

TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH

KHOA HỌC CƠ BẢN



TRƯỜNG ĐẠI HỌC MỞ TP. HỒ CHÍ MINH
Cơ hội học tập cho mọi người

TIỂU LUẬN MÔN
MÁY HỌC

ĐỀ TÀI

**DỰ ĐOÁN NGUY CƠ MẮC BỆNH TIM MẠCH DỰA
TRÊN CÁC CHỈ SỐ SỨC KHỎE CÁ NHÂN**

GV hướng dẫn: TH.S HÀ MINH TUẤN

Nhóm sinh viên thực hiện:

Họ và tên

MSSV

Nguyễn Thảo Quyên

2351060031

Nguyễn Phạm Triệu Vỹ

2351060042

Hồ Chí Minh, 2026

Contents

DANH MỤC HÌNH ẢNH	3
DANH MỤC BẢNG	4
DANH MỤC VIẾT TẮT	5
1 Giới thiệu	1
1.1 Giới thiệu đề tài và ứng dụng máy học	1
1.2 Lý do chọn đề tài	1
1.3 Mục tiêu nghiên cứu	1
2 Khảo sát tài liệu	2
2.1 Lược khảo các nghiên cứu liên quan	2
3 Mô hình học máy	3
3.1 Logistic Regression	3
3.1.1 Cơ sở toán học và thuật toán	3
3.1.2 Thuật toán Logistic Regression	4
3.1.3 Siêu tham số quan trọng	5
3.1.4 Điều chỉnh Overfitting và Underfitting	6
3.1.5 Bài toán minh họa	6
3.2 Random Forest	7
3.2.1 Nguyên lý hoạt động	7
3.2.2 Các bài toán Random Forest giải quyết	8
3.2.3 Cơ sở toán học và thuật toán	8
3.2.4 Kết hợp các cây (Ensemble)	9
3.2.5 Out-of-Bag Error và Feature Importance	9
3.2.6 Thuật toán Random Forest	9
3.2.7 Siêu tham số quan trọng	10
3.2.8 Điều chỉnh Overfitting và Underfitting	10
3.2.9 Bài toán minh họa	11
3.3 Mô Hình XGBoost	12
3.3.1 Mục tiêu và khai triển Taylor bậc hai	13
3.3.2 Nghiệm tối ưu của từng lá và công thức Gain	14
3.3.3 Ý nghĩa trực giác	14
3.3.4 Phân loại nhị phân (Log-loss)	15
3.3.5 Chiến lược điều chỉnh tham số cho Overfitting và Underfitting: . . .	17

4	Phương pháp nghiên cứu	19
4.1	Tóm tắt ưu nhược điểm của các thuật toán:	19
4.2	Thuật toán tối ưu hóa	20
4.3	Chỉ số đánh giá hiệu suất	21
5	Chuẩn bị dữ liệu	21
5.1	Nguồn dữ liệu	21
5.1.1	Các biến đầu vào	22
5.1.2	Biến đầu ra	22
5.2	Các bước tiền xử lý dữ liệu	23
5.2.1	Kết hợp dữ liệu	23
5.2.2	Làm sạch và chuyển đổi dữ liệu	23
5.2.3	Thống kê mô tả và ý nghĩa các biến	24
5.2.4	Kiểm tra tính hợp lệ dữ liệu	25
5.2.5	Tách đặc trưng đầu vào và biến đầu ra	25
5.3	EDA – Khám phá dữ liệu	26
5.3.1	Vẽ biểu đồ tròn cho biến đầu ra	26
5.3.2	Kiểm tra các feature không nhất quán	26
5.3.3	Kiểm tra mức độ phân phối của các biến định lượng	28
5.3.4	Vẽ histogram cho các biến định lượng	30
5.3.5	Kiểm tra mức độ phân phối và sự cân bằng của các biến phân loại	34
5.3.6	Tính hệ số tương quan giữa các biến đầu vào và đầu ra	35
6	Triển khai và huấn luyện mô hình	43
6.1	Thư viện học máy sử dụng	43
6.2	Quá trình triển khai và huấn luyện mô hình	43
6.3	LogisticRegression	44
6.4	RandomForestClassifier	44
6.5	XGBClassifier	45
6.6	Đánh giá và trực quan hóa	46
6.7	Kết quả huấn luyện mô hình học máy	46
7	Đánh giá và kết luận	52

Danh mục hình ảnh

List of Figures

1	Tỷ lệ mắc bệnh tim mạch cardio	26
2	Kiểm tra các feature không nhất quán	26
3	Mức độ phân phối của các biến định lượng	28
4	Histogram age	30
5	Histogram Height	30
6	Histogram Weight	31
7	Histogram ApHi	31
8	Histogram ApLo	32
9	Histogram BMI	33
10	Mức độ phân phối và sự cân bằng của các biến phân loại	34
11	Hệ số tương quan giữa các biến đầu vào và đầu ra	35
12	Mối quan hệ giữa BMI và huyết áp tâm thu	36
13	Phân phối tuổi theo tình trạng bệnh nhân	37
14	Tỷ lệ mắc bệnh tim theo nhóm BMI	38
15	Tỷ lệ mắc bệnh tim theo mức độ Cholesterol	39
16	Tỷ lệ mắc bệnh tim theo mức độ Glucose	40
17	Tỷ lệ mắc bệnh tim mạch theo vận động thể chất	41
18	Tương quan chiều cao và cân nặng (Phân nhóm bệnh tim)	42
19	Đường cong ROC của mô hình Logistic Regression	47
20	Ma trận nhầm lẫn của mô hình Logistic Regression tại threshold = 0.40	48
21	Đường cong ROC của mô hình Random Forest	49
22	Ma trận nhầm lẫn của mô hình Random Forest tại threshold = 0.35	50
23	Đường cong ROC của mô hình XGBoost	51
24	Ma trận nhầm lẫn của mô hình XGBoost tại threshold = 0.35	52

Danh mục bảng

List of Tables

1	Tổng hợp hướng điều chỉnh tham số XGBoost	19
2	Kết quả mô hình Logistic Regression trên tập kiểm tra (Threshold = 0.35) . .	47
3	Kết quả đánh giá mô hình Random Forest trên tập kiểm tra (Threshold = 0.35)	49
4	Kết quả đánh giá mô hình XGBoost trên tập kiểm tra (Threshold = 0.35) . .	51
5	So sánh hiệu suất các mô hình trên tập kiểm tra	52

DANH MỤC VIẾT TẮT

Viết tắt	Diễn giải
LR	Logistic Regression
RF	Random Forest
XGBoost	Extreme Gradient Boosting
ROC	Receiver Operating Characteristic
AUC	Area Under the Curve
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

1 Giới thiệu

1.1 Giới thiệu đề tài và ứng dụng máy học

Trong bối cảnh xã hội hiện đại, các bệnh lý tim mạch đang trở thành một trong những nguyên nhân gây tử vong hàng đầu trên toàn thế giới. Sự gia tăng nhanh chóng của các yếu tố nguy cơ như lối sống ít vận động, chế độ ăn uống không lành mạnh, hút thuốc, sử dụng rượu bia và áp lực cuộc sống đã làm cho tỷ lệ mắc bệnh tim mạch ngày càng gia tăng, đặc biệt tại các quốc gia đang phát triển. Do đó, việc phát hiện sớm và dự đoán nguy cơ mắc bệnh tim mạch có ý nghĩa quan trọng trong công tác phòng ngừa, chẩn đoán và điều trị, góp phần nâng cao chất lượng cuộc sống và giảm gánh nặng cho hệ thống y tế.

Trong những năm gần đây, trí tuệ nhân tạo và khoa học dữ liệu ngày càng được ứng dụng rộng rãi trong lĩnh vực y tế, đặc biệt là trong việc phân tích các bộ dữ liệu y sinh phức tạp. So với các phương pháp thống kê truyền thống, các mô hình học máy và học sâu có khả năng học được những mối quan hệ phức tạp giữa nhiều yếu tố nguy cơ, từ đó giúp dự đoán bệnh chính xác hơn.

Xuất phát từ thực tế đó, đề tài tập trung xây dựng các mô hình Logistic Regression, Random Forest và XGBoost có khả năng xử lý hiệu quả dữ liệu và khai thác mối quan hệ giữa các yếu tố nguy cơ. Việc nghiên cứu và so sánh các mô hình này giúp đánh giá mức độ phù hợp của từng phương pháp trong bài toán dự đoán bệnh tim mạch dựa trên dữ liệu sức khỏe và thói quen sinh hoạt của bệnh nhân.

1.2 Lý do chọn đề tài

Việc lựa chọn đề tài này xuất phát từ thực tế rằng bệnh tim mạch ngày càng phổ biến và ảnh hưởng nghiêm trọng đến sức khỏe cũng như chất lượng cuộc sống của con người. Nhiều yếu tố nguy cơ của bệnh tim mạch có thể được phát hiện sớm nếu được phân tích đúng cách từ dữ liệu sức khỏe cá nhân. Vì vậy, việc xây dựng mô hình dự đoán nguy cơ mắc bệnh tim mạch là cần thiết và có ý nghĩa trong công tác phòng ngừa và hỗ trợ chẩn đoán sớm.

1.3 Mục tiêu nghiên cứu

Mục tiêu của đề tài là xây dựng và đánh giá các mô hình học máy nhằm dự đoán nguy cơ mắc bệnh tim mạch dựa trên dữ liệu sức khỏe của bệnh nhân. Cụ thể, đề tài tập trung vào các mục tiêu sau:

- Phân tích và tiền xử lý bộ dữ liệu, bao gồm làm sạch dữ liệu, xử lý các giá trị bất thường và chuẩn hóa các đặc trưng đầu vào.

- Xây dựng các mô hình Logistic Regression, Random Forest và XGBoost để dự đoán khả năng mắc bệnh tim mạch.
- So sánh và đánh giá hiệu quả của các mô hình thông qua các chỉ số đo lường như Accuracy, Precision, Recall, F1-score và ROC–AUC.
- Xác định yếu tố ảnh hưởng lớn nhất đến kết quả chẩn đoán bệnh tim mạch.

2 Khảo sát tài liệu

2.1 Lược khảo các nghiên cứu liên quan

Nghiên cứu sử dụng dữ liệu từ UCI Machine Learning Repository (2019) đã áp dụng các mô hình Logistic Regression (LR), Decision Tree (DT) và Random Forest (RF) để dự đoán nguy cơ mắc bệnh tim mạch dựa trên các chỉ số sinh học và thông tin nhân khẩu học. Với chiến lược chia dữ liệu train/test = 70/30, kết quả cho thấy Logistic Regression đạt độ chính xác khoảng 82–84%, nổi bật ở khả năng diễn giải và tính ổn định. Trong khi đó, Random Forest đạt độ chính xác cao hơn, khoảng 86–88%, nhờ khả năng mô hình hóa các mối quan hệ phi tuyến và giảm hiện tượng quá khớp so với cây quyết định đơn lẻ [1].

Trong nghiên cứu của Alizadehsani et al. (2020), các mô hình Random Forest và XGBoost được đánh giá trên dữ liệu bệnh tim với tỷ lệ chia huấn luyện/kiểm tra = 80/20. Kết quả thực nghiệm cho thấy XGBoost đạt độ chính xác cao nhất, khoảng 89–92%, vượt trội so với Random Forest (khoảng 85–88%), đặc biệt trong trường hợp dữ liệu có nhiều đặc trưng và mối quan hệ phi tuyến phức tạp. Tuy nhiên, nhóm tác giả cũng chỉ ra rằng XGBoost yêu cầu quá trình tinh chỉnh siêu tham số cẩn thận để đạt được hiệu quả tối ưu [2].

Theo Rajkomar et al. (2018), cả các mô hình học sâu và học máy truyền thống đều có thể đạt hiệu suất cao trong dự đoán bệnh lý khi được huấn luyện trên dữ liệu phù hợp. Tuy vậy, trong nhiều ứng dụng y tế thực tế, các mô hình như Logistic Regression và Random Forest vẫn thường được ưu tiên sử dụng nhờ tính ổn định, khả năng tổng quát tốt và dễ giải thích, mặc dù độ chính xác có thể thấp hơn so với một số mô hình phức tạp hơn [3].

Bên cạnh đó, nghiên cứu của Chen & Guestrin (2016) đã chứng minh hiệu quả vượt trội của XGBoost trong nhiều bài toán phân loại y sinh. Trên các bộ dữ liệu lớn với nhiều biến đầu vào, XGBoost thường cải thiện độ chính xác từ 3–6% so với các mô hình cây quyết định đơn lẻ, nhờ cơ chế gradient boosting và khả năng tối ưu hóa hàm mất mát hiệu quả [4].

3 Mô hình học máy

3.1 Logistic Regression

Logistic Regression (LR) là một mô hình học máy tuyến tính được sử dụng phổ biến trong các bài toán phân loại nhị phân, đặc biệt trong các lĩnh vực như y sinh, tài chính và khoa học dữ liệu ứng dụng.

Về mặt lý thuyết, Logistic Regression mô hình hóa mối quan hệ giữa biến đầu vào X và xác suất xảy ra của biến mục tiêu Y thông qua hàm logistic (sigmoid).

Logistic Regression được sử dụng rộng rãi trong:

- **Y sinh:** dự đoán nguy cơ mắc bệnh (Hosmer et al., 2013).
- **Tài chính:** đánh giá rủi ro tín dụng.
- **Khoa học dữ liệu:** phân loại nhị phân tổng quát.

Các nghiên cứu cho thấy Logistic Regression:

- Hoạt động hiệu quả khi mối quan hệ giữa các đặc trưng và log-odds gần tuyến tính.
- Có khả năng diễn giải cao, đặc biệt quan trọng trong các bài toán y tế và xã hội (James et al., 2013; Bishop, 2006).

Vì vậy, Logistic Regression thường được sử dụng như một *baseline model* trong các bài toán phân loại.

3.1.1 Cơ sở toán học và thuật toán

Logistic Regression mô hình hóa xác suất:

$$P(Y = 1 | X) = \sigma(z), \quad z = \beta_0 + \sum_{j=1}^p \beta_j x_j$$

Trong đó hàm sigmoid được định nghĩa:

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

Hàm sigmoid đảm bảo đầu ra của mô hình luôn nằm trong khoảng $[0, 1]$.

Quá trình huấn luyện Logistic Regression được thực hiện bằng cách tối ưu hàm mất mát log-loss (cross-entropy).

Hàm mất mát:

$$\mathcal{L}(\beta) = -\frac{1}{n} \sum_{i=1}^n [y^{(i)} \log(p^{(i)}) + (1 - y^{(i)}) \log(1 - p^{(i)})]$$

3.1.2 Thuật toán Logistic Regression

Input:

- Dữ liệu huấn luyện: $(x^{(i)}, y^{(i)})$, với $y^{(i)} \in \{0, 1\}$
- Learning rate: η
- Số vòng lặp: T
- (Tùy chọn) Regularization: L1 hoặc L2

Output:

- Tham số mô hình: β

Bước 1: Chuẩn hoá dữ liệu

Bước 2: Khởi tạo tham số

Khởi tạo tập trọng số $\beta = (\beta_0, \beta_1, \dots, \beta_p)$ bằng 0 hoặc các giá trị ngẫu nhiên nhỏ.

Bước 3: Huấn luyện mô hình

Với mỗi mẫu $x^{(i)}$, giá trị tuyến tính được tính:

$$z^{(i)} = \beta_0 + \sum_{j=1}^p \beta_j x_j^{(i)}$$

Sau đó đưa qua hàm sigmoid để thu được xác suất dự đoán:

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

Bước 4: Regularization

Để hạn chế hiện tượng overfitting, có thể bổ sung regularization vào hàm mất mát:

- **L2 (Ridge)**: phạt các trọng số có giá trị lớn, giúp mô hình ổn định và phân bố trọng số đều hơn.
- **L1 (Lasso)**: khuyến khích nhiều trọng số tiến về 0, giúp chọn lọc đặc trưng quan trọng.

Bước 5: Tính gradient

Gradient của hàm mất mát theo từng tham số:

$$\nabla_{\beta_j} \mathcal{L} = \frac{1}{n} \sum_{i=1}^n (\hat{p}^{(i)} - y^{(i)}) x_j^{(i)}$$

Bước 6: Cập nhật tham số

Các trọng số được cập nhật bằng Gradient Descent:

$$\beta_j := \beta_j - \eta \nabla_{\beta_j} \mathcal{L}$$

3.1.3 Siêu tham số quan trọng

Trong thư viện sklearn, Logistic Regression có các siêu tham số quan trọng:

- **Penalty:** l1, l2, elasticnet

L1 Regularization:

$$\mathcal{L}_{L1}(\beta) = \mathcal{L}(\beta) + \lambda \sum_{j=1}^p |\beta_j|$$

Gradient của chuẩn L1

$$\frac{\partial}{\partial \beta_j} |\beta_j| = \begin{cases} 1, & \beta_j > 0 \\ -1, & \beta_j < 0 \\ [-1, 1], & \beta_j = 0 \end{cases}$$

L2 Regularization:

$$\mathcal{L}_{L2}(\beta) = \mathcal{L}(\beta) + \lambda \sum_{j=1}^p \beta_j^2$$

Gradient của chuẩn L2

$$\frac{\partial}{\partial \beta_j} (\lambda \beta_j^2) = 2\lambda \beta_j$$

Elastic Net:

$$\mathcal{L}_{\text{ElasticNet}}(\beta) = \mathcal{L}(\beta) + \lambda \left(\alpha \sum_{j=1}^p |\beta_j| + (1 - \alpha) \sum_{j=1}^p \beta_j^2 \right)$$

Gradient của chuẩn ElasticNet

$$\frac{\partial}{\partial \beta_j} [\lambda (\alpha |\beta_j| + (1 - \alpha) \beta_j^2)] = \lambda (\alpha \text{sign}(\beta_j) + 2(1 - \alpha) \beta_j)$$

Tham số C : nghịch đảo của λ , kiểm soát mức độ regularization.

- C nhỏ \Rightarrow regularization mạnh \Rightarrow giảm overfitting.
- C lớn \Rightarrow ít regularization \Rightarrow dễ overfitting.

Solver: lbfgs, liblinear, saga (liên quan penalty và tốc độ hội tụ).

3.1.4 Điều chỉnh Overfitting và Underfitting

Overfitting:

- Giảm $C \rightarrow$ tăng regularization.
- Tăng L2 (Ridge) \rightarrow làm nhỏ đồng đều các trọng số
- Sử dụng Elastic Net \rightarrow ổn định mô hình + loại bỏ đặc trưng kém
- Tăng L1 để loại bỏ đặc trưng \rightarrow ép nhiều trọng số về 0 (feature selection)
- Giảm số lượng đặc trưng.
- Loại bỏ đặc trưng nhiều.
- Tăng kích thước tập huấn luyện.

Underfitting:

- Tăng $C \rightarrow$ giảm regularization.
- Giảm L1, L2.
- Thêm đặc trưng tương tác.
- Thêm đặc trưng đa thức.
- Mở rộng feature engineering dựa trên domain knowledge.

3.1.5 Bài toán minh họa

Xét hai đặc trưng:

$$x_1 = \frac{ap_hi}{100}, \quad x_2 = cholesterol$$

Thêm intercept $x_0 = 1$.

i	x_1	x_2	y
1	1.10	1	0
2	1.40	3	1
3	1.30	3	1
4	1.50	1	1
5	1.00	1	0
6	1.20	2	0
7	1.30	3	0
8	1.30	3	1
9	1.10	1	0
10	1.10	1	0

Khởi tạo:

$$\beta_0 = \beta_1 = \beta_2 = 0 \Rightarrow \hat{p} = 0.5$$

Gradient (không regularization):

$$\nabla \mathcal{L} = \frac{1}{n} \sum (p - y)$$

Tính được:

$$\nabla_{\beta_0} = 0.10, \quad \nabla_{\beta_1} = 0.065, \quad \nabla_{\beta_2} = -0.05$$

Với $\eta = 1$, cập nhật:

$$\beta_0 = 0.1, \quad \beta_1 = 0.065, \quad \beta_2 = -0.05$$

Dự đoán mẫu đầu tiên:

$$z = -0.10 - 0.065 \times 1.10 + 0.05 \times 1 = -0.1215$$

$$\hat{p} = \sigma(z) \approx 0.47 \Rightarrow \hat{y} = 0$$

3.2 Random Forest

Random Forest (RF) là một mô hình học máy thuộc nhóm *ensemble learning*, được xây dựng bằng cách kết hợp nhiều cây quyết định (Decision Tree) nhằm cải thiện độ chính xác và khả năng tổng quát hóa của mô hình. Thuật toán này lần đầu tiên được đề xuất bởi Breiman (2001) và nhanh chóng trở thành một trong những phương pháp phổ biến trong các bài toán phân loại và hồi quy.

3.2.1 Nguyên lý hoạt động

Nguyên lý hoạt động của Random Forest dựa trên hai cơ chế chính (Breiman, 2001):

- **Bagging (Bootstrap Aggregating):** Nhiều tập dữ liệu con được tạo ra bằng cách lấy mẫu ngẫu nhiên có hoàn lại từ tập dữ liệu gốc. Mỗi cây quyết định được huấn luyện độc lập trên một tập con, giúp giảm phương sai của mô hình tổng thể.
- **Chọn đặc trưng ngẫu nhiên tại mỗi nút chia:** Thay vì xem xét toàn bộ tập đặc trưng, mỗi cây chỉ sử dụng một tập con ngẫu nhiên các biến đầu vào khi xây dựng các nút chia. Cơ chế này giúp giảm sự phụ thuộc giữa các cây và nâng cao khả năng tổng quát hóa.

Đối với bài toán phân loại, dự đoán cuối cùng của Random Forest được xác định thông qua cơ chế bỏ phiếu đa số (*majority voting*) từ các cây thành viên, giúp mô hình ổn định hơn so với cây quyết định đơn lẻ, đặc biệt khi dữ liệu có nhiễu (Hastie, Tibshirani & Friedman, 2009).

3.2.2 Các bài toán Random Forest giải quyết

Random Forest được áp dụng hiệu quả cho nhiều loại bài toán:

- **Bài toán phân loại (Classification):** Ví dụ: dự đoán có/không mắc bệnh, phát hiện gian lận.
- **Bài toán hồi quy (Regression):** Ví dụ: dự đoán giá nhà.
- **Bài toán chọn đặc trưng (Feature Selection):** Ví dụ: xác định biến quan trọng trong y sinh, phân tích dữ liệu nhiều chiều.

Random Forest phù hợp vì:

- Xử lý tốt các mối quan hệ phi tuyến.
- Ít bị overfitting hơn so với cây quyết định đơn.
- Hoạt động ổn định với dữ liệu nhiễu.

3.2.3 Cơ sở toán học và thuật toán

Bootstrap sampling:

Tập dữ liệu gốc:

$$\mathcal{D} = \{(x^{(i)}, y^{(i)})\}_{i=1}^n$$

Mỗi cây được huấn luyện trên một tập con:

$$\mathcal{D}^{(t)} = \text{Bootstrap}(\mathcal{D})$$

Chọn đặc trưng ngẫu nhiên tại mỗi nút:

- Tổng số đặc trưng: p
- Số đặc trưng xét tại mỗi nút: m

$$m = \begin{cases} \sqrt{p}, & \text{bài toán phân loại} \\ \frac{p}{3}, & \text{bài toán hồi quy} \end{cases}$$

Tiêu chí chia nút:

- **Phân loại – Gini:**

$$\text{Gini}(S) = 1 - \sum_{k=1}^K p_k^2$$

$$\Delta \text{Gini} = \text{Gini}(S) - \sum_{v \in \text{children}} \frac{|S_v|}{|S|} \text{Gini}(S_v)$$

- **Phân loại – Entropy:**

$$\text{Entropy}(S) = - \sum_{k=1}^K p_k \log_2 p_k \quad \text{Information Gain} = \text{Entropy}(S) - \sum_v \frac{|S_v|}{|S|} \text{Entropy}(S_v)$$

- **Hồi quy – MSE:**

$$\text{MSE}(S) = \frac{1}{|S|} \sum_{i \in S} (y_i - \bar{y})^2$$

3.2.4 Kết hợp các cây (Ensemble)

Dự đoán của cây thứ t :

$$\hat{y}^{(t)}(x)$$

Phân loại (majority voting):

$$\hat{Y}(x) = \text{mode}\{\hat{y}^{(1)}(x), \dots, \hat{y}^{(T)}(x)\}$$

Hồi quy (trung bình):

$$\hat{Y}(x) = \frac{1}{T} \sum_{t=1}^T \hat{y}^{(t)}(x)$$

3.2.5 Out-of-Bag Error và Feature Importance

- **Out-of-Bag (OOB) Error:** sử dụng các mẫu không được chọn trong bootstrap để ước lượng lỗi tổng quát.
- **Feature Importance:** đo mức độ đóng góp của từng đặc trưng dựa trên mức giảm Gini trung bình.

3.2.6 Thuật toán Random Forest

Input:

- Dữ liệu huấn luyện: $(x^{(i)}, y^{(i)})$
- Với phân loại: y là nhãn lớp
- Với hồi quy: y là giá trị liên tục
- Số cây: T
- Số đặc trưng chọn tại mỗi nút: `max_features`
- (Tuỳ chọn) các tham số cây: `max_depth`, `min_samples_split`, `min_samples_leaf`

Output:

- Mô hình Random Forest (tập hợp các cây quyết định)
- Kết quả dự đoán cho dữ liệu mới

3.2.7 Siêu tham số quan trọng

- `n_estimators`: số cây trong rừng
- `max_depth`: độ sâu tối đa của cây
- `min_samples_split`: số mẫu tối thiểu để tách nút
- `min_samples_leaf`: số mẫu tối thiểu tại lá
- `max_features`: số đặc trưng xét tại mỗi split
- `class_weight`: xử lý dữ liệu mất cân bằng (tuỳ chọn)

3.2.8 Điều chỉnh Overfitting và Underfitting

Overfitting:

- Giảm `max_depth`
- Tăng `min_samples_leaf`
- Giảm `max_features`
- Tăng `n_estimators` (giúp ổn định mô hình)

Underfitting:

- Tăng max_depth
- Giảm min_samples_leaf
- Tăng max_features
- Tăng n_estimators (chỉ khi mô hình chưa ổn định)

3.2.9 Bài toán minh họa

Bước 1: Tính Gini trước khi chia

Trong 10 mẫu, số mẫu $y = 1$ là 4:

$$p = \frac{4}{10} = 0.4$$

$$\text{Gini} = 2p(1 - p) = 2 \times 0.4 \times 0.6 = 0.48$$

Bước 2: Thử split theo ap_hi

Chọn ngưỡng chia theo thuộc tính $ap_hi = 125$:

- **Nhánh trái** ($ap_hi \leq 125$) gồm các mẫu $\{1, 5, 6, 9, 10\} \Rightarrow y = \{0, 0, 0, 0, 0\}$

$$p = \frac{0}{5} = 0 \quad \Rightarrow \quad \text{Gini}_{\text{trái}} = 1 - (1^2 + 0^2) = 0$$

- **Nhánh phải** ($ap_hi > 125$) gồm các mẫu $\{2, 3, 4, 7, 8\} \Rightarrow y = \{1, 1, 1, 0, 1\}$

$$p = \frac{4}{5} = 0.8$$

$$\text{Gini}_{\text{phải}} = 1 - (0.8^2 + 0.2^2) = 2 \times 0.8 \times 0.2 = 0.32$$

$$\text{Gini}_{\text{sau split}} = \frac{5}{10} \times 0 + \frac{5}{10} \times 0.32 = 0.16$$

Giảm mạnh từ 0.48 xuống 0.16 \Rightarrow split rất tốt.

Cây quyết định đơn giản (stump):

- Nếu $ap_hi \leq 125 \Rightarrow \hat{y} = 0$
- Nếu $ap_hi > 125 \Rightarrow \hat{y} = 1$

Bước 3: Random Forest = nhiều cây + bỏ phiếu

Ví dụ với 3 cây:

- Tree 1: split theo *ap_hi*
- Tree 2: split theo *cholesterol* với ngưỡng 2.5
- Tree 3: split theo *ap_hi* (khác ngưỡng do bootstrap)

Dự đoán mẫu số 3 (*ap_hi* = 130, *chol* = 3):

$$\text{Tree 1} = 1, \quad \text{Tree 2} = 1, \quad \text{Tree 3} = 1$$

$$\Rightarrow \text{majority vote} = 1$$

3.3 Mô Hình XGBoost

XGBoost (Extreme Gradient Boosting) là một hệ thống học máy tiên tiến được đề xuất bởi Chen và Guestrin [14], được thiết kế nhằm mở rộng và tối ưu hóa phương pháp Gradient Tree Boosting cả về mặt thuật toán lẫn hệ thống. XGBoost không chỉ được sử dụng như một mô hình dự đoán độc lập mà còn được tích hợp trong nhiều hệ thống sản xuất thực tế, chẳng hạn như dự đoán tỷ lệ nhấp chuột quảng cáo (Click-Through Rate – CTR), xếp hạng tìm kiếm và phát hiện gian lận.

Hiệu quả của XGBoost đã được chứng minh rõ ràng thông qua các cuộc thi học máy quy mô lớn. Theo báo cáo của Chen và Guestrin [14], trong số các lời giải chiến thắng trên nền tảng Kaggle trong năm 2015, phần lớn đều sử dụng XGBoost như mô hình học chính. Điều này cho thấy XGBoost đạt được kết quả tiên tiến (state-of-the-art) trên nhiều bài toán thực tế khác nhau, bao gồm dự đoán doanh số bán hàng, phân loại văn bản web, phân loại sự kiện vật lý năng lượng cao, phát hiện phần mềm độc hại và dự đoán hành vi khách hàng.

Nguyên nhân cốt lõi dẫn đến thành công của XGBoost nằm ở khả năng mở rộng vượt trội của hệ thống. Cụ thể, XGBoost được thiết kế với nhiều cải tiến quan trọng, bao gồm: (i) một thuật toán học cây mới có khả năng xử lý dữ liệu thưa (sparsity-aware) hiệu quả; (ii) phương pháp *weighted quantile sketch* có cơ sở lý thuyết vững chắc, cho phép xây dựng cây xấp xỉ trong trường hợp dữ liệu lớn và có trọng số; (iii) khả năng huấn luyện song song và phân tán giúp tăng tốc đáng kể quá trình học; và (iv) hỗ trợ *out-of-core computation*, cho phép xử lý dữ liệu có kích thước vượt quá bộ nhớ chính.

XGBoost (Extreme Gradient Boosting) mở rộng từ Gradient Boosting bằng việc dùng khai triển Taylor bậc hai của hàm mất mát.

3.3.1 Mục tiêu và khai triển Taylor bậc hai

Tại vòng lặp thứ t , XGBoost bổ sung một cây quyết định mới f_t vào mô hình hiện tại:

$$F_t(x) = F_{t-1}(x) + f_t(x)$$

Giả sử cây f_t có T_t lá, mỗi lá j tương ứng với một vùng R_{tj} và giá trị dự đoán w_{tj} . Thành phần chính quy hóa của cây được định nghĩa là:

$$\Omega(f_t) = \gamma T_t + \frac{1}{2} \lambda \sum_{j=1}^{T_t} w_{tj}^2$$

Hàm mục tiêu tại vòng lặp t có dạng:

$$\mathcal{L}^{(t)} = \sum_{i=1}^n l(y_i, F_{t-1}(x_i) + f_t(x_i)) + \Omega(f_t)$$

Do hàm mất mát $l(\cdot, \cdot)$ là hàm khả vi, XGBoost sử dụng khai triển Taylor bậc hai quanh $F_{t-1}(x_i)$:

$$l(y_i, F_{t-1}(x_i) + f_t(x_i)) \approx l(y_i, F_{t-1}(x_i)) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)$$

Trong đó:

$$g_i = \left. \frac{\partial l(y_i, \hat{y}_i)}{\partial \hat{y}_i} \right|_{\hat{y}_i = F_{t-1}(x_i)}, \quad h_i = \left. \frac{\partial^2 l(y_i, \hat{y}_i)}{\partial \hat{y}_i^2} \right|_{\hat{y}_i = F_{t-1}(x_i)}$$

Do hạng tử $l(y_i, F_{t-1}(x_i))$ không phụ thuộc vào f_t , ta có thể loại bỏ khi tối ưu. Khi đó, hàm mục tiêu xấp xỉ là:

$$\tilde{\mathcal{L}}^{(t)} = \sum_{j=1}^{T_t} \left[G_{tj} w_{tj} + \frac{1}{2} (H_{tj} + \lambda) w_{tj}^2 \right] + \gamma T_t$$

với:

$$G_{tj} = \sum_{i \in R_{tj}} g_i, \quad H_{tj} = \sum_{i \in R_{tj}} h_i$$

3.3.2 Nghiệm tối ưu của từng lá và công thức Gain

Xét một lá j với vùng R_{tj} . Hàm mất mát xấp xỉ của lá có dạng:

$$\tilde{\mathcal{L}}(w_{tj}) = G_{tj}w_{tj} + \frac{1}{2}(H_{tj} + \lambda)w_{tj}^2$$

Lấy đạo hàm theo w_{tj} và cho bằng 0:

$$\frac{\partial \tilde{\mathcal{L}}}{\partial w_{tj}} = G_{tj} + (H_{tj} + \lambda)w_{tj} = 0$$

Suy ra nghiệm tối ưu của trọng số lá:

$$w_{tj}^* = -\frac{G_{tj}}{H_{tj} + \lambda}$$

Thay nghiệm này vào hàm mục tiêu, ta thu được giá trị mất mát tối ưu của cây:

$$\tilde{\mathcal{L}}^{(t)*} = -\frac{1}{2} \sum_{j=1}^{T_t} \frac{G_{tj}^2}{H_{tj} + \lambda} + \gamma T_t$$

Khi chia một vùng R thành hai vùng con R_L và R_R , độ lợi (Gain) của phép tách được xác định là:

$$\text{Gain} = \frac{1}{2} \left(\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{G^2}{H + \lambda} \right) - \gamma$$

XGBoost chọn phép tách có giá trị Gain lớn nhất. Similarity Score và ý nghĩa trực giác

Từ công thức Gain, ta định nghĩa *Similarity Score* của một vùng R như sau:

$$\text{Score}(R) = \frac{G(R)^2}{H(R) + \lambda}$$

Similarity Score phản ánh mức độ giảm hàm mất mát mà một lá có thể mang lại.

3.3.3 Ý nghĩa trực giác

- Nếu các gradient g_i trong lá có cùng dấu, tổng $G(R)$ có độ lớn lớn, dẫn đến Score cao. Điều này cho thấy lá đồng nhất và cập nhật hiệu quả.
- Nếu các gradient trái dấu và triệt tiêu lẫn nhau, $G(R)$ nhỏ, Score thấp, lá không đồng nhất.

Khi chia một vùng cha P thành hai vùng con L và R , công thức Gain có thể viết lại (theo Score):

$$\text{Gain} = \frac{1}{2}(\text{Score}(L) + \text{Score}(R) - \text{Score}(P)) - \gamma$$

Do đó, một phép tách tốt là phép tách làm tăng tổng Similarity Score. **Gradient, Hessian và Residual trong XGBoost**

Trường hợp tổng quát Với mọi hàm mất mát khả vi, XGBoost sử dụng gradient và Hessian:

$$G = \sum g_i, \quad H = \sum h_i, \quad w^* = -\frac{G}{H + \lambda}, \quad \text{Score} = \frac{G^2}{H + \lambda}$$

3.3.4 Phân loại nhị phân (Log-loss)

$$g_i = p_i - y_i, \quad h_i = p_i(1 - p_i)$$

Nếu đặt residual $r_i = y_i - p_i = -g_i$ thì:

$$\text{Score}(R) = \frac{(\sum r_i)^2}{\sum p_i(1 - p_i) + \lambda}$$

Hồi quy MSE Với MSE, Hessian $h_i = 1$ với mọi i , khi đó:

$$w^* = \frac{\sum r_i}{|R| + \lambda}, \quad \text{Score}(R) = \frac{(\sum r_i)^2}{|R| + \lambda}$$

Do Hessian là hằng số, ta có thể làm việc trực tiếp với residual mà không cần viết g_i, h_i .

*Ví dụ minh họa XGBoost với hàm mất mát bình phương

Dữ liệu: 4 mẫu, 1 đặc trưng x , nhãn y .

Instance	x	y
1	1.0	2.0
2	2.0	3.0
3	3.0	5.0
4	4.0	6.0

Siêu tham số:

$$\text{Loss: } l(y, \hat{y}) = (y - \hat{y})^2, \quad \eta = 0.5,$$

$$\lambda = 1, \quad \gamma = 0, \quad \text{max depth} = 2$$

Khởi tạo mô hình:

$$F_0(x) = \bar{y} = \frac{2 + 3 + 5 + 6}{4} = 4$$

Vòng lặp thứ nhất ($t = 1$)

Với hàm mất mát bình phương, gradient và Hessian tại mỗi mẫu là:

$$g_i = 2(F_0(x_i) - y_i), \quad h_i = 2$$

Instance	$F_0(x_i)$	g_i	h_i
1	4	$2(4 - 2) = 4$	2
2	4	$2(4 - 3) = 2$	2
3	4	$2(4 - 5) = -2$	2
4	4	$2(4 - 6) = -4$	2

Xét phép tách theo $x \leq 2.5$

Nút trái ($x \leq 2.5$, mẫu 1,2):

$$G_L = 4 + 2 = 6, \quad H_L = 2 + 2 = 4$$

Nút phải ($x > 2.5$, mẫu 3,4):

$$G_R = -2 + (-4) = -6, \quad H_R = 4$$

Trọng số tối ưu của các lá:

$$w_L^* = -\frac{G_L}{H_L + \lambda} = -\frac{6}{4 + 1} = -1.2$$

$$w_R^* = -\frac{G_R}{H_R + \lambda} = -\frac{-6}{4 + 1} = 1.2$$

Độ lợi của phép tách

$$\text{Gain} = \frac{1}{2} \left(\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right)$$

Do $G_L + G_R = 0$, ta có:

$$\text{Gain} = \frac{1}{2} \left(\frac{36}{5} + \frac{36}{5} \right) = 7.2$$

Tree Structure

Vùng	Giá trị dự đoán $f_1(x)$
$x \leq 2.5$	-1.2
$x > 2.5$	1.2

Cập nhật dự đoán

$$F_1(x_i) = F_0(x_i) + \eta f_1(x_i)$$

Instance	x	y	$f_1(x_i)$	$F_1(x_i)$
1	1.0	2.0	-1.2	$4 - 0.6 = 3.4$
2	2.0	3.0	-1.2	3.4
3	3.0	5.0	1.2	4.6
4	4.0	6.0	1.2	4.6

Kết quả cho thấy mô hình đã điều chỉnh dự đoán theo hướng giảm sai số bình phương, đúng với nguyên lý của Gradient Boosting.

3.3.5 Chiến lược điều chỉnh tham số cho Overfitting và Underfitting:

Trong thực tế huấn luyện mô hình XGBoost, việc cân bằng giữa độ lệch (bias) và phương sai (variance) là yếu tố then chốt. Dưới đây là các chiến lược điều chỉnh siêu tham số (hyperparameters) để giải quyết hai vấn đề phổ biến là Overfitting và Underfitting.

Xử lý Overfitting (Quá khớp)

Overfitting xảy ra khi mô hình quá phức tạp, học cả nhiễu (noise) của dữ liệu huấn luyện, dẫn đến khả năng tổng quát hóa kém trên dữ liệu kiểm thử. Để giảm Overfitting, ta cần hạn chế độ phức tạp của mô hình:

- **Giảm `max_depth`:** Độ sâu của cây quyết định mức độ phức tạp của các tương tác đặc trưng. Giảm độ sâu giúp mô hình đơn giản hơn.
- **Tăng `min_child_weight`:** Tham số này quy định tổng trọng số Hessian tối thiểu cần thiết ở một nút con. Tăng giá trị này sẽ ngăn chặn cây phân chia các nút quá cụ thể (chỉ áp dụng cho một nhóm nhỏ mẫu), giúp mô hình bảo thủ hơn.
- **Tăng `gamma` (γ):** Đây là mức giảm hàm mất mát tối thiểu cần thiết để thực hiện một phép tách. Tăng γ làm cho thuật toán trở nên bảo thủ, chỉ tách nút khi lợi ích thực sự lớn.

- **Tăng λ và α :** Đây là các tham số chính quy hóa L2 và L1 trên trọng số lá. Tăng các giá trị này sẽ phạt các trọng số lớn, giúp mô hình mượt mà hơn.
- **Giảm η và tăng `num_round`:** Giảm tốc độ học (learning rate) buộc mô hình phải học chậm hơn và cần nhiều cây hơn để hội tụ, giúp giảm thiểu rủi ro bỏ qua các mẫu tổng quát.
- **Sử dụng Subsampling:** Thiết lập `subsample` và `colsample_bytree` nhỏ hơn 1.0 (ví dụ: 0.8) để thêm tính ngẫu nhiên, giúp mô hình bền vững hơn trước nhiễu.

Xử lý Underfitting (Chưa khớp)

Underfitting xảy ra khi mô hình quá đơn giản để nắm bắt được cấu trúc của dữ liệu, dẫn đến sai số cao trên cả tập huấn luyện và tập kiểm thử. Để khắc phục, ta cần tăng độ phức tạp của mô hình:

- **Tăng `max_depth`:** Cho phép cây phát triển sâu hơn để học các mối quan hệ phức tạp hơn giữa các đặc trưng.
- **Giảm `min_child_weight`:** Cho phép thuật toán học từ các nhóm mẫu nhỏ hơn, giúp mô hình nhạy bén hơn với các chi tiết cục bộ.
- **Giảm γ :** Giảm ngưỡng yêu cầu để tách nút, cho phép cây phát triển đầy đủ hơn.
- **Giảm λ và α :** Giảm bớt sự phạt lên trọng số, cho phép mô hình linh hoạt hơn trong việc khớp dữ liệu.
- **Tăng η :** Đôi khi việc tăng tốc độ học (đi kèm với số lượng cây phù hợp) giúp mô hình thoát khỏi các điểm tối ưu cục bộ nhanh hơn, tuy nhiên cần thận trọng để tránh phân kỳ.

Early Stopping: Bên cạnh việc điều chỉnh các siêu tham số cấu trúc, XGBoost còn hỗ trợ kỹ thuật *early stopping* nhằm hạn chế hiện tượng overfitting trong quá trình huấn luyện. Cụ thể, mô hình được đánh giá trên một tập dữ liệu xác thực (validation set) sau mỗi vòng lặp. Nếu hàm mất mát trên tập validation không được cải thiện sau một số vòng lặp liên tiếp (gọi là `early_stopping_rounds`), quá trình huấn luyện sẽ tự động dừng lại. Cơ chế này giúp lựa chọn số lượng cây tối ưu, đồng thời ngăn mô hình học quá mức vào dữ liệu huấn luyện, từ đó cải thiện khả năng tổng quát hóa.

Bảng tổng hợp tác động của tham số

Bảng dưới đây tóm tắt hướng điều chỉnh các tham số quan trọng trong XGBoost:

Table 1: Tổng hợp hướng điều chỉnh tham số XGBoost

Tham số	Vai trò	Giảm Overfitting	Giảm Underfitting
max_depth	Độ sâu tối đa của cây	Giảm (\downarrow)	Tăng (\uparrow)
min_child_weight	Trọng số Hessian tối thiểu	Tăng (\uparrow)	Giảm (\downarrow)
gamma (γ)	Ngưỡng tách nút tối thiểu	Tăng (\uparrow)	Giảm (\downarrow)
lambda (λ)	Chính quy hóa L2	Tăng (\uparrow)	Giảm (\downarrow)
alpha (α)	Chính quy hóa L1	Tăng (\uparrow)	Giảm (\downarrow)
eta (η)	Tốc độ học (Learning Rate)	Giảm (\downarrow)	Tăng (\uparrow)
subsample	Tỷ lệ mẫu dữ liệu	Giảm (< 1.0)	Tăng ($\rightarrow 1.0$)

4 Phương pháp nghiên cứu

4.1 Tóm tắt ưu nhược điểm của các thuật toán:

Dựa trên các nghiên cứu liên quan và đặc điểm của bài toán dự đoán nguy cơ mắc bệnh tim mạch, các mô hình Logistic Regression, Random Forest và XGBoost được lựa chọn trong đề tài đều có những ưu điểm và hạn chế riêng. Việc phân tích các đặc điểm này giúp làm rõ cơ sở lựa chọn mô hình và định hướng đánh giá kết quả nghiên cứu.

Logistic Regression

Ưu điểm:

Logistic Regression có cấu trúc đơn giản, dễ triển khai và đặc biệt có khả năng diễn giải cao, phù hợp với các bài toán y tế yêu cầu tính minh bạch trong quyết định. Các hệ số của mô hình cho phép đánh giá mức độ ảnh hưởng của từng yếu tố nguy cơ đến khả năng mắc bệnh tim mạch, hỗ trợ cho việc phân tích và ra quyết định lâm sàng. Ngoài ra, mô hình có thời gian huấn luyện nhanh và hoạt động ổn định với các tập dữ liệu có kích thước vừa và lớn.

Nhược điểm:

Do giả định mối quan hệ tuyến tính giữa các đặc trưng đầu vào và logit của biến mục tiêu, Logistic Regression gặp hạn chế trong việc mô hình hóa các quan hệ phi tuyến phức tạp. Khi dữ liệu chứa nhiều tương tác không tuyến tính giữa các yếu tố nguy cơ, hiệu suất dự báo của mô hình có thể bị suy giảm.

Random Forest

Ưu điểm:

Random Forest là mô hình học máy dựa trên tập hợp nhiều cây quyết định, giúp giảm hiện tượng quá khớp và nâng cao khả năng tổng quát hóa. Mô hình này xử lý tốt các mối quan hệ phi tuyến và ít nhạy cảm với nhiễu trong dữ liệu. Ngoài ra, Random Forest còn cung cấp thông tin về mức độ quan trọng của các đặc trưng, giúp phân tích vai trò của từng yếu tố trong dự đoán bệnh tim mạch.

Nhược điểm:

Có độ phức tạp cao hơn và yêu cầu nhiều tài nguyên tính toán, đặc biệt khi số lượng cây trong

mô hình tăng lên. Bên cạnh đó, tính diễn giải của mô hình cũng bị giảm do kết quả dự báo được tổng hợp từ nhiều cây khác nhau.

XGBoost

Ưu điểm:

XGBoost là mô hình boosting tiên tiến, có khả năng học các mẫu dữ liệu phức tạp thông qua việc kết hợp nhiều cây quyết định theo cơ chế tăng cường dần. Mô hình này thường đạt độ chính xác cao trong các bài toán phân loại và được đánh giá cao về khả năng xử lý dữ liệu lớn, dữ liệu phi tuyến cũng như dữ liệu có nhiều đặc trưng đầu vào. Nhờ cơ chế tối ưu hóa gradient, XGBoost cho phép cải thiện hiệu suất dự báo đáng kể so với các mô hình cây truyền thống.

Nhược điểm:

XGBoost đòi hỏi quá trình tinh chỉnh siêu tham số cẩn thận để đạt hiệu quả tối ưu, điều này có thể làm tăng thời gian huấn luyện và độ phức tạp trong quá trình triển khai. Ngoài ra, do cấu trúc mô hình phức tạp, khả năng diễn giải kết quả của XGBoost thường thấp hơn so với các mô hình khác.

4.2 Thuật toán tối ưu hóa

Trong nghiên cứu này, ba mô hình học máy gồm Logistic Regression, Random Forest và XGBoost được áp dụng cho bài toán phân loại nhị phân. Do sự khác biệt về cơ chế huấn luyện, mỗi mô hình sử dụng phương pháp tối ưu hóa phù hợp với đặc điểm riêng của nó. Logistic Regression được huấn luyện bằng cách tối ưu hàm mất mát log-loss thông qua các thuật toán tối ưu lồi nhằm đảm bảo quá trình hội tụ ổn định. Random Forest không sử dụng tối ưu dựa trên gradient mà xây dựng nhiều cây quyết định độc lập dựa trên kỹ thuật lấy mẫu ngẫu nhiên dữ liệu và đặc trưng. Trong khi đó, XGBoost huấn luyện mô hình theo cơ chế boosting, xây dựng các cây quyết định tuần tự để giảm dần sai số của mô hình và kết hợp regularization nhằm hạn chế hiện tượng overfitting.

*Log Loss – Binary Classification

$$L = -\frac{1}{n} \sum_{i=1}^n [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)]$$

*Log Loss – Multiclass Classification

$$L = -\frac{1}{n} \sum_{i=1}^n \sum_{k=1}^K y_{ik} \log(p_{ik})$$

4.3 Chỉ số đánh giá hiệu suất

- **Accuracy:** Tỷ lệ dự đoán đúng trên toàn bộ dữ liệu; chỉ mang tính tham khảo khi dữ liệu mất cân bằng, do có thể che giấu sai lệch giữa các lớp [5, 10, 13].
- **Balanced Accuracy:** Trung bình cộng của Recall (Sensitivity) và Specificity (True Negative Rate); phản ánh khả năng phân loại cân bằng giữa cả hai lớp, đặc biệt phù hợp trong bối cảnh dữ liệu y sinh mất cân bằng khi số ca bệnh và không bệnh chênh lệch lớn [10, 11, 13].
- **Precision:** Đo mức độ chính xác của các dự đoán dương tính; phản ánh khả năng giảm báo động giả (False Positive), đặc biệt quan trọng khi chi phí chẩn đoán sai là cao [6, 11].
- **Recall (Sensitivity):** Đo khả năng phát hiện đúng các ca bệnh; quan trọng nhất trong y sinh nhằm hạn chế bỏ sót bệnh nhân mắc bệnh (False Negative) [5, 7, 13].
- **F1-score:** Trung bình điều hòa giữa Precision và Recall; được sử dụng khi cần cân bằng giữa khả năng phát hiện bệnh và hạn chế báo động giả, đặc biệt với dữ liệu mất cân bằng [6, 10].
- **ROC–AUC:** Đánh giá khả năng phân biệt tổng quát giữa hai lớp thông qua diện tích dưới đường cong ROC; không phụ thuộc vào ngưỡng phân loại cụ thể [9, 12].
- **Log-loss (Cross-Entropy Loss):** Đánh giá chất lượng xác suất dự đoán của mô hình; giá trị càng nhỏ cho thấy mô hình dự đoán xác suất càng chính xác [8, 11].
- **Confusion Matrix:** Cung cấp cái nhìn chi tiết về các lỗi TP, TN, FP, FN, hỗ trợ phân tích sai lệch mô hình và lựa chọn chỉ số đánh giá phù hợp trong bối cảnh y sinh [5, 13].

5 Chuẩn bị dữ liệu

5.1 Nguồn dữ liệu

Dữ liệu nghiên cứu được sử dụng trong đề tài này được lấy từ bộ *Cardiovascular Disease Dataset*, được công bố công khai trên nền tảng Kaggle bởi người dùng *sulianova*. Bộ dữ liệu bao gồm một tệp Excel chứa các thông tin lâm sàng đã được ẩn danh của bệnh nhân và được sử dụng phổ biến trong nhiều nghiên cứu học máy liên quan đến dự đoán nguy cơ mắc bệnh tim mạch.

Tập dữ liệu gồm 13 biến đầu vào và 1 biến đầu ra, phản ánh các đặc điểm nhân khẩu học, chỉ số sinh học và thói quen sinh hoạt của bệnh nhân.

5.1.1 Các biến đầu vào

- **age**: Tuổi của bệnh nhân, được tính bằng số ngày kể từ khi sinh.
- **gender**: Giới tính của bệnh nhân, được mã hóa dưới dạng biến phân loại.
- **height**: Chiều cao của bệnh nhân, đơn vị centimet (cm).
- **weight**: Cân nặng của bệnh nhân, đơn vị kilogram (kg).
- **ap_hi**: Huyết áp tâm thu, phản ánh áp lực máu trong động mạch khi tim co bóp.
- **ap_lo**: Huyết áp tâm trương, phản ánh áp lực máu trong động mạch khi tim giãn ra.
- **cholesterol**: Mức cholesterol trong máu, được mã hóa theo ba mức:
 - 1 – Bình thường
 - 2 – Cao hơn mức bình thường
 - 3 – Cao hơn nhiều so với mức bình thường
- **gluc**: Mức glucose trong máu, được mã hóa theo ba mức:
 - 1 – Bình thường
 - 2 – Cao hơn mức bình thường
 - 3 – Cao hơn nhiều so với mức bình thường
- **smoke**: Thói quen hút thuốc của bệnh nhân, biểu diễn dưới dạng biến nhị phân.
- **alco**: Thói quen sử dụng rượu bia của bệnh nhân, biểu diễn dưới dạng biến nhị phân.
- **active**: Mức độ hoạt động thể chất của bệnh nhân, biểu diễn dưới dạng biến nhị phân.

5.1.2 Biến đầu ra

- **cardio**: Biến mục tiêu biểu thị sự hiện diện hoặc không của bệnh tim mạch. Giá trị 1 cho biết bệnh nhân có nguy cơ hoặc mắc bệnh tim mạch, trong khi giá trị 0 cho biết bệnh nhân không mắc bệnh tim mạch.

5.2 Các bước tiền xử lý dữ liệu

5.2.1 Kết hợp dữ liệu

Tập dữ liệu sử dụng trong nghiên cứu được thu thập từ một tệp dữ liệu tim mạch và được đọc vào môi trường phân tích bằng thư viện Pandas. Sau khi tải dữ liệu, một bản sao của *DataFrame* gốc được tạo ra nhằm đảm bảo dữ liệu ban đầu không bị thay đổi trong quá trình tiền xử lý.

Cột `id` được loại bỏ do không mang ý nghĩa dự báo và không ảnh hưởng đến quá trình huấn luyện các mô hình học máy. Sau bước này, tập dữ liệu còn lại bao gồm các biến nhân trắc học, lâm sàng, hành vi và biến mục tiêu `cardio`, tạo thành một *DataFrame* thống nhất phục vụ cho các bước xử lý tiếp theo.

5.2.2 Làm sạch và chuyển đổi dữ liệu

Kiểm tra giá trị thiếu và trùng lặp: Kết quả kiểm tra cho thấy tập dữ liệu không tồn tại giá trị thiếu (*missing values*). Tuy nhiên, dữ liệu có xuất hiện các bản ghi trùng lặp, do đó các dòng trùng lặp được loại bỏ nhằm tránh gây nhiễu và sai lệch cho quá trình huấn luyện mô hình.

Chuẩn hóa và chuyển đổi đơn vị: Biến `age` ban đầu được lưu trữ dưới dạng số ngày tuổi. Để thuận tiện cho việc phân tích và diễn giải, biến này được chuyển đổi sang đơn vị năm theo công thức:

$$\text{Age (years)} = \frac{\text{Age (days)}}{365}$$

Sau đó, giá trị tuổi được ép kiểu về số nguyên.

Xử lý dữ liệu huyết áp bất thường: Hai biến huyết áp tâm thu (`ap_hi`) và huyết áp tâm trương (`ap_lo`) được kiểm tra và làm sạch dựa trên các tiêu chí sinh học thực tế:

- Lấy giá trị tuyệt đối để loại bỏ các giá trị âm không hợp lệ.
- Loại bỏ các giá trị ngoài ngưỡng sinh lý:
 - `ap_hi`: từ 60 đến 245 mmHg
 - `ap_lo`: từ 40 đến 160 mmHg
- Loại bỏ các bản ghi không hợp lý khi huyết áp tâm trương lớn hơn huyết áp tâm thu.

Bước này giúp đảm bảo dữ liệu phản ánh đúng các điều kiện y sinh thực tế, tránh gây sai lệch cho mô hình học máy.

Làm sạch dữ liệu chiều cao, cân nặng và BMI: Từ hai biến chiều cao (*height*) và cân nặng (*weight*), một biến mới BMI (Body Mass Index) được tính toán theo công thức:

$$\text{BMI} = \frac{\text{Weight (kg)}}{(\text{Height (m)})^2}$$

Trong đó, chiều cao được chuyển đổi từ centimet sang mét trước khi tính toán.

Sau đó, các giá trị không hợp lý được loại bỏ theo các ngưỡng sinh học:

- Chiều cao: 140 – 210 cm
- Cân nặng: 40 – 180 kg
- BMI: 15 – 50

Việc này giúp loại bỏ các ngoại lai phi thực tế và nâng cao độ tin cậy của dữ liệu.

5.2.3 Thống kê mô tả và ý nghĩa các biến

Sau khi làm sạch, các chỉ số thống kê mô tả bao gồm trung bình, trung vị, độ lệch chuẩn, giá trị nhỏ nhất và lớn nhất được tính toán cho các biến quan trọng.

Age – Tuổi

- Giá trị trung bình: khoảng 53 tuổi
- Phạm vi: 29 – 64 tuổi

Đây là nhóm tuổi có nguy cơ cao mắc bệnh tim mạch, phù hợp với mục tiêu nghiên cứu.

Height và Weight – Chiều cao và cân nặng

- Chiều cao trung bình: khoảng 165 cm
- Cân nặng trung bình: khoảng 74 kg

Dữ liệu ổn định và không còn giá trị ngoại lai sau quá trình làm sạch.

BMI – Chỉ số khối cơ thể

- Giá trị trung bình: khoảng 27.4

Giá trị này thuộc nhóm thừa cân (*Overweight*) theo phân loại của WHO. BMI cao có mối liên hệ chặt chẽ với tăng huyết áp và nguy cơ mắc bệnh tim mạch.

ap_hi và ap_lo – Huyết áp tâm thu và tâm trương

- Huyết áp tâm thu trung vị: khoảng 120 mmHg
- Huyết áp tâm trương trung vị: khoảng 80 mmHg

Một tỷ lệ đáng kể đối tượng có huyết áp cao, đây là yếu tố nguy cơ quan trọng của bệnh tim mạch.

Biến mục tiêu – Cardio Biến cardio có phân phối tương đối cân bằng giữa hai lớp (0: không bệnh, 1: có bệnh). Đây là đặc điểm thuận lợi cho bài toán phân loại nhị phân, giúp các mô hình học máy hoạt động hiệu quả mà không cần áp dụng các kỹ thuật xử lý mất cân bằng lớp phức tạp.

5.2.4 Kiểm tra tính hợp lệ dữ liệu

Sau toàn bộ quá trình tiền xử lý, dữ liệu được kiểm tra lại để đảm bảo:

- Không tồn tại giá trị âm ở các biến đo lường vật lý.
- Các biến huyết áp, chiều cao, cân nặng và BMI đều nằm trong ngưỡng sinh học hợp lý.
- Tất cả các biến đều ở dạng số, sẵn sàng cho quá trình huấn luyện mô hình.

5.2.5 Tách đặc trưng đầu vào và biến đầu ra

Cuối cùng, tập dữ liệu được chia thành:

- **Tập đặc trưng đầu vào (X):** bao gồm các biến nhân trắc học, lâm sàng và hành vi.
- **Biến đầu ra (y):** biến nhị phân cardio, đại diện cho tình trạng mắc bệnh tim mạch.

5.3 EDA – Khám phá dữ liệu

5.3.1 Vẽ biểu đồ tròn cho biến đầu ra

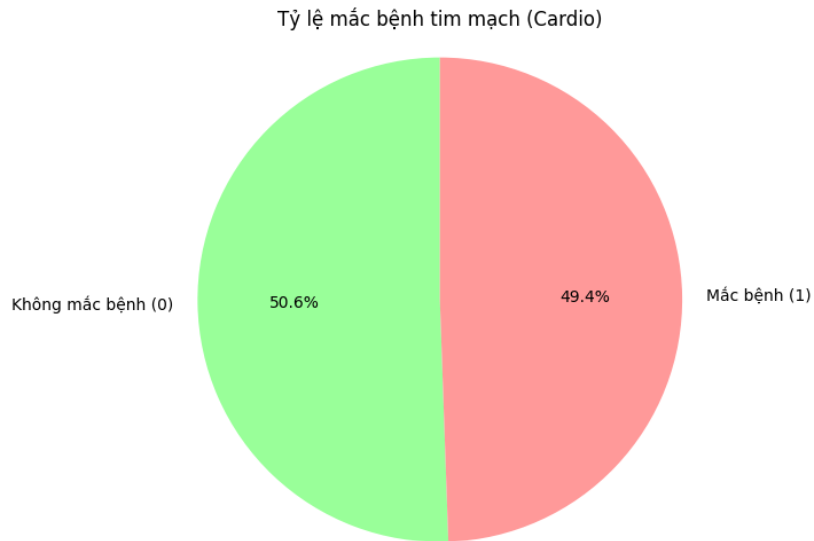


Figure 1: Tỷ lệ mắc bệnh tim mạch cardio

Nhận xét: Biểu đồ tròn cho thấy biến mục tiêu cardio có phân phối gần như cân bằng giữa hai lớp, trong đó nhóm không mắc bệnh tim mạch chiếm khoảng 50.6% và nhóm mắc bệnh chiếm khoảng 49.4%. Mức chênh lệch giữa hai lớp là không đáng kể, cho thấy tập dữ liệu không gặp vấn đề mất cân bằng nghiêm trọng. Đây là điều kiện thuận lợi cho bài toán phân loại nhị phân, giúp các mô hình học máy hạn chế hiện tượng thiên lệch về một lớp cụ thể và cho phép đánh giá kết quả dự đoán một cách khách quan và đáng tin cậy.

5.3.2 Kiểm tra các feature không nhất quán

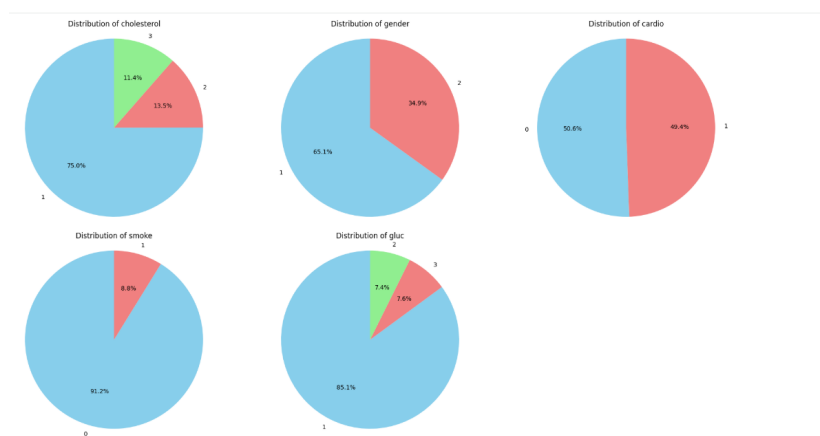


Figure 2: Kiểm tra các feature không nhất quán

Biến cholesterol Mức cholesterol bình thường (mức 1) chiếm tỷ lệ cao nhất, khoảng 75%, trong khi các mức trên bình thường và cao (mức 2 và mức 3) chiếm tỷ lệ thấp hơn. Phân phối này phản ánh tương đối đúng thực tế dân số, trong đó phần lớn đối tượng có mức cholesterol trong ngưỡng cho phép, tuy nhiên vẫn tồn tại một nhóm đáng kể có nguy cơ tim mạch cao.

Biến gender Nữ giới (mã 1) chiếm khoảng 65.1%, cao hơn so với nam giới (khoảng 34.9%). Sự chênh lệch này có thể ảnh hưởng đến quá trình học của mô hình và cần được xem xét khi phân tích kết quả dự đoán theo giới tính.

Biến cardio (biến mục tiêu) Tỷ lệ đối tượng mắc bệnh tim mạch (khoảng 49.4%) và không mắc bệnh (khoảng 50.6%) gần như cân bằng. Đây là một ưu điểm đáng kể của tập dữ liệu, giúp bài toán phân loại nhị phân không gặp vấn đề mất cân bằng lớp và cho phép đánh giá mô hình một cách khách quan và đáng tin cậy.

Biến smoke Phần lớn đối tượng không hút thuốc (khoảng 91.2%), trong khi tỷ lệ hút thuốc chỉ chiếm khoảng 8.8%. Mặc dù biến này có phân phối lệch mạnh, nhưng vẫn mang ý nghĩa y sinh quan trọng do hút thuốc là một yếu tố nguy cơ rõ rệt của bệnh tim mạch.

Biến gluc Đa số đối tượng có mức đường huyết bình thường (khoảng 85.1%), trong khi các mức trên bình thường và cao chiếm tỷ lệ nhỏ (khoảng 15%). Điều này cho thấy biến gluc có phân phối không cân bằng, tuy nhiên vẫn cần được giữ lại trong mô hình do mối liên hệ chặt chẽ giữa rối loạn đường huyết và nguy cơ mắc bệnh tim mạch.

5.3.3 Kiểm tra mức độ phân phối của các biến định lượng

	age	height	weight	ap_hi \
count	68302.000000	68302.000000	68302.000000	68302.000000
mean	52.825788	164.477834	74.004135	126.655427
std	6.769508	7.789990	13.904419	16.659478
min	29.000000	140.000000	40.000000	60.000000
25%	48.000000	159.000000	65.000000	120.000000
50%	53.000000	165.000000	72.000000	120.000000
75%	58.000000	170.000000	82.000000	140.000000
max	64.000000	207.000000	180.000000	240.000000
skew	-0.303640	0.154882	0.816983	0.929728
kurtosis	-0.821221	0.067750	1.221511	1.831318

	ap_lo	BMI
count	68302.000000	68302.000000
mean	81.295584	27.385359
std	9.417059	5.028603
min	40.000000	15.035584
25%	80.000000	23.875115
50%	80.000000	26.306318
75%	90.000000	30.110991
max	160.000000	50.000000
skew	0.326258	0.990069
kurtosis	1.687110	1.223016

Figure 3: Mức độ phân phối của các biến định lượng

Nhóm biến nhân trắc học (Age, Height, Weight, BMI)

Age (Tuổi): Tuổi của các đối tượng trong tập dữ liệu nằm trong khoảng từ 29 đến 64 tuổi, với giá trị trung bình xấp xỉ 53 tuổi.

Nhận xét: Đây là nhóm độ tuổi có nguy cơ cao mắc bệnh tim mạch, phù hợp với mục tiêu nghiên cứu. Phân phối của biến Age có xu hướng lệch trái nhẹ (skewness ≈ -0.3), cho thấy số lượng đối tượng lớn tuổi chiếm tỷ lệ cao hơn so với nhóm trẻ tuổi. Đặc điểm này là hợp lý vì bệnh tim mạch thường xuất hiện phổ biến ở người cao tuổi.

Height (Chiều cao):

Chiều cao của các đối tượng dao động trong khoảng từ 140 cm đến 207 cm.

Nhận xét: Phân phối của biến Height gần với phân phối chuẩn (skewness ≈ 0.15), không tồn tại các giá trị ngoại lai cực đoan. Độ lệch chuẩn tương đối thấp (khoảng 7.8 cm), cho thấy dữ liệu chiều cao ổn định và đồng nhất.

Weight và BMI (Cân nặng và chỉ số khối cơ thể):

Giá trị trung bình của cân nặng vào khoảng 74 kg, trong khi chỉ số BMI trung bình xấp xỉ 27.4.

Nhận xét: Theo phân loại của Tổ chức Y tế Thế giới (WHO), giá trị BMI trung bình này thuộc nhóm *thừa cân (Overweight)*. Phân phối của hai biến Weight và BMI đều có độ lệch dương rõ rệt (skewness $\approx 0.8-0.99$), cho thấy đuôi phân phối kéo dài về phía giá trị lớn. Điều này phản ánh sự tồn tại của một nhóm đáng kể đối tượng bị béo phì và béo phì nặng, vốn là các yếu tố nguy cơ quan trọng của bệnh tim mạch. Đặc điểm này giúp mô hình học máy dễ dàng nhận diện nhóm đối tượng có nguy cơ cao.

Nhóm biến sức khỏe tim mạch (ap_hi, ap_lo) ap_hi (Huyết áp tâm thu):

Giá trị huyết áp tâm thu nằm trong khoảng từ 70 đến 240 mmHg, với trung vị (khoảng phân vị 50%) là 120 mmHg. Khoảng 75% đối tượng có huyết áp tâm thu không vượt quá 140 mmHg, trong khi khoảng 25% còn lại có giá trị lớn hơn 140 mmHg, tương ứng với nhóm tăng huyết áp.

Nhận xét: Đây là nhóm đối tượng tiềm năng thuộc lớp mắc bệnh tim mạch (cardio = 1) mà mô hình học máy có khả năng phát hiện hiệu quả.

ap_lo (Huyết áp tâm trương):

Huyết áp tâm trương dao động trong khoảng từ 40 đến 150 mmHg, với trung vị khoảng 80 mmHg.

Nhận xét: Các giá trị đều nằm trong ngưỡng sinh học hợp lý, cho thấy dữ liệu đã được làm sạch hiệu quả và phản ánh đúng tình trạng sức khỏe thực tế của đối tượng nghiên cứu.

Hình dạng phân phối (Skewness và Kurtosis) Skewness (Độ lệch):

Hầu hết các biến quan trọng như Weight, BMI và ap_hi đều có độ lệch dương (positive skew). Điều này cho thấy phần lớn đối tượng có chỉ số ở mức bình thường, trong khi tồn tại một nhóm nhỏ có giá trị rất cao (béo phì hoặc huyết áp cao). Đây là đặc điểm thuận lợi cho các mô hình học máy, vì các trường hợp mắc bệnh thường nằm ở phần đuôi trên của phân phối.

Kurtosis (Độ nhọn):

Giá trị kurtosis của các biến đều tương đối thấp (nhỏ hơn 2), cho thấy dữ liệu không quá tập trung quanh giá trị trung bình và cũng không xuất hiện các đuôi phân phối quá dài. Điều này cho thấy tập dữ liệu ít ngoại lai cực đoan và có tính ổn định cao, giúp quá trình huấn luyện mô hình diễn ra hiệu quả.

5.3.4 Vẽ histogram cho các biến định lượng

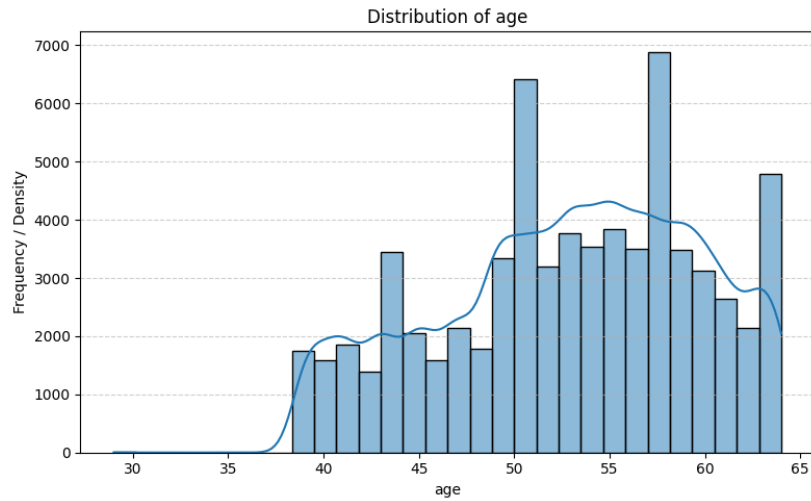


Figure 4: Histogram age

- **Trục X:** Tuổi của các cá nhân trong tập dữ liệu (đơn vị: năm).
- **Trục Y:** Tần suất (số lượng quan sát) hoặc mật độ phân bố của tuổi.

Nhận xét: Phân bố tuổi tập trung chủ yếu ở nhóm trung niên, khoảng từ 40 đến 60 tuổi, trong khi số lượng đối tượng trẻ tuổi và cao tuổi chiếm tỷ lệ thấp hơn. Phân phối không hoàn toàn đối xứng, cho thấy mẫu nghiên cứu có xu hướng thiên về nhóm tuổi trung bình.

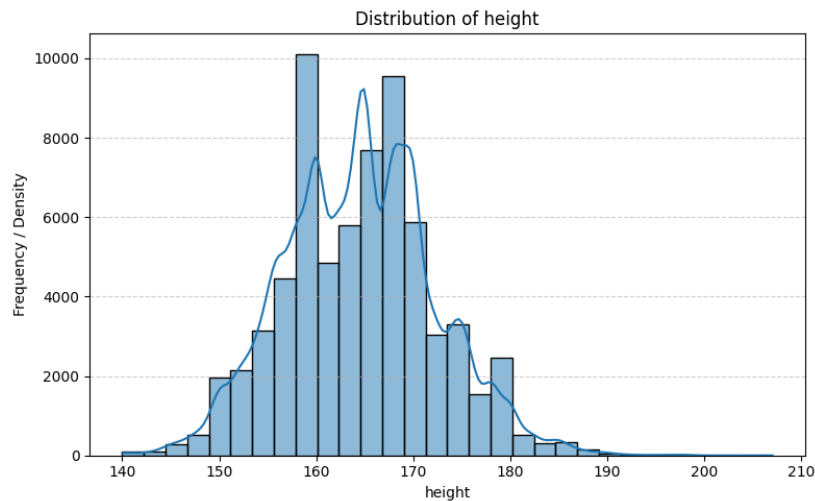


Figure 5: Histogram Height

- **Trục X:** Chiều cao của các cá nhân (cm)
- **Trục Y:** Tần suất hoặc mật độ phân bố chiều cao.

Nhận xét: Chiều cao có phân bố khá cân đối và gần với phân bố chuẩn, tập trung quanh giá trị trung bình. Các giá trị quá thấp hoặc quá cao xuất hiện ít, cho thấy dữ liệu chiều cao tương đối ổn định.

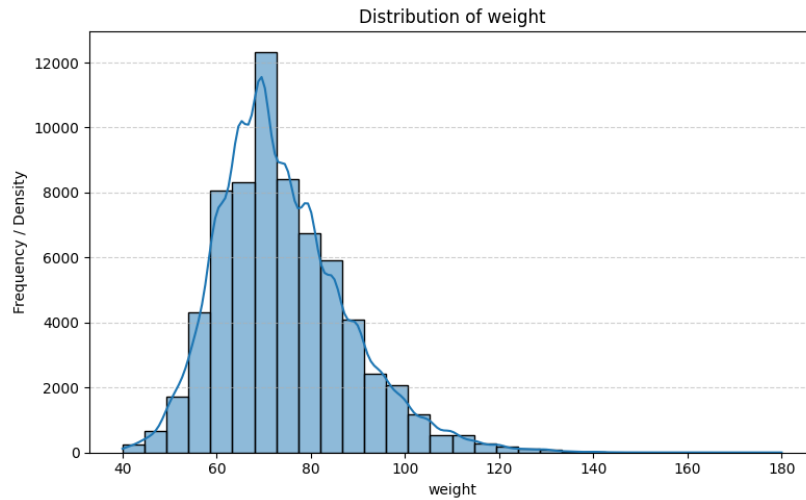


Figure 6: Histogram Weight

- **Trục X:** Cân nặng của các cá nhân (kg).
- **Trục Y:** Tần suất hoặc mật độ phân bố cân nặng

Nhận xét: Cân nặng có xu hướng lệch phải, phần lớn tập trung ở mức trung bình nhưng vẫn tồn tại một số giá trị lớn, phản ánh sự khác biệt rõ rệt về thể trạng giữa các cá nhân.

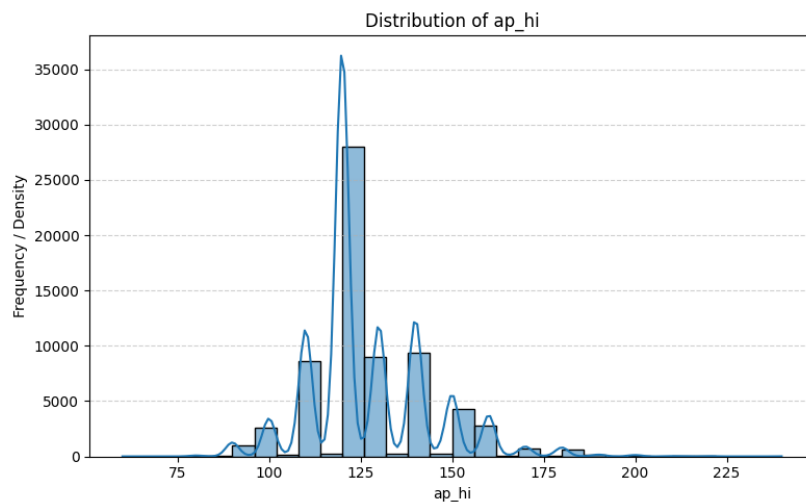


Figure 7: Histogram ApHi

- **Trục X:** Giá trị huyết áp tâm thu (mmHg).

- **Trục Y:** Tần suất hoặc mật độ phân bố huyết áp tâm thu.

Nhận xét: Huyết áp tâm thu (aphi) tập trung chủ yếu trong khoảng 110–140 mmHg, với đỉnh phân bố quanh 120 mmHg, phù hợp với mức huyết áp phổ biến trong dân số. Phân phối có xu hướng lệch phải, thể hiện sự tồn tại của một nhóm nhỏ đối tượng có huyết áp cao. Nhóm này tuy chiếm tỷ lệ không lớn nhưng có ý nghĩa quan trọng vì liên quan trực tiếp đến nguy cơ mắc bệnh tim mạch.

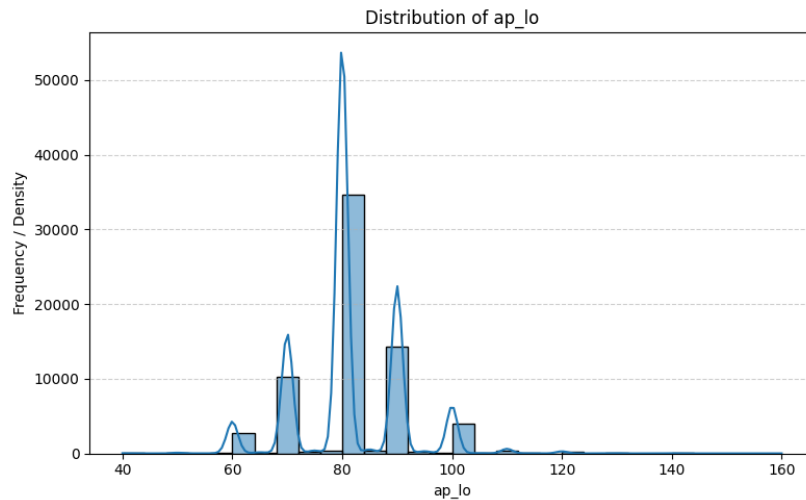


Figure 8: Histogram ApLo

- **Trục X:** Giá trị huyết áp tâm trương của các cá nhân (đơn vị: mmHg).
- **Trục Y:** Tần suất hoặc mật độ phân bố của huyết áp tâm trương.

Nhận xét: Phân bố huyết áp tâm trương không mượt và hình thành nhiều cụm giá trị khác nhau, phản ánh sự khác biệt về thói quen sinh hoạt cũng như tình trạng sức khỏe giữa các cá nhân. Đặc điểm này cho thấy biến `ap_lo` có khả năng cung cấp thông tin phân biệt hữu ích cho mô hình học máy trong việc dự đoán nguy cơ mắc bệnh tim mạch.

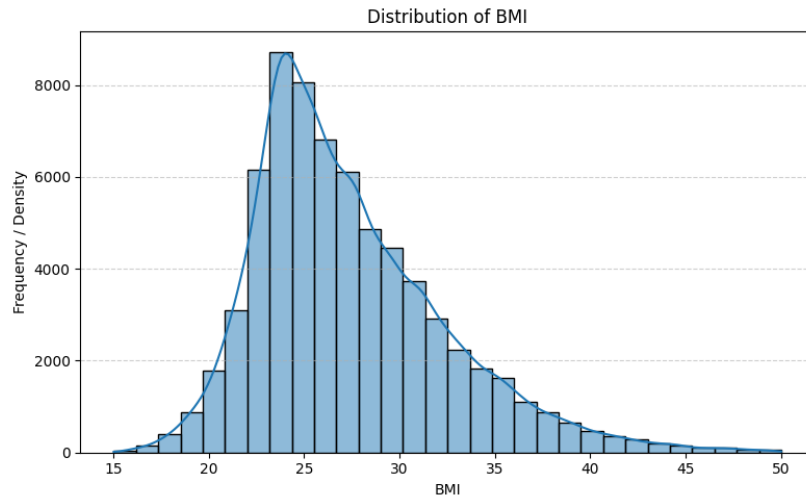


Figure 9: Histogram BMI

- **Trục X:** Chỉ số khối cơ thể BMI (đơn vị: kg/m^2).
- **Trục Y:** Tần suất hoặc mật độ phân bố của BMI.

Nhận xét: Phân phối của biến BMI có xu hướng lệch phải, với phần lớn giá trị tập trung trong khoảng từ mức bình thường đến thừa cân nhẹ. Tuy nhiên, sự xuất hiện của một số giá trị BMI cao cho thấy tồn tại một nhóm đối tượng có nguy cơ béo phì. Đây là một yếu tố nguy cơ quan trọng liên quan trực tiếp đến bệnh tim mạch và có khả năng đóng vai trò đáng kể trong quá trình dự đoán của mô hình học máy.

5.3.5 Kiểm tra mức độ phân phối và sự cân bằng của các biến phân loại

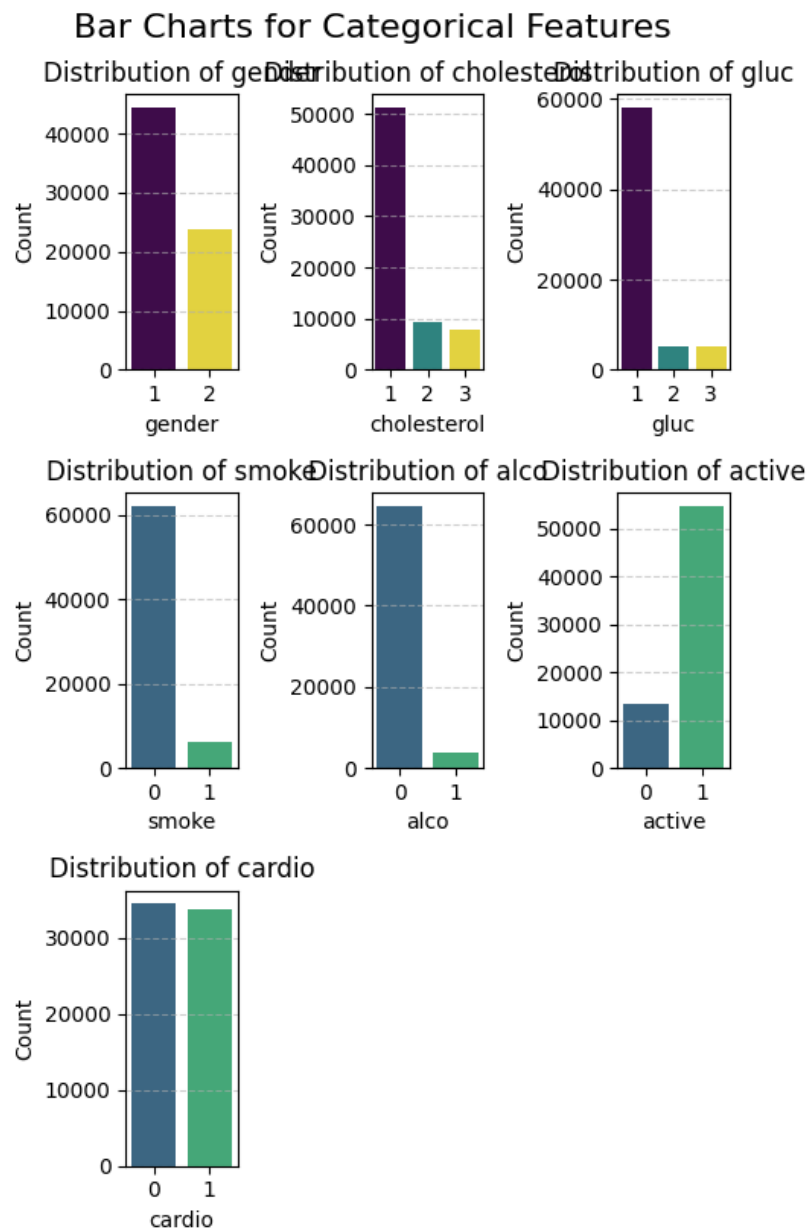


Figure 10: Mức độ phân phối và sự cân bằng của các biến phân loại

Biến Gender có tỷ lệ nữ cao hơn nam nhưng mức chênh lệch không lớn và vẫn chấp nhận được. Biến Cholesterol chủ yếu tập trung ở mức bình thường, các mức cao chiếm tỷ lệ thấp hơn. Biến Glucose (gluc) cho thấy mức bình thường chiếm ưu thế rõ rệt, các mức cao rất ít, thể hiện sự mất cân bằng đáng kể. Các biến hành vi như Smoke và Alcohol intake (alco) có tỷ lệ không hút thuốc và không uống rượu rất cao, cho thấy phân phối lệch mạnh. Biến Physical activity (active) có đa số đối tượng hoạt động thể chất thường xuyên, mức mất cân bằng ở mức trung bình. Riêng biến mục tiêu Cardio có phân phối tương đối cân bằng giữa

hai lớp, thuận lợi cho bài toán phân loại. Nhìn chung, nhiều biến phân loại bị mất cân bằng, đặc biệt là các biến hành vi, tuy nhiên biến mục tiêu cardio gần cân bằng nên không cần áp dụng kỹ thuật xử lý mất cân bằng lớp.

5.3.6 Tính hệ số tương quan giữa các biến đầu vào và đầu ra

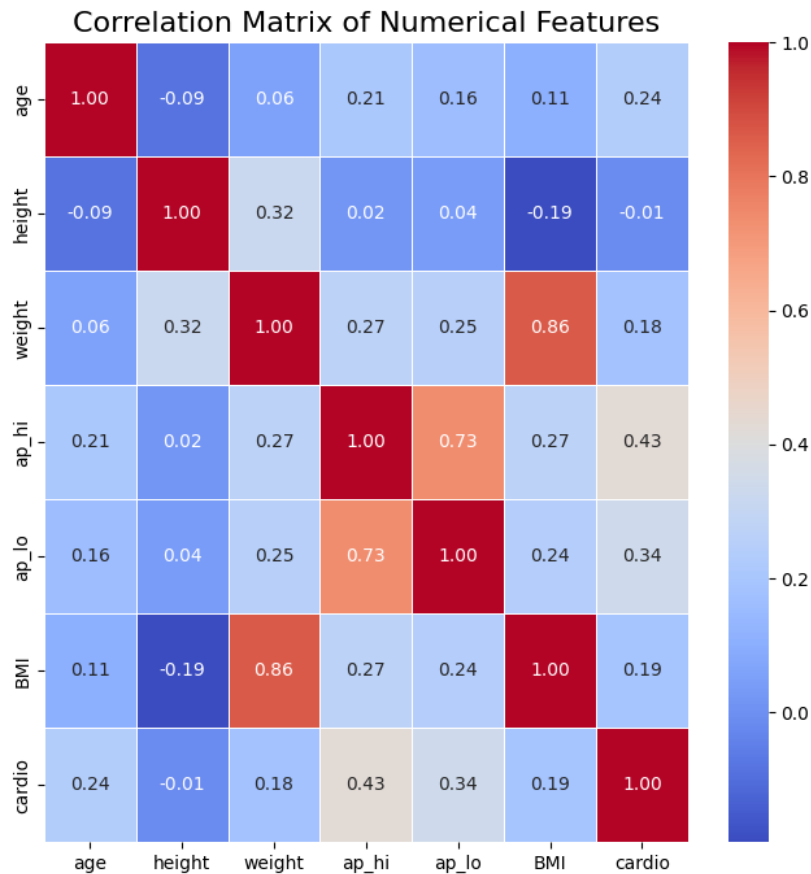


Figure 11: Hệ số tương quan giữa các biến đầu vào và đầu ra

Hiện tượng Đa cộng tuyến (Multicollinearity) giữa các biến hình thể. Quan sát: Có một sự tương quan thuận cực kỳ mạnh (hệ số tương quan $r > 0.8$) giữa weight (cân nặng) và BMI. Đồng thời, height (chiều cao) cũng có tương quan nhất định với BMI. Nguyên nhân: Điều này là hiển nhiên vì công thức $BMI = \frac{Weight}{Height^2}$. Biến BMI được tạo ra trực tiếp từ hai biến kia, dẫn đến việc thông tin bị lặp lại redundancy. Rủi ro: Việc giữ cả 3 biến này sẽ gây ra hiện tượng đa cộng tuyến, khiến mô hình bị nhiễu, khó xác định mức độ quan trọng thực sự của từng biến Feature Importance và làm tăng chi phí tính toán không cần thiết.

Tương quan giữa Huyết áp ap_hi, ap_lo và Biến mục tiêu cardio Quan sát: Cả ap_hi và ap_lo đều có sự tương quan mạnh với nhau (điều này hợp lý về mặt sinh học). Đặc biệt, hai chỉ số này có tương quan dương (màu nóng) rõ rệt với biến mục tiêu cardio. Điều này xác nhận

Huyết áp là một “Predictor” (biến dự báo) quan trọng nhất trong việc phát hiện bệnh tim mạch.

Tương quan giữa Tuổi age và BMI: Tuổi tác age có tương quan dương nhẹ với cardio, phản ánh đúng thực tế: tuổi càng cao, nguy cơ mắc bệnh tim càng lớn. BMI cũng có tương quan dương với ap_hi và ap_lo (người béo phì thường có huyết áp cao).

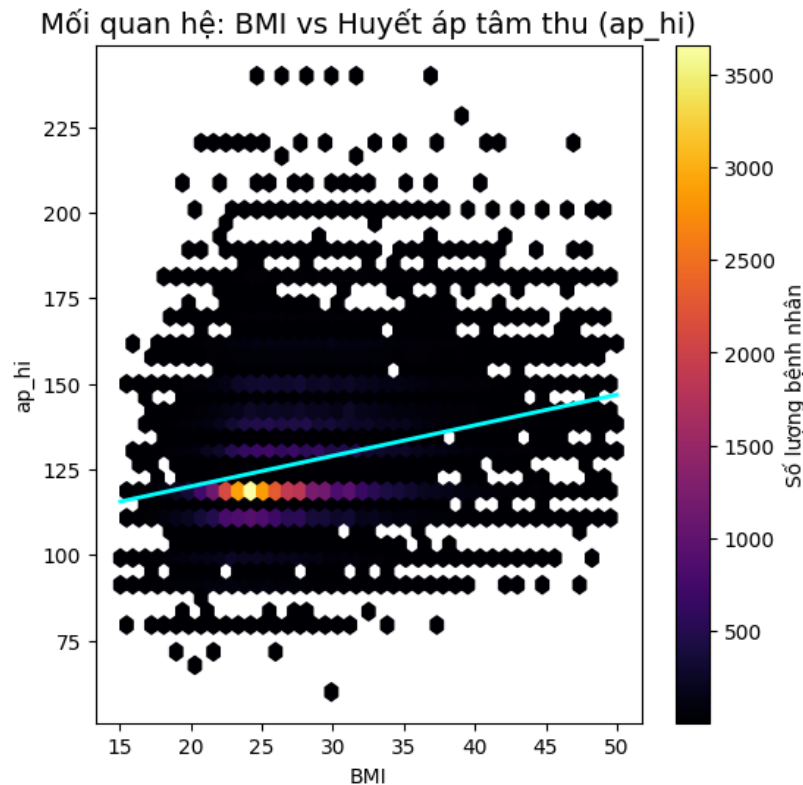


Figure 12: Mối quan hệ giữa BMI và huyết áp tâm thu

- **Trục X:** Chỉ số khối cơ thể BMI.
- **Trục Y:** Huyết áp tâm thu (ap_hi, mmHg).

Biểu đồ phân tán thể hiện mối quan hệ giữa BMI và huyết áp tâm thu (ap_hi) – hai biến định lượng liên tục (*num vs. num*). Đường hồi quy cho thấy xu hướng đồng biến nhẹ, tức là khi BMI tăng thì huyết áp tâm thu có xu hướng tăng theo. Điều này phù hợp với cơ sở y sinh học, vì tình trạng thừa cân và béo phì thường làm gia tăng áp lực lên hệ tim mạch.

Tuy nhiên, mức độ phân tán của các điểm dữ liệu khá lớn. Ở cùng một mức BMI, giá trị ap_hi có thể dao động trong một khoảng rộng, từ mức bình thường đến mức tăng cao. Điều này cho thấy BMI không phải là yếu tố duy nhất ảnh hưởng đến huyết áp, mà còn chịu tác động bởi nhiều yếu tố khác như tuổi, giới tính, mức độ hoạt động thể chất, thói quen sinh hoạt và các bệnh nền.

Ngoài ra, mật độ điểm dữ liệu tập trung nhiều nhất ở nhóm BMI trung bình (khoảng 20–30) và huyết áp tâm thu trong ngưỡng 110–130 mmHg, phản ánh đặc điểm phổ biến của quần thể nghiên cứu. Ở các mức BMI cao hơn (béo phì), số lượng quan sát ít hơn nhưng xu hướng huyết áp tăng rõ rệt hơn, cho thấy nguy cơ tim mạch có xu hướng gia tăng khi BMI vượt ngưỡng bình thường.

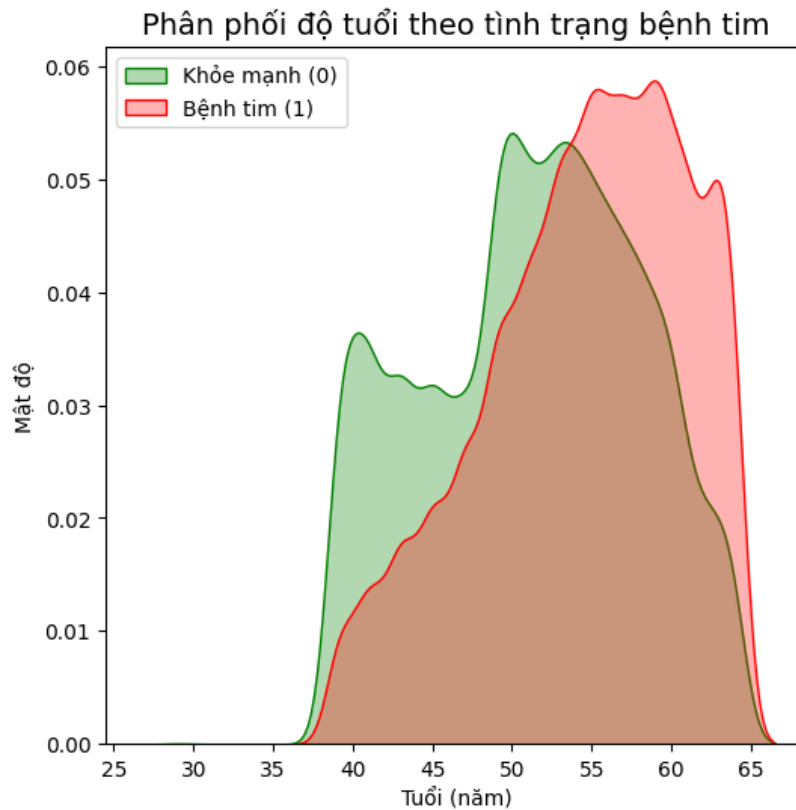


Figure 13: Phân phối tuổi theo tình trạng bệnh nhân

- **Trục X:** Tuổi của đối tượng nghiên cứu (đơn vị: năm).
- **Trục Y:** Mật độ phân bố (Density).

Biểu đồ mật độ xác suất (Kernel Density Estimation – KDE) cho thấy sự khác biệt rõ rệt về phân phối tuổi giữa hai nhóm mắc bệnh và không mắc bệnh tim mạch. Tuổi trung bình của nhóm mắc bệnh cao hơn đáng kể so với nhóm không mắc bệnh. Dựa trên kết quả thống kê từ dữ liệu, mức chênh lệch tuổi trung bình giữa hai nhóm thường dao động khoảng từ 2 đến 3 tuổi hoặc cao hơn.

Sự chênh lệch này có ý nghĩa thống kê và phù hợp với quy luật sinh học, khi hệ tim mạch có xu hướng suy giảm chức năng theo thời gian, dẫn đến nguy cơ mắc bệnh tim mạch gia tăng ở người cao tuổi.

Biểu đồ KDE xác nhận mối tương quan thuận giữa tuổi tác và nguy cơ mắc bệnh tim mạch. Tỷ lệ mắc bệnh tăng dần theo độ tuổi và đạt mức cao ở nhóm người cao tuổi (*Senior*). Do đó, biến *age* được xem là một biến dự báo (*predictor*) quan trọng và không thể thiếu trong quá trình huấn luyện mô hình học máy.

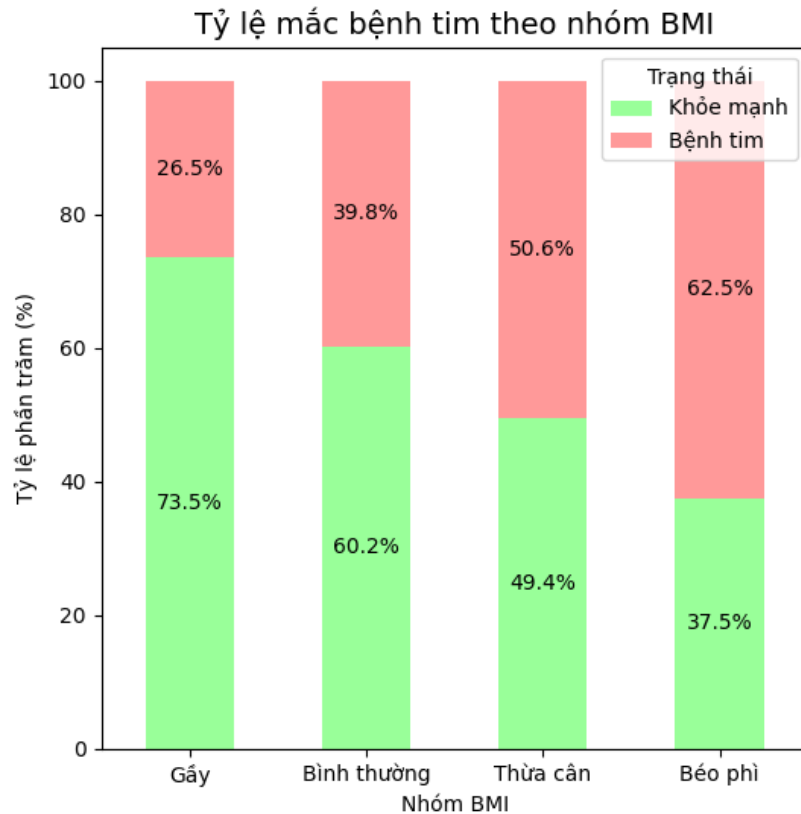


Figure 14: Tỷ lệ mắc bệnh tim theo nhóm BMI

- **Trục X:** Các nhóm chỉ số khối cơ thể BMI.
- **Trục Y:** Tỷ lệ bệnh nhân khỏe mạnh và mắc bệnh tim mạch.

Bảng số liệu và biểu đồ cho thấy tỷ lệ mắc bệnh tim mạch tăng rõ rệt theo từng nhóm BMI, thể hiện một xu hướng nhất quán và có ý nghĩa thực tiễn.

Cụ thể, nhóm gầy có tỷ lệ mắc bệnh tim mạch thấp nhất, chỉ khoảng 26.4%, trong khi 73.6% đối tượng còn lại không mắc bệnh. Ở nhóm BMI bình thường, tỷ lệ mắc bệnh tăng lên đáng kể, đạt mức 39.8%, cho thấy nguy cơ tim mạch đã cao hơn so với nhóm gầy. Đối với nhóm thừa cân, tỷ lệ mắc bệnh tim mạch vượt ngưỡng 50% (với giá trị khoảng 50.6%), tức là số người mắc bệnh đã chiếm đa số trong nhóm này. Nhóm béo phì ghi nhận tỷ lệ mắc bệnh tim mạch cao nhất, lên tới 62.5%, trong khi tỷ lệ người không mắc bệnh chỉ còn 37.5%.

Xu hướng trên cho thấy chỉ số BMI càng cao thì nguy cơ mắc bệnh tim mạch càng lớn, hoàn toàn phù hợp với các bằng chứng y học về mối liên hệ giữa thừa cân – béo phì và các yếu tố

nguy cơ tim mạch như tăng huyết áp, rối loạn lipid máu và kháng insulin.

Kết quả này cũng giải thích mối tương quan dương giữa biến BMI và biến mục tiêu cardio, đồng thời khẳng định rằng BMI là một đặc trưng quan trọng cần được giữ lại trong mô hình dự đoán bệnh tim mạch.

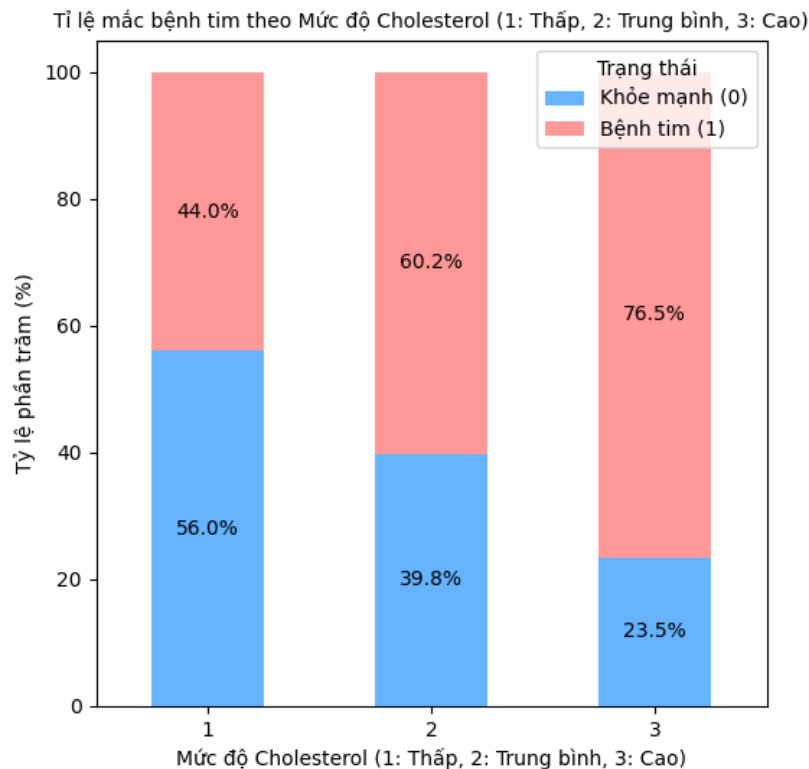


Figure 15: Tỷ lệ mắc bệnh tim theo mức độ Cholesterol

- **Trục X:** Mức độ cholesterol.
- **Trục Y:** Tỷ lệ phần trăm bệnh nhân khỏe mạnh và mắc bệnh tim mạch.

Biểu đồ cho thấy tỷ lệ mắc bệnh tim mạch tăng rõ rệt theo mức độ cholesterol. Cụ thể, ở nhóm cholesterol thấp, tỷ lệ mắc bệnh tim mạch chiếm khoảng 44%. Đối với nhóm cholesterol trung bình, tỷ lệ này tăng lên đáng kể, đạt khoảng 60.2%. Đáng chú ý, ở nhóm cholesterol cao, tỷ lệ mắc bệnh tim mạch tăng mạnh và đạt tới 76.5%.

Kết quả này cho thấy cholesterol là một yếu tố nguy cơ quan trọng đối với bệnh tim mạch. Mức cholesterol càng cao thì khả năng mắc bệnh tim mạch càng lớn, qua đó khẳng định vai trò cần thiết của biến cholesterol trong các mô hình dự đoán nguy cơ bệnh tim mạch.

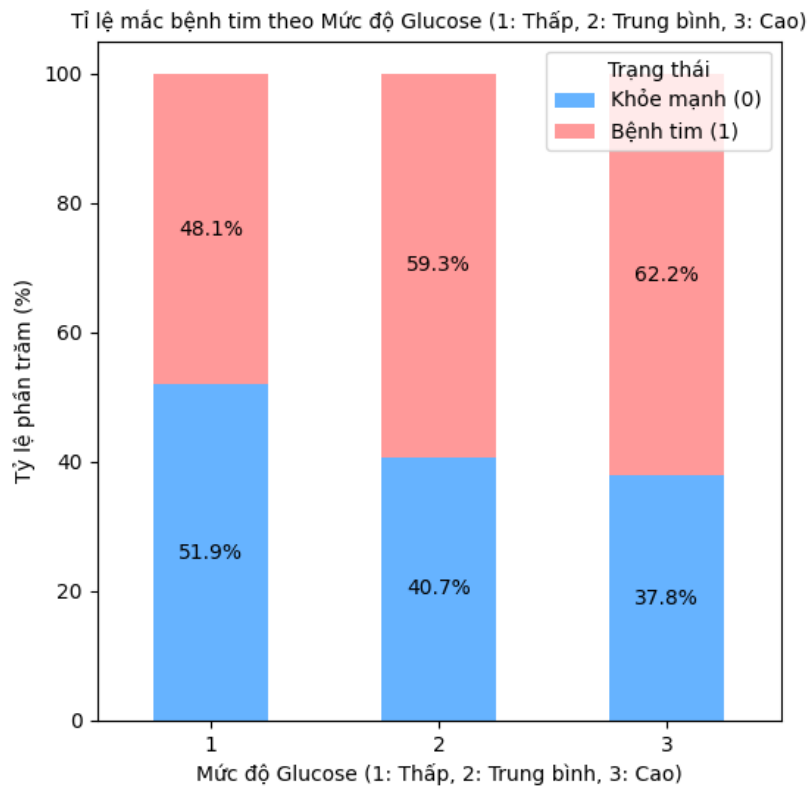


Figure 16: Tỷ lệ mắc bệnh tim theo mức độ Glucose

- **Trục X:** Mức độ glucose trong máu.
- **Trục Y:** Tỷ lệ phần trăm bệnh nhân khỏe mạnh và mắc bệnh tim mạch.

Biểu đồ cho thấy tỷ lệ mắc bệnh tim mạch tăng dần theo mức độ glucose. Cụ thể, ở nhóm glucose thấp, tỷ lệ mắc bệnh tim mạch vào khoảng 48.1%, tăng lên 59.3% ở mức trung bình và đạt 62.2% ở mức cao. Xu hướng này cho thấy đường huyết cao có mối liên hệ với nguy cơ mắc bệnh tim mạch. Tuy nhiên, mức độ gia tăng theo glucose không quá đột biến khi so sánh với biến cholesterol.

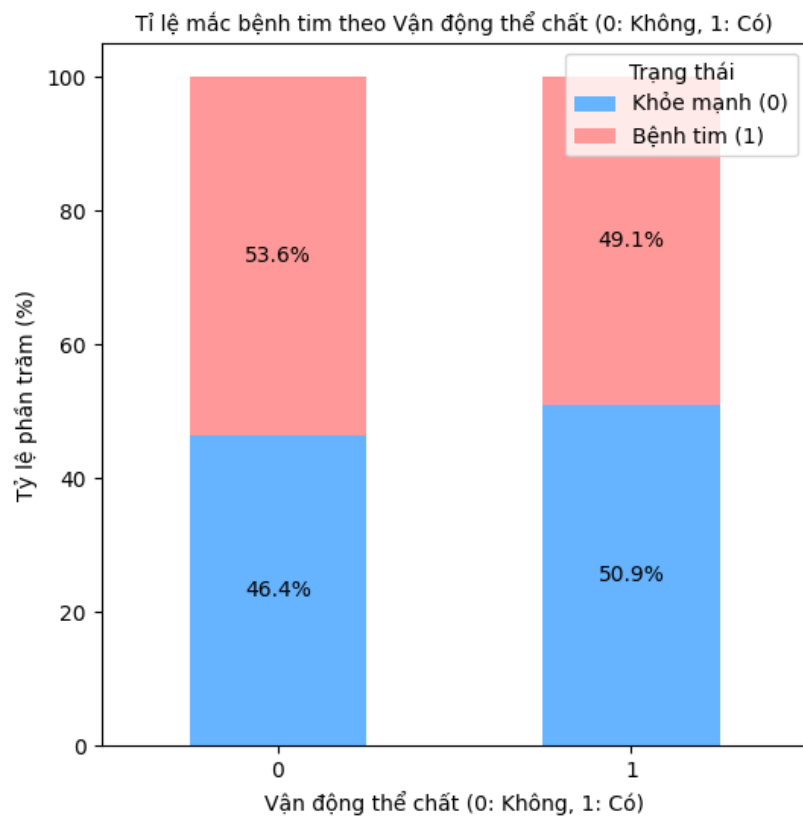


Figure 17: Tỷ lệ mắc bệnh tim mạch theo vận động thể chất

- **Trục X:** Vận động thể chất.
- **Trục Y:** Tỷ lệ phần trăm bệnh nhân khỏe mạnh và mắc bệnh tim mạch.

Biểu đồ cho thấy nhóm không vận động thể chất có tỷ lệ mắc bệnh tim mạch cao hơn, khoảng 53.6%, so với nhóm có vận động thể chất, với tỷ lệ khoảng 49.1%. Mặc dù mức chênh lệch giữa hai nhóm không quá lớn, kết quả này vẫn cho thấy vận động thể chất có tác dụng tích cực trong việc giảm nguy cơ mắc bệnh tim mạch.

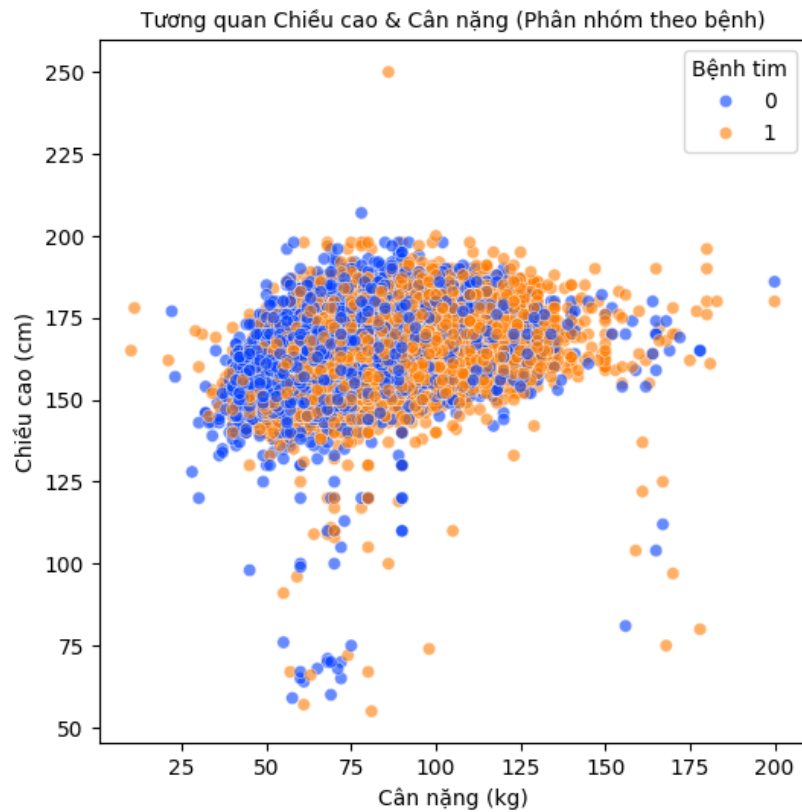


Figure 18: Tương quan chiều cao và cân nặng (Phân nhóm bệnh tim)

Biểu đồ phân tán cho thấy chiều cao và cân nặng có mối quan hệ thuận, trong đó khi cân nặng tăng thì chiều cao có xu hướng tăng theo. Điều này được thể hiện rõ qua cụm điểm dữ liệu tập trung chủ yếu ở vùng chiều cao trung bình và cân nặng trung bình. Các điểm dữ liệu của hai nhóm mắc bệnh tim và không mắc bệnh tim phân bố xen kẽ, không hình thành ranh giới tách biệt rõ ràng, cho thấy chiều cao và cân nặng khi xét riêng lẻ chưa đủ khả năng phân biệt tình trạng bệnh tim mạch.

Tuy nhiên, nhóm mắc bệnh tim có xu hướng xuất hiện nhiều hơn ở vùng cân nặng cao, đặc biệt trong các trường hợp chiều cao không tăng tương ứng. Điều này phản ánh vai trò của thể trạng và tình trạng thừa cân – béo phì trong việc làm gia tăng nguy cơ mắc bệnh tim mạch. Ngoài ra, biểu đồ vẫn xuất hiện một số điểm ngoại lai với giá trị cân nặng hoặc chiều cao bất thường, có thể bắt nguồn từ sai số đo lường hoặc các trường hợp cá biệt cần được xem xét và xử lý trong bước tiền xử lý dữ liệu.

Nhìn chung, kết quả phân tích cho thấy chiều cao và cân nặng chỉ đóng vai trò hỗ trợ trong việc dự đoán bệnh tim mạch. Để nâng cao khả năng phân biệt và độ chính xác của mô hình, cần kết hợp thêm các biến có tính thông tin cao hơn như chỉ số BMI, huyết áp và các chỉ số sinh hóa.

6 Triển khai và huấn luyện mô hình

Môi trường lập trình:

- Ngôn ngữ lập trình: Python được sử dụng làm ngôn ngữ chính để triển khai các mô hình học máy và học sâu.
- Môi trường phát triển: Sử dụng Google Colab để viết và chạy mã nguồn.

6.1 Thư viện học máy sử dụng

scikit-learn:

- StandardScaler, OneHotEncoder, OrdinalEncoder: Chuẩn hóa dữ liệu đầu vào và đầu ra.
- train_test_split: Chia dữ liệu thành các tập huấn luyện, kiểm tra, và validation.
- LogisticRegression, RandomForestClassifier, XGBClassifier: Các mô hình LR, RF, XGB.
- MultiOutputRegressor: Hỗ trợ dự đoán đa đầu ra.
- GridSearchCV, KFold, RandomizedSearchCV, Optuna: Tối ưu hóa siêu tham số và đánh giá chéo.
- accuracy_score, balanced_accuracy_score, precision_score, recall_score, f1_score, roc_auc_score, roc_curve, log_loss, brier_score_loss: Đánh giá hiệu suất mô hình.
- Pipeline, ColumnTransformer: Quản lý tiền xử lý theo cột và huấn luyện mô hình trong một quy trình thống nhất.

6.2 Quá trình triển khai và huấn luyện mô hình

Đầu vào và đầu ra:

- Đầu vào (X): Tất cả các cột trong DataFrame df trừ cột target là cardio.
- Đầu ra (y): Cột target cardio.

Chuẩn hóa dữ liệu:

- Sử dụng StandardScaler để chuẩn hóa cả đầu vào (X) và đầu ra (y) nhằm đưa dữ liệu về phân phối chuẩn (mean = 0, std = 1).

Chia dữ liệu:

Dữ liệu được chia thành ba tập:

- Tập huấn luyện (train): 80% dữ liệu.
- Tập kiểm tra (test): 20% dữ liệu gốc.

Sử dụng `train_test_split` với `random_state = 42` để đảm bảo tính tái lập.

6.3 LogisticRegression

Mô hình: `LogisticRegression`

Tối ưu hóa: `GridSearchCV`

Tham số:

- `penalty`: ['l1', 'l2']
- `C`: [0.001, 0.01, 0.1, 1, 10, 100]
- `class_weight`: [None, 'balanced']

Cấu hình cố định:

- `solver` = 'liblinear'
- `max_iter` = 1000

Đánh giá:

- `scoring` = 'f1' (threshold mặc định 0.5)
- Cross-validation: 5-fold

6.4 RandomForestClassifier

Mô hình: `RandomForestClassifier`

Tối ưu hóa: `RandomizedSearchCV`

Tham số:

- `n_estimators`: [100, 300, 500, 800, 1000]
- `max_depth`: [5, 10, 20, 30, None]
- `min_samples_split`: [2, 5, 10, 20]
- `min_samples_leaf`: [1, 5, 10, 20]

- max_features: ['sqrt', 'log2']
- bootstrap: [True, False]
- class_weight: [None, 'balanced']

Đánh giá:

- Cross-validation: 5-fold

6.5 XGBClassifier

Mô hình: XGBClassifier

Tối ưu hóa: Optuna

Tham số:

- n_estimators: [300, 1000]
- max_depth: [3, 20]
- learning_rate: [0.01, 0.15]
- subsample: [0.6, 1.0]
- colsample_bytree: [0.6, 1.0]
- gamma: [0, 5]
- min_child_weight: [1, 10]
- reg_alpha: [0, 5]
- reg_lambda: [1, 10]
- scale_pos_weight: [0.5, 2.0]

Cấu hình cố định:

- objective = 'binary:logistic'
- eval_metric = 'logloss'
- early_stopping_rounds = 50

6.6 Đánh giá và trực quan hóa

- Accuracy: Tỷ lệ dự đoán đúng tổng thể.
- Precision: Đánh giá độ chính xác của các dự đoán dương tính, phản ánh khả năng hạn chế báo động giả (False Positive).
- Recall (Sensitivity): Đo khả năng phát hiện đúng các trường hợp dương tính (False Negative).
- F1-score: Trung bình điều hòa giữa Precision và Recall; phù hợp cho bài toán dữ liệu mất cân bằng.
- ROC–AUC: Đánh giá khả năng phân biệt hai lớp tổng quát, không phụ thuộc vào ngưỡng phân loại.
- Log-loss: Đo chất lượng xác suất dự đoán; giá trị càng nhỏ thể hiện mô hình dự đoán càng tốt.
- Confusion Matrix: Phân tích chi tiết các loại lỗi TP, TN, FP, FN, hỗ trợ đánh giá và so sánh mô hình.

Trực quan hóa:

- Confusion Matrix
- Biểu đồ loss: ROC và AUC
- Biểu đồ cho các Feature Importance

6.7 Kết quả huấn luyện mô hình học máy

Mô hình Logistic Regression.

Mô hình Logistic Regression được huấn luyện trên bộ dữ liệu đã được chia thành tập huấn luyện và tập kiểm tra theo tỷ lệ 80%–20%. Quá trình tối ưu siêu tham số được thực hiện bằng GridSearchCV nhằm lựa chọn cấu hình mô hình phù hợp nhất. Sau đó, mô hình tiếp tục được hiệu chỉnh ngưỡng phân loại (threshold) để tối ưu hiệu suất trên tập kiểm tra.

Trong bối cảnh bài toán y tế, chỉ số *Recall* được ưu tiên do hệ quả nghiêm trọng của lỗi âm tính giả (False Negative), tức trường hợp bệnh nhân mắc bệnh nhưng bị dự đoán là không mắc bệnh. So với lỗi dương tính giả (False Positive), loại sai lệch này gây ảnh hưởng lớn hơn đến an toàn và sức khỏe của bệnh nhân. Vì vậy, Recall được lựa chọn làm chỉ số đánh giá chính trong quá trình lựa chọn ngưỡng phân loại.

Kết quả hiệu chỉnh cho thấy ngưỡng phân loại tối ưu là 0.35, tại đó mô hình đạt được Recall cao (0.82) đồng thời vẫn kiểm soát được mức độ báo động giả, thể hiện qua Precision đạt 0.68. Chỉ số F1-score đạt 0.73, cho thấy sự cân bằng hợp lý giữa khả năng phát hiện bệnh và độ chính xác của dự đoán. Bên cạnh đó, chỉ số Balanced Accuracy đạt 0.71, phản ánh hiệu suất ổn định của mô hình trong điều kiện dữ liệu mất cân bằng.

Chỉ số	Accuracy	Balanced Acc.	Precision	Recall	F1-score
Giá trị	0.71	0.71	0.68	0.82	0.73

Table 2: Kết quả mô hình Logistic Regression trên tập kiểm tra (Threshold = 0.35)

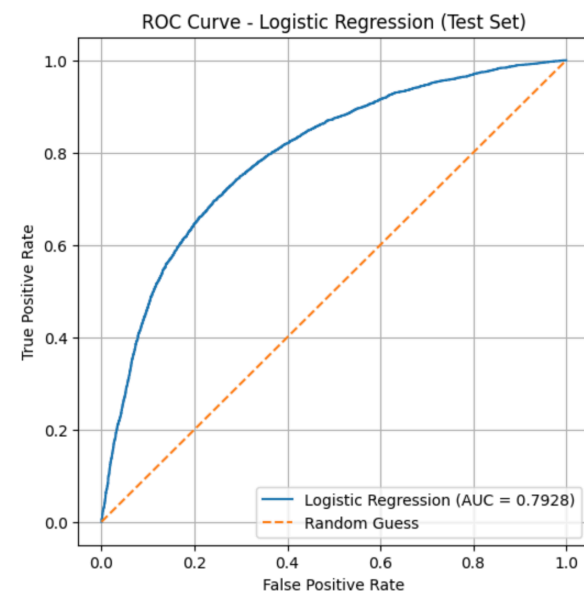


Figure 19: Đường cong ROC của mô hình Logistic Regression

Đường cong ROC cho thấy mô hình Logistic Regression đạt giá trị AUC = 0.7928, phản ánh khả năng phân biệt giữa hai lớp ở mức khá tốt. Kết quả này cho thấy mô hình có năng lực tổng quát trong việc xếp hạng xác suất bệnh nhân mắc và không mắc bệnh, độc lập với ngưỡng phân loại cụ thể.

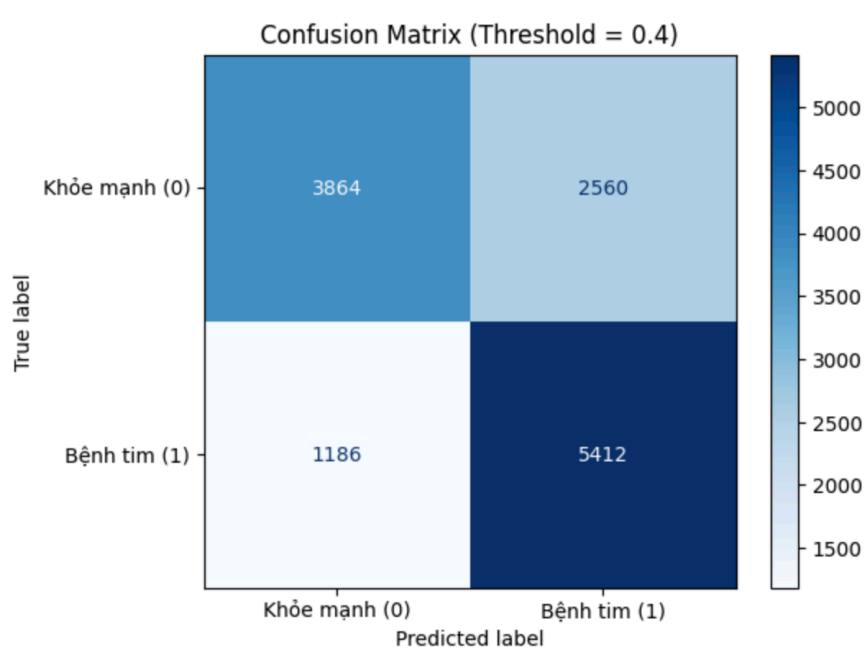


Figure 20: Ma trận nhầm lẫn của mô hình Logistic Regression tại threshold = 0.40

Ma trận nhầm lẫn cho thấy mô hình đạt số lượng lớn dự đoán đúng cho cả hai lớp. Cụ thể, mô hình dự đoán chính xác 5 412 trường hợp mắc bệnh (True Positive) và 3 864 trường hợp không mắc bệnh (True Negative).

Số lượng âm tính giả (False Negative) là 1 186 trường hợp, cho thấy mặc dù mô hình đã được hiệu chỉnh nhằm ưu tiên chỉ số Recall, vẫn còn tồn tại một tỷ lệ nhất định các ca bệnh bị bỏ sót. Tuy nhiên, mức độ sai lệch này được xem là chấp nhận được trong bối cảnh sàng lọc y tế, nơi mục tiêu chính là phát hiện phần lớn các trường hợp có nguy cơ.

Ngược lại, số lượng dương tính giả (False Positive) là 2 560 trường hợp, phản ánh chiến lược đánh đổi giữa độ chính xác và khả năng phát hiện bệnh. Việc chấp nhận số lượng báo động giả cao hơn giúp giảm thiểu nguy cơ bỏ sót bệnh nhân mắc bệnh, đồng thời phù hợp với yêu cầu an toàn trong các ứng dụng y sinh, nơi hậu quả của lỗi âm tính giả nghiêm trọng hơn lỗi dương tính giả.

Mô hình Random Forest

Mô hình Random Forest được huấn luyện trên bộ dữ liệu sau khi chia thành tập huấn luyện và tập kiểm tra theo tỷ lệ 80%–20%. Quá trình lựa chọn siêu tham số được thực hiện thông qua phương pháp RandomizedSearchCV nhằm tìm ra cấu hình tối ưu, giúp cân bằng giữa khả năng tổng quát hóa và nguy cơ quá khớp. Sau đó, mô hình tiếp tục được hiệu chỉnh ngưỡng phân loại (threshold) để phù hợp với mục tiêu của bài toán.

Trong bối cảnh bài toán y tế, chỉ số *Recall* được ưu tiên do hậu quả nghiêm trọng của lỗi âm tính giả (False Negative), tức trường hợp bệnh nhân thực sự mắc bệnh nhưng bị mô hình dự đoán là không mắc bệnh. So với lỗi dương tính giả (False Positive), loại sai lệch này tiềm ẩn

rủi ro cao hơn đối với sức khỏe bệnh nhân, do có thể làm chậm trễ quá trình chẩn đoán và điều trị. Vì vậy, Recall được lựa chọn làm tiêu chí chính trong quá trình hiệu chỉnh ngưỡng phân loại.

Kết quả hiệu chỉnh cho thấy ngưỡng phân loại tối ưu là 0.35. Tại ngưỡng này, mô hình đạt Recall = 0.85, thể hiện khả năng phát hiện phần lớn các trường hợp mắc bệnh. Đồng thời, Precision đạt 0.65, cho thấy tỷ lệ báo động giả vẫn được kiểm soát ở mức chấp nhận được. Chỉ số F1-score đạt 0.74 phản ánh sự cân bằng tương đối tốt giữa Recall và Precision, trong khi Balanced Accuracy đạt 0.69 cho thấy mô hình duy trì hiệu suất ổn định trong điều kiện dữ liệu mất cân bằng.

Table 3: Kết quả đánh giá mô hình Random Forest trên tập kiểm tra (Threshold = 0.35)

Chỉ số	Accuracy	Balanced Acc.	Precision	Recall	F1-score	ROC-AUC
Giá trị	0.70	0.69	0.65	0.85	0.74	0.79

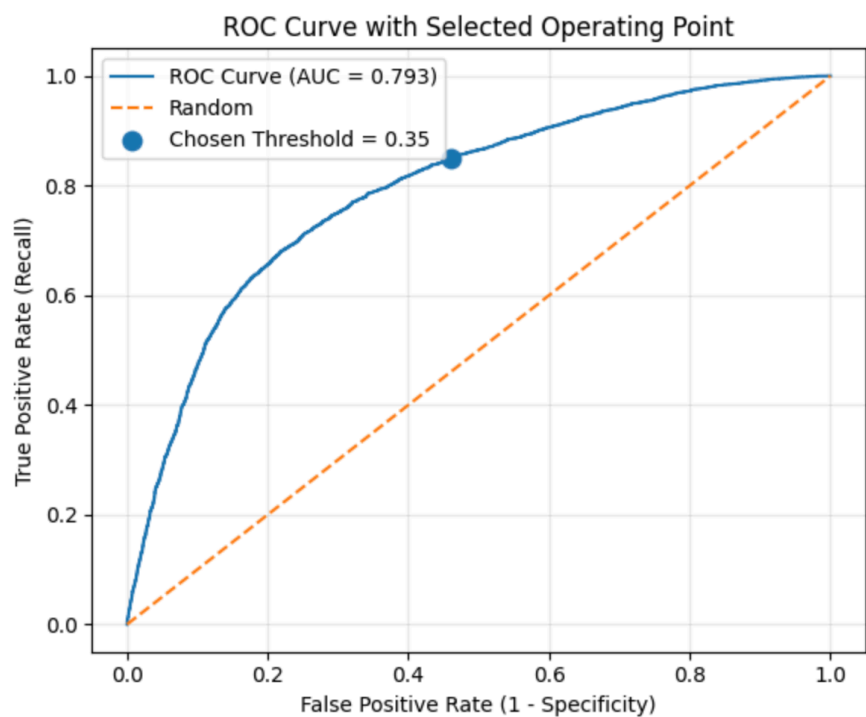


Figure 21: Đường cong ROC của mô hình Random Forest

Đường cong ROC cho thấy mô hình Random Forest đạt giá trị AUC = 0.793, phản ánh khả năng phân biệt giữa hai lớp ở mức khá tốt. Điều này cho thấy mô hình có năng lực tổng quát trong việc xếp hạng xác suất bệnh nhân mắc và không mắc bệnh, độc lập với ngưỡng phân loại cụ thể.

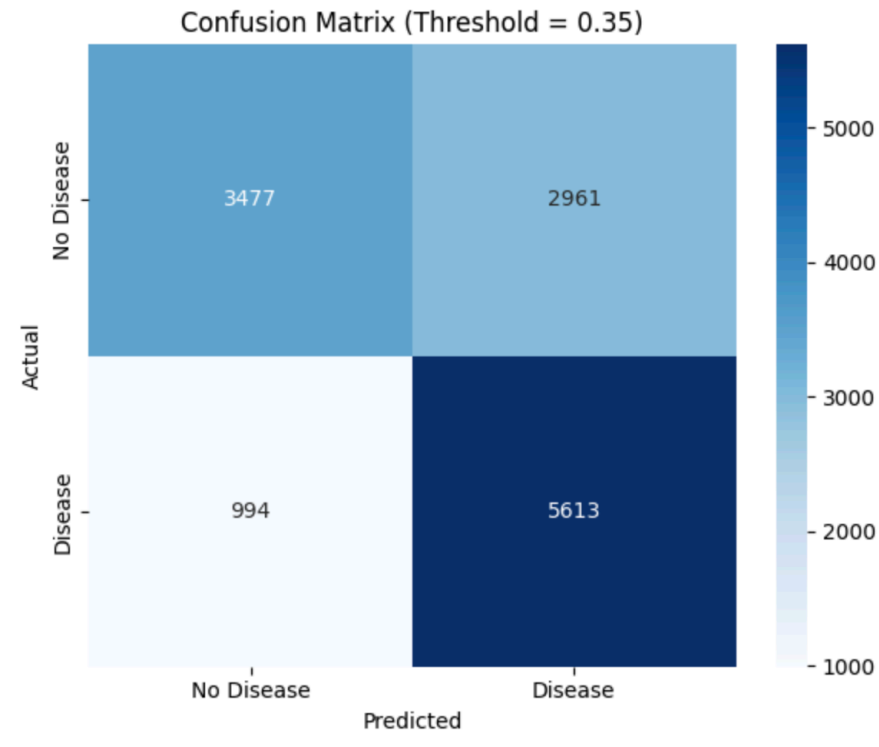


Figure 22: Ma trận nhầm lẫn của mô hình Random Forest tại threshold = 0.35

Ma trận nhầm lẫn cho thấy mô hình dự đoán đúng 5 613 trường hợp mắc bệnh (True Positive) và 3 477 trường hợp không mắc bệnh (True Negative). Số lượng âm tính giả là 994 trường hợp, phản ánh rằng mặc dù mô hình ưu tiên Recall, vẫn tồn tại một tỷ lệ nhất định các ca bệnh bị bỏ sót. Tuy nhiên, so với tổng số bệnh nhân mắc bệnh, tỷ lệ này vẫn ở mức chấp nhận được trong bối cảnh sàng lọc y tế.

Bên cạnh đó, mô hình ghi nhận 2 961 trường hợp dương tính giả, cho thấy Random Forest có xu hướng đánh đổi độ chính xác để tăng khả năng phát hiện bệnh. Chiến lược đánh đổi này phù hợp với mục tiêu của bài toán, nơi việc phát hiện sớm và hạn chế bỏ sót ca bệnh được ưu tiên hơn so với việc giảm số lượng báo động giả. **Mô hình XGBoost**

Mô hình XGBoost được huấn luyện trên bộ dữ liệu được chia theo tỷ lệ 80% cho tập huấn luyện và 20% cho tập kiểm tra. Trong quá trình huấn luyện, 20% dữ liệu của tập huấn luyện tiếp tục được tách ra làm tập xác thực (validation) nhằm phục vụ cho việc theo dõi hiệu suất và tối ưu mô hình. Việc lựa chọn siêu tham số được thực hiện bằng phương pháp Optuna, giúp tìm ra cấu hình tối ưu một cách hiệu quả trong không gian tham số lớn của XGBoost.

Đối với bộ dữ liệu ban đầu chỉ bao gồm các đặc trưng cơ bản, mô hình cho kết quả chưa thực sự cao và thiếu ổn định. Do đó, trong nghiên cứu này, một bước *feature engineering* đã được thực hiện nhằm bổ sung các đặc trưng mang ý nghĩa y sinh, bao gồm: *pulse_pressure*, *MAP*, *BMI_class*, *BP_class*, *hypertension* và *metabolic_risk*. Các đặc trưng này được xây dựng dựa trên các chỉ số sinh lý quan trọng, giúp mô hình khai thác tốt hơn mối quan hệ tiềm

ảnh giữa các yếu tố nguy cơ và tình trạng bệnh. Thực nghiệm cho thấy mô hình huấn luyện trên bộ dữ liệu đã được mở rộng đặc trưng cho hiệu suất cao hơn và ổn định hơn so với dữ liệu gốc, do đó được lựa chọn cho các phân tích tiếp theo.

Kết quả hiệu chỉnh ngưỡng phân loại cho thấy threshold tối ưu là 0.35. Tại ngưỡng này, mô hình đạt Recall = 0.83, cho thấy khả năng phát hiện phần lớn các trường hợp mắc bệnh. Đồng thời, Precision đạt 0.68, phản ánh rằng tỷ lệ báo động giả vẫn được kiểm soát ở mức chấp nhận được. Chỉ số F1-score đạt 0.75, thể hiện sự cân bằng tương đối tốt giữa Recall và Precision. Bên cạnh đó, Balanced Accuracy đạt 0.71 cho thấy mô hình duy trì hiệu suất ổn định trong điều kiện dữ liệu mất cân bằng.

Table 4: Kết quả đánh giá mô hình XGBoost trên tập kiểm tra (Threshold = 0.35)

Chỉ số	Accuracy	Balanced Acc.	Precision	Recall	F1-score
Giá trị	0.71	0.71	0.68	0.83	0.75

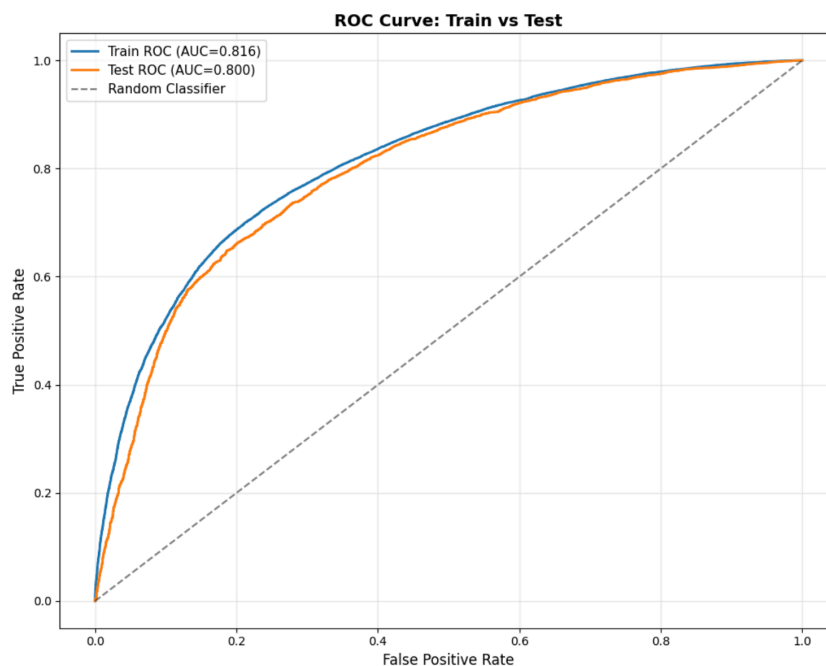


Figure 23: Đường cong ROC của mô hình XGBoost

Đường cong ROC cho thấy mô hình XGBoost đạt giá trị AUC xấp xỉ 0.80, phản ánh khả năng phân biệt giữa hai lớp ở mức khá tốt. Kết quả này cho thấy mô hình có năng lực tổng quát trong việc xếp hạng xác suất bệnh nhân mắc và không mắc bệnh, tương đối độc lập với ngưỡng phân loại được lựa chọn.

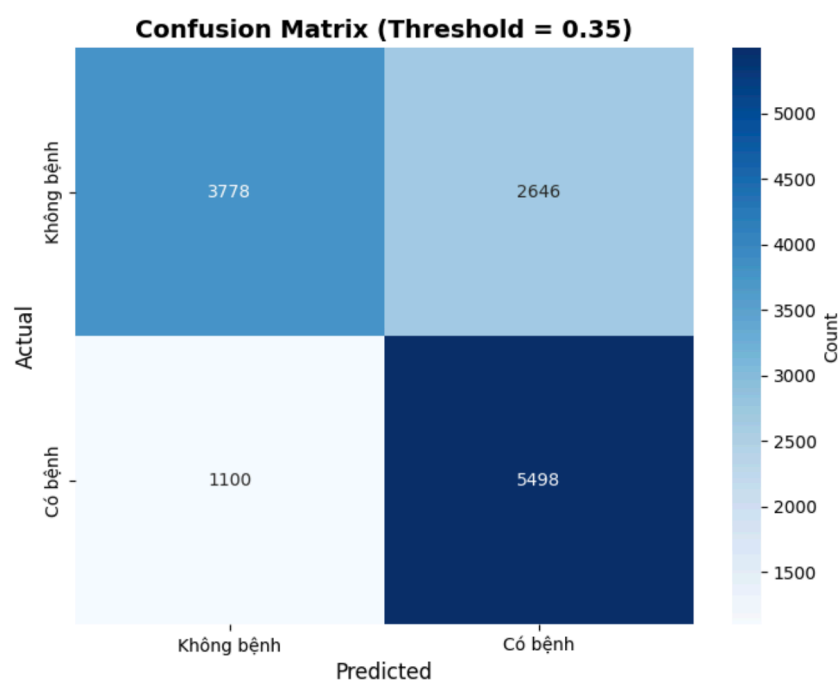


Figure 24: Ma trận nhầm lẫn của mô hình XGBoost tại threshold = 0.35

Ma trận nhầm lẫn cho thấy mô hình dự đoán đúng 5 613 trường hợp mắc bệnh (True Positive) và 3 477 trường hợp không mắc bệnh (True Negative). Số lượng âm tính giả là 994 trường hợp, cho thấy mặc dù mô hình ưu tiên Recall, vẫn tồn tại một số ca bệnh bị bỏ sót. Tuy nhiên, tỷ lệ này được xem là chấp nhận được trong bối cảnh sàng lọc y tế.

Ngoài ra, mô hình ghi nhận 2 961 trường hợp dương tính giả, phản ánh chiến lược đánh đổi giữa độ chính xác và khả năng phát hiện bệnh. Chiến lược này phù hợp với mục tiêu của bài toán, trong đó việc hạn chế bỏ sót bệnh nhân mắc bệnh được ưu tiên hơn so với việc giảm số lượng báo động giả.

7 Đánh giá và kết luận

Table 5: So sánh hiệu suất các mô hình trên tập kiểm tra

Mô hình	Accuracy	Balanced Acc.	Precision	Recall	F1-score	ROC-AUC
Logistic Regression	0.71	0.71	0.68	0.82	0.73	0.7928
Random Forest	0.70	0.69	0.65	0.85	0.74	0.79
XGBoost	0.71	0.71	0.68	0.83	0.75	0.8

So sánh chi tiết hiệu suất các mô hình

Bảng kết quả cho thấy ba mô hình Logistic Regression, Random Forest và XGBoost đều đạt hiệu suất tốt khi được hiệu chỉnh ngưỡng phân loại nhằm duy trì Recall lớn hơn 0.8. Tuy nhiên, mỗi mô hình thể hiện những đặc điểm khác nhau về khả năng cân bằng giữa độ chính xác, khả năng phát hiện bệnh và tính ổn định trong bối cảnh dữ liệu mất cân bằng.

Mô hình Logistic Regression (threshold = 0.4) đạt Accuracy và Balanced Accuracy đều bằng 0.71, cho thấy hiệu suất tương đối đồng đều trên cả hai lớp. Với Recall = 0.82 và Precision = 0.68, mô hình duy trì khả năng phát hiện phần lớn các ca bệnh trong khi vẫn kiểm soát được mức độ báo động giả ở mức chấp nhận được. Ưu điểm nổi bật của Logistic Regression nằm ở tính đơn giản, khả năng diễn giải cao và sự ổn định, khiến mô hình này phù hợp với các hệ thống hỗ trợ quyết định lâm sàng yêu cầu tính minh bạch.

Random Forest đạt Recall cao nhất trong ba mô hình (0.85), cho thấy khả năng phát hiện bệnh nhân mắc bệnh là tốt nhất. Tuy nhiên, Precision chỉ đạt 0.65 và Balanced Accuracy đạt 0.69, phản ánh việc mô hình phải đánh đổi bằng số lượng báo động giả cao hơn để đạt được mức Recall này. Điều này cho thấy Random Forest có xu hướng thiên về độ nhạy, phù hợp với các kịch bản sàng lọc ban đầu nhưng có thể làm gia tăng chi phí kiểm tra bổ sung trong thực tế lâm sàng.

XGBoost thể hiện hiệu suất tổng thể tốt nhất khi đạt F1-score cao nhất (0.75) và ROC-AUC lớn nhất (0.80). Với Recall = 0.83 và Precision = 0.68, mô hình cho thấy khả năng cân bằng hiệu quả giữa việc hạn chế bỏ sót bệnh nhân và kiểm soát báo động giả. Balanced Accuracy đạt 0.71 cho thấy mô hình duy trì hiệu suất ổn định trên cả hai lớp, ngay cả trong điều kiện dữ liệu mất cân bằng. Kết quả này phản ánh ưu thế của XGBoost trong việc học các mối quan hệ phi tuyến và khai thác hiệu quả các đặc trưng đã được thiết kế thêm thông qua quá trình feature engineering.

Tổng hợp các kết quả trên, có thể nhận thấy rằng mặc dù Random Forest đạt Recall cao nhất, XGBoost lại là mô hình có hiệu suất cân bằng và khả năng phân biệt tốt nhất, trong khi Logistic Regression đóng vai trò là mô hình nền tảng với tính diễn giải cao và hiệu suất ổn định. Việc lựa chọn mô hình cuối cùng do đó phụ thuộc vào mức độ ưu tiên giữa độ nhạy, chi phí báo động giả và yêu cầu về tính minh bạch trong ứng dụng y tế cụ thể.

References

- [1] UCI Machine Learning Repository. *Heart Disease Dataset*. University of California, Irvine, 2019.
- [2] Alizadehsani, R., et al. Machine learning-based coronary artery disease diagnosis. *Computer Methods and Programs in Biomedicine*, 2020.
- [3] Rajkomar, A., et al. Scalable and accurate deep learning for electronic health records. *npj Digital Medicine*, 2018.
- [4] Chen, T., & Guestrin, C. XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD Conference*, 2016.
- [5] Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. *Applied Logistic Regression*. Wiley, 2013.
- [6] Saito, T., & Rehmsmeier, M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 2015.
- [7] Chowdary, C. R., et al. Heart disease prediction using machine learning. *Journal of Critical Reviews*, 2020.
- [8] James, G., Witten, D., Hastie, T., & Tibshirani, R. *An Introduction to Statistical Learning*. Springer, 2013.
- [9] Fawcett, T. An introduction to ROC analysis. *Pattern Recognition Letters*, 2006.
- [10] Powers, D. M. W. Evaluation: From precision, recall and F-measure to ROC. *Journal of Machine Learning Technologies*, 2011.
- [11] Japkowicz, N., & Shah, M. *Evaluating Learning Algorithms*. Cambridge University Press, 2011.
- [12] Hand, D. J., & Till, R. J. A generalisation of the area under the ROC curve. *Machine Learning*, 2001.
- [13] Steyerberg, E. W., et al. Assessing the performance of prediction models. *Epidemiology*, 2010.
- [14] Chen, T. and Guestrin, C. (2016). *XGBoost: A scalable tree boosting system*. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), pp. 785–794.