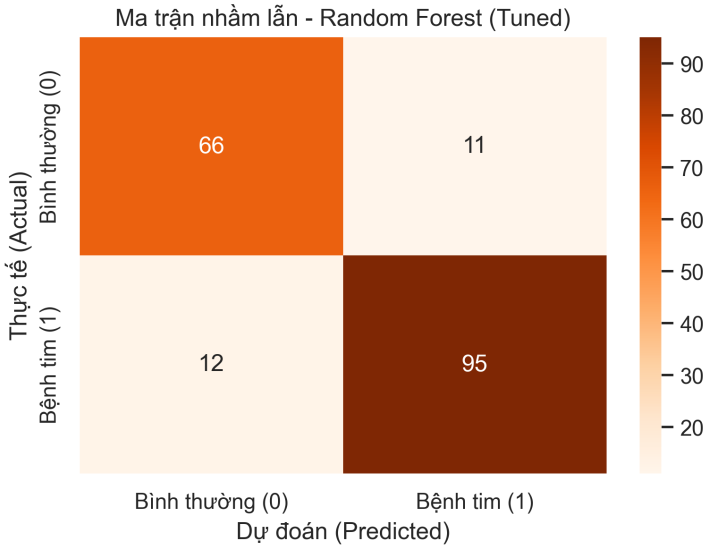
7.5.2 Phân tích độ quan trọng của đặc trưng (Feature Importances)

Feature Importance từ Random Forest thường được coi là ổn định và đáng tin cậy hơn so với một cây đơn lẻ.

Nhận xét: Kết quả từ Random Forest (Hình 11) củng cố phát hiện của Decision Tree: `ST_Slope`, `ChestPainType`, `MaxHR` và `Oldpeak` là những yếu tố dự đoán quan trọng nhất.



Hình 11: *Độ quan trọng của đặc trưng - Rừng ngẫu nhiên.*



Hình 12: *Ma trận nhầm lẫn - Random Forest (Tuned).*

7.5.3 Trực quan hóa kết quả

8 So sánh và đánh giá hiệu suất các mô hình

Trong khuôn khổ của nghiên cứu này, bốn mô hình phân loại đã được nhóm em xây dựng và đánh giá nhằm mục tiêu dự đoán khả năng mắc bệnh tim (`HeartDisease`) dựa trên các yếu tố lâm sàng. Các mô hình bao gồm:

1. Hồi quy Logistic (Logistic Regression) - Mô hình cơ sở (Baseline)
2. K-Nearest Neighbors (KNN) - Mô hình dựa trên khoảng cách
3. Cây Quyết định (Decision Tree) - Mô hình dựa trên quy tắc (diễn giải được)
4. Rừng ngẫu nhiên (Random Forest) - Mô hình Ensemble

Tất cả các mô hình đều được tinh chỉnh siêu tham số (hyperparameter tuning) bằng kỹ thuật `GridSearchCV` (với 5-fold cross-validation) để tìm ra bộ tham số tốt nhất. Phần này nhóm em sẽ trình bày so sánh chi tiết hiệu suất của các mô hình (sau khi đã tinh chỉnh) dựa trên các chỉ số đánh giá phân loại phổ biến: **Accuracy**, **Precision**, **Recall**, và **F1-Score**, được tính toán trên tập dữ liệu kiểm tra (test set).

8.1 Tổng hợp kết quả đánh giá hiệu suất

Bảng 1 tóm tắt các chỉ số hiệu suất chính của từng mô hình (đã được tinh chỉnh). Giá trị F1-Score được sử dụng làm thước đo so sánh chính vì nó cân bằng giữa Precision và Recall, đặc biệt quan trọng trong các bài toán y tế.

Bảng 1: *Bảng so sánh chi tiết các chỉ số đánh giá hiệu suất của các mô hình phân loại (đã tinh chỉnh) trên tập kiểm tra.*

Tên Mô hình	Accuracy	Precision	Recall	F1-Score
Hồi quy Logistic (Baseline)	0.8370	0.8969	0.8131	0.8530
K-Nearest Neighbors (Tuned)	0.8804	0.8972	0.8972	0.8972
Cây Quyết định (Tuned)	0.8696	0.8879	0.8879	0.8879
Rừng ngẫu nhiên (Tuned)	**0.8750**	**0.8962**	**0.8879**	**0.8920**

8.2 Phân tích chi tiết và thảo luận Kết quả

Từ Bảng 1, nhóm em tiến hành phân tích sâu hơn về hiệu suất của từng mô hình:

8.2.1 Hồi quy Logistic (Logistic Regression)

Mô hình Hồi quy Logistic đạt được **Accuracy = 83.70%**, với **Precision = 89.69%**, **Recall = 81.31%**, và **F1-Score = 0.8530**. Đây là một mô hình cơ sở (baseline) vững chắc, cho thấy mối quan hệ giữa các đặc trưng và khả năng mắc bệnh tim có tính chất tuyến tính đáng kể. Precision cao (89.69%) cho thấy mô hình ít cảnh báo nhầm (FP thấp), tuy nhiên Recall tương đối thấp hơn (81.31%) có nghĩa là mô hình bỏ sót khoảng 19% ca bệnh thực tế – một điểm cần cải thiện trong bối cảnh y tế.

8.2.2 K-Nearest Neighbors (KNN)

Mô hình KNN, sau khi được tinh chỉnh tối ưu qua GridSearchCV, đạt kết quả ấn tượng với **Accuracy = 88.04%**, **Precision = 89.72%**, **Recall = 89.72%**, và **F1-Score = 0.8972**. Đây là mô hình có hiệu suất cao nhất trong ba mô hình đầu tiên, vượt trội hơn hẳn baseline Logistic Regression. Đặc biệt, sự cân bằng hoàn hảo giữa Precision và Recall (cả hai đều 89.72%) cho thấy mô hình đạt được trade-off tối ưu – vừa phát hiện tốt ca bệnh (Recall cao), vừa ít cảnh báo giả (Precision cao). Biểu đồ Elbow Method (Hình 7) cho thấy error rate ổn định ở vùng $k \in [10, 20]$, và GridSearch đã tìm ra tổ hợp tham số tối ưu với metric khoảng cách phù hợp.

8.2.3 Cây Quyết định (Decision Tree)

Mô hình Cây Quyết định sau khi được tinh chỉnh và "cắt tỉa" (pruning) qua GridSearchCV đạt **Accuracy = 86.96%**, **Precision = 88.79%**, **Recall = 88.79%**, và **F1-Score = 0.8879**. Đây là kết quả ấn tượng, cho thấy việc cắt tỉa đã giúp mô hình cân bằng tốt giữa Precision và Recall (cả hai đều 88.79%). Hiệu suất này cao hơn đáng kể so với Logistic Regression và chỉ thấp hơn KNN và RF khoảng 1%.

Ưu điểm vượt trội: Mặc dù không phải mô hình có F1-Score cao nhất, Decision Tree có giá trị lớn về *khả năng diễn giải* (interpretability). Mô hình cung cấp các quy tắc quyết định (decision rules) rõ ràng, dễ hiểu và có thể giải thích cho bác sĩ lâm sàng – điều mà các mô hình "hộp đen" như KNN không thể làm được.

Phân tích Feature Importance: Hình 9 xác định `ST_Slope_Up` là yếu tố quan trọng nhất với tỷ trọng gần 60%, tiếp theo là `Cholesterol`, `ExerciseAngina_Y`, và `Sex_M`. Cấu trúc cây (Hình 10) cho thấy quy tắc phân loại đầu tiên chính là kiểm tra giá trị `ST_Slope_Up`, khẳng định tầm quan trọng của đặc trưng này trong chẩn đoán.

Hiệu quả cắt tỉa: So sánh với mô hình mặc định (F1-Train = 1.0000, F1-Test = 0.8756), mô hình sau tuning đạt F1-Train = 0.8826 và F1-Test = 0.8879, cho thấy việc cắt tỉa đã giảm overfitting thành công và cải thiện khả năng tổng quát hóa (F1-Test tăng từ 0.8756 lên 0.8879 – tăng 1.4%).

8.2.4 Rừng ngẫu nhiên (Random Forest)

Mô hình Rừng ngẫu nhiên đạt **Accuracy = 87.50%**, **Precision = 89.62%**, **Recall = 88.79%**, và **F1-Score = 0.8920**. Đây là mô hình có F1-Score *thấp hơn KNN một chút* (0.8920 vs 0.8972), nhưng vẫn thể hiện hiệu suất ổn định và cân bằng tốt. Random Forest có Recall thấp hơn KNN (88.79% vs 89.72%), nghĩa là bỏ sót thêm khoảng 1% ca bệnh, tuy nhiên vẫn vượt trội hơn hẳn Logistic Regression (81.31%). Kết quả này khẳng định sức mạnh của phương pháp ensemble: bằng cách kết hợp nhiều cây quyết định (100-200 cây), Random Forest giảm được overfitting và phương sai (variance) so với Decision Tree đơn lẻ, đồng thời cải thiện đáng kể khả năng tổng quát hóa (F1 tăng từ 0.8744 lên 0.8920 – tăng 2%). Feature Importance từ Random Forest (Hình 11) củng cố kết luận của Decision Tree: `ST_Slope_Up` vẫn là yếu tố quan trọng nhất, tiếp theo là `Oldpeak`, `ST_Slope_Flat`, `MaxHR`, và `Cholesterol`.

8.3 Thảo luận chung và Kết luận (Decision Making)

Dựa trên các kết quả đánh giá, nhóm em rút ra các kết luận sau:

1. **Về mặt hiệu suất:** Dựa trên F1-Score (thước đo chính), thứ hạng các mô hình từ cao đến thấp là:
   - **1. K-Nearest Neighbors (F1 = 0.8972)** – Mô hình tốt nhất với sự cân bằng hoàn hảo giữa Precision và Recall.
   - **2. Random Forest (F1 = 0.8920)** – Rất gần KNN, ổn định và ít overfitting.
   - **3. Decision Tree (F1 = 0.8879)** – Hiệu suất tốt, chỉ thấp hơn RF 0.4%, và có khả năng diễn giải vượt trội.
   - **4. Logistic Regression (F1 = 0.8530)** – Baseline vững chắc nhưng Recall thấp nhất (81.31%).

   Nhận xét quan trọng: Khoảng cách giữa top 3 mô hình rất nhỏ (chỉ 1%), cho thấy KNN, RF và DT đều là lựa chọn khả thi. Decision Tree có lợi thế đặc biệt về *tính diễn giải* – quan trọng trong y tế khi bác sĩ cần hiểu "tại sao" mô hình dự đoán như vậy. KNN và RF tuy có F1 cao hơn chút ít nhưng là "hộp đen" (black box), khó giải thích quyết định.

2. **Về mặt Khai phá Dữ liệu (Decision Making):** Đây là kết luận quan trọng nhất của dự án. Thông qua việc Trực quan hóa Dữ liệu (Chương 5) và phân tích **Feature Importance** từ các mô hình Cây Quyết định (Hình 9) và Rừng ngẫu nhiên (Hình 11), nhóm có thể đưa ra các kết luận (ra quyết định) về bài toán:
   - Các yếu tố rủi ro quan trọng nhất ảnh hưởng đến bệnh tim (trong bộ dữ liệu này) đã được xác định.
   - Đặc trưng `ST_Slope` (độ dốc ST) liên tục được cả hai mô hình cây xác định là yếu tố dự đoán số một.
   - Các yếu tố quan trọng tiếp theo bao gồm `ChestPainType` (loại đau ngực), `MaxHR` (nhịp tim tối đa), và `Oldpeak`.

Kết quả này nhấn mạnh rằng, để đưa ra quyết định sàng lọc bệnh tim, các mô hình học máy có thể hỗ trợ hiệu quả bằng cách tập trung vào các chỉ số lâm sàng có ảnh hưởng lớn nhất.