

PREDICTION DISEASE HEART

Dinh-Phuc Pham

Faculty of Information Technology
Hung Yen University of Technology and Education

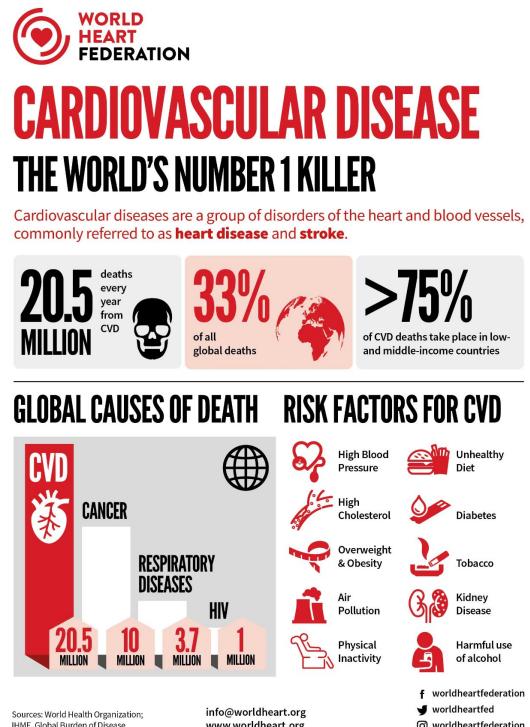
Instructor: Van-Hau Nguyen

Table of Contents

1. Introduction
2. EDA (Exploratory Data Analysis)
3. Data Preprocessing
4. Model Training (baseline models)
5. Hyperparameter Tuning
6. Evaluation on Testset
7. Demo Inference on Gradio

1. Introduction

- Bệnh tim mạch là nguyên nhân tử vong số 1 toàn cầu (~20,5M ca/năm, ~33% tổng tử vong; >75% ở nước thu nhập thấp–trung bình).
- Suy tim rất phổ biến (~26M ca; ước tính >37,7M nếu tính cả chưa chẩn đoán), nguy cơ cả đời ~1/5, và >50% tái nhập viện trong 6 tháng.
- Nhiều yếu tố nguy cơ có thể theo dõi và can thiệp (huyết áp, cholesterol, tiểu đường, BMI, hút thuốc, lối sống...). Vì vậy, dùng Machine Learning để dự đoán sớm giúp sàng lọc nhóm nguy cơ cao, ưu tiên can thiệp và giảm biến cố tim mạch

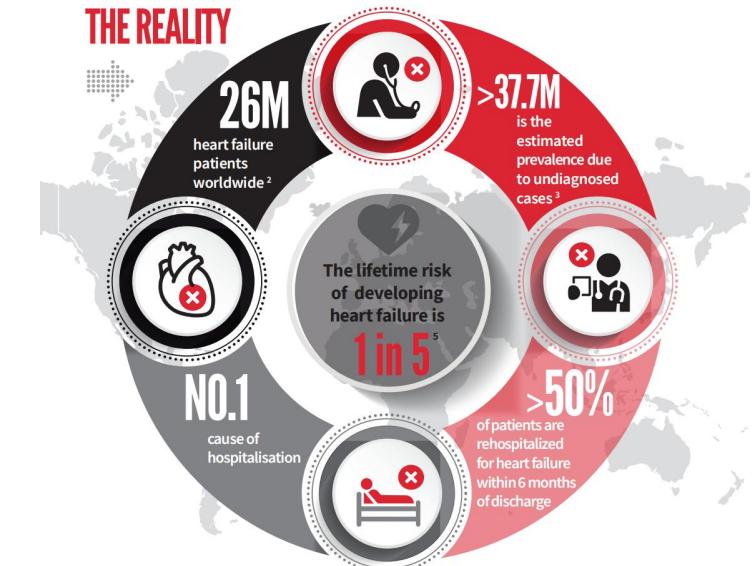


PEOPLE WITH HEART FAILURE ARE MORE VULNERABLE TO COVID-19

WHAT IS HEART FAILURE?

Heart failure is a severe failure of the heart to pump enough blood around the body
Symptoms include breathlessness, fatigue and swollen limbs

THE REALITY



Dataset

Feature	Kiểu	Ý nghĩa	Giá trị / Range (theo data)	Missing
gender	categorical	Giới tính sinh học (Male/Female).	Male, Female	0.51%
age	numeric	Tuổi (năm).	32 → 70	0.68%
education	numeric	Trình độ học vấn (mã hoá 1–4, dạng thứ bậc/ordinal).	1 → 4	0.00%
currentSmoker	categorical	Hiện tại có hút thuốc không (Yes/No).	No, Yes	0.00%
cigsPerDay	numeric	Số điếu thuốc hút mỗi ngày.	0 → 70	0.42%
BPMeds	categorical	Đang dùng thuốc huyết áp (Blood Pressure Meds) (Yes/No).	No, Yes	0.00%
prevalentStroke	categorical	Tiền sử đột quỵ (Yes/No).	No, Yes	0.00%
prevalentHyp	categorical	Tăng huyết áp (Yes/No).	No, Yes	1.17%
diabetes	categorical	Tiểu đường (Yes/No).	No, Yes	1.97%
totChol	numeric	Cholesterol toàn phần (thường mg/dL).	107 → 696	0.36%
sysBP	numeric	Huyết áp tâm thu (thường mmHg).	83.5 → 295	0.00%
diaBP	numeric	Huyết áp tâm trương (thường mmHg).	48 → 142.5	3.75%
BMI	numeric	Body Mass Index (kg/m ²).	15.54 → 56.8	0.59%
heartRate	numeric	Nhip tim (beats per minute).	44 → 143	0.44%
glucose	numeric	Đường huyết (thường mg/dL).	40 → 394	0.00%
TenYearCHD	numeric	Nhận: nguy cơ bệnh mạch vành trong 10 năm (0/1).	0 → 1	3.28%

Data split under 5-fold CV and held-out test set.

Statistic	CV = 5		Test
	Train	Val	
Count	6,809	1,702	2,128
Percent	64%	16%	20%

End-to-end ML Pipeline

Bài Toán: Classification (2 labels)

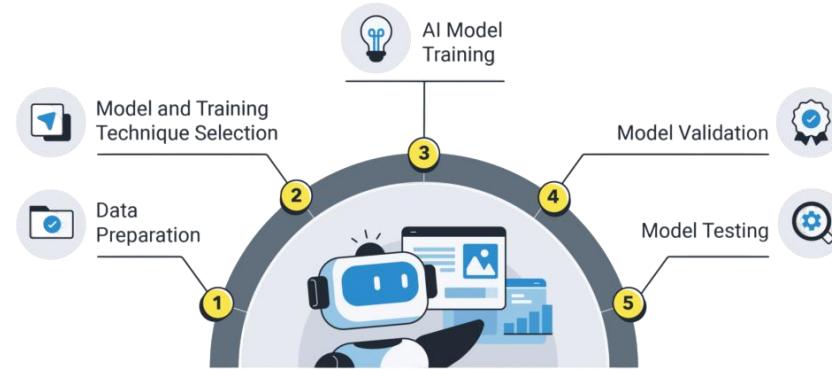
Dữ liệu:

- Phân tích dữ liệu
- Tiền xử lý dữ liệu
- Chuẩn hóa dữ liệu



Các mô hình học máy:

- Logistic Regression
- Gaussian Naive Bayes
- K-Nearest Neighbors
- Support Vector Classifier
- Decision Tree Classifier
- Random Forest Classifier
- Extra Trees Classifier
- AdaBoost Classifier
- Gradient Boosting Classifier
- Histogram Gradient Boosting Classifier
- XGBoost Classifier
- LightGBM Classifier



Đánh giá mô hình

- Confusion matrix
- Classification Metrics:
 - Accuracy
 - Precision
 - Recall
 - F1-score

2. EDA

Sample records (first 5 rows).

#	gender	age	edu	smoker	cigs	meds	stroke	hyp	diab	chol	sys	dia	BMI	HR	glu	CHD
0	Male	39.0	4.0	No	0.0	No	No	No	No	195.0	106.0	70.0	26.97	80.0	77.0	0.0
1	Female	46.0	2.0	No	0.0	No	No	No	No	250.0	121.0	81.0	28.73	95.0	76.0	0.0
2	Male	48.0	1.0	Yes	20.0	No	No	No	No	245.0	127.5	80.0	25.34	75.0	70.0	0.0
3	Female	61.0	3.0	Yes	30.0	No	No	Yes	No	225.0	150.0	95.0	28.58	65.0	103.0	1.0
4	Female	46.0	3.0	Yes	23.0	No	No	No	No	285.0	130.0	84.0	23.10	85.0	85.0	0.0

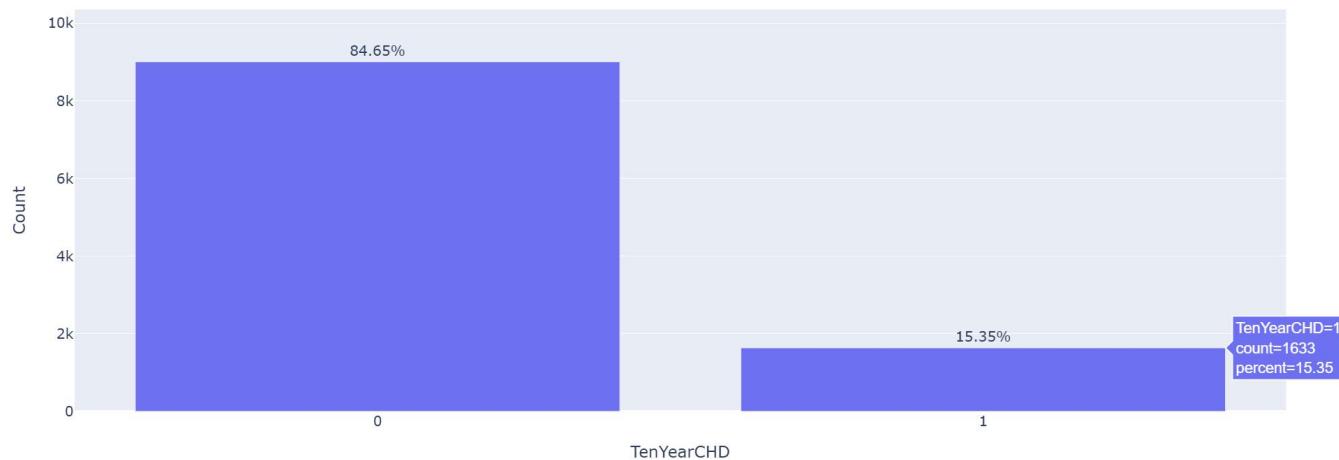
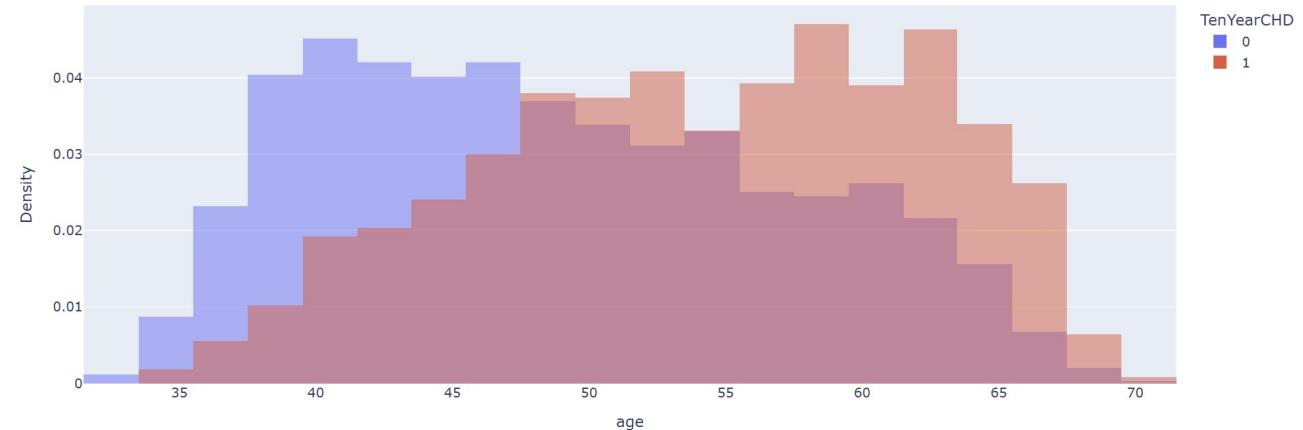
Dataset overview. The dataset contains $11,000 \times 16$ entries (samples \times variables). The first five records are shown below.

Feature data types.

Feature	Dtype
gender	object
age	float64
education	float64
currentSmoker	object
cigsPerDay	float64
BPMeds	object
prevalentStroke	object
prevalentHyp	object
diabetes	object
totChol	float64
sysBP	float64
diaBP	float64
BMI	float64
heartRate	float64
glucose	float64
TenYearCHD	float64

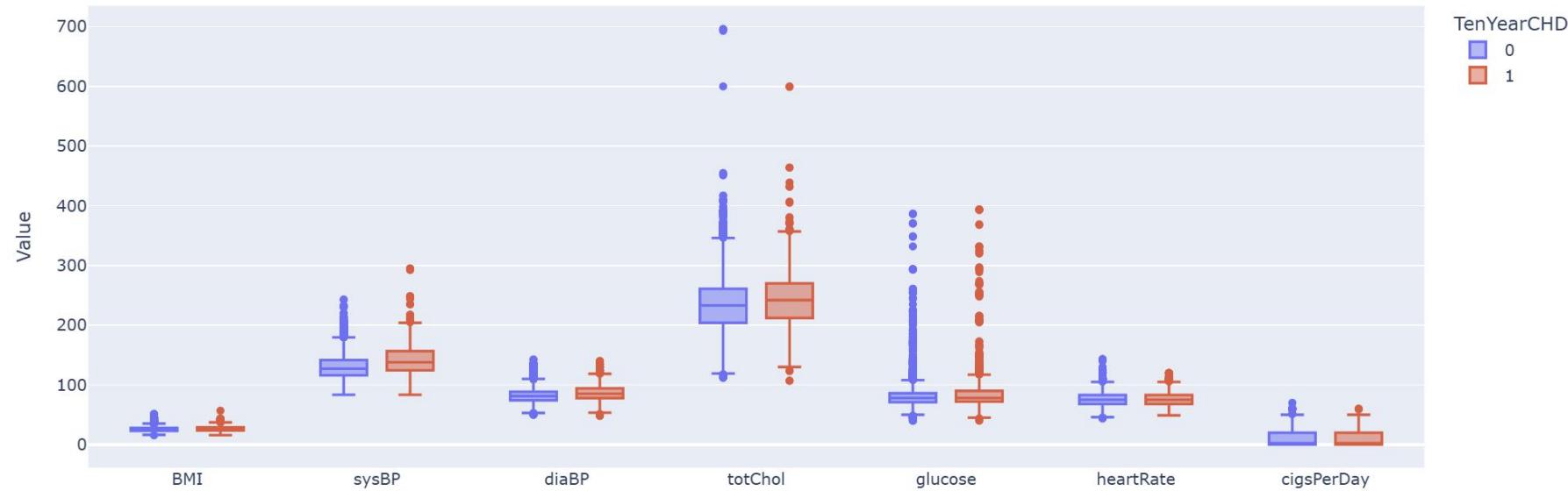
Distribution for Label

- Biểu đồ histogram so sánh phân phối tuổi giữa hai nhóm TenYearCHD=0 và TenYearCHD=1 (chồng lên nhau, trục Y là mật độ)
- Biểu đồ cột này dùng để đếm số lượng (count) và tính tỷ lệ (%) của hai lớp TenYearCHD (0/1), sau đó vẽ biểu đồ cột Plotly: trục Y là count, và % được hiển thị trên đầu mỗi cột.



Boxplot Outlier Check

- Biểu đồ boxplot dùng để so sánh các biến số (BMI, huyết áp, cholesterol, glucose, ...) giữa hai nhóm TenYearCHD = 0/1, giúp nhìn nhanh trung vị (median), độ phân tán (IQR) và phát hiện outlier (điểm bất thường ngoài "râu" boxplot).



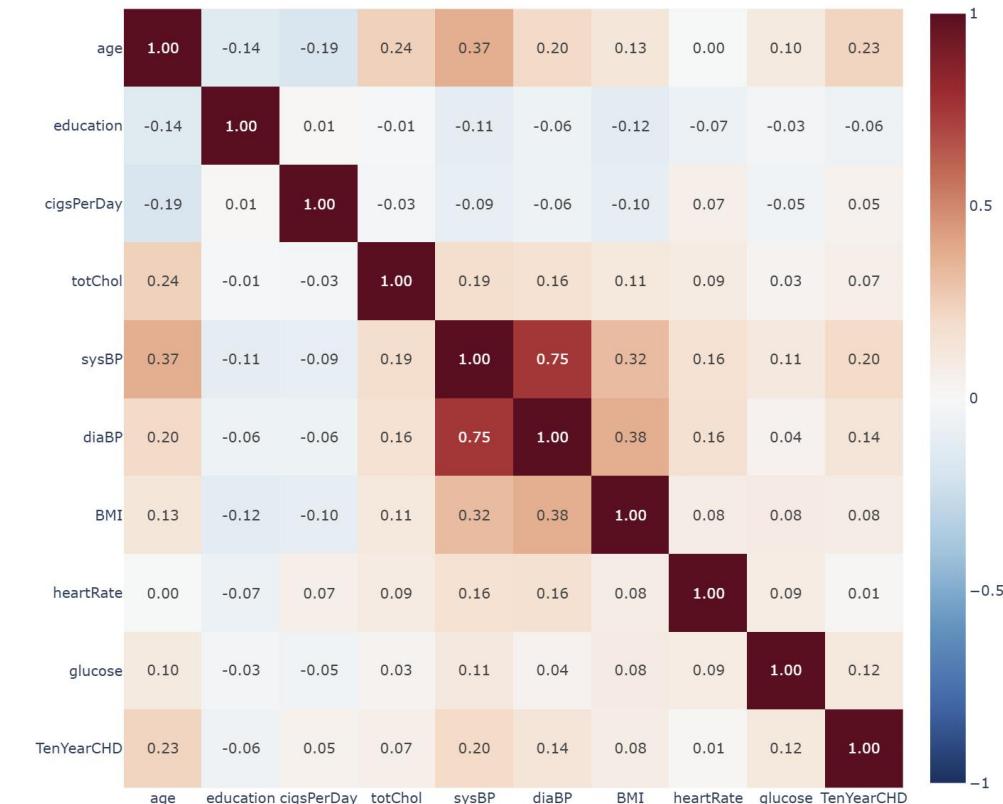
Correlation Heatmap

- Heatmap tương quan cho thấy mức độ "đi cùng nhau" giữa các biến số ($\text{corr} \in [-1, 1]$): gần +1 là cùng tăng/giảm mạnh, gần -1 là ngược chiều mạnh, gần 0 là ít liên hệ tuyến tính.
- Nó hữu ích để phát hiện các cặp feature tương quan quá cao (đa cộng tuyến) có thể cân nhắc bỏ bớt 1 biến hoặc dùng regularization (đặc biệt với mô hình tuyến tính). Heatmap chỉ để tham khảo, không nên chọn feature chỉ dựa vào correlation vì nó chủ yếu phản ánh feature-feature, không nói trực tiếp feature nào dự đoán target tốt.

$$r = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y}$$

$$\text{Cov}(X, Y) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

$$\sigma_X = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$$

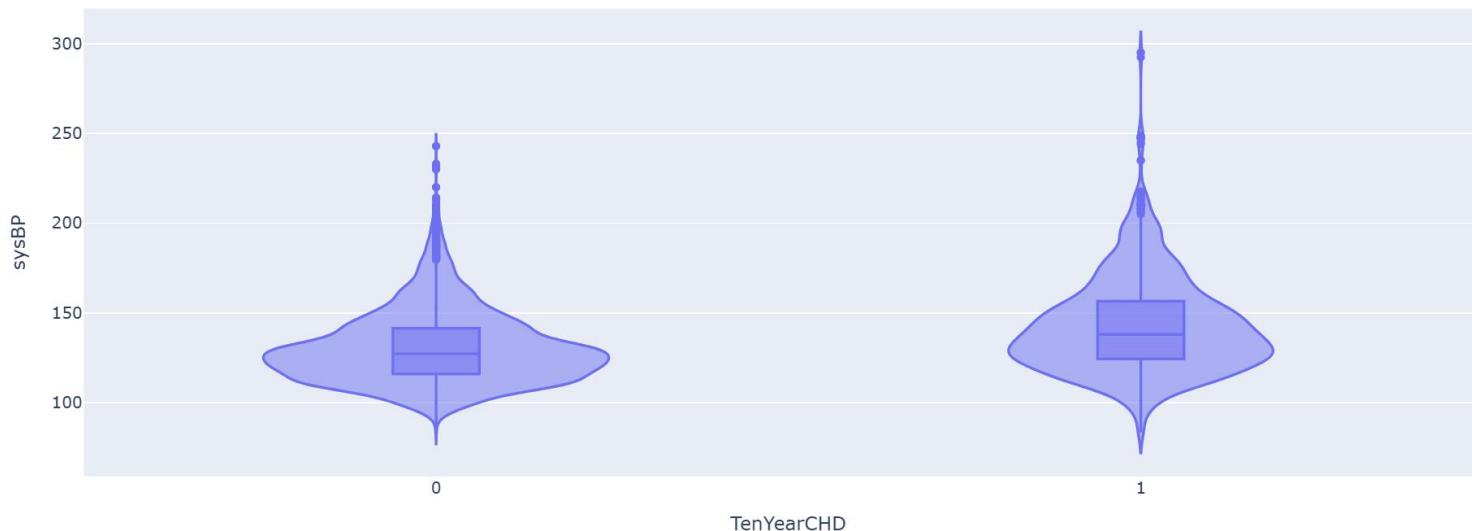


Violinplot + Boxplot

Violin plot dùng để so sánh phân phối của một biến số (ví dụ sysBP hoặc age) giữa 2 nhóm TenYearCHD = 0/1.

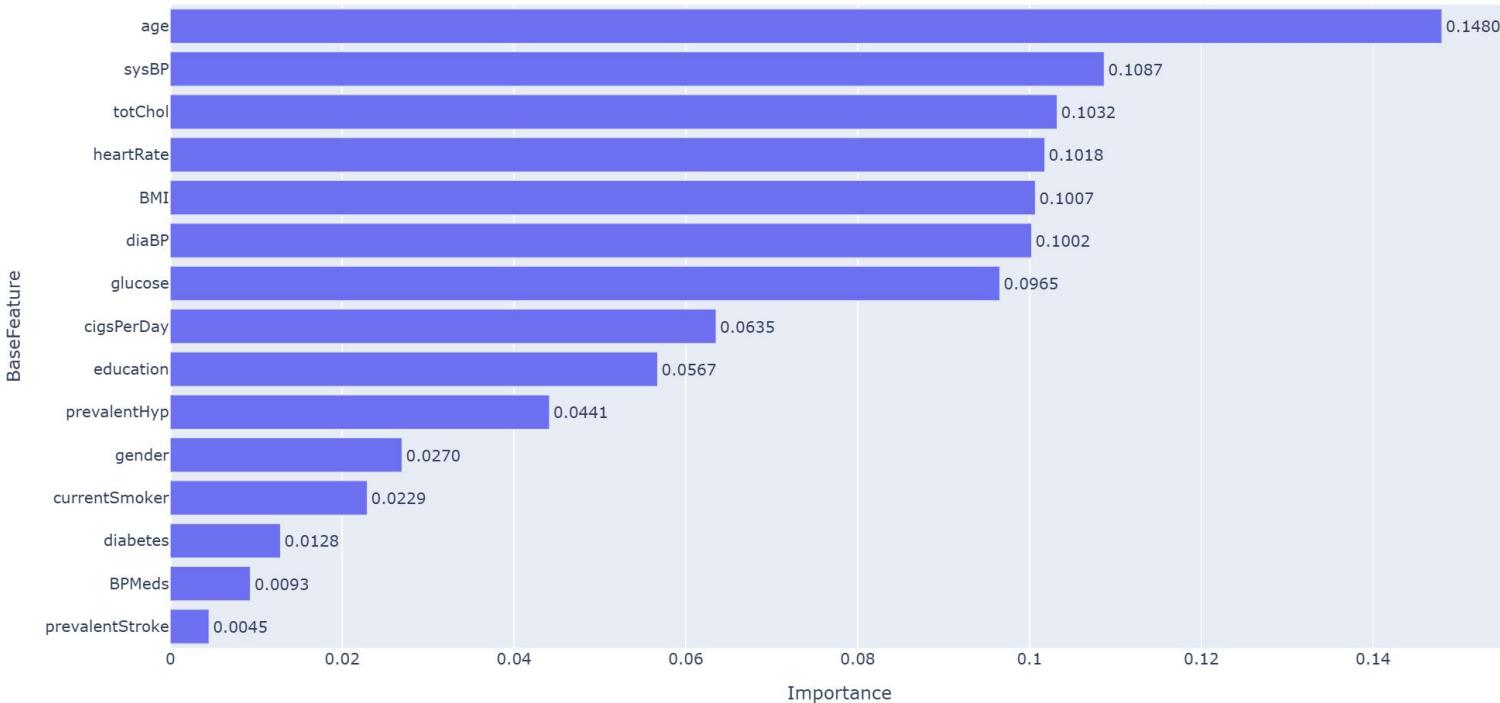
- Phần “violin” thể hiện mật độ phân phối (phình to = dữ liệu tập trung nhiều).
- Bên trong thường kèm boxplot: median (đường giữa) và IQR Q1–Q3 (hộp).
- Có thể hiện outlier nếu bật điểm.

Nhìn xem violin/box của nhóm TenYearCHD=1 có dịch lên cao hay phân tán rộng hơn so với nhóm 0 để đánh giá biến đó khác biệt như thế nào giữa 2 nhóm.



Feature Importance

- Feature importance (ExtraTrees) cho thấy mô hình dự đoán CHD chủ yếu dựa vào age (quan trọng nhất), tiếp theo là các chỉ số huyết áp (sysBP, diaBP), mỡ máu (totChol), nhịp tim, BMI và glucose. Các yếu tố như hút thuốc, học vấn, giới tính, tiền sử bệnh/thuốc có ảnh hưởng thấp hơn trong mô hình này.



3. Data Processing

StandardScaler (chuẩn hoá dữ liệu số)

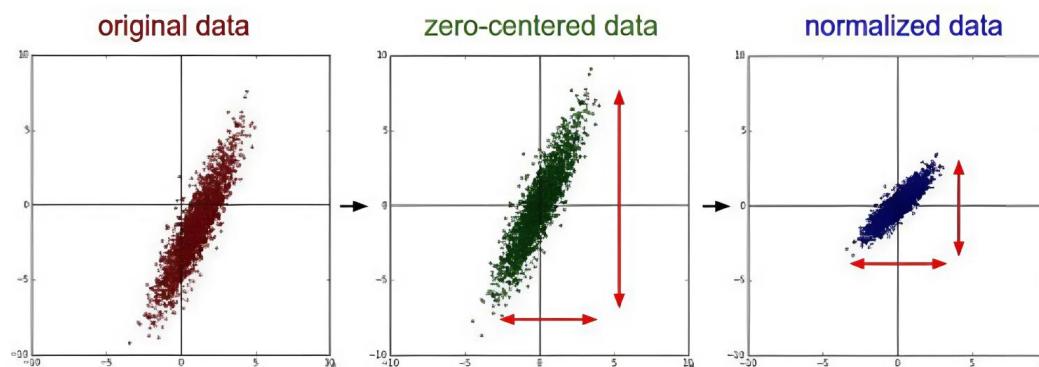
- Khái niệm: Chuẩn hoá các biến số về cùng một thang đo.
- Ý nghĩa: Giúp các feature như tuổi, huyết áp, cholesterol, BMI... không bị lệch do khác đơn vị/độ lớn, từ đó model học ổn định và công bằng hơn giữa các feature.

class_weight="balanced" (cân bằng trọng số lớp)

- Khái niệm: Gán trọng số khác nhau cho từng lớp khi huấn luyện để bù cho dữ liệu mất cân bằng.
- Ý nghĩa: Giảm thiên vị về lớp chiếm đa số ($CHD=0$), tăng khả năng phát hiện lớp hiếm ($CHD=1$), thường cải thiện Recall cho lớp dương tính.

StandardScaler (Z-score standardization)

StandardScaler là kỹ thuật chuẩn hoá theo Z-score (standardization) trong tiền xử lý dữ liệu, dùng để tuyến tính hóa từng đặc trưng số sao cho chúng có trung bình (mean) xấp xỉ 0 và độ lệch chuẩn (standard deviation) xấp xỉ 1. Mục tiêu là đưa các biến về cùng thang đo, giúp các thuật toán nhạy với độ lớn của feature



$$x'_i = \frac{x_i - \mu}{\sigma}$$

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2}$$

- x_i : giá trị gốc (raw value) của một quan sát thứ i trong một cột feature số
- μ (mean): **trung bình** của feature đó, tính trên **tập train**:
- σ (standard deviation): **độ lệch chuẩn** của feature đó, tính trên **tập train**:
- z_i : giá trị sau chuẩn hoá (standardized value).

4. Models Training [Baseline]

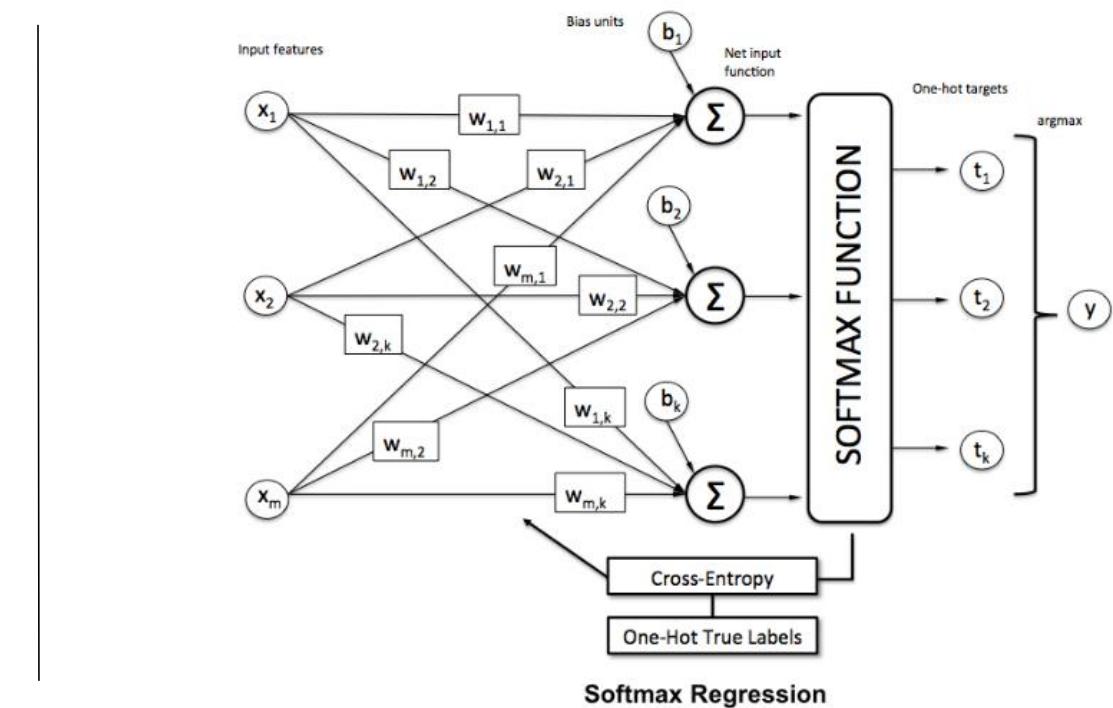
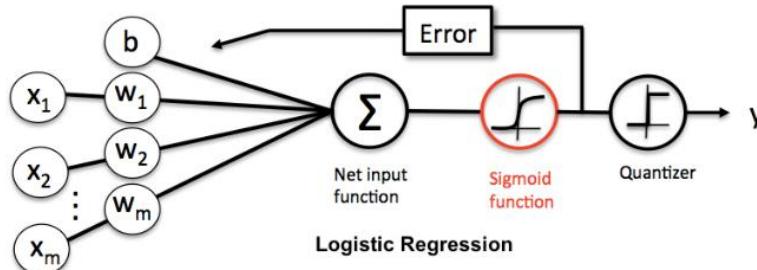
Models		
Base Models	Ensemble (Bagging)	Ensemble (Boosting)
Logistic Regression	Random Forest Classifier	AdaBoost Classifier
Gaussian Naive Bayes	Extra Trees Classifier	Gradient Boosting Classifier
K-Nearest Neighbors		Histogram Gradient Boosting
Support Vector Classifier		XGBoost Classifier
Decision Tree Classifier		LightGBM Classifier

Baseline Models

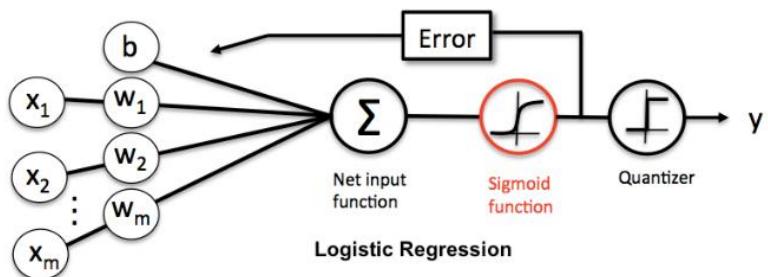
- Logistic Regression
- Gaussian Naive Bayes
- K-Nearest Neighbors
- Support Vector Classifier
- Decision Tree Classifier

Logistic Regression

Hồi quy logistic là mô hình tuyến tính được sử dụng cho các nhiệm vụ phân loại, dự đoán xác suất của một đầu vào thuộc về một trong nhiều lớp bằng cách áp dụng hàm softmax hoặc hàm sigmoid



Behind the Scenes for Lo.R



$X \xrightarrow{\text{linear combo}} z \xrightarrow{\text{sigmoid}} p \xrightarrow{\text{threshold}} \hat{y} \xrightarrow{\text{loss + gradient}} \beta \text{ update} \rightarrow \text{lặp lại}$

$$z = w^T x + b$$

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

$$\mathcal{L}(w, b) = -\frac{1}{N} \sum_{i=1}^N \left(y_i \log \hat{y}_i + (1 - y_i) \log (1 - \hat{y}_i) \right)$$

$$W := W - \alpha \cdot \frac{\partial J}{\partial W}$$

$$b := b - \alpha \cdot \frac{\partial J}{\partial b}$$

$$w \leftarrow w - \eta \frac{\partial \mathcal{L}}{\partial w}, \quad b \leftarrow b - \eta \frac{\partial \mathcal{L}}{\partial b}$$

Baseline Models: Pros & Cons

Baseline model	Ưu điểm	Nhược điểm
Logistic Regression (LR)	Nhanh, baseline mạnh; dễ giải thích (hệ số); dễ deploy	Giả định gần tuyến tính; cần scaling ; kém với quan hệ phi tuyến nếu không feature engineering
Gaussian Naive Bayes (GNB)	Rất nhanh; ít tham số; hợp dataset nhỏ	Giả định độc lập feature + Gaussian → dễ lệch thực tế; kém khi feature tương quan mạnh
K-Nearest Neighbors (KNN)	Đơn giản; học được phi tuyến; không “train” nặng	Cần scaling ; predict chậm khi dữ liệu lớn; nhạy nhiễu/outlier & chọn k ; giảm hiệu quả khi nhiều chiều
Support Vector Classifier (SVC)	Mạnh trên dataset vừa/nhỏ; kernel bắt phi tuyến tốt	Train chậm khi dữ liệu lớn; nhạy tham số (C, γ); cần scaling ; xác suất cần bật/calibration
Decision Tree	Dễ hiểu; bắt tương tác & phi tuyến; không cần scaling	Dễ overfit nếu không giới hạn; không ổn định; thường thua ensemble (RF/GBM)

Ensamble Learning

Bagging

- Cách làm: train nhiều model độc lập trên các mẫu dữ liệu bootstrap (lấy mẫu có hoán lại), rồi vote/average kết quả.
- Mục tiêu: giảm variance, ổn định mô hình, hạn chế overfitting.
- Ví dụ: **Random Forest, Extra Trees**.

Boosting

- Cách làm: train model theo chuỗi, mỗi model sau tập trung sửa lỗi của model trước (tăng trọng số cho điểm bị dự đoán sai / fit phần dư).
- Mục tiêu: giảm bias (và thường cũng giảm lỗi tổng), đạt độ chính xác cao.
- Ví dụ: **AdaBoost, Gradient Boosting, XGBoost, LightGBM, HistGB**.

Stacking

Bagging Algorithms

Ensemble Learning là một phương pháp học máy mạnh mẽ dựa trên nguyên lý kết hợp nhiều mô hình (được gọi là base learners) để tạo ra một mô hình tổng hợp có khả năng dự đoán chính xác hơn.

1) Bootstrapped Samples (Mẫu Bootstrap)

- Lấy mẫu có hoán lại từ tập train D (n quan sát) để tạo B tập con $D_1 \dots D_B$, mỗi tập kích thước vẫn là n nhưng có thể trùng phần tử.
- Tạo đa dạng dữ liệu để các mô hình cơ sở nhìn "lát cắt" khác nhau → giảm overfitting, tăng khả năng tổng quát hóa.

2) Base Learners (Mô hình cơ sở)

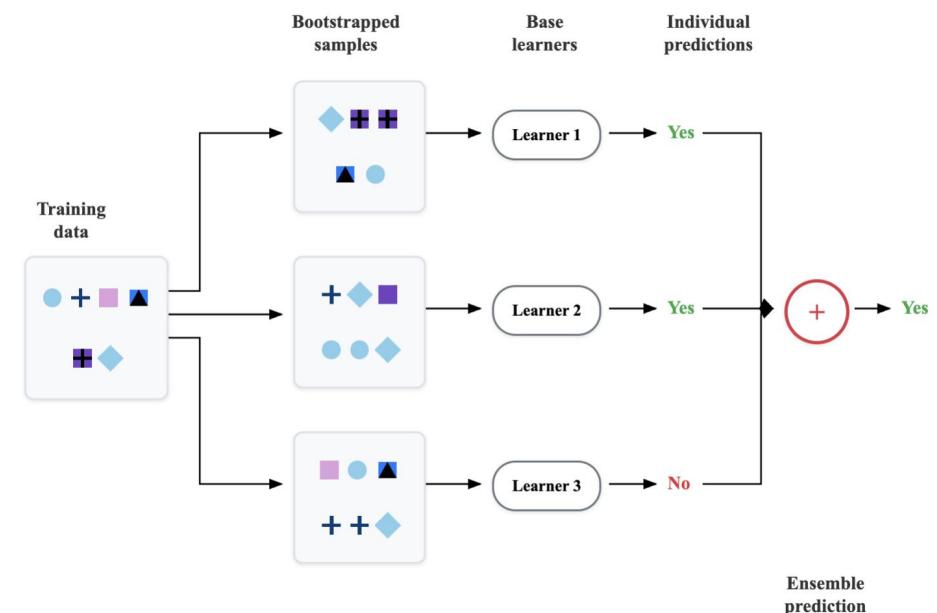
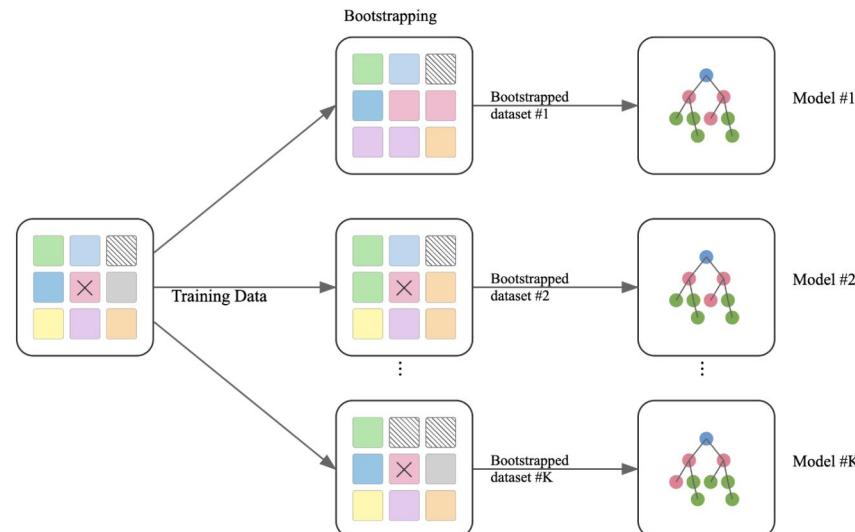
- Huấn luyện một base learner h_b trên từng mẫu bootstrap D_b .
- Base learners có thể cùng loại (vd: nhiều cây quyết định) hoặc khác loại (vd: tree + SVM + NN).

3) Individual Predictions (Dự đoán cá nhân)

- Với điểm mới x , mỗi h_b đưa ra dự đoán riêng (nhận Yes/No hoặc giá trị số).
- Các dự đoán có thể khác nhau giữa các learner do dữ liệu train khác nhau.

4) Ensemble Prediction (Dự đoán tổng hợp)

- Gộp dự đoán của các learner để ra kết quả cuối (voting/averaging).
- Cách gộp phổ biến: Hard voting (đa số), Soft voting (trung bình xác suất), hoặc Weighted voting (có trọng số theo hiệu suất).

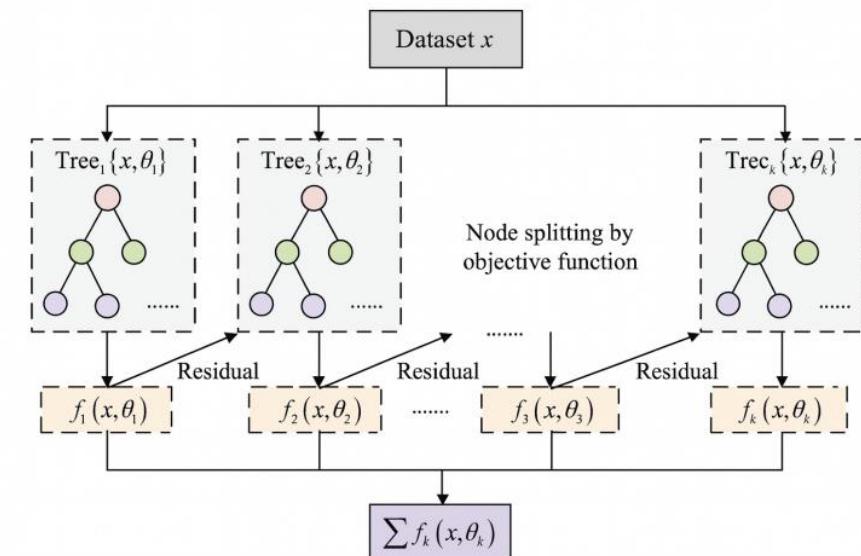


Boosting Algorithms

- Boosting là một kỹ thuật ensemble kết hợp nhiều mô hình yếu được huấn luyện theo chuỗi tuần tự để tạo ra mô hình mạnh.
- Mỗi mô hình mới được thêm vào để sửa lỗi của mô hình trước (tập trung vào mẫu khó hoặc tối ưu trực tiếp hàm măt/mát/gradient), nên thường cho hiệu quả cao; ví dụ: AdaBoost, Gradient Boosting, XGBoost, LightGBM, CatBoost.

Ý tưởng chính của Gradient Boosting dựa trên hai nguyên tắc:

- Boosting: Xây dựng mô hình mạnh từ việc kết hợp nhiều mô hình yếu
- Gradient Descent: Sử dụng gradient descent trong không gian hàm để tối ưu hóa hàm măt/mát



Initialization: Initial Model Prediction

- Khởi tạo dự đoán ban đầu bằng **mean** của nhãn (regression) / **log-odds** (classification):

$$\mu = \frac{1}{N} \sum_{i=1}^N y_i$$

- N : số mẫu train, y_i : giá trị thật mẫu i , μ : dự đoán ban đầu.

Step 1: Calculate Residuals

- Tính **residuals** = sai số còn lại giữa giá trị thật và dự đoán hiện tại:

$$r_1 = y - \hat{y} = y - \mu, \quad r_2 = y - \hat{y}_{new}$$

- Residuals là phần mô hình chưa học được → mục tiêu cho cây tiếp theo.

Step 2: Build a Decision Tree $h(x)$

- Train **decision tree** để dự đoán residuals từ features X : $\hat{r} = h(x)$.
- Mục tiêu: minimize squared error:

$$h(x) = \arg \min_h \sum_{i=1}^N (r_i - h(x_i))^2$$

Step 3: Compute Residual Outputs

- Với mỗi leaf j , tìm γ_j tối ưu để giảm loss:

$$\gamma_j = \arg \min_{\gamma} \sum_{x_i \in leaf_j} L(y_i, F_{m-1}(x_i) + \gamma)$$

- Với squared loss, γ_j thường là trung bình residual trong leaf j .

Step 4: Update Predictions

- Cập nhật dự đoán bằng learning rate:

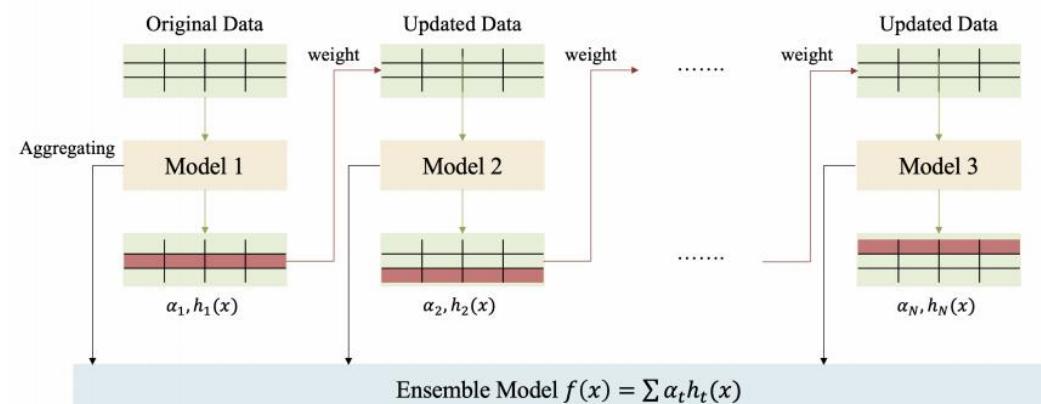
$$\hat{y}_{new} = \hat{y} + lr \times \hat{r}$$

- lr nhỏ giúp tránh overfitting và tăng generalization.

Loop and Convergence

- Lặp lại các bước trên đến khi đủ số trees / residual nhỏ / early stopping.
- Mô hình cuối:

$$F_M(x) = \mu + \sum_{m=1}^M lr \times h_m(x)$$



Classification Evaluation Metrics

Confusion Matrix (TP, FP, TN, FN)

Khái niệm. Ma trận nhầm lẫn (confusion matrix) là bảng 2×2 tóm tắt kết quả phân loại bằng cách đếm số dự đoán đúng/sai theo từng lớp. Đây là nền tảng để suy ra hầu hết các chỉ số đánh giá.

- TP (True Positive): dự đoán có bệnh và thực tế có bệnh
- FP (False Positive): dự đoán có bệnh nhưng thực tế không bệnh (báo động giả)
- TN (True Negative): dự đoán không bệnh và thực tế không bệnh
- FN (False Negative): dự đoán không bệnh nhưng thực tế có bệnh (bỏ sót)

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

$$N = TP + FP + TN + FN$$

Classification Metrics Cheat Sheet

Metric	Khái niệm	Công thức	Ghi chú nhanh
Confusion Matrix	Bảng 2×2 đếm đúng/sai theo lớp.	$N = TP + FP + TN + FN$ $\frac{TP + TN}{N}$	Nền tảng để tính hầu hết các metric còn lại.
Accuracy	Tỷ lệ đúng tổng thể.	$\frac{TP}{TP + FP}$ $\frac{TP}{TP + FN}$ $\frac{TN}{TN + FP}$	Dễ “ảo” khi lớp dương hiếm → không nên dùng một mình.
Precision	Trong dự đoán dương, đúng thật bao nhiêu.		Cao → ít báo nhầm dương tính (giảm FP).
Recall	Trong dương tính thật, bắt đúng bao nhiêu.		Cao → ít bỏ sót dương tính (giảm FN).
Specificity	Trong âm tính thật, nhận đúng bao nhiêu.		Cao → ít dương tính giả ở nhóm không bệnh.
FPR	Âm tính thật bị dự đoán nhầm dương.	$\frac{FP}{FP + TN} = 1 - \text{Specificity}$	Thấp → hạn chế báo động giả.
F1-score	Cân bằng Precision & Recall.	$\frac{2PR}{P + R} = \frac{2TP}{2TP + FP + FN}$	Hữu ích khi dữ liệu mất cân bằng (imbalanced).

Medical-Specific Metrics & Curves for CHD

Metric	Khái niệm	Công thức	Ý nghĩa trong CHD (y tế)
Recall	Ưu tiên bắt đúng ca bệnh.	$\frac{TP}{TP + FN}$	Rất quan trọng: cao → giảm bỏ sót người có nguy cơ CHD (giảm FN).
Specificity	Kiểm soát báo động giả.	$\frac{TN}{TN + FP}$	Cao → giảm dương tính giả (giảm FP), tránh gây lo lắng/xét nghiệm không cần thiết.
ROC Curve	Trade-off theo threshold.	$x = \text{FPR}, y = \text{TPR}$	Khi thay đổi ngưỡng, ta đánh đổi giữa bắt bệnh (TPR) và báo nhầm (FPR).
ROC-AUC	Diện tích dưới ROC.	$\int_0^1 \text{TPR}(\text{FPR}) d(\text{FPR})$	Đo khả năng phân biệt tổng quát; đôi khi vẫn “đẹp” dù lớp dương hiếm → nên xem thêm PR-AUC.
PR Curve	Trade-off Precision–Recall.	$x = \text{Recall}, y = \text{Precision}$	Phù hợp lớp dương hiếm: phản ánh trực tiếp chất lượng phát hiện ca bệnh và kiểm soát báo nhầm.
PR-AUC	Diện tích dưới PR curve.	$\int_0^1 \text{Precision}(\text{Recall}) d(\text{Recall})$	Rất phù hợp CHD imbalanced: tổng hợp khả năng phát hiện ca bệnh mà vẫn kiểm soát FP.

ROC Curve (Đường cong ROC)

Khái niệm: ROC curve là đồ thị thể hiện sự đánh đổi giữa TPR (Recall/Sensitivity) và FPR (tỷ lệ dương tính giả) khi ta thay đổi ngưỡng phân loại (threshold) trên điểm số/xác suất dự đoán của mô hình.

Ý nghĩa: Cho thấy model hoạt động thế nào trên nhiều threshold khác nhau và giúp chọn threshold phù hợp tùy mục tiêu (ưu tiên bắt bệnh hay giảm báo nhầm).

ROC-AUC

Khái niệm: ROC-AUC (Area Under the ROC Curve) là diện tích dưới đường ROC, dùng để tóm tắt khả năng phân biệt hai lớp của mô hình một cách tổng quát, không phụ thuộc vào một threshold cố định.

Ý nghĩa: AUC càng cao thì mô hình càng có xu hướng xếp hạng mẫu dương tính cao hơn mẫu âm tính. Tuy nhiên, với dữ liệu lệch lớp mạnh, ROC-AUC đôi khi vẫn “đẹp” nên thường cần xem thêm PR-AUC.

PR Curve (Đường cong Precision–Recall)

Khái niệm: PR curve là đồ thị thể hiện sự đánh đổi giữa Precision và Recall khi thay đổi threshold.

Ý nghĩa: PR curve đặc biệt hữu ích khi lớp dương tính hiếm (imbalanced) vì tập trung đánh giá chất lượng dự đoán trên lớp dương: vừa bắt được nhiều ca bệnh (Recall), vừa hạn chế báo động giả (Precision).

PR-AUC (Average Precision)

Khái niệm: PR-AUC là diện tích dưới PR curve, thường được báo cáo dưới dạng Average Precision (AP) để tóm tắt hiệu năng mô hình trên toàn bộ các threshold.

Ý nghĩa: PR-AUC càng cao thì mô hình càng tốt trong việc phát hiện lớp dương tính hiếm mà vẫn giữ dự đoán dương tính “đáng tin” (cân bằng giữa Recall và Precision), rất phù hợp cho bài toán y tế như dự đoán nguy cơ bệnh tim.

Baseline results Top-5 models are highlighted

Model	ROC-AUC	PR-AUC	Recall	Precision	F1	Thr
ET	0.9462	0.8565	0.6914	0.8616	0.7672	0.3375
RF	0.9381	0.8435	0.7175	0.8043	0.7584	0.2800
LGBM	0.8879	0.7256	0.6524	0.7018	0.6762	0.2468
HGB	0.8621	0.6490	0.5965	0.6308	0.6131	0.2682
XGB	0.7772	0.4780	0.5046	0.4318	0.4654	0.2393
KNN	0.7716	0.3552	0.5360	0.3610	0.4314	0.2667
SVC	0.7713	0.3720	0.5260	0.4003	0.4547	0.2600
GB	0.7530	0.4365	0.5482	0.3503	0.4275	0.1987
DT	0.7262	0.3646	0.5306	0.5513	0.5408	1.0000
LR	0.7216	0.3419	0.6133	0.2822	0.3866	0.5323
ADA	0.7126	0.3160	0.6210	0.2711	0.3775	0.2559
GNB	0.7016	0.2891	0.4954	0.2860	0.3627	0.0108

Top-5 models selected for tuning

Top-performing baseline models.

Model	ROC-AUC	PR-AUC	Recall	Precision	F1	Thr
ET	<u>0.9462</u>	<u>0.8565</u>	0.6914	<u>0.8616</u>	<u>0.7672</u>	0.3375
RF	0.9381	0.8435	<u>0.7175</u>	0.8043	0.7584	0.2800
LGBM	0.8879	0.7256	0.6524	0.7018	0.6762	0.2468
HGB	0.8621	0.6490	0.5965	0.6308	0.6131	0.2682
XGB	0.7772	0.4780	0.5046	0.4318	0.4654	0.2393

5. Hyperparameter Tuning

Before/After Tuning Metrics

Stage	Model	Metrics					
		ROC-AUC	PR-AUC	Acc	Precision	Recall	F1
Before tuning	<u>ET</u>	0.9462	0.8565	0.9356	0.8616	0.6914	0.7672
	RF	0.9381	0.8435	0.9299	0.8043	0.7175	0.7584
	HGB	0.8879	0.7256	0.9041	0.7018	0.6524	0.6762
	LGBM	0.8621	0.6490	0.8845	0.6308	0.5965	0.6131
	XGB	0.7772	0.4780	0.8221	0.4318	0.5046	0.4654
After tuning	<u>ET</u>	0.9388	0.8463	0.9300	0.8024	0.7213	0.7597
	RF	0.9456	0.8565	0.8850	0.5863	0.8507	0.6942
	HGB	0.9253	0.8316	0.8697	0.5486	0.8507	0.6671
	LGBM	0.9188	0.8237	0.8306	0.4716	0.8637	0.6101
	XGB	0.8862	0.7248	0.7421	0.3571	0.8507	0.5031

HyperParameter for Tuning

Baseline vs. tuned hyperparameters for ET and RF.

Stage	Model	thr	Hyperparameters
Baseline	ET	0.338	n_estimators=400, max_depth=None, max_features=sqrt, min_samples_split=2, min_samples_leaf=1, bootstrap=False, class_weight=balanced
	RF	0.280	n_estimators=400, max_depth=None, max_features=sqrt, min_samples_split=2, min_samples_leaf=1, bootstrap=True, class_weight=balanced
Tuned	ET	0.180	n_estimators=659, max_depth=30, max_features=sqrt, min_samples_split=2, min_samples_leaf=1, bootstrap=False, class_weight=balanced
	RF	0.277	n_estimators=859, max_depth=29, max_features=sqrt, min_samples_split=2, min_samples_leaf=1, bootstrap=True, class_weight=balanced

6. Evaluation on Testset

Test set performance comparison between RF and ET.

Model	ROC-AUC	PR-AUC	Accuracy	Precision	Recall	F1
RF	0.9655	0.9040	<u>0.9445</u>	<u>0.8276</u>	0.8073	<u>0.8173</u>
ET	<u>0.9705</u>	<u>0.9114</u>	0.9023	0.6258	<u>0.9052</u>	0.7400

Confusion matrix of RF on the test set.

	Pred 0	Pred 1
True 0	1746	55
True 1	63	264

Confusion matrix of ET on the test set.

	Pred 0	Pred 1
True 0	1624	177
True 1	31	296

7. Demo Inference on Gradio