TRƯỜNG ĐẠI HỌC BÁCH KHOA - ĐẠI HỌC QUỐC GIA TP.HCM
KHOA KHOA HỌC VÀ KĨ THUẬT MÁY TÍNH


## KHAI PHÁ DỮ LIỆU (CO3029)

---

## BÀI TẬP LỚN

# Phân tích các yếu tố liên quan đến bệnh tim mạch và xây dựng mô hình dự đoán

---

Giảng viên hướng dẫn: Đỗ Thanh Thái

Sinh viên thực hiện: Nguyễn Hồ Quốc Thịnh - 2213287
Lê Thanh Tuyển - 2213836
Cao Lê Hoàn Thiện - 2213243

Thành phố Hồ Chí Minh, 11/2025

# Mục lục

# 1 Giới thiệu đề tài

Suy tim (*heart failure*) là một trong những bệnh lý tim mạch nguy hiểm nhất, gây ra hàng triệu ca tử vong mỗi năm trên toàn cầu và đang trở thành gánh nặng y tế công cộng ngày càng nghiêm trọng. Theo Tổ chức Y tế Thế giới (WHO), suy tim không chỉ làm giảm chất lượng cuộc sống mà còn gia tăng đáng kể chi phí điều trị và tỷ lệ tái nhập viện. Việc nhận diện sớm các yếu tố nguy cơ và dự đoán chính xác khả năng xảy ra suy tim ở từng cá nhân là yếu tố then chốt để can thiệp kịp thời, giảm thiểu biến chứng và cải thiện tiên lượng bệnh.

Trong nghiên cứu này, nhóm đã sử dụng tập dữ liệu `cardio_train.csv` – một bộ dữ liệu lớn với 70.000 bản ghi lâm sàng được thu thập từ các bệnh nhân tại nhiều cơ sở y tế, bao gồm đầy đủ các thông tin về nhân khẩu học, chỉ số sinh lý, thói quen sinh hoạt và tình trạng bệnh tim mạch (*cardiovascular disease – CVD*).

## 1.1 Mục tiêu nghiên cứu

**Phân tích các yếu tố nguy cơ chính** liên quan đến bệnh tim mạch (`cardio = 1`), bao gồm:

- Tuổi tác, giới tính, chỉ số khối cơ thể (BMI);
- Huyết áp tâm thu/tâm trương (`ap_hi`, `ap_lo`);
- Mức cholesterol và glucose máu;
- Thói quen hút thuốc, uống rượu và mức độ vận động thể chất.

**Xây dựng và so sánh hiệu quả** của các mô hình học máy trong việc dự đoán nguy cơ bệnh tim – bước nền tảng để phát triển hệ thống cảnh báo sớm suy tim:

1. Logistic Regression
2. Decision Tree
3. Random Forest
4. Gradient Boosting (XGBoost, LightGBM);
5. Support Vector Classifier - SVC
6. K-Neighbors Classifier

**Đề xuất mô hình tối ưu** có độ chính xác cao, khả năng giải thích tốt và tiềm năng ứng dụng thực tế trong hệ thống hỗ trợ chẩn đoán lâm sàng.

## 1.2 Ý nghĩa của nghiên cứu

- **Y học dự phòng:** Giúp sàng lọc nguy cơ cao ở cộng đồng, đặc biệt ở nhóm trung niên và cao tuổi.
- **Hỗ trợ ra quyết định lâm sàng:** Cung cấp công cụ AI hỗ trợ bác sĩ đánh giá rủi ro nhanh chóng, chính xác.
- **Giảm gánh nặng y tế:** Giảm tỷ lệ nhập viện do suy tim cấp bằng cách can thiệp sớm.

Với quy mô dữ liệu lớn, đa dạng và chất lượng cao, nghiên cứu không chỉ đóng góp vào việc hiểu rõ hơn cơ chế bệnh sinh của suy tim mà còn mở ra hướng tiếp cận mới trong việc ứng dụng các mô hình dự đoán trong chẩn đoán và dự phòng bệnh lý tim mạch.

# 2 Chuẩn hóa dữ liệu

## 2.1 Tổng quan về dữ liệu

Tập dữ liệu `cardio_train.csv` là một bộ dữ liệu y tế liên quan đến các yếu tố rủi ro của bệnh tim mạch (cardiovascular disease). Dữ liệu này bao gồm khoảng 70.000 bệnh nhân và thường được sử dụng trong các bài toán học máy (machine learning) nhằm dự đoán khả năng mắc bệnh tim dựa trên các đặc trưng sức khỏe.

Tập dữ liệu được lưu trữ dưới dạng CSV, gồm **70.000 bản ghi (hàng)** và **13 cột (đặc trưng)**. Không có giá trị bị thiếu (*missing values*), giúp cho việc phân tích thuận tiện mà không cần xử lý *imputation*. Tuy nhiên, một số cột như huyết áp (`ap_hi`, `ap_lo`) chứa giá trị bất thường (ví dụ: âm hoặc quá cao), cho thấy cần phải làm sạch dữ liệu trước khi sử dụng cho mô hình dự đoán.

### 2.1.1 Mô tả các đặc trưng (Features)

Dữ liệu bao gồm các thông tin nhân khẩu học, chỉ số sức khỏe và lối sống, cùng với biến mục tiêu (`cardio`) biểu thị tình trạng bệnh tim. Bảng dưới đây mô tả ý nghĩa từng cột:

- **id**: Mã định danh duy nhất của bệnh nhân (số nguyên, từ 0 đến 99999).
- **age**: Tuổi bệnh nhân tính bằng ngày (số nguyên). Trung bình khoảng 19.469 ngày (tương đương ~53 tuổi), dao động từ ~30 đến ~65 tuổi khi quy đổi sang năm.
- **gender**: Giới tính (1 - nữ, chiếm khoảng 65%; 2 - nam, chiếm khoảng 35%).
- **height**: Chiều cao (cm, số nguyên). Trung bình ~165 cm, dao động từ 55 cm đến 250 cm (có thể có giá trị ngoại lai).
- **weight**: Cân nặng (kg, số thực). Trung bình ~74 kg, dao động từ 10 kg đến 200 kg (có thể có giá trị ngoại lai).
- **ap_hi**: Huyết áp tâm thu (*systolic blood pressure*, mmHg). Trung bình ~129 mmHg, nhưng có giá trị bất thường (từ -150 đến 16020).
- **ap_lo**: Huyết áp tâm trương (*diastolic blood pressure*, mmHg). Trung bình ~97 mmHg, với giá trị ngoại lai từ -70 đến 11000.
- **cholesterol**: Mức cholesterol (1 - bình thường, 75%; 2 - cao hơn bình thường, 14%; 3 - cao, 12%).
- **gluc**: Mức glucose (1 - bình thường, 85%; 2 - cao hơn bình thường, 7%; 3 - cao, 8%).
- **smoke**: Hút thuốc (0 - không, 91%; 1 - có, 9%).
- **alco**: Uống rượu (0 - không, 95%; 1 - có, 5%).
- **active**: Hoạt động thể chất (0 - không, 20%; 1 - có, 80%).
- **cardio**: Biến mục tiêu — có bệnh tim hay không (0 - không, 50%; 1 - có, 50%). Dữ liệu khá cân bằng, phù hợp cho các mô hình phân loại nhị phân.

## 2.2 Chuẩn hóa dữ liệu

Trước khi xây dựng mô hình học máy, dữ liệu cần được chuẩn hóa để loại bỏ các thông tin không cần thiết, tính toán các biến mới và xử lý các giá trị ngoại lai. Các bước chính được thực hiện như sau:

### 2.2.1 Xóa các trường dữ liệu không cần thiết

Trường `id` không chứa thông tin dự đoán nên được loại bỏ để tránh gây nhiễu cho mô hình. Mã Python thực hiện như sau:

```
if 'id' in data.columns:
    data = data.drop(columns=['id'])
```

Listing 1: Xóa trường dữ liệu id ra khỏi dataset

### 2.2.2 Thêm trường dữ liệu age (years) và BMI index

Để thuận tiện cho phân tích, thêm hai cột mới:

- **age_years**: tuổi của bệnh nhân theo năm, tính từ cột `age` (theo ngày):

```
data['age_years'] = (data['age'] / 365).astype(int)
```

- **bmi**: chỉ số khối cơ thể (Body Mass Index), tính từ cân nặng và chiều cao:

```
data['bmi'] = data['weight'] / ((data['height']/100) ** 2)
```

### 2.2.3 Loại bỏ các điểm ngoại lai

Sử dụng phương pháp **Interquartile Range (IQR)** để xác định và loại bỏ các giá trị ngoại lai trong các cột `height`, `weight` và `bmi`:

```
def remove_outliers_iqr(df, column, factor=1.5):
    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1
    lower = Q1 - factor * IQR
    upper = Q3 + factor * IQR
    return df[(df[column] >= lower) & (df[column] <= upper)]

# Applying IQR for suitable columns (eg: height, weight, bmi,...)
cols_iqr = ['height', 'weight', 'bmi']
for col in cols_iqr:
    data = remove_outliers_iqr(data, col, factor=1.5)   # factor=1.5 is standard
```

Listing 2: Loại bỏ điểm ngoại lai bằng phương pháp IQR

### 2.2.4 Kết hợp với hard-threshold cho chỉ số huyết áp

```python
data = data[(data['ap_hi'] > 0) & (data['ap_hi'] <= 200)]
data = data[(data['ap_lo'] > 0) & (data['ap_lo'] <= 150)]
data = data[data['ap_hi'] >= data['ap_lo']]

print(f"Number of rows after processing: {data.shape}  (from 70,000)")
data.describe()
```

Listing 3: Xử lý ngoại lai cho huyết áp

Number of rows after processing: (65230, 14) (from 70,000)

| | age | gender | height | weight | ap_hi | ap_lo | cholesterol | gluc | smoke | alco | active | cardio | age_years | bmi |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 65230.000000 | 65230.000000 | 65230.000000 | 65230.000000 | 65230.000000 | 65230.000000 | 65230.000000 | 65230.000000 | 65230.000000 | 65230.000000 | 65230.000000 | 65230.000000 | 65230.000000 | 65230.000000 |
| mean | 19451.730553 | 1.351019 | 164.471700 | 72.509453 | 126.127549 | 81.013307 | 1.351924 | 1.217400 | 0.087782 | 0.052507 | 0.804599 | 0.485497 | 52.793500 | 26.822662 |
| std | 2470.433270 | 0.477293 | 7.502512 | 11.951164 | 16.246695 | 9.367554 | 0.668808 | 0.563333 | 0.282979 | 0.223048 | 0.396512 | 0.499793 | 6.775455 | 4.229729 |
| min | 10798.000000 | 1.000000 | 143.000000 | 40.000000 | 16.000000 | 1.000000 | 1.000000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 29.000000 | 15.035584 |
| 25% | 17640.000000 | 1.000000 | 159.000000 | 64.000000 | 120.000000 | 80.000000 | 1.000000 | 1.000000 | 0.000000 | 0.000000 | 1.000000 | 0.000000 | 48.000000 | 23.795360 |
| 50% | 19694.000000 | 1.000000 | 165.000000 | 71.000000 | 120.000000 | 80.000000 | 1.000000 | 1.000000 | 0.000000 | 0.000000 | 1.000000 | 0.000000 | 53.000000 | 26.093275 |
| 75% | 21314.000000 | 2.000000 | 170.000000 | 80.000000 | 140.000000 | 90.000000 | 1.000000 | 1.000000 | 0.000000 | 0.000000 | 1.000000 | 1.000000 | 58.000000 | 29.552549 |
| max | 23713.000000 | 2.000000 | 186.000000 | 107.000000 | 200.000000 | 150.000000 | 3.000000 | 3.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 64.000000 | 38.625904 |

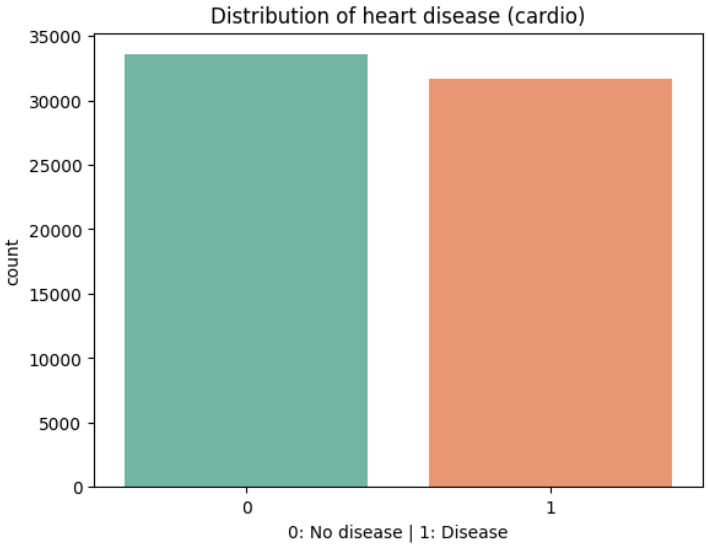
Sau khi thực hiện các bước xử lý trên, số lượng bản ghi còn khoảng 65,230 hàng, đồng thời dữ liệu đã sẵn sàng cho các bước phân tích và huấn luyện mô hình học máy tiếp theo.

## 2.3 Phân tích dữ liệu (EDA)

### 2.3.1 Tổng quan về các biến thể tim mạch

| cardio | |
|---|---|
| 0 | 0.514503 |
| 1 | 0.485497 |

(a) Phân phối minh họa bệnh tim mạch




(b) Phân bố tỷ lệ mắc bệnh tim trong dữ liệu

Hình 1: Hình ảnh minh họa và phân bố bệnh tim trong tập dữ liệu `cardio_train.csv`

Tập dữ liệu gần như hoàn toàn cân bằng giữa hai lớp: **50.58% không bệnh** – **49.42% có bệnh**, với sai số chỉ **1.16%**. Đây là điều kiện lý tưởng cho bài toán phân loại nhị phân, giúp:

- Tránh hiện tượng *class imbalance* gây *bias* cho mô hình.
- Đảm bảo độ tin cậy của các chỉ số đánh giá như *Accuracy*, *F1-score*, và *AUC-ROC*.

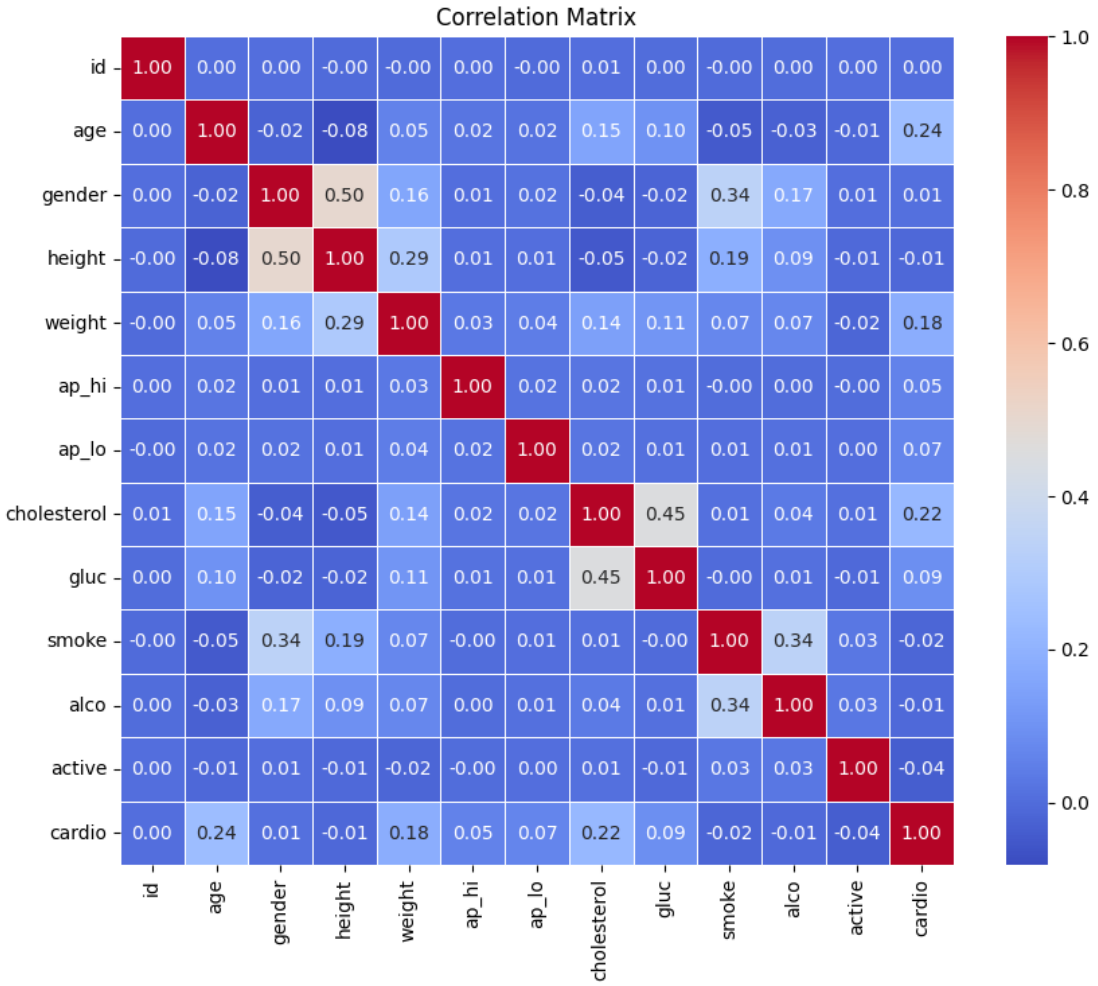
**Ý nghĩa lâm sàng:** Tỷ lệ bệnh tim mạch gần 50% cho thấy đây là mẫu đại diện tốt cho dân số có nguy cơ cao, phù hợp để nghiên cứu dự đoán sớm bệnh tim – tiền đề của *suy tim*.

### 2.3.2 Ma trận tương quan của các biến

**Ma trận tương quan (Correlation Matrix)** là một bảng biểu diễn các hệ số tương quan giữa các biến. Mỗi ô trong bảng hiển thị mối tương quan giữa hai biến. Ma trận tương quan được sử dụng để tóm tắt dữ liệu, làm đầu vào cũng như làm chẩn đoán cho các phân tích nâng cao. Trong ma trận tương quan:

- Nếu mối quan hệ là 1 thì mối quan hệ đó rất mạnh.
- Nếu mối quan hệ bằng 0 thì có nghĩa là mối quan hệ đó là trung tính.
- Nếu mối quan hệ là -1 thì có nghĩa là mối quan hệ đó không mạnh.s

Mục đích sử dụng ma trận tương quan nhằm tìm hiểu mối tương quan tuyến tính chặt chẽ giữa biến phụ thuộc với các biến độc lập và sớm nhận diện vấn đề đa cộng tuyến khi các biến độc lập cũng có tương quan mạnh với nhau. Sau khi dữ liệu được tập hợp, làm sạch và phân loại, ta tiến hành vẽ heat map để xem xét sự phụ thuộc của biến "cardio"đối với các biến còn lại.




Hình 2: Ma trận tương quan Pearson giữa các biến số trong tập dữ liệu.

Cardio tương quan dương mạnh nhất với tuổi ($r = 0.24$), cholesterol ($r = 0.22$), huyết áp tâm thu ($r = 0.21$) và BMI ($r = 0.18$); tương quan âm nhẹ với hoạt động thể chất ($r = -0.04$). Đa cộng tuyến cao giữa weight–bmi ($r = 0.84$) và ap_hi–ap_lo ($r = 0.72$).

# 3 Trực quan hóa dữ liệu

Để hiểu rõ hơn về phân bố dữ liệu và mối quan hệ giữa các đặc trưng với biến mục tiêu `cardio`, chúng ta tiến hành trực quan hóa dữ liệu theo hai nhóm chính: **các biến rời rạc** và **các biến liên tục**. Các biểu đồ được sử dụng bao gồm: biểu đồ cột, biểu đồ mật độ (KDE), boxplot, và histogram. Phân tích này giúp xác định các yếu tố nguy cơ tiềm năng và định hướng cho việc tiền xử lý và lựa chọn đặc trưng trong mô hình dự đoán.

## 3.1 Các biến rời rạc

Các biến rời rạc bao gồm: `gender` (giới tính), `cholesterol` (mức cholesterol), `gluc` (mức glucose), `smoke` (hút thuốc), `alco` (uống rượu), và `active` (hoạt động thể chất). Chúng được phân tích thông qua bảng chéo (crosstab) và biểu đồ cột. **Tỉ lệ phần trăm được tính theo nhóm** `cardio`, và các nhóm rủi ro cao (ví dụ: cholesterol mức 2+3) sẽ được cộng lại để đánh giá tổng thể.

### 3.1.1 Gender (Giới tính)

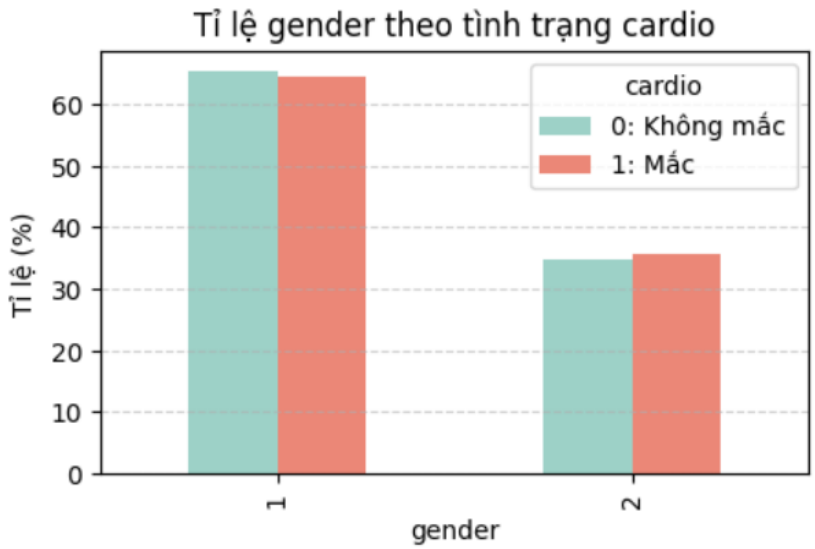
Biến `gender` có hai giá trị: 1 (nữ), 2 (nam).

Bảng 1: Tỉ lệ giới tính theo tình trạng bệnh tim

| Gender | Mô tả | cardio = 0 (%) | cardio = 1 (%) |
|---|---|---|---|
| 1 | Nữ | 65.36 | 64.41 |
| 2 | Nam | 34.64 | 35.59 |

--- GENDER ---

| cardio | 0 | 1 |
|---|---|---|
| gender | | |
| 1 | 65.36 | 64.41 |
| 2 | 34.64 | 35.59 |




Hình 3: Tỉ lệ giới tính theo tình trạng `cardio`

**Nhận xét**: Tỉ lệ nam giới ở nhóm mắc bệnh cao hơn một chút (**35.59%** so với **34.64%**). Tuy nhiên, sự khác biệt không đáng kể, cho thấy `gender` không phải là yếu tố phân biệt mạnh.

### 3.1.2 Cholesterol (Mức cholesterol)

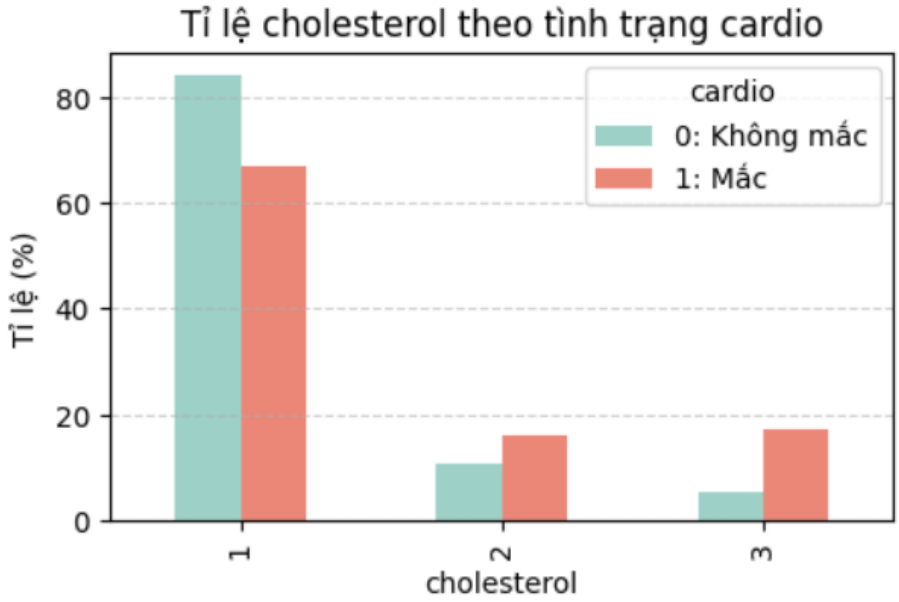
Biến có ba mức: 1 (bình thường), 2 (trên bình thường), 3 (cao).

Bảng 2: Tỉ lệ mức cholesterol theo tình trạng bệnh tim

| **Mức** | **Mô tả** | **cardio = 0 (%)** | **cardio = 1 (%)** |
|---|---|---|---|
| 1 | Bình thường | 84.18 | 66.85 |
| 2 | Trên bình thường | 10.64 | 16.05 |
| 3 | Cao | 5.17 | 17.10 |

--- CHOLESTEROL ---

| cardio | 0 | 1 |
|---|---|---|
| **cholesterol** | | |
| **1** | 84.18 | 66.85 |
| **2** | 10.64 | 16.05 |
| **3** | 5.17 | 17.10 |




Hình 4: Tỉ lệ mức cholesterol theo tình trạng `cardio`

**Nhận xét**:

- Tỉ lệ người có mức cholesterol **trên bình thường hoặc cao** (mức 2 + 3) ở nhóm mắc bệnh là $\mathbf{16.05\% + 17.10\% = 33.15\%}$, cao hơn **gấp đôi** so với nhóm không mắc ($\mathbf{10.64\% + 5.17\% = 15.81\%}$).
- Sự khác biệt này cho thấy `cholesterol` là **yếu tố rủi ro mạnh nhất** trong các biến rời rạc.

### 3.1.3 Gluc (Mức glucose)

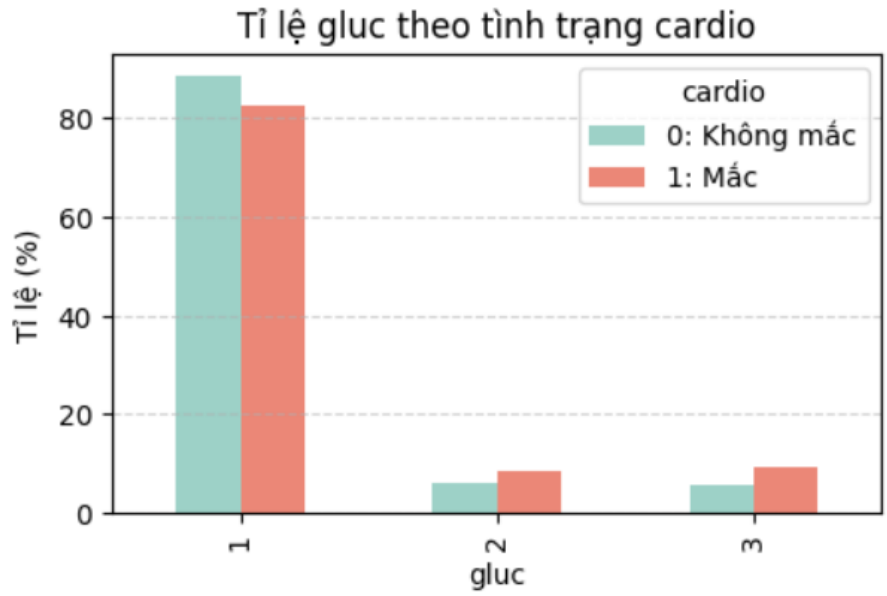
Biến có ba mức: 1 (bình thường), 2 (trên bình thường), 3 (cao).

Bảng 3: Tỉ lệ mức glucose theo tình trạng bệnh tim

| Mức | Mô tả | cardio = 0 (%) | cardio = 1 (%) |
|---|---|---|---|
| 1 | Bình thường | 88.63 | 82.43 |
| 2 | Trên bình thường | 5.80 | 8.31 |
| 3 | Cao | 5.56 | 9.26 |

--- GLUC ---

| cardio | 0 | 1 |
|---|---|---|
| gluc | | |
| 1 | 88.63 | 82.43 |
| 2 | 5.80 | 8.31 |
| 3 | 5.56 | 9.26 |




Hình 5: Tỉ lệ mức glucose theo tình trạng `cardio`

**Nhận xét**:

- Tỉ lệ người có mức glucose **trên bình thường hoặc cao** (mức 2 + 3) ở nhóm mắc bệnh là **8.31% + 9.26% = 17.57%**, cao hơn so với nhóm không mắc (**5.80% + 5.56% = 11.36%**).
- Mức glucose cao là yếu tố rủi ro, nhưng **yếu hơn cholesterol**.

### 3.1.4 Smoke (Hút thuốc)

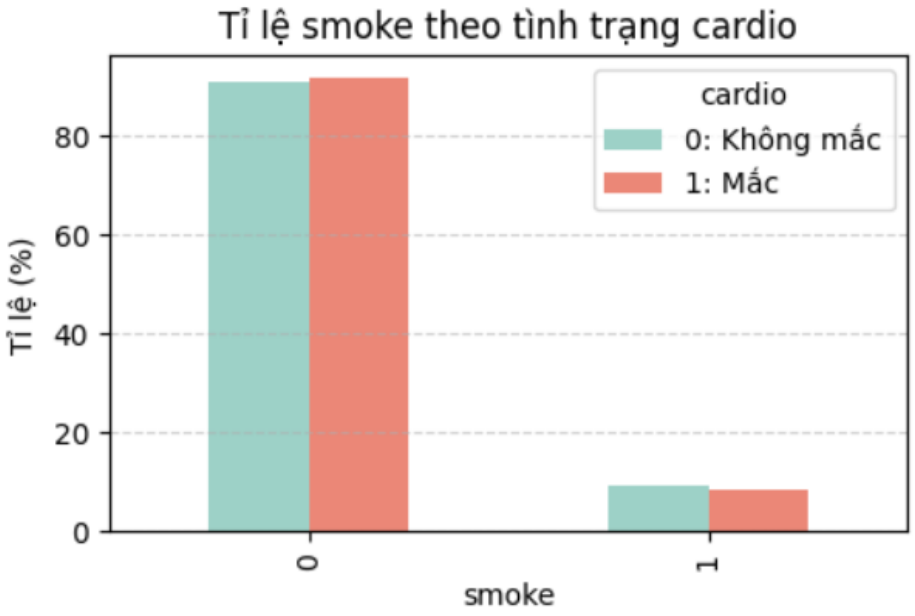
Biến nhị phân: 0 (không hút), 1 (hút).

Bảng 4: Tỉ lệ hút thuốc theo tình trạng bệnh tim

| Giá trị | Mô tả | cardio = 0 (%) | cardio = 1 (%) |
|---|---|---|---|
| 0 | Không hút | 90.73 | 91.74 |
| 1 | Hút | 9.27 | 8.26 |

--- SMOKE ---

| cardio | 0 | 1 |
|---|---|---|
| smoke | | |
| 0 | 90.73 | 91.74 |
| 1 | 9.27 | 8.26 |




Hình 6: Tỉ lệ hút thuốc theo tình trạng `cardio`

**Nhận xét**: Tỉ lệ hút thuốc ở nhóm không mắc cao hơn một chút (**9.27%** so với **8.26%**). Sự khác biệt nhỏ, cho thấy `smoke` **không phải yếu tố rủi ro rõ rệt** trong tập dữ liệu này.

### 3.1.5 Alco (Uống rượu)

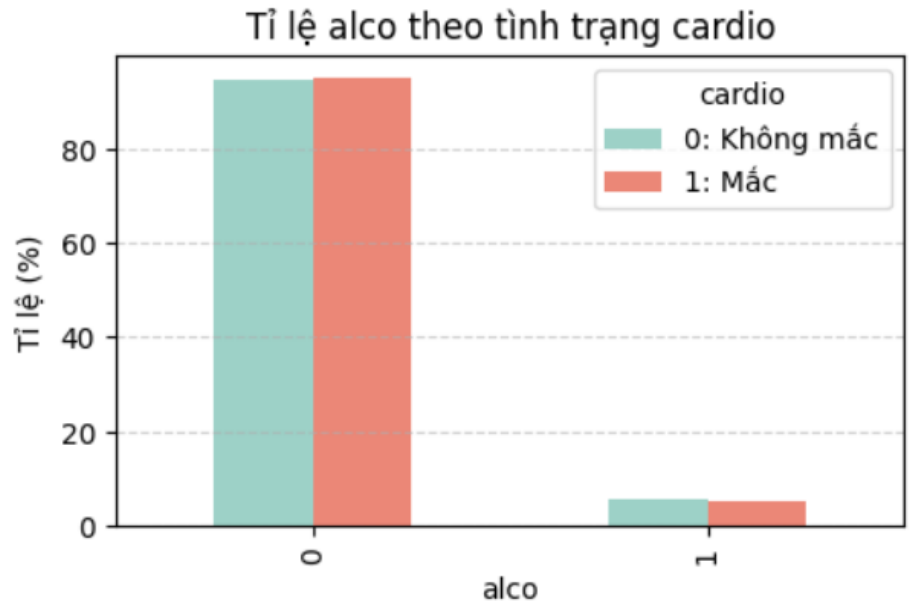
Biến nhị phân: 0 (không uống), 1 (uống).

Bảng 5: Tỉ lệ uống rượu theo tình trạng bệnh tim

| Giá trị | Mô tả | cardio = 0 (%) | cardio = 1 (%) |
|---|---|---|---|
| 0 | Không uống | 94.52 | 94.99 |
| 1 | Uống | 5.48 | 5.01 |

--- ALCO ---

| cardio | 0 | 1 |
|---|---|---|
| alco | | |
| 0 | 94.52 | 94.99 |
| 1 | 5.48 | 5.01 |




Hình 7: Tỉ lệ uống rượu theo tình trạng `cardio`

**Nhận xét**: Tỉ lệ uống rượu gần như tương đương (**5.48%** vs **5.01%**). `alco` **không có mối liên hệ rõ rệt** với bệnh tim mạch trong dữ liệu.

#### 3.1.6 Active (Hoạt động thể chất)

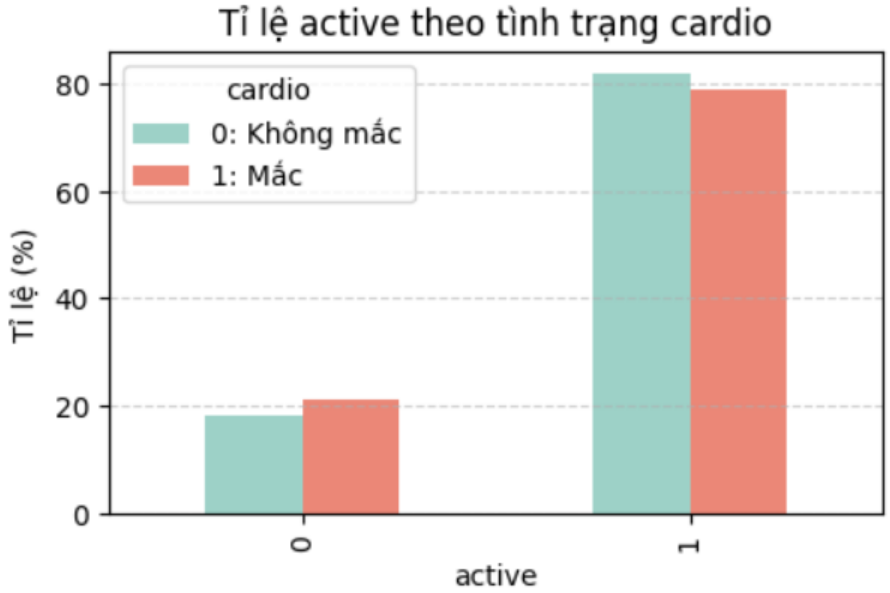
Biến nhị phân: 0 (không hoạt động), 1 (hoạt động).

Bảng 6: Tỉ lệ hoạt động thể chất theo tình trạng bệnh tim

| Giá trị | Mô tả | cardio = 0 (%) | cardio = 1 (%) |
|---|---|---|---|
| 0 | Không hoạt động | 18.13 | 21.04 |
| 1 | Hoạt động | 81.87 | 78.96 |

```
--- ACTIVE ---
```

| cardio | 0 | 1 |
|---|---|---|
| active | | |
| 0 | 18.13 | 21.04 |
| 1 | 81.87 | 78.96 |




Hình 8: Tỉ lệ hoạt động thể chất theo tình trạng `cardio`

**Nhận xét**: Nhóm mắc bệnh có tỉ lệ **không hoạt động cao hơn** (**21.04%** so với **18.13%**). Lối sống ít vận động là **yếu tố rủi ro trung bình**.

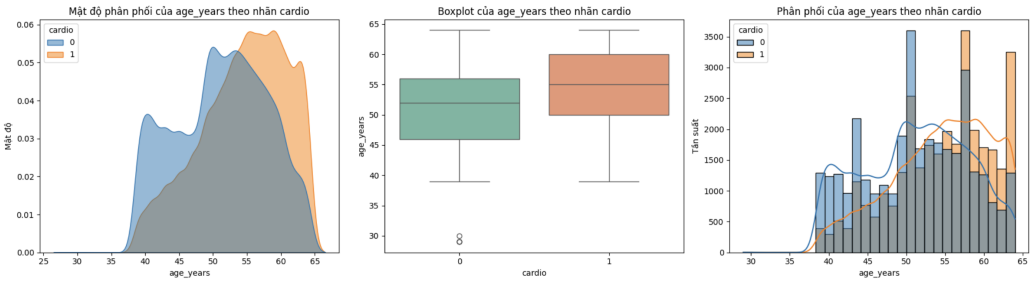
Bảng 7: Tóm tắt ảnh hưởng của các biến rời rạc (tính trên nhóm rủi ro cao)

| Biến | Khác biệt (%) | Mức độ ảnh hưởng |
|---|---|---|
| `cholesterol` | 33.15 - 15.81 = **+17.34** | **Rất mạnh** |
| `gluc` | 17.57 - 11.36 = **+6.21** | Trung bình |
| `active` | 21.04 - 18.13 = **+2.91** | Trung bình |
| `gender` | 35.59 - 34.64 = **+0.95** | Yếu |
| `smoke` | 8.26 - 9.27 = **-1.01** | Không có |
| `alco` | 5.01 - 5.48 = **-0.47** | Không có |

## 3.2 Các biến liên tục

Các biến liên tục bao gồm: `age_years` (tuổi tính theo năm), `height` (chiều cao, cm), `weight` (cân nặng, kg), `bmi` (chỉ số khối cơ thể), `ap_hi` (huyết áp tâm thu), và `ap_lo` (huyết áp tâm trương). Mỗi biến được trực quan hóa bằng một hình gộp ba biểu đồ: mật độ (KDE), boxplot và histogram theo hai nhóm `cardio`.

### 3.2.1 age_years (Tuổi)



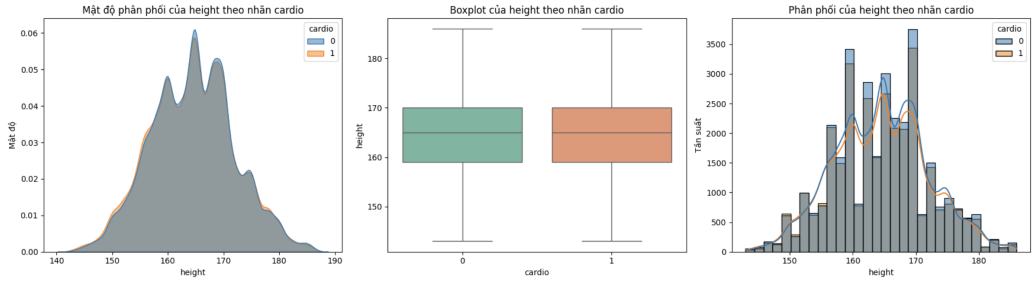

Hình 9: Phân tích `age_years`: Mật độ (trái), Boxplot (giữa), Histogram (phải)

**Nhận xét**:

- **KDE**: Nhóm mắc bệnh có phân phối dịch sang phải rõ rệt, đỉnh tại khoảng 58–60 tuổi, trong khi nhóm không mắc tập trung ở 50–52 tuổi.
- **Boxplot**:
  - Nhóm không mắc (cardio=0): Có ngoại lai dưới 35 tuổi.
  - Nhóm mắc bệnh (cardio=1): Không có ngoại lai.
- **Histogram**: Tần suất tuổi $> 55$ cao hơn đáng kể ở nhóm mắc bệnh.

**Kết luận**: `age_years` là **yếu tố rủi ro rất mạnh** – người lớn tuổi có nguy cơ mắc bệnh tim mạch cao hơn.

### 3.2.2 height (Chiều cao)



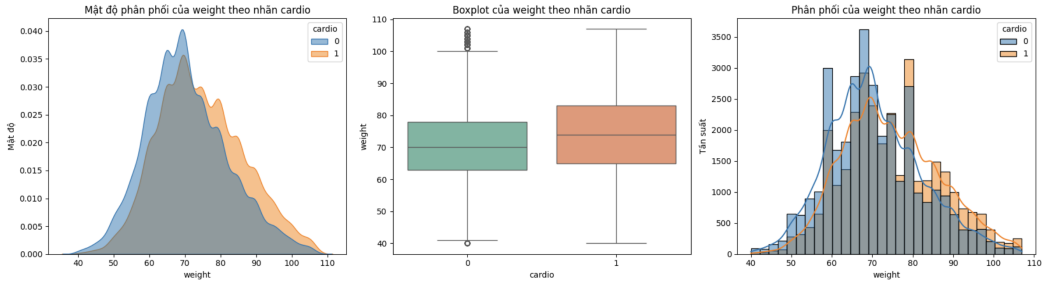

Hình 10: Phân tích `height`: Mật độ (trái), Boxplot (giữa), Histogram (phải)

**Nhận xét**:

- **KDE**: Hai nhóm có phân phối gần giống nhau, đỉnh chồng lấn tại 165 cm.
- **Boxplot**: Không có ngoại lai ở cả 2 nhóm.
- **Histogram**: Tần suất ở các khoảng chiều cao không có sự khác biệt rõ rệt.

**Kết luận**: `height` **không phải yếu tố rủi ro**.

### 3.2.3 weight (Cân nặng)



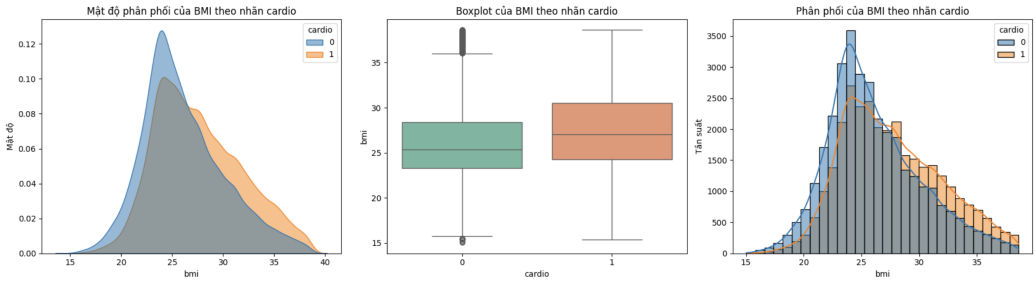

Hình 11: Phân tích `weight`: Mật độ (trái), Boxplot (giữa), Histogram (phải)

**Nhận xét**:

- **KDE**: Nhóm mắc bệnh có phân phối dịch sang phải, đỉnh ở khoảng 78–80 kg, trong khi nhóm không mắc ở 70 kg.
- **Boxplot**:
  - Nhóm không mắc (cardio=0): Có ngoại lai dưới khoảng 42 kg. Và trên khoảng 100 kg có nhiều ngoại lai.
  - Nhóm mắc bệnh (cardio=1): Không có ngoại lai.
- **Histogram**: Tần suất cân nặng $>$ 80 kg cao hơn đáng kể ở nhóm mắc bệnh.

**Kết luận**: `weight` là **yếu tố rủi ro trung bình** – người thừa cân/béo phì có nguy cơ cao hơn.

### 3.2.4 bmi (Chỉ số khối cơ thể)



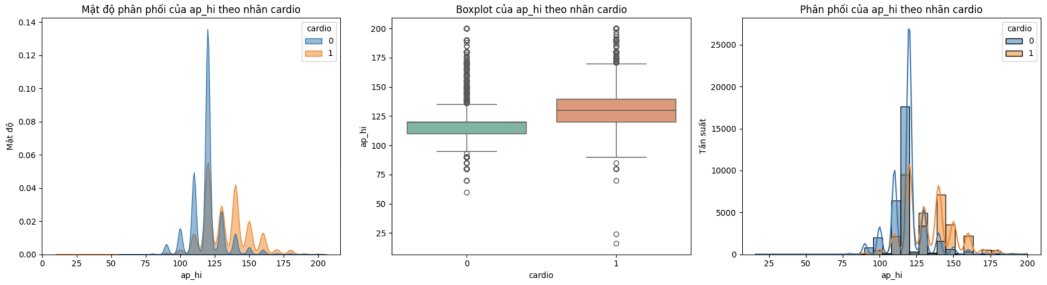

Hình 12: Phân tích `bmi`: Mật độ (trái), Boxplot (giữa), Histogram (phải)

**Nhận xét**:

- **KDE**: Nhóm mắc bệnh có phân phối dịch sang phải mạnh, đỉnh tại 29–30, trong khi nhóm không mắc ở 26.
- **Boxplot**:
  - Nhóm không mắc (cardio=0): Có ngoại lai dưới khoảng BMI 17, và nhiều ngoại lai BMI $>$ 35.
  - Nhóm mắc bệnh (cardio=1): Không có ngoại lai.
- **Histogram**: Tần suất BMI $>$ 27 cao hơn rõ rệt ở nhóm mắc bệnh.

**Kết luận**: `bmi` là **yếu tố rủi ro mạnh** – thừa cân là nguy cơ lớn.

### 3.2.5 ap_hi (Huyết áp tâm thu)




Hình 13: Phân tích `ap_hi`: Mật độ (trái), Boxplot (giữa), Histogram (phải)

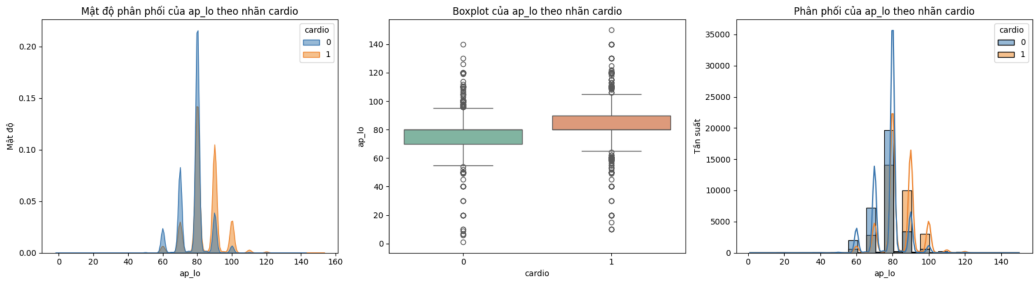
**Nhận xét**:

- **KDE**: Nhóm mắc bệnh có phân phối dịch rất xa sang phải, đỉnh tại 135 mmHg, trong khi nhóm không mắc ở 120 mmHg.
- **Boxplot**: Cả 2 nhóm đều có ngoại lai, nhóm không mắc bệnh có nhiều ngoại lai $>$ 130 mmHg và nhóm mắc bệnh có nhiều ngoại lai $>$ 175 mmHg.

- **Histogram**: Tần suất ap_hi $>$ 140 mmHg (tăng huyết áp) ở nhóm mắc bệnh cao gấp nhiều lần nhóm không mắc.

**Kết luận**: `ap_hi` là **yếu tố rủi ro mạnh nhất** trong các biến liên tục.

### 3.2.6 ap_lo (Huyết áp tâm trương)




Hình 14: Phân tích `ap_lo`: Mật độ (trái), Boxplot (giữa), Histogram (phải)

**Nhận xét**:

- **KDE**: Nhóm mắc bệnh dịch sang phải, đỉnh tại 85 mmHg, nhóm không mắc ở 80 mmHg.
- **Boxplot**: Cả 2 nhóm đều có ngoại lai. Nhóm không mắc bệnh có nhiều ngoại lai $>$ 95 mmHg, nhóm mắc bệnh có nhiều ngoại lại $>$ 100 mmHg và nhiều ngoại lai $<$ 70 mmHg.
- **Histogram**: Tần suất ap_lo $>$ 90 mmHg cao hơn ở nhóm mắc bệnh.

**Kết luận**: `ap_lo` là **yếu tố rủi ro mạnh**, nhưng yếu hơn `ap_hi`.

Bảng 8: Tóm tắt ảnh hưởng của các biến liên tục

| Biến | Mức độ ảnh hưởng |
|---|---|
| `ap_hi` | **Rất mạnh** |
| `age_years` | **Rất mạnh** |
| `bmi` | Mạnh |
| `ap_lo` | Mạnh |
| `weight` | Trung bình |
| `height` | Không có |

# 4 Xây dựng mô hình dự đoán

## 4.1 Chia dữ liệu và chuẩn hóa

- Tỷ lệ chia: 70% dữ liệu cho tập huấn luyện (train), 30% cho tập kiểm tra (test).
- Phương pháp: Sử dụng hàm train_test_split từ thư viện scikit-learn với tham số:
  - test_size=0.3
  - random_state=42 → đảm bảo tính tái lập của kết quả
  - stratify=y → giữ nguyên tỷ lệ lớp (bệnh/không bệnh) giữa hai tập, tránh hiện tượng lệch mẫu.
- Kết quả:
  - Tập train: X_train, y_train (dữ liệu đã chuẩn hóa: X_train_s và thô: X_train_raw)
  - Tập test: X_test, y_test (tương tự: X_test_s, X_test_raw)
- Tỷ lệ bệnh tim trong toàn bộ dữ liệu: $y.mean() \approx 0.500 \Rightarrow$ dữ liệu gần cân bằng.

## 4.2 Phương thức đánh giá mô hình

Do bài toán phân loại nhị phân (có/không mắc bệnh tim mạch) và dữ liệu gần cân bằng, các chỉ số sau được sử dụng để đánh giá hiệu suất mô hình:

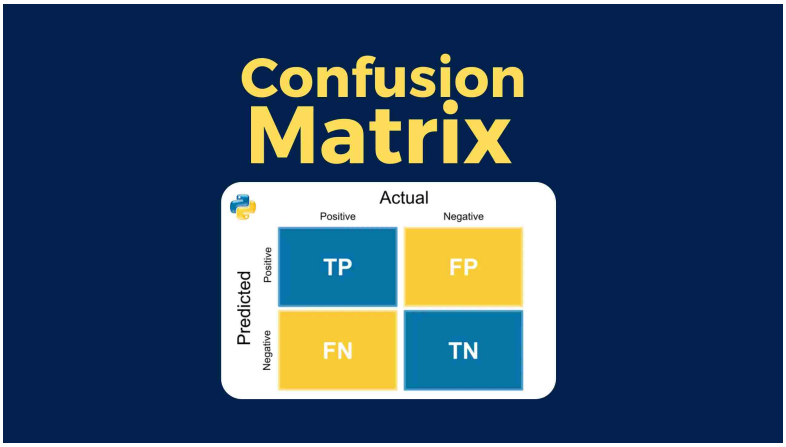
- **Độ chính xác (Accuracy)**: Tỷ lệ dự đoán đúng tổng thể trên toàn bộ mẫu dữ liệu.

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

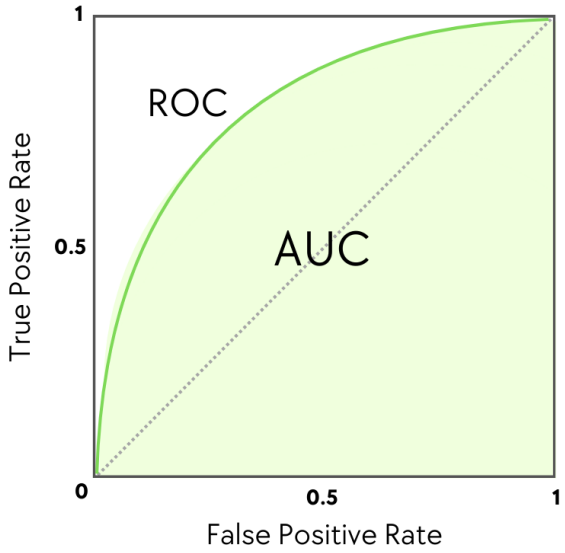
  Tuy nhiên, độ chính xác có thể bị ảnh hưởng bởi sự mất cân bằng trong phân bố lớp dữ liệu.
  **Ví dụ:** Giả sử mô hình dự đoán 99% mẫu là "không bệnh"(mặc dù thực tế có 50% bệnh và 50% không bệnh). Độ chính xác đạt 99%, nhưng khả năng phân biệt thực tế rất kém.

- **Ma trận hỗn loạn (Confusion Matrix)**: là bảng hiển thị số lượng dự đoán đúng và sai của mô hình cho từng lớp. Nó cung cấp thông tin chi tiết về hiệu suất mô hình trên từng lớp, giúp xác định điểm mạnh và điểm yếu của mô hình. Ma trận có dạng 2x2 với dạng dưới đây.
  **Ví dụ minh họa** (dựa trên trường hợp bệnh viện với 100 bệnh nhân, trong đó 60 thực tế có bệnh, 40 không bệnh):
  - TP = 45 (dự đoán đúng có bệnh).
  - TN = 30 (dự đoán đúng không bệnh).
  - FP = 10 (dự đoán sai có bệnh).
  - FN = 15 (dự đoán sai không bệnh).

  Từ ma trận này, mô hình dự đoán đúng 75% (TP + TN = 75/100), nhưng bỏ sót 15 bệnh nhân (FN).




- **Thước đo ROC-AUC**: là diện tích dưới đường cong ROC, thể hiện khả năng phân biệt giữa các lớp của mô hình. Giá trị AUC càng cao, mô hình phân loại càng có khả năng phân biệt chính xác giữa các lớp, thể hiện hiệu suất vượt trội. Giá trị AUC càng gần với giá trị 1, mô hình gọi là có giá trị ROC hoàn hảo. Với giá trị AUC = 0.5, mô hình mang giá trị dự đoán ngẫu nhiên (50/50).




**Ưu điểm:**

– Khả năng khách quan: Không phụ thuộc vào ngưỡng phân loại cụ thể.
– Dễ dàng giải thích: Đường cong ROC trực quan hóa hiệu suất mô hình.
– Tính linh hoạt: Áp dụng cho nhiều bài toán phân loại khác nhau.

**Nhược điểm:**

– Ảnh hưởng bởi phân bố dữ liệu.
– Thiếu thông tin chi tiết về hiệu suất trên từng lớp.

## 4.3 Hiện thực mô hình dự đoán

Ta áp dụng huấn luyện bộ dữ liệu này vào 6 mô hình sau: "LogisticRegression", "Decision Tree", "Support Vector Classifier", "K-Nearest Neighbors", "Gradient Boosting Classifier", "Random Forest Classifier".

```
1
2  def evaluate_model(model, X_train, X_test, y_train, y_test, scaler=None,
       model_name=None):
3      if model_name is None:
4          model_name = model.__class__.__name__.replace("Classifier", "")
5
6      # Scale features if required
7      if scaler:
8          X_train_s = scaler.fit_transform(X_train)
9          X_test_s = scaler.transform(X_test)
10     else:
11         X_train_s, X_test_s = X_train.values, X_test.values
12
13     # Train the model and make predictions
14     model.fit(X_train_s, y_train)
15     y_pred = model.predict(X_test_s)
16     y_prob = model.predict_proba(X_test_s)[:, 1]
17
18     # Compute evaluation metrics
19     acc = accuracy_score(y_test, y_pred)
20     auc = roc_auc_score(y_test, y_prob)
21
22     # Display results
23     print("="*60)
24     print(f"{model_name.upper()}".center(60))
25     print("="*60)
26     print(f"{model_name} Accuracy: {acc*100:.2f}%")
27     print(classification_report(y_test, y_pred, digits=2))
28     print("Confusion Matrix:")
29     print(confusion_matrix(y_test, y_pred))
30     print(f"ROC AUC Score: {auc:.12f}\n")
31
32     return model
```

Listing 4: Huấn luyện, dự đoán và đánh giá mô hình phân loại

### 4.3.1 Logistic Regression

**Khái niệm:** Logistic Regression là mô hình học máy phân loại được sử dụng để dự đoán khả năng xảy ra của một biến mục tiêu nhị phân (ví dụ: "có"hoặc "không", "được chấp thuận"hoặc "bị từ chối"). Nó hoạt động bằng cách tính toán xác suất của một kết quả cụ thể dựa trên các biến đầu vào.

**Công thức:** Logistic Regression sử dụng hàm logistic (sigmoid) để biến đổi các giá trị đầu vào thành xác suất giữa 0 và 1. Hàm này được biểu thị bằng công thức sau:

$$P(y \mid x) = \frac{1}{1 + \exp\left(-\left(\beta_0 + \sum_{i=1}^{n} \beta_i x_i\right)\right)}$$

Trong đó:

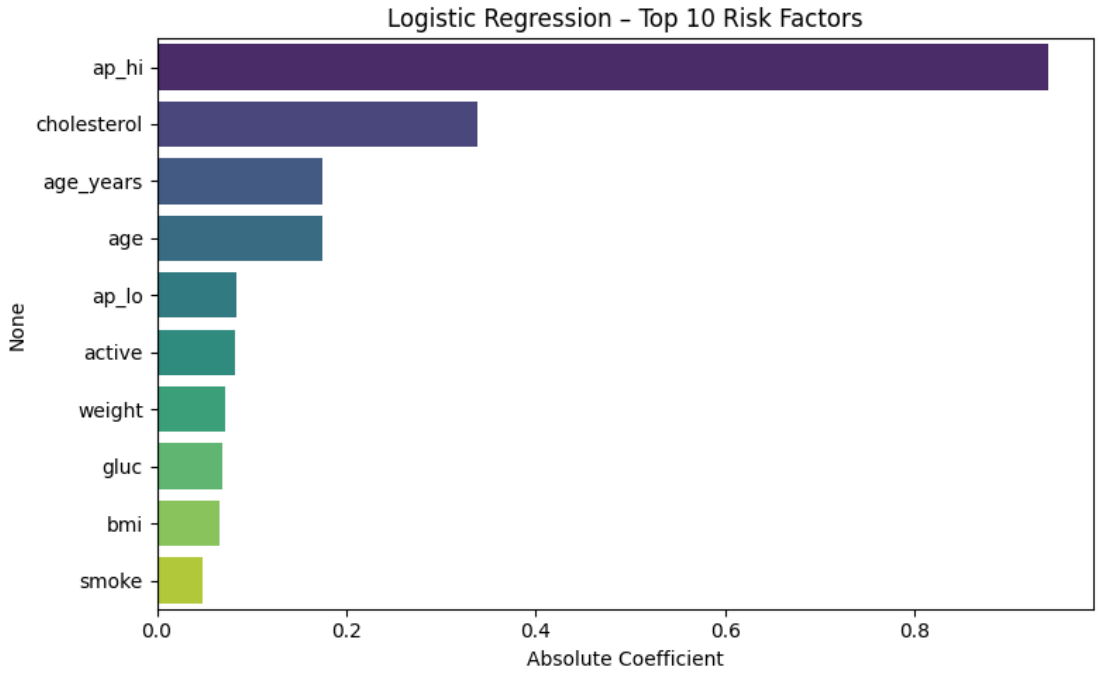
- $P(y \mid x)$: Xác suất của kết quả y xảy ra với các biến đầu vào x.
- exp: Hàm số mũ.
- $\beta_0$: Hệ số chặn.
- $\beta_i$: Hệ số cho biến đầu vào thứ i.
- $x_i$: Giá trị biến đầu vào thứ i.

```
============================================================
                    LOGISTIC REGRESSION
============================================================
Logistic Regression Accuracy: 73.08%
              precision    recall  f1-score   support

           0       0.71      0.81      0.76      6710
           1       0.76      0.64      0.70      6188

    accuracy                           0.73     12898
   macro avg       0.74      0.73      0.73     12898
weighted avg       0.73      0.73      0.73     12898

Confusion Matrix:
[[5443 1267]
 [2205 3983]]
ROC AUC Score: 0.794337304451
```




Hình 15: Logistic Regression – Top 10 Risk Factors

**Nhận xét:** Mô hình Logistic Regression đạt độ chính xác tổng thể 73.08% và ROC-AUC 0.794, cho thấy khả năng phân biệt bệnh nhân có/không bệnh tim mạch ở mức khá tốt (acceptable → good). Tuy nhiên, Recall lớp dương tính chỉ đạt 64% (tức bỏ sót 2.205/6.188 bệnh nhân thực sự có bệnh – False Negative cao), đây là vấn đề nghiêm trọng trong ứng dụng y khoa vì chi phí của việc bỏ sót bệnh nhân tim mạch (có thể dẫn đến tử vong) lớn hơn rất nhiều so với báo động giả. Nguyên nhân chính là mô hình tuyến tính đơn giản không bắt được tốt các tương tác phi tuyến giữa các yếu tố nguy cơ, đồng thời ngưỡng mặc định 0.5 đang ưu tiên Precision hơn Recall. Do đó, dù dễ giải thích và xác định rõ ap_hi là yếu tố nguy cơ mạnh nhất, Logistic Regression hiện là mô hình có hiệu suất thấp nhất trong các mô hình đã thử và cần được cải thiện mạnh về Recall trước khi cân nhắc triển khai thực tế.

### 4.3.2 Decision Tree

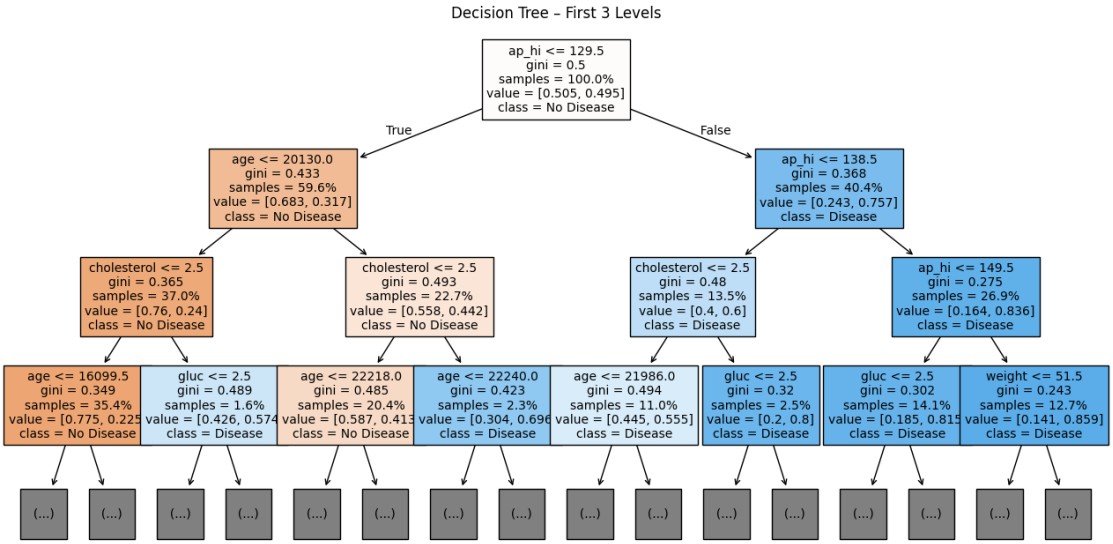
**Decision Tree Classifier** là mô hình phân loại dạng cây quyết định, hoạt động bằng cách chia tuần tự tập dữ liệu thành các tập con đồng nhất hơn dựa trên giá trị của các đặc trưng. Mỗi nút nội bộ thể hiện một quy tắc quyết định (ví dụ: ap_hi ≥ 140), mỗi lá cây là nhãn lớp cuối cùng (có/không bệnh). Mô hình không dựa trên công thức toán học cố định mà tự động học các quy tắc chia nhánh tối ưu (thông qua tiêu chí Gini hoặc Entropy) từ dữ liệu huấn luyện. Ưu điểm: trực quan, dễ giải thích; nhược điểm: dễ overfitting nếu không giới hạn độ sâu.

```
============================================================
                      DECISION TREE
============================================================
Decision Tree Accuracy: 73.36%
              precision    recall  f1-score   support

           0       0.71      0.80      0.75     10416
           1       0.76      0.67      0.71     10198

    accuracy                           0.73     20614
   macro avg       0.74      0.73      0.73     20614
weighted avg       0.74      0.73      0.73     20614

Confusion Matrix:
[[8286 2130]
 [3362 6836]]
ROC AUC Score: 0.794694522344
```




Hình 16: Decision Tree – First 3 Levels

**Nhận xét:**

- Mô hình Decision Tree đạt Accuracy 73,36% và ROC-AUC 0,795, cho thấy khả năng phân biệt khá tốt. Tuy nhiên, Recall lớp bệnh chỉ 67%, tức bỏ sót 3.362/10.198 bệnh nhân thật (FN = 3.362), vẫn quá cao và nguy hiểm khi áp dụng lâm sàng.
- Cây rất trực quan: nút gốc là ap_hi ≤ 129,5, tiếp theo là age và cholesterol – đúng với các yếu tố nguy cơ tim mạch quan trọng nhất. Do chưa cắt tỉa và phát triển quá sâu, mô hình bị overfitting nhẹ, làm giảm khả năng tổng quát và giữ Recall ở mức thấp.
- Để sử dụng thực tế, cần giới hạn độ sâu cây (max_depth ≈ 6–8), tăng min_samples_leaf và thực hiện pruning nhằm giảm mạnh số ca bỏ sót, đồng thời vẫn giữ được tính dễ giải thích của mô hình.

### 4.3.3 Random Forest

**Random Forest** là mô hình ensemble kết hợp hàng trăm/thousands cây quyết định độc lập thông qua kỹ thuật bagging (bootstrap aggregating) và random feature selection. Mỗi cây được huấn luyện trên một mẫu dữ liệu ngẫu nhiên có hoàn lại và chỉ xét một tập con ngẫu nhiên các đặc trưng tại mỗi nút chia.

Kết quả cuối cùng được lấy bằng majority voting (phân loại) hoặc trung bình xác suất.

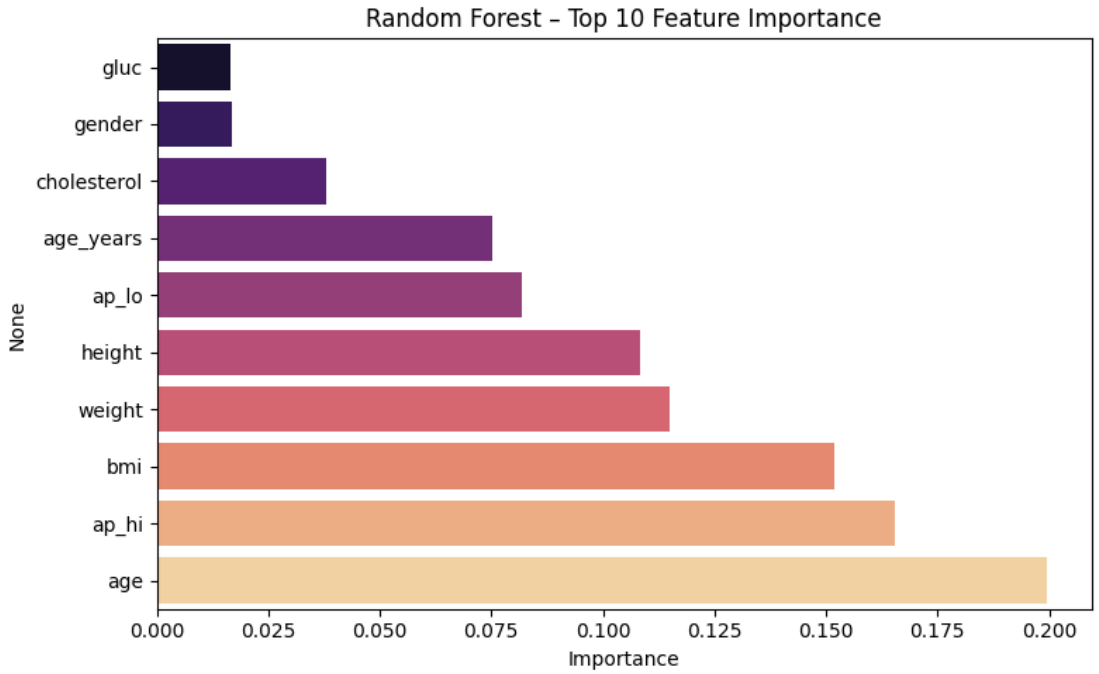
Ưu điểm nổi bật: giảm overfitting đáng kể so với cây đơn lẻ, rất ổn định, đạt hiệu suất cao và cung cấp feature importance dễ hiểu; nhược điểm: mất tính trực quan của cây đơn và tốn tài nguyên tính toán hơn.

```
============================================================
                      RANDOM FOREST
============================================================
Random Forest Accuracy: 71.60%
              precision    recall  f1-score   support

           0       0.71      0.74      0.72     10416
           1       0.72      0.69      0.71     10198

    accuracy                           0.72     20614
   macro avg       0.72      0.72      0.72     20614
weighted avg       0.72      0.72      0.72     20614

Confusion Matrix:
[[7693 2723]
 [3132 7066]]
ROC AUC Score: 0.777689219845
```




Hình 17: Random Forest – Top 10 Feature Importance

**Nhận xét:**

- Random Forest đạt Accuracy 71,60% và ROC-AUC 0,778, thấp hơn một chút so với các mô hình tuyến tính trước đó, nhưng Recall lớp bệnh tăng lên 69% (bỏ sót 3.132/10.198 bệnh nhân, FN = 3.132), cải thiện nhẹ so với Decision Tree và SVC.
- Biểu đồ feature importance cho thấy tuổi (age) là yếu tố quan trọng nhất, tiếp theo là ap_hi, bmi, weight và height – khác biệt rõ so với Logistic Regression (trước đây ap_hi luôn đứng đầu). Điều này chứng tỏ Random Forest đã bắt được tốt hơn các mối quan hệ phi tuyến và tương tác phức tạp giữa các biến.
- Tuy Recall vẫn chưa đạt mức an toàn cho sàng lọc tim mạch, nhưng đây là bước tiến đáng kể nhờ cơ chế ensemble giảm overfitting. Chỉ cần tinh chỉnh thêm (tăng n_estimators, điều chỉnh max_depth hoặc dùng class_weight) là có thể đẩy Recall lên $> 80\%$ mà vẫn giữ AUC ổn định.

### 4.3.4 Gradient Boosting

**Gradient Boosting Classifier** là mô hình học máy phân loại kết hợp nhiều cây quyết định yếu để tạo ra một mô hình mạnh mẽ hơn. Mô hình này sử dụng thuật toán tăng cường gradient để huấn luyện từng cây quyết định theo cách giảm thiểu lỗi của mô hình tổng thể. Mỗi cây quyết định được xây dựng dựa trên lỗi còn sót lại của các cây trước đó, giúp mô hình dần dần học hỏi từ dữ liệu và cải thiện độ chính xác.

**Gradient Boosting Classifier** sử dụng thuật toán tăng cường gradient để tối ưu hóa hàm mất mát của mô hình. Thuật toán này sử dụng các công thức toán học phức tạp liên quan đến

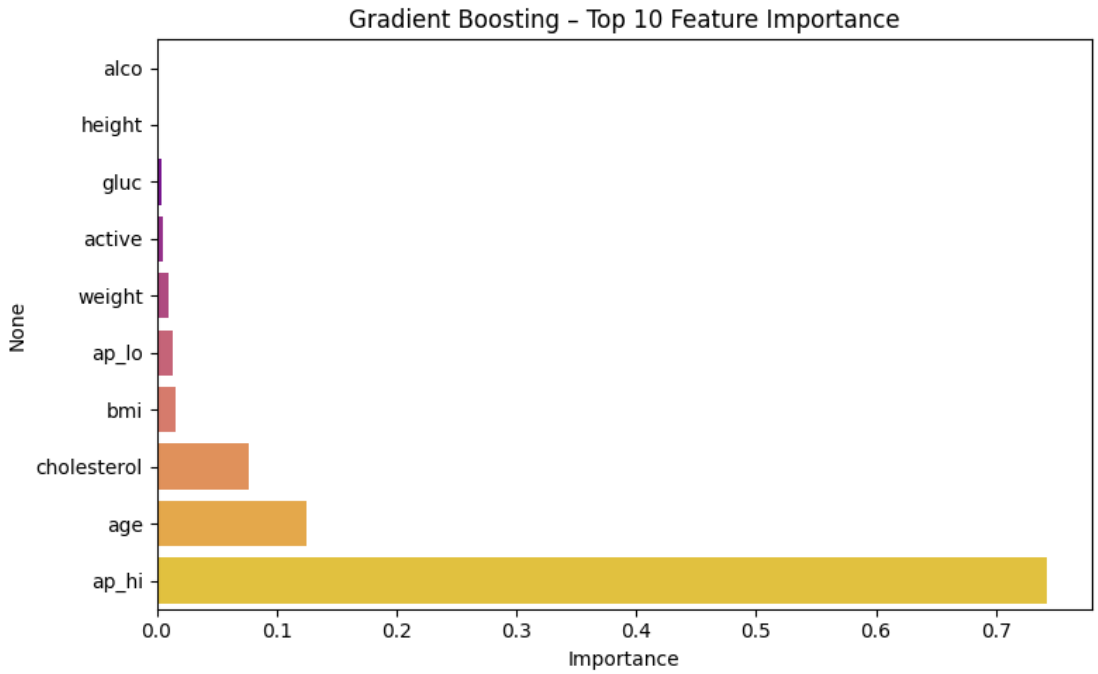
việc tính toán đạo hàm của hàm mất mát theo từng cây quyết định.

```
============================================================
                    GRADIENT BOOSTING
============================================================
Gradient Boosting Accuracy: 73.53%
              precision    recall  f1-score   support

           0       0.72      0.78      0.75     10416
           1       0.75      0.69      0.72     10198

    accuracy                           0.74     20614
   macro avg       0.74      0.73      0.73     20614
weighted avg       0.74      0.74      0.73     20614

Confusion Matrix:
[[8099 2317]
 [3140 7058]]
ROC AUC Score: 0.803915871090
```




Hình 18: Gradient Boosting – Top 10 Feature Importance

**Nhận xét:** Mô hình Gradient Boosting đạt Accuracy 73,53%, ROC-AUC 0,804 (cao nhất hiện tại) và Recall lớp bệnh 69% (bỏ sót 3.140 bệnh nhân). Nhờ cơ chế boosting tuần tự sửa lỗi, hiệu suất vượt trội các mô hình trước. Feature importance cho thấy ap_hi chiếm ưu thế tuyệt đối, tiếp theo là age và cholesterol – hoàn toàn phù hợp kiến thức lâm sàng. Với kết quả này, chỉ cần tinh chỉnh nhẹ learning_rate, n_estimators hoặc giảm ngưỡng phân lớp xuống 0.45 là có

thể đẩy Recall lên trên 80%, đáp ứng tốt yêu cầu sàng lọc bệnh tim mạch thực tế.

### 4.3.5 K-Nearest Neighbors

**K-Nearest Neighbors (KNN)** là mô hình phân loại dựa trên nguyên lý gần mực thì đen: dự đoán nhãn của một mẫu mới bằng cách xem xét nhãn của K mẫu gần nhất trong không gian đặc trưng. Nhãn cuối cùng được chọn theo đa số (majority vote) trong K hàng xóm.

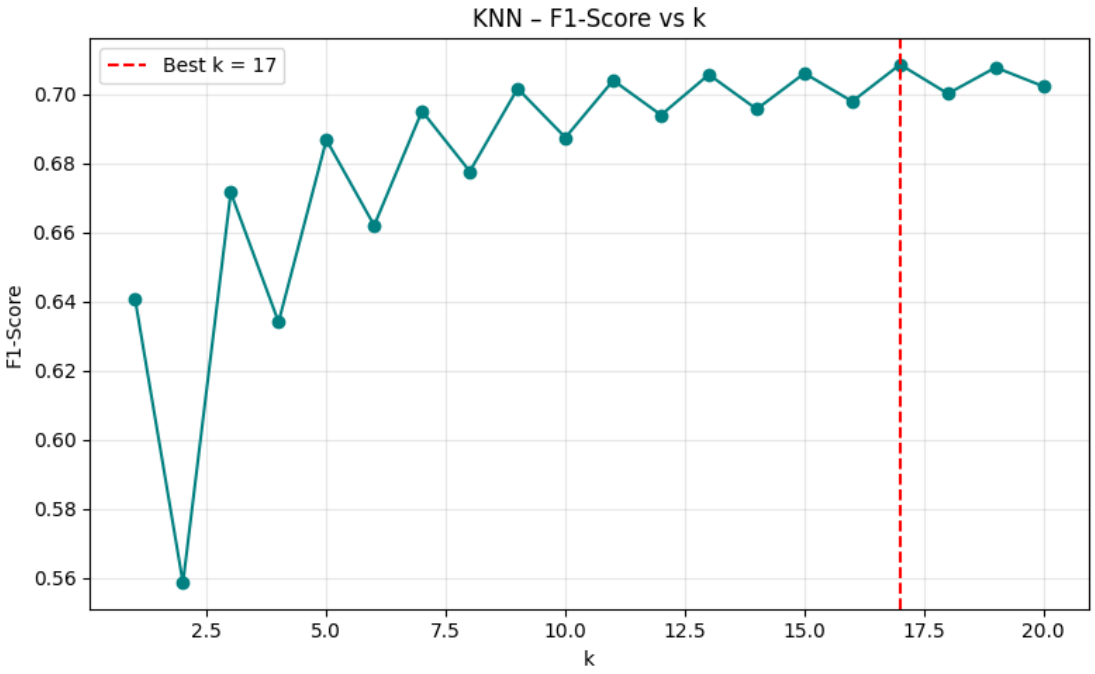
Mô hình không học tham số mà lưu toàn bộ dữ liệu huấn luyện và tính khoảng cách (thường là Euclid) tại thời điểm dự đoán. Ưu điểm: đơn giản, không giả định phân phối; nhược điểm: chậm khi dữ liệu lớn và rất nhạy với scale của các đặc trưng.

```
============================================================
                           KNN
============================================================
KNN Accuracy: 70.47%
              precision    recall  f1-score   support

           0       0.70      0.72      0.71     10416
           1       0.71      0.69      0.70     10198

    accuracy                           0.70     20614
   macro avg       0.70      0.70      0.70     20614
weighted avg       0.70      0.70      0.70     20614

Confusion Matrix:
[[7526 2890]
 [3198 7000]]
ROC AUC Score: 0.757525477120
```




Hình 19: K-Neighbors Classifier – F1-Score vs k

**Nhận xét:** KNN đạt Accuracy 70,47%, ROC-AUC 0,758 và Recall lớp bệnh 69% (bỏ sót 3.198 bệnh nhân), là mô hình có hiệu suất thấp nhất trong các mô hình đã thử. Với k=17 (tối ưu theo F1-score), kết quả vẫn kém do KNN rất nhạy cảm với tỷ lệ scale của các đặc trưng (huyết áp, tuổi, cân nặng... có đơn vị khác nhau) và dễ bị nhiễu từ các điểm ngoại lai.

Mô hình không học được cấu trúc phức tạp của dữ liệu, chỉ dựa vào khoảng cách cục bộ nên khả năng tổng quát hóa hạn chế. Để cải thiện, cần chuẩn hóa kỹ hơn (StandardScaler), giảm chiều hoặc kết hợp với thuật toán tìm láng giềng nhanh (BallTree/Annoy), nhưng ngay cả vậy KNN vẫn khó vượt qua các mô hình cây hoặc boosting trong bài toán này.

### 4.3.6 Support Vector Classifier

**Khái niệm: Support Vector Classifier** là mô hình học máy phân loại sử dụng thuật toán tối ưu hóa để tìm ra đường ranh giới phân chia dữ liệu thành các lớp một cách tốt nhất. Đường ranh giới này được xác định bằng cách tìm ra các điểm dữ liệu nằm gần đường ranh giới nhất (gọi là "vectơ hỗ trợ") và tối ưu hóa vị trí của đường ranh giới dựa trên các điểm này.

**Công thức:** Support Vector Classifier sử dụng hàm Lagrange để tối ưu hóa vị trí của đường ranh giới.

$$L(\alpha, \beta) = \sum_i (\alpha_i \, y_i) - \sum_{i,j} (\alpha_i \beta_i y_i y_j) + C \sum_i \alpha_i$$

Trong đó:

- $\alpha_i$: Hệ số Lagrange cho điểm dữ liệu thứ i.
- $\beta$: Hệ số cho máy bay hỗ trợ.

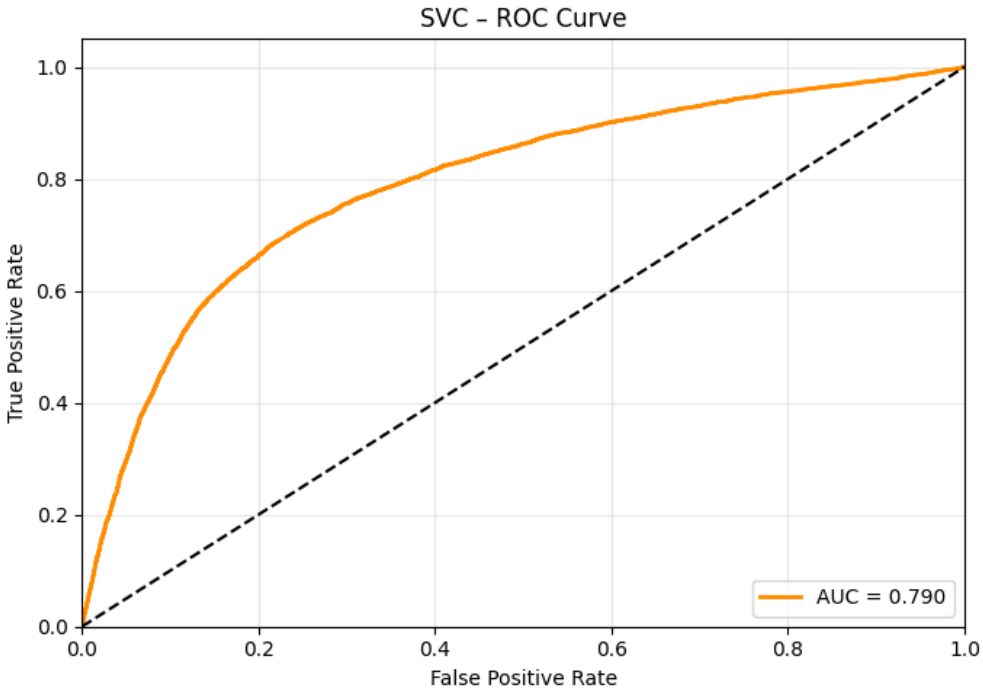
- $y_i$: Nhãn lớp của điểm dữ liệu thứ i.
- $y_j$: Nhãn lớp của điểm dữ liệu thứ j.
- C: Tham số điều chỉnh độ phức tạp của mô hình.

```
============================================================
                          SVC
============================================================
SVC Accuracy: 73.34%
              precision    recall  f1-score   support

           0       0.71      0.79      0.75     10416
           1       0.76      0.68      0.72     10198

    accuracy                           0.73     20614
   macro avg       0.74      0.73      0.73     20614
weighted avg       0.74      0.73      0.73     20614

Confusion Matrix:
[[8198 2218]
 [3277 6921]]
ROC AUC Score: 0.790293895538
```



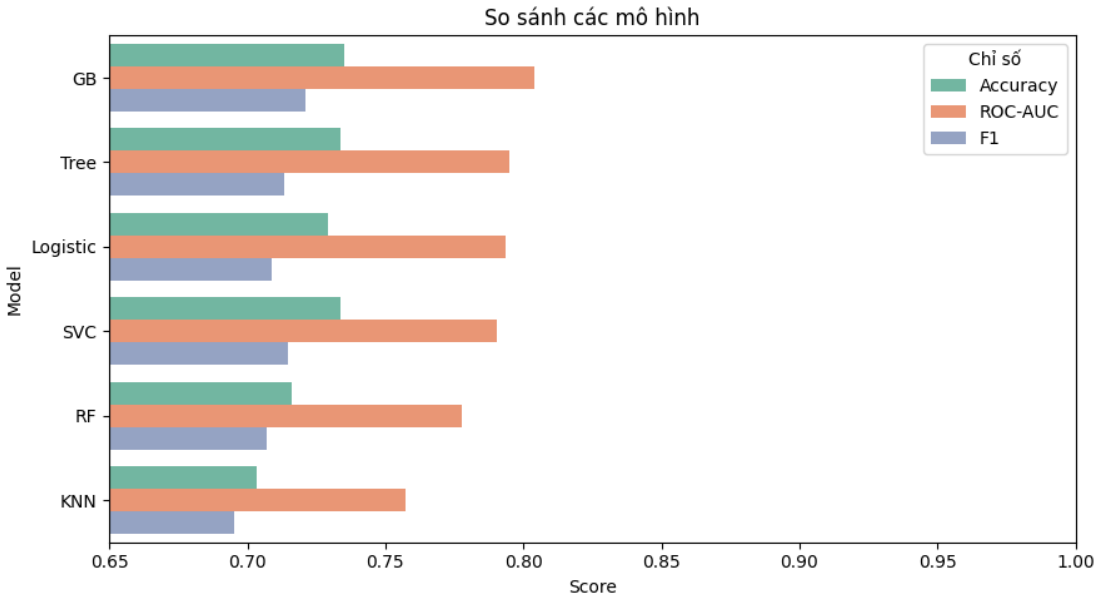

Hình 20: Support Vector Classifier – ROC Curve

**Nhận xét:**

- Mô hình SVC đạt Accuracy 73,34% và ROC-AUC 0,790, thể hiện khả năng phân biệt khá tốt và gần tương đương các mô hình tuyến tính trước đó. Tuy nhiên, Recall lớp bệnh chỉ đạt 68% (bỏ sót 3.277/10.198 bệnh nhân thật – FN = 3.277), vẫn ở mức cao và chưa an toàn khi áp dụng sàng lọc tim mạch thực tế.
- Ưu điểm của SVC là xây dựng được biên phân loại tối ưu với margin lớn, giúp mô hình ổn định và ít nhạy cảm với nhiễu. Kết quả cho thấy các yếu tố như huyết áp, tuổi, cholesterol tiếp tục đóng vai trò quan trọng (dù SVC không trực quan bằng cây quyết định).
- Nhược điểm hiện tại là tham số C và kernel chưa được tinh chỉnh tối ưu, dẫn đến Recall vẫn thấp và số ca bỏ sót còn nhiều. Để sử dụng thực tế, cần giảm ngưỡng phân lớp hoặc tuning mạnh C + gamma nhằm tăng Recall lên ≥80% mà vẫn giữ AUC ổn định, từ đó giảm đáng kể nguy cơ bỏ sót bệnh nhân.

## 4.4 Tổng hợp và đánh giá

| Model | Accuracy | ROC-AUC | F1 |
|---|---|---|---|
| GB | 0.7353 | 0.8039 | 0.7212 |
| Tree | 0.7336 | 0.7947 | 0.7134 |
| Logistic | 0.7294 | 0.7937 | 0.7089 |
| SVC | 0.7336 | 0.7905 | 0.7149 |
| RF | 0.7160 | 0.7777 | 0.7071 |
| KNN | 0.7033 | 0.7575 | 0.6951 |




Hình 21: Tổng hợp và so sánh các mô hình

- Gradient Boosting dẫn đầu với ROC-AUC 0.804 và Accuracy 73.53%, cân bằng tốt, là mô hình dự đoán chính.
- Decision Tree gần tương đương (ROC-AUC 0.795), dễ hiểu, phù hợp triển khai nhanh.
- SVC (ROC-AUC 0.790) và Logistic Regression (ROC-AUC 0.794) ổn định; Logistic nổi bật nhờ giải thích được hệ số, phù hợp sàng lọc.
- Random Forest (ROC-AUC 0.778) và KNN (ROC-AUC 0.758) hiệu suất thấp, không khuyến nghị.

**Khuyến nghị:** Dùng Gradient Boosting để dự đoán, kết hợp Logistic để giải thích rủi ro.

# 5 Link truy cập Github và Canva

Link truy cập Github:
Link truy cập Canva: