

**TRƯỜNG ĐẠI HỌC MỞ THÀNH PHỐ HỒ CHÍ MINH**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI BÁO CÁO**

**PHÂN TÍCH TÌNH TRẠNG, SỨC KHỎE CỦA NHỮNG NGƯỜI  
CÓ NGUY CƠ MẮC BỆNH ĐAU TIM**

**Tên thành viên báo cáo:**

Trương Đình Nhật Cường - 2151050045

Trịnh Tông Hiệp - 2151050138

Lê Tấn Đạt - 2151050087

**Giảng viên hướng dẫn:** Nguyễn Văn Bảy

## MỤC LỤC

<b>A. PHẦN MỞ ĐẦU.....</b>	<b>2</b>
1. Lý do chọn chủ đề.....	2
2. Giới thiệu bệnh tim mạch.....	3
<b>B. PHẦN BÁO CÁO.....</b>	<b>4</b>
I. CHUẨN BỊ DỮ LIỆU.....	4
1. Bộ dữ liệu được sử dụng.....	4
2. Mô tả thuộc tính.....	4
3. Nhập thư viện và tải tập dữ liệu.....	6
4. Một số thông tin cơ bản về tập dữ liệu.....	6
5. Mô tả thống kê.....	8
6. Làm sạch dữ liệu.....	9
II. PHÂN TÍCH DỮ LIỆU KHÁM PHÁ.....	11
1. Phân tích đơn biến.....	11
Hình 1.1. Biểu đồ tỉ lệ người mắc bệnh và không mắc bệnh tim mạch	
.....	11
Hình 1.2. Biểu đồ thể hiện số lượng người mắc bệnh tim mạch và	
không mắc bệnh tim mạch.....	12
Hình 1.3. Biểu đồ phân bố các đặc điểm phân loại.....	14
Hình 1.4. Biểu đồ phân phối xác suất.....	16
2. Phân tích hai biến.....	17
2.1 Correlation matrix.....	17
Hình 2.1. Ma trận tương quan giữa các đặc trưng.....	18
Hình 2.2. Ma trận tương quan của các đặc trưng với ‘Target’ .....	19
2.2 Box_Plot.....	20
Hình 2.3. Biểu đồ hộp.....	21
Hình 2.4 Pair-plot.....	22
III. HUẤN LUYỆN DỮ LIỆU.....	24
1. Xử lý và điều chỉnh dữ liệu chuẩn bị cho việc huấn luyện.....	24
2. Đào tạo và phân chia dữ liệu.....	24
3. Huấn luyện dữ liệu với các mô hình cơ sở.....	25
4. Điều chỉnh siêu tham số và báo cáo phân loại.....	26
IV. PHÂN CỤM BỆNH NHÂN TIM MẠCH BẰNG PHÂN CỤM K MEANS	
.....	26
1. Phân tích phân cụm.....	26
Hình 4.1. Biểu đồ Elbow.....	28

Hình 4.2. Biểu đồ phân tán trực quan hóa mối quan hệ giữa Age và Thalach.....	29
2. Gom cụm.....	29
Hình 4.3. Biểu đồ gom cụm giữa 'Age' và 'Trestbps' .....	30
V. THUẬT TOÁN PHÂN LOẠI.....	30
• Confusion Matrix.....	30
Hình 5.1. Biểu đồ Ma trận hỗn loạn.....	31
• Features Importance.....	32
Hình 5.2. Biểu đồ Top 10 thuộc tính quan trọng.....	33
VI. CÂY QUYẾT ĐỊNH.....	33
1. Mô tả lại thuộc tính.....	33
2. Tạo biến ảo.....	35
3. Tách dữ liệu thành tập huấn luyện và tập thử nghiệm.....	35
4. Mô hình cuối cùng.....	36
5. Dự đoán mẫu dữ liệu mới.....	37
6. Trực quan hóa cây quyết định.....	37
Hình 6.1. Biểu đồ Cây quyết định.....	38
7. Độ chính xác của cây quyết định.....	39
VII. TỔNG KẾT.....	40
1. Kết quả thu được.....	40
2. Hạn chế.....	41
3. Ứng dụng.....	41

## **A. PHẦN MỞ ĐẦU**

### **1. Lý do chọn chủ đề**

Bệnh tim mạch đứng trong số những mối nguy hiểm đáng kể trên toàn cầu, không chỉ là một căn bệnh đơn lẻ mà còn là nguyên nhân của nhiều vấn đề sức khỏe nghiêm trọng khác như tiểu đường, mù lòa, và nhiều bệnh lý khác. Thường thì quá trình chuẩn đoán bệnh tiểu đường yêu cầu bệnh nhân phải đến trung tâm y tế, tìm kiếm ý kiến bác sĩ và chờ đợi một khoảng thời gian đáng kể để nhận được bản báo cáo chuẩn đoán.

Một trong những lý do chính là bệnh tim mạch không chỉ là một căn bệnh cụ thể, mà còn đóng vai trò quan trọng như một "nguyên nhân đa nhiệm" gây ra nhiều vấn đề sức khỏe khác. Nếu không được phát hiện và điều trị kịp thời, bệnh tim mạch có thể dẫn đến nhiều hệ lụy nghiêm trọng như bệnh tim, mù lòa và nhiều bệnh lý khác. Do đó, tập trung vào việc chuẩn đoán sớm và hiệu quả bệnh tim mạch có thể đồng thời giảm nguy cơ phát triển các vấn đề sức khỏe phụ khác.

### **2. Giới thiệu bệnh tim mạch**

Bệnh tim mạch, hay còn được gọi là các rối loạn tim mạch, là một trong những vấn đề sức khỏe lớn đang ngày càng gia tăng trên toàn cầu. Bệnh tim mạch không chỉ đơn thuần là một căn bệnh đặc trưng, mà còn đóng vai trò quan trọng là nguyên nhân chính gây ra nhiều vấn đề sức khỏe nghiêm trọng khác nhau. Đây là một tình trạng mà những tế bào và cơ bắp của tim không nhận đủ lượng máu, dẫn đến những biến đổi lớn trong chức năng tim.

Bệnh tim mạch bao gồm nhiều loại khác nhau như đau thắt ngực, nhồi máu cơ tim, và nhồi máu mạch máu não. Nguyên nhân chủ yếu thường là do tắc nghẽn mạch máu do chất béo, xơ vữa, và các tác nhân

khác. Đối diện với những thách thức này, việc hiểu rõ về bệnh tim mạch, nguyên nhân, và các yếu tố nguy cơ là quan trọng để phòng ngừa và quản lý tình trạng sức khỏe này một cách hiệu quả.

## B. PHẦN BÁO CÁO

### I. CHUẨN BỊ DỮ LIỆU

#### 1. Bộ dữ liệu được sử dụng

**Bộ dữ liệu:** Dữ liệu được lấy và tham khảo trên trang <https://www.kaggle.com>

**Nội dung bộ dữ liệu:** Bộ dữ liệu này chứa các thuộc tính như tuổi, giới tính, loại công việc, huyết áp, hạng mục cholesterol, đường huyết, đau ngực, đồng vị cân bằng thallium, tình trạng mạch và nhiều thuộc tính khác. Các thuộc tính này được sử dụng để phân loại và dự đoán khả năng mắc bệnh đau tim.

#### 2. Mô tả thuộc tính

- Age : Tuổi của bệnh nhân
- Sex : Giới tính của bệnh nhân
- cp : Loại đau ngực
  - o value 1: Đau ngực điển hình
  - o value 2: Đau ngực không điển hình
  - o value 3: Đau ngực không phải do tim
  - o value 4: Không có triệu chứng đau
- Trestbps : Huyết áp nghỉ (tính bằng mm/Hg)
- Chol : Cholesterol trong máu (tính bằng mg/dl)
- Fbs : (Đường huyết nhanh > 120 mm/dl) (đúng = 1, sai = 0)
- Restecg : Kết quả điện tâm đồ nghỉ
  - o value 0: Bình thường
  - o value 1: Có bất thường sóng ST-T (đảo ngược sóng T và/hoặc tăng hoặc giảm ST > 0.05 mV)

- o value 2: dự đoán hoặc xác định tăng thể tích tử cung trái theo tiêu chí của Estes
- Thalach: Nhịp tim tối đa đạt được
- Exang : Tình trạng đau ngực do vận động (có = 1, không = 0)
- Oldpeak : chênh xuống do luyện tập so với nghỉ ngơi
- Slope :
- Ca : Số mạch chủ chính (0-3)
- Thal :
  - o value 0 : Bình thường
  - o value 1 : Khuyết tật cố định
  - o value 2 : Khuyết tật có thể đảo ngược
- Target :
  - o value 0: Nguy cơ đau tim ít
  - o value 1: Nguy cơ đau tim cao

### 3. Nhập thư viện và tải tập dữ liệu

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split, GridSearchCV, StratifiedKFold, cross_val_score
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier, AdaBoostClassifier
import xgboost as xg
from sklearn.metrics import accuracy_score, recall_score, precision_score, f1_score, roc_auc_score
from sklearn.metrics import confusion_matrix, classification_report, RocCurveDisplay, ConfusionMatrixDisplay
from sklearn.base import clone

import warnings
# Ignore warnings
warnings.filterwarnings('ignore')
```

```
data=pd.read_csv('/content/drive/MyDrive/Khai phá dữ liệu/data/heart.csv')
data.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

### 4. Một số thông tin cơ bản về tập dữ liệu

Bộ dữ liệu ‘heart’ này chứa:

- thông tin từ 303 bệnh nhân đau tim.
- Chúng ta có 14 biến trong đó có 13 biến độc lập và 1 biến phụ thuộc là đầu ra.
- Chúng tôi có 9 biến phân loại: sex, cp, fbs, restecg, exng, slp, ca, thal, target
- Chúng ta có 5 biến số: age, trtbps, chol, thalach, oldpeak



```
[54] data.shape
```

```
(303, 14)
```

```
[55] data.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 303 entries, 0 to 302  
Data columns (total 14 columns):  
#   Column      Non-Null Count  Dtype  
---  ---  
0   age         303 non-null    int64  
1   sex         303 non-null    int64  
2   cp          303 non-null    int64  
3   trestbps    303 non-null    int64  
4   chol        303 non-null    int64  
5   fbs         303 non-null    int64  
6   restecg     303 non-null    int64  
7   thalach     303 non-null    int64  
8   exang       303 non-null    int64  
9   oldpeak     303 non-null    float64  
10  slope       303 non-null    int64  
11  ca          303 non-null    int64  
12  thal        303 non-null    int64  
13  target      303 non-null    int64  
dtypes: float64(1), int64(13)  
memory usage: 33.3 KB
```

## 5. Mô tả thống kê



```
data.describe().T
```

	count	mean	std	min	25%	50%	75%	max
age	303.0	54.366337	9.082101	29.0	47.5	55.0	61.0	77.0
sex	303.0	0.683168	0.466011	0.0	0.0	1.0	1.0	1.0
cp	303.0	0.966997	1.032052	0.0	0.0	1.0	2.0	3.0
trestbps	303.0	131.623762	17.538143	94.0	120.0	130.0	140.0	200.0
chol	303.0	246.264026	51.830751	126.0	211.0	240.0	274.5	564.0
fbs	303.0	0.148515	0.356198	0.0	0.0	0.0	0.0	1.0
restecg	303.0	0.528053	0.525860	0.0	0.0	1.0	1.0	2.0
thalach	303.0	149.646865	22.905161	71.0	133.5	153.0	166.0	202.0
exang	303.0	0.326733	0.469794	0.0	0.0	0.0	1.0	1.0
oldpeak	303.0	1.039604	1.161075	0.0	0.0	0.8	1.6	6.2
slope	303.0	1.399340	0.616226	0.0	1.0	1.0	2.0	2.0
ca	303.0	0.729373	1.022606	0.0	0.0	0.0	1.0	4.0
thal	303.0	2.313531	0.612277	0.0	2.0	2.0	3.0	3.0
target	303.0	0.544554	0.498835	0.0	0.0	1.0	1.0	1.0

Nhận xét:

Age

- Độ tuổi trung bình trong tập dữ liệu là 54,5 tuổi
- Người lớn tuổi nhất là 77 tuổi, người trẻ nhất là 29 tuổi.

Cholesterol:

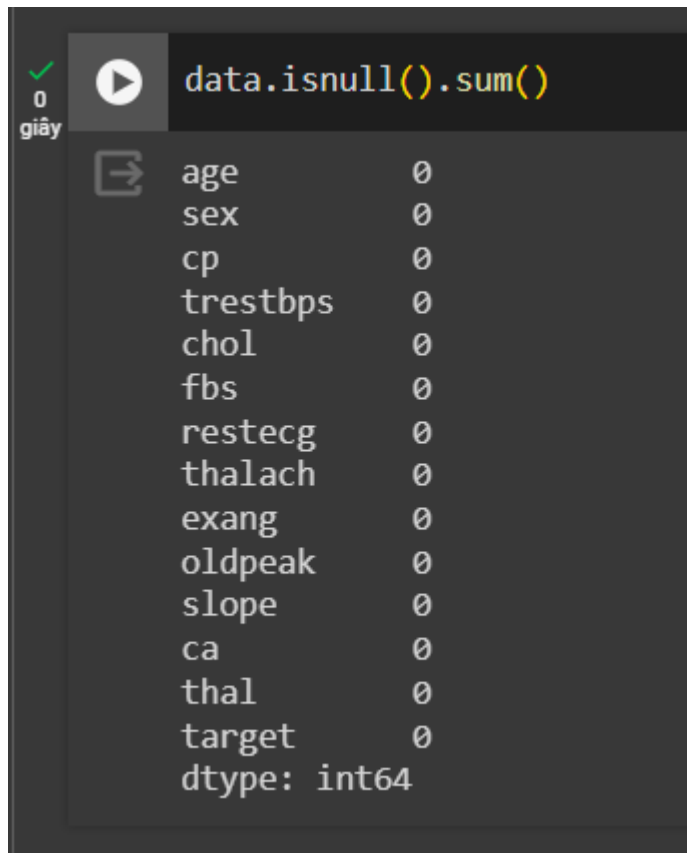
- Mức cholesterol đăng ký trung bình là 246,26
- Cấp độ tối đa là 564 và cấp độ tối thiểu là 126.
- Mức cholesterol khỏe mạnh là  $< 200\text{mg/dl}$  và thường mức cholesterol cao có liên quan đến bệnh tim.

Trestbps : Trung bình 131, tối đa 200 và 94 phút

Thalach : Nhịp tim tối đa trung bình được đăng ký là 149,6 bpm. Tối đa và tối thiểu lần lượt là 202 và 71 bpm.

## 6. Làm sạch dữ liệu

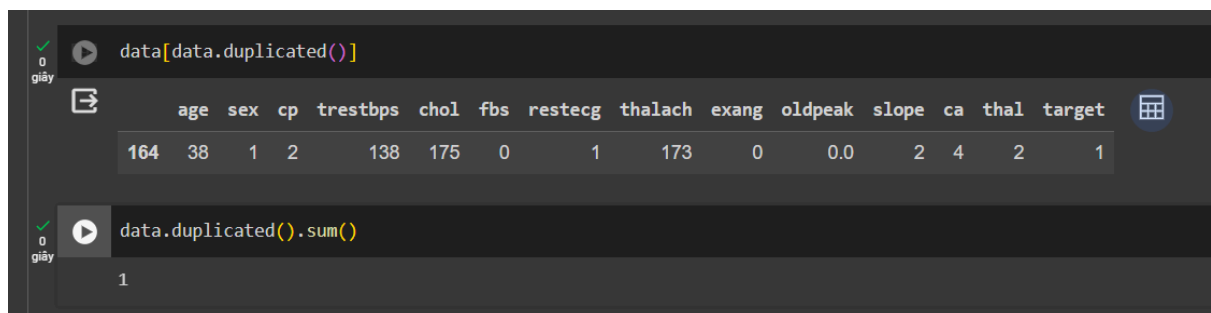
- Kiểm tra giá trị null



A Jupyter Notebook cell showing the execution of `data.isnull().sum()`. The output is a Series with 14 variables, all having a sum of 0, indicating no missing values. The dtype is int64.

```
data.isnull().sum()
age      0
sex      0
cp       0
trestbps 0
chol     0
fbs      0
restecg  0
thalach  0
exang    0
oldpeak  0
slope    0
ca       0
thal     0
target   0
dtype: int64
```

- Kiểm tra dữ liệu trùng lặp



A Jupyter Notebook cell showing the execution of `data[data.duplicated()]`. The output is a DataFrame with 14 columns and 1 row, representing a duplicate record. Below it, another cell shows `data.duplicated().sum()` returning 1.

```
data[data.duplicated()]
  age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  slope  ca  thal  target
164  38   1   2      138   175    0         1     173     0       0.0    2   4      2         1

data.duplicated().sum()
1
```

Có 1 dữ liệu trùng lặp

- Xóa các dữ liệu bị trùng lặp

✓ 0 giây [▶] `data = data.drop_duplicates()`

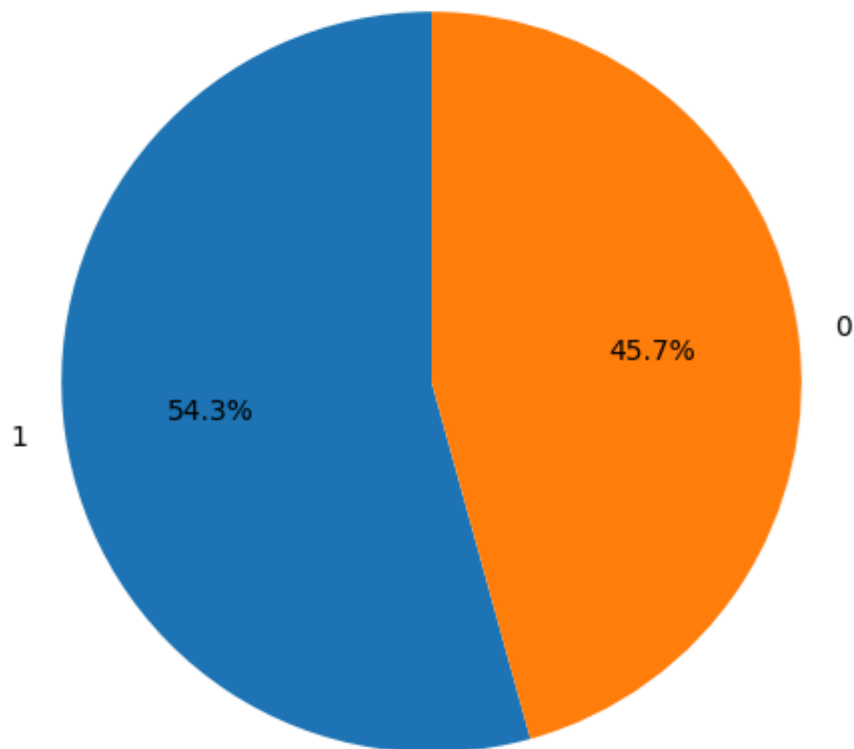
✓ 0 giây [▶] `data.info()`

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 302 entries, 0 to 302
Data columns (total 14 columns):
#   Column      Non-Null Count  Dtype
---  -
0   age         302 non-null   int64
1   sex         302 non-null   int64
2   cp          302 non-null   int64
3   trestbps    302 non-null   int64
4   chol        302 non-null   int64
5   fbs         302 non-null   int64
6   restecg     302 non-null   int64
7   thalach     302 non-null   int64
8   exang       302 non-null   int64
9   oldpeak     302 non-null   float64
10  slope       302 non-null   int64
11  ca          302 non-null   int64
12  thal        302 non-null   int64
13  target      302 non-null   int64
dtypes: float64(1), int64(13)
memory usage: 35.4 KB
```

## II. PHÂN TÍCH DỮ LIỆU KHÁM PHÁ

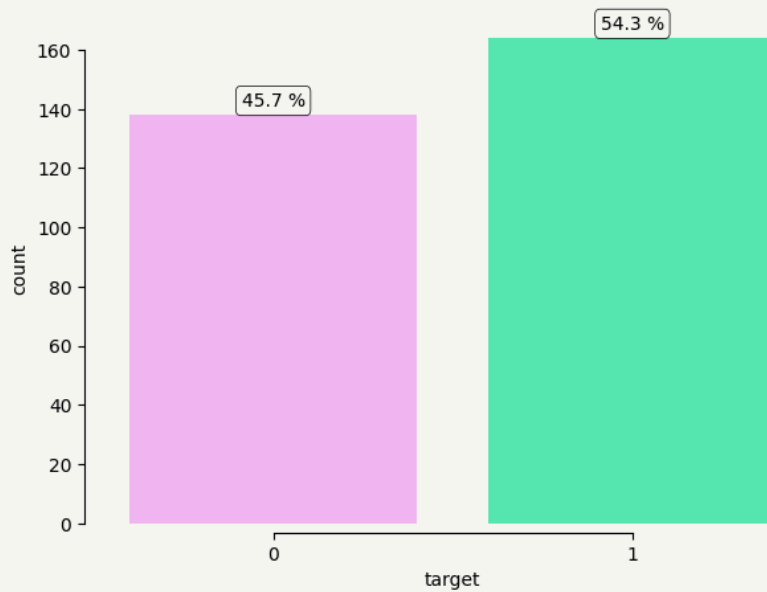
### 1. Phân tích đơn biến

Tỉ lệ bệnh nhân có nguy cơ nhồi máu cơ tim



*Hình 1.1. Biểu đồ tỉ lệ người mắc bệnh và không mắc bệnh tim mạch*

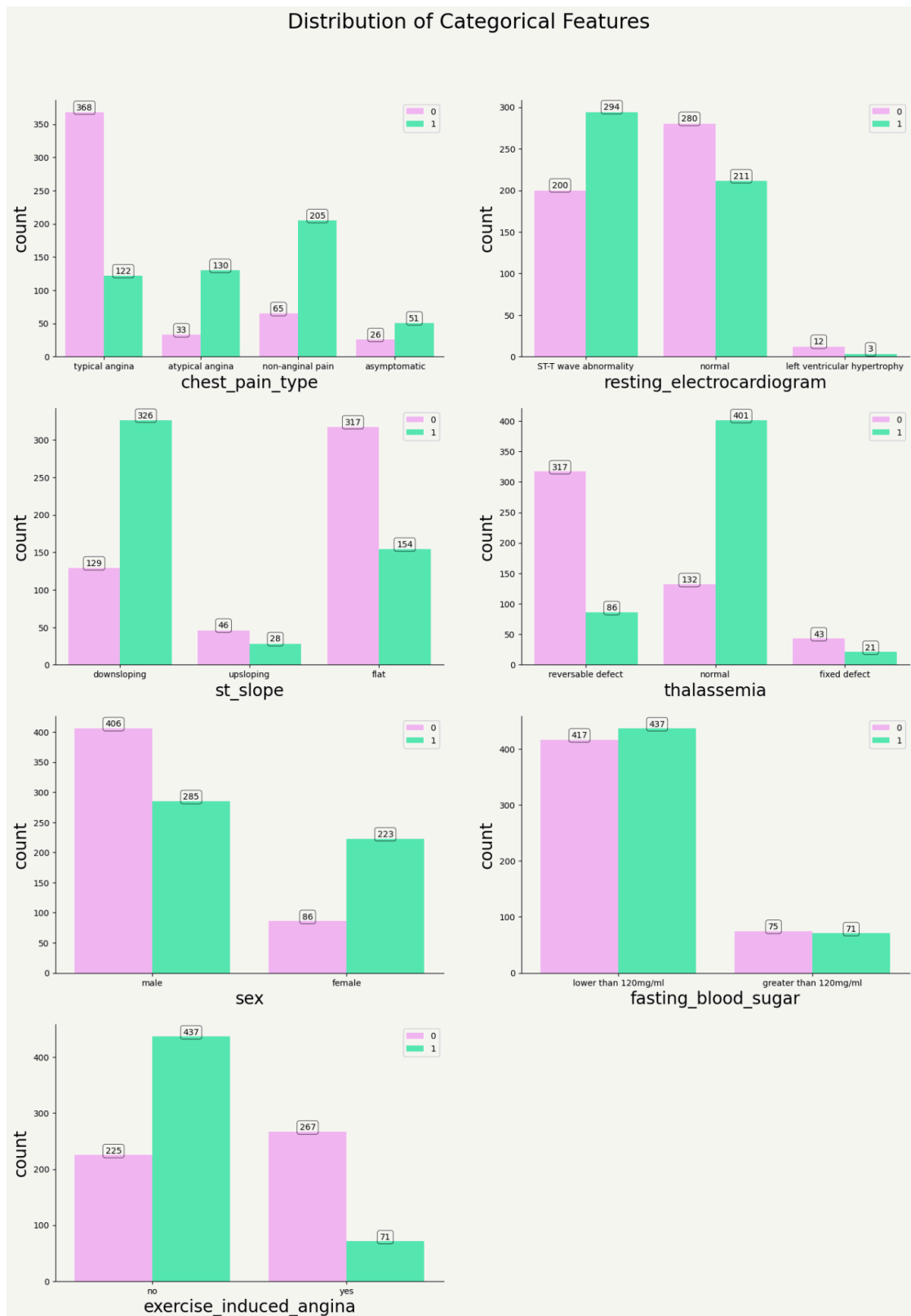
### Số lượng người có nguy cơ đau tim cao và nguy cơ đau tim thấp



a.

*Hình 1.2. Biểu đồ thể hiện số lượng người mắc bệnh tim mạch và không mắc bệnh tim mạch*

**Nhận xét:** Từ biểu đồ cho thấy được tỉ lệ người bị mắc nguy cơ nhồi máu cơ tim đang cao hơn người ít bị nguy cơ nhồi máu cơ tim. Tập dữ liệu này thu thập được cho thấy sức khỏe của con người có nguy cơ đi xuống và mất tỉ lệ cân bằng nhưng không quá đáng kể



*Hình 1.3. Biểu đồ phân bố các đặc điểm phân loại*

## **Nhận xét:**

### **Chest\_pain\_type**

- Bệnh đau thắt ngực: Theo thống kê 390 người, có nguy cơ cao mắc phải chiếm 122, số còn lại không có nguy cơ
- Bệnh đau thắt ngực không điển hình: Theo thống kê có 163 người, trong đó có 130 người có nguy cơ cao mắc phải.
- Bị đau tức ngực không phải do tim: Thống kê có 370 người , trong đó có 205 người có nguy cơ cao mắc phải.
- Không có triệu chứng đau: Theo thống kê có 77 người, trong đó có 51 người có nguy cơ cao bị mắc phải.

### **Resting\_electrocardiogram**

- Có bất thường sóng ST-T : Theo thống kê có 494 người, trong đó có 294 người có nguy cơ mắc phải cao.
- Bình thường: Theo thống kê có 491 người, trong đó có 211 người có nguy cơ bị mắc bệnh cao, 280 người còn lại không có nguy cơ mắc phải.
- Tăng thể tích tử cung trái theo tiêu chí của Estes: Theo thống kê có 15 người trong đó chỉ có 3 người có nguy cơ mắc bệnh cao.

### **ST-Slope**

- Dốc xuống: Theo thống kê có 455 ca, trong đó có 329 ca có nguy cơ cao mắc phải.
- Dốc lên: Theo thống kê có 64 ca, trong đó có 28 ca có nguy cơ cao mắc phải.
- Phẳng: Theo thống kê có 471 ca, trong đó có 154 ca có nguy cơ mắc phải cao

### **Thalassemia**

- Bình thường: Theo thống kê có 403, trong đó có 86 ca có nguy cơ cao bị mắc bệnh
- Khuyết tật cố định: Theo thống kê có 533, trong đó có 86 ca có nguy cơ cao bị mắc bệnh



- Khuyết tật có thể đảo ngược: Theo thống kê có 64, trong đó có 86 ca có nguy cơ cao bị mắc bệnh

### **Sex**

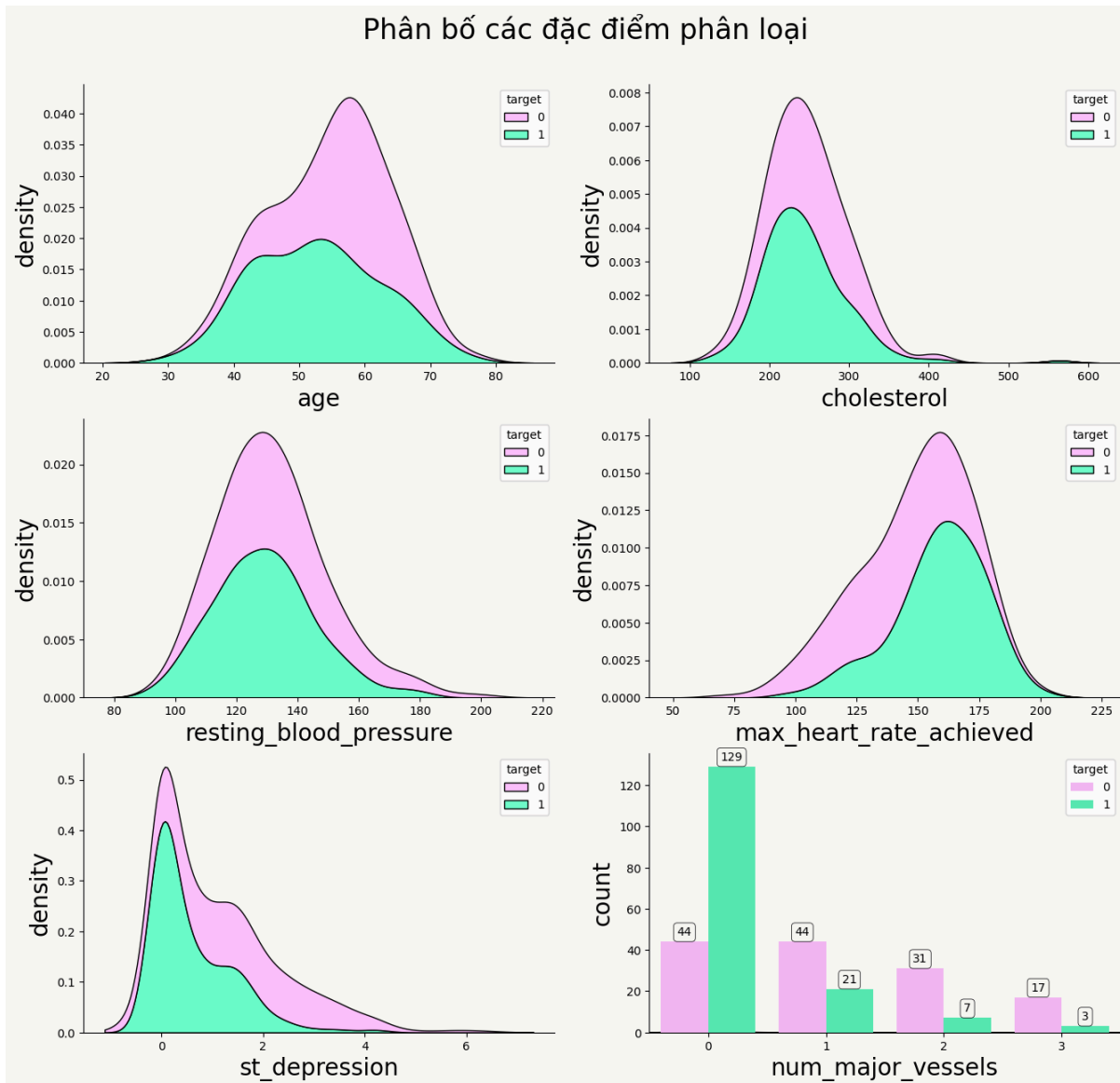
- Nam: Theo thống kê có 691, trong đó có 285 ca có nguy cơ cao bị mắc bệnh
- Nữ: Theo thống kê có 309, trong đó có 223 ca có nguy cơ cao bị mắc bệnh

### **Fasting\_blood\_sugar**

- Đường huyết nhanh  $> 120$  mm/dl: Theo thống kê có 854, trong đó có 437 ca có nguy cơ cao bị mắc bệnh
- Đường huyết nhanh  $< 120$  mm/dl: Theo thống kê có 146, trong đó có 71 ca có nguy cơ cao bị mắc bệnh

### **Exercise\_induced\_angina ( Tập thể dục gây đau thắt ngực)**

- Có: Theo thống kê có 662, trong đó có 437 ca có nguy cơ cao bị mắc phải.
- Không: Theo thống kê có 338, trong đó có 71 ca có nguy cơ cao bị mắc bệnh



*Hình 1.4. Biểu đồ phân phối xác suất*

## **Nhận xét các chỉ số về người ít có khả năng bị bệnh tim**

### **Age:**

Biểu đồ này cho thấy sự phân bố của biến độ tuổi trong tập dữ liệu. Từ biểu đồ ta thấy tỷ trọng của những người có nguy cơ đau tim cao thường tập trung ở độ tuổi từ 40 đến 60. Vì vậy biểu đồ cho thấy những người có nguy cơ đau tim cao có xu hướng ở độ tuổi cao.

### **Cholesterol:**

Biểu đồ này cho thấy sự phân bố của biến nồng độ cholesterol trong máu trong tập dữ liệu.

Biểu đồ này cho thấy người có nguy cơ đau tim cao có xu hướng có nồng độ cholesterol cao hơn so với những người có nguy cơ đau tim thấp.

**Restricting Blood Pressure:**

Biểu đồ này cho thấy sự phân bố của biến `resting_blood_pressure` trong tập dữ liệu.

Biểu đồ này cho thấy rằng những người có huyết áp lúc nghỉ ngơi cao hơn có nguy cơ đau tim cao hơn.

**Max Heart Rate Achieved:**

Biểu đồ này cho thấy sự phân bố của biến `max_heart_rate_achieved` trong tập dữ liệu.

Biểu đồ này cho thấy rằng những người có nhịp tim tối đa đạt cao hơn có nguy cơ đau tim cao hơn.

**ST Depression:**

Biểu đồ này cho thấy sự phân bố của biến `st_depression` trong tập dữ liệu.

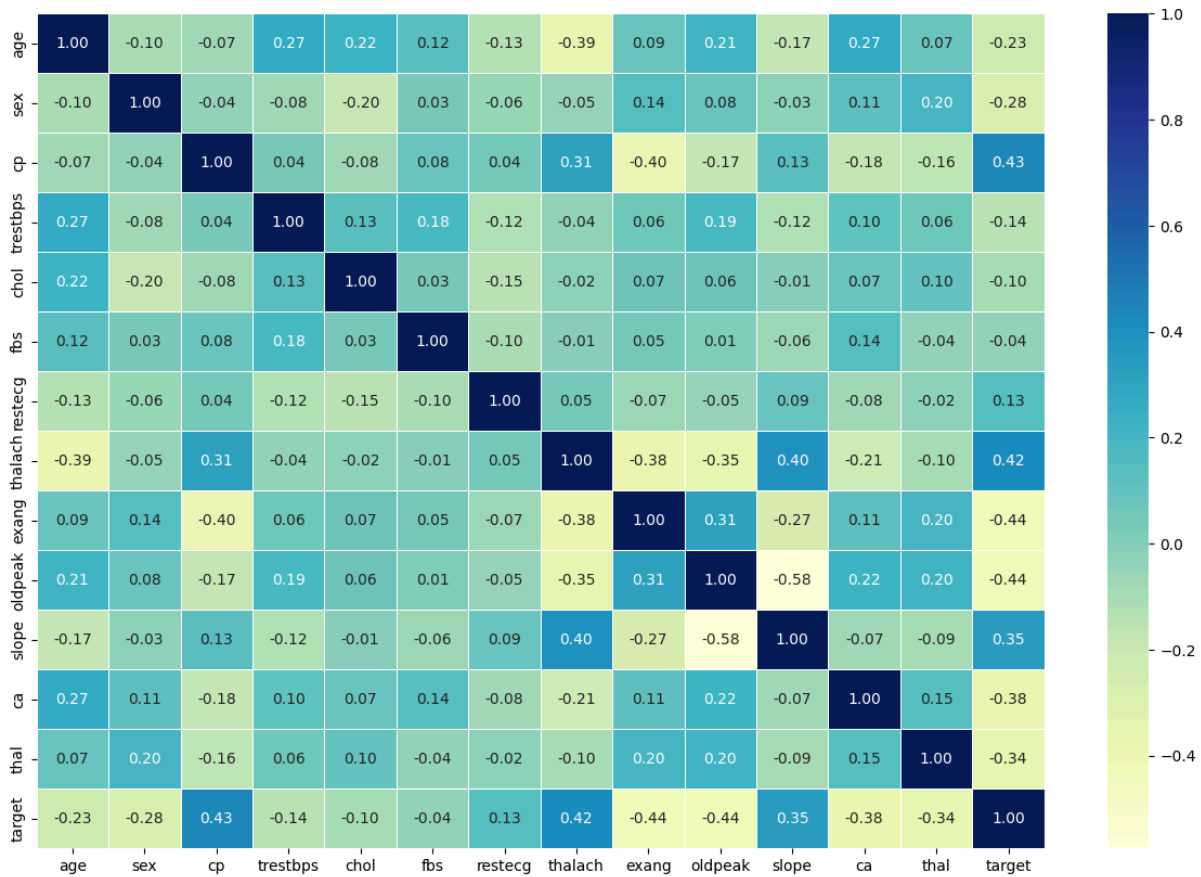
Biểu đồ cho thấy có nhiều bệnh nhân bị đau tim có ST Depression cao hơn so với những bệnh nhân không bị đau tim..

**Num Major Vessels:**

Biểu đồ trên cho thấy mối quan hệ giữa số lượng mạch máu chính và nguy cơ đau tim. Có thể thấy rằng, những người có số lượng mạch máu chính càng cao thì nguy cơ đau tim càng cao. Cụ thể, những người có 0 mạch máu chính có nguy cơ đau tim thấp nhất, trong khi những người có 3 mạch máu chính có nguy cơ đau tim cao nhất.

## 2. Phân tích hai biến

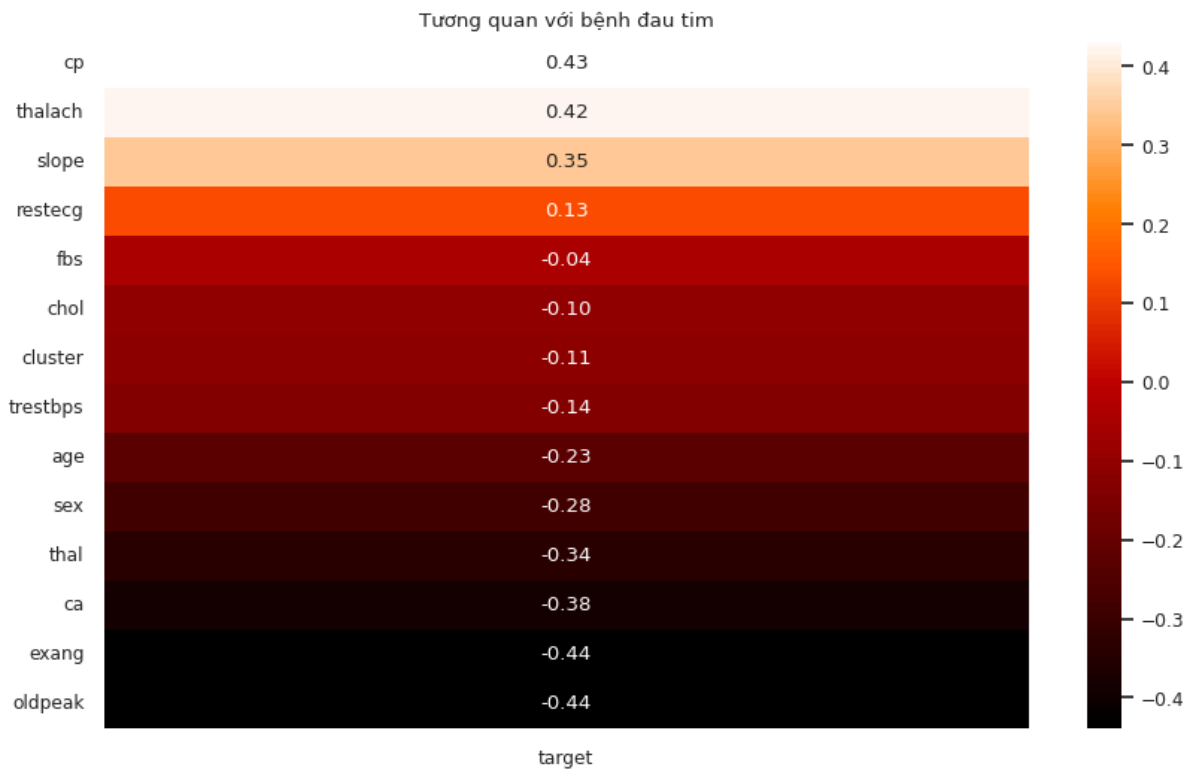
### 2.1 Correlation matrix



*Hình 2.1. Ma trận tương quan giữa các đặc trưng*

Mối tương quan giữa các biến đặc trưng trong tập dữ liệu *thấp*, không có cặp biến nào có mối tương quan cao. Để hiểu rõ hơn về ảnh hưởng của từng đặc trưng đối với mục tiêu ('target').

Ngoài ra, mối tương quan trung bình giữa 'cp' và 'Thalach' chỉ làm rõ thêm ý tưởng thông thường rằng người có bệnh đau về ngực xu hướng nhịp tim tăng để đạt tối đa hơn.



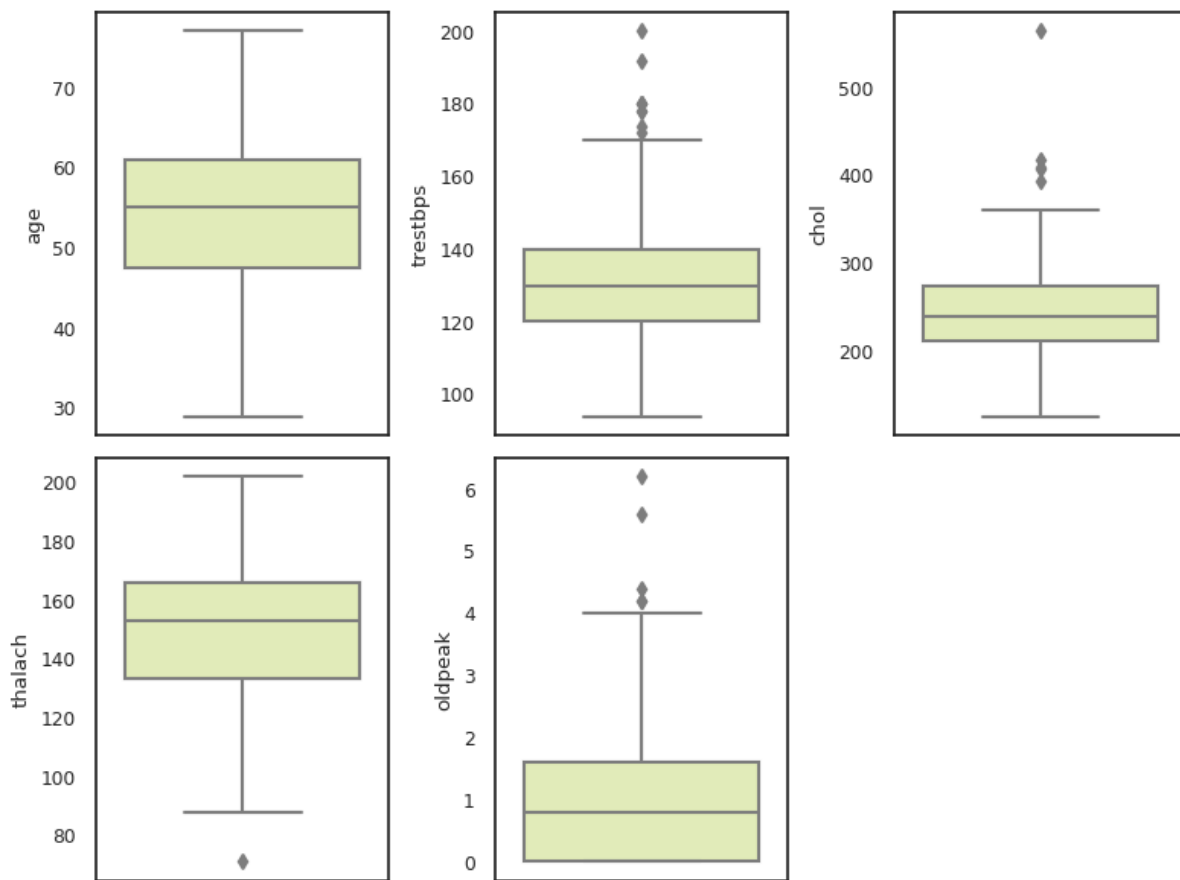
**Hình 2.2. Ma trận tương quan của các đặc trưng với 'Target'**

Nhìn vào kết quả tương quan, chúng ta có thể thấy rằng các Đặc điểm khác nhau có mức độ tương quan khác nhau với kết quả (bệnh tim mạch).

- cp: Với hệ số tương quan là 0,43, đây là đặc điểm có mối tương quan chặt chẽ nhất với kết quả. Điều này cho thấy đau tức ngực nhiều có thể là một dấu hiệu quan trọng của bệnh tim mạch.
- Thalach: Đặc điểm này có mối tương quan 0,42 với kết quả. Mặc dù không mạnh bằng cp nhưng đây vẫn là mối tương quan vừa phải, cho thấy chỉ số nhịp tim tối đa của cơ thể cũng có thể là một yếu tố quan trọng trong bệnh tim mạch.
- Slope: Mạch chính chủ có mối tương quan 0,35 với kết quả. Điều này cho thấy những người có mạch chính chủ cao có nhiều khả năng mắc bệnh tim hơn.
- Restecg: Đặc điểm này có mối tương quan 0,13 với kết quả. Điều này có thể gợi ý rằng chỉ số điện tim đồ có ảnh hưởng đến sự tăng nguy cơ mắc bệnh tim mạch.

- Fbs: Với hệ số tương quan là -0,04, đặc điểm này có mối tương quan thấp với kết quả. Điều này cho thấy lượng đường huyết nhanh hay chậm không có ảnh hưởng đến bệnh tim.
- Chol: Tính năng này có mối tương quan -0,10 với kết quả. Đây là mối tương quan thấp, cho thấy rằng Cholesterol trong máu không phải là yếu tố dự báo mạnh mẽ về bệnh tim mạch.
- Trestbps: Với hệ số tương quan là -0,14, đặc điểm này có mối tương quan rất thấp với kết quả. Điều này cho thấy huyết áp nghỉ không phải là yếu tố quan trọng gây ra bệnh tim mạch.
- Age: Tính năng này có mối tương quan rất thấp là -0,23 với kết quả. Điều này cho thấy tuổi không phải là yếu tố quan trọng gây ra bệnh tim mạch.
- Sex: Tính năng này có mối tương quan rất thấp là -0,28 với kết quả. Điều này cho thấy giới tính không phải là yếu tố quan trọng gây ra bệnh tim mạch.
- Thal: Tính năng này có mối tương quan rất thấp là -0,34 với kết quả. Điều này cho thấy Thalassemia không phải là yếu tố quan trọng gây ra bệnh tim mạch.
- Ca: Tính năng này có mối tương quan rất thấp là -0,38 với kết quả. Điều này cho thấy số mạch chủ chính không phải là yếu tố quan trọng gây ra bệnh tim mạch.
- Exang: Tính năng này có mối tương quan rất thấp là -0,44 với kết quả. Điều này cho thấy tình trạng đau ngực do vận động không phải là yếu tố quan trọng gây ra bệnh tim mạch.
- Oldpeak: Tính năng này có mối tương quan rất thấp là -0,44 với kết quả. Điều này cho thấy chênh xuống do luyện tập không phải là yếu tố quan trọng gây ra bệnh tim mạch.

## 2.2 Box\_Plot



*Hình 2.3. Biểu đồ hộp*

### Phân tích biểu đồ hộp:

Age:

- Có khoảng 60% đối tượng dưới 50 tuổi và 40% trên 60 tuổi.

Trestbps:

- Có khoảng 50% chiếm 140 và 50% chiếm 120

Chol:

- Có khoảng 45% chiếm 290 và 55% chiếm 210

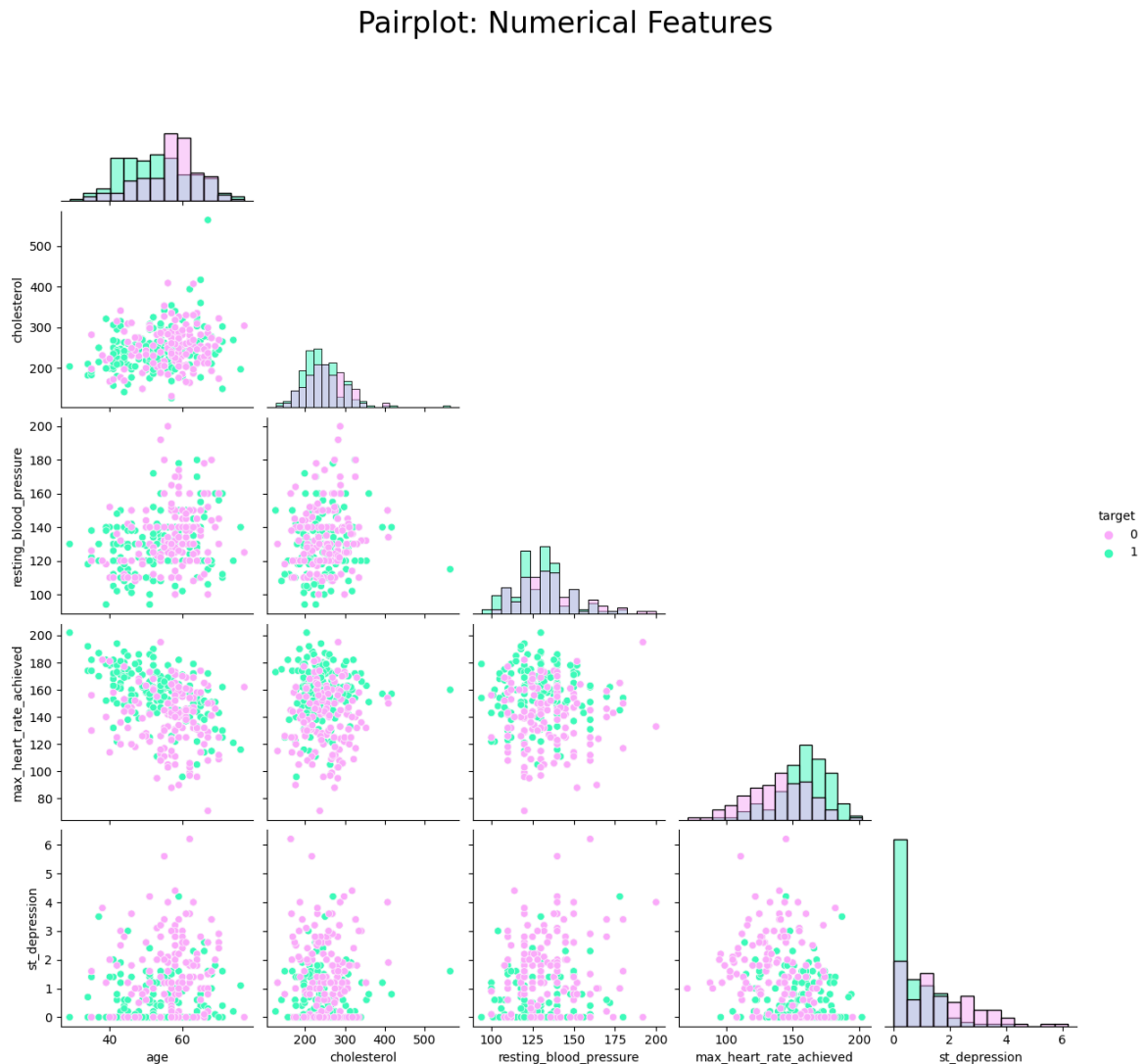
Thalanh:

- Có khoảng 65% chiếm 130 và 35% chiếm 170

Oldpeak:

- Có khoảng 48% chiếm 0,1 và 52% chiếm 1,9

**Bây giờ, hãy thực hiện phân tích hai biến của các đặc trưng số. Cụ thể, chúng ta sẽ xem xét mối tương quan giữa các biến đặc trưng số với nhau.**



*Hình 2.4 Pair-plot*

**Nhận xét:**

- Age cao thì resting\_blood\_pressure cao thì khả năng mắc bệnh tim càng cao.
- Cholesterol cao thì max\_heart\_rate\_achieved cao khả năng mắc bệnh tim càng cao.
- Ngoài ra các tác nhân khác không ảnh hưởng nhiều.



### III. HUẤN LUYỆN DỮ LIỆU

#### 1. Xử lý và điều chỉnh dữ liệu chuẩn bị cho việc huấn luyện

- Tách biệt biến đặc điểm và biến mục tiêu

```
data = pd.read_csv('/content/drive/MyDrive/BTL_KPDL/data/heart.csv')

X = data.drop(['target'], axis = 1)
X.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2

```
y = data['target']
y.head()
```

```
0    1
1    1
2    1
3    1
4    1
Name: target, dtype: int64
```

#### 2. Đào tạo và phân chia dữ liệu

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# In ra hình dạng của các bộ dữ liệu
print("X_train.shape:", X_train.shape)
print("X_test.shape:", X_test.shape)
print("y_train.shape:", y_train.shape)
print("y_test.shape:", y_test.shape)
```

```
X_train.shape: (242, 13)
X_test.shape: (61, 13)
y_train.shape: (242,)
y_test.shape: (61,)
```

### 3. Huấn luyện dữ liệu với các mô hình cơ sở

```
models = {
    "Logistic Regression" : LogisticRegression() ,
    "Naive Bayes" : GaussianNB(),
    "Random Forest Classifier" : RandomForestClassifier(),
    "Extreme Gradient Boost" : XGBClassifier(),
    "KNN" : KNeighborsClassifier() ,
    "DecisionTreeClassifier" : DecisionTreeClassifier(),
    "Random Forest" : RandomForestClassifier(),
    "Support Vector Classifier" : SVC(kernel='rbf', C=2)
}

# Tạo một danh sách để lưu trữ điểm chuẩn xác của các mô hình
model_scores = []

# Lặp qua danh sách các mô hình
for name, model in models.items():
    # Huấn luyện mô hình
    model.fit(X_train, y_train)

    # Đánh giá mô hình
    score = model.score(X_test, y_test)

    # Lưu trữ điểm chuẩn xác của mô hình
    model_scores.append((name, score))

# Sắp xếp các mô hình theo điểm chuẩn xác giảm dần
model_scores.sort(key=lambda x: x[1], reverse=True)

# In ra tên và điểm chuẩn xác của các mô hình
print("Điểm chuẩn xác của các mô hình:")
for name, score in model_scores:
    print(f"{name}: {score}")
```

```
Điểm chuẩn xác của các mô hình:
Logistic Regression: 0.8852459016393442
Random Forest Classifier: 0.8852459016393442
Naive Bayes: 0.8688524590163934
Random Forest: 0.8524590163934426
DecisionTreeClassifier: 0.8360655737704918
Extreme Gradient Boost: 0.819672131147541
Support Vector Classifier: 0.7049180327868853
KNN: 0.6885245901639344
```

## 4. Điều chỉnh siêu tham số và báo cáo phân loại

```
[18] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

rf = RandomForestClassifier(random_state=42)

param_grid = {
    'n_estimators': [50, 100, 150],
    'max_depth': [None, 10, 20],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4]
}

grid_search = GridSearchCV(estimator=rf, param_grid=param_grid, cv=3, scoring='accuracy')
grid_search.fit(X_train, y_train)

rf_best = grid_search.best_estimator_
y_pred = rf_best.predict(X_test)

print("Classification Report:")
print(classification_report(y_test, y_pred))
```

➡ Classification Report:

	precision	recall	f1-score	support
0	0.86	0.83	0.84	29
1	0.85	0.88	0.86	32
accuracy			0.85	61
macro avg	0.85	0.85	0.85	61
weighted avg	0.85	0.85	0.85	61

## IV. PHÂN CỤM BỆNH NHÂN TIM MẠCH BẰNG PHÂN CỤM K MEANS

### 1. Phân tích phân cụm

#### Chuẩn hóa dữ liệu

Tiền xử lý là bước quan trọng trước khi huấn luyện mô hình. Ở đây, đặc điểm số được chuẩn hóa và đặc điểm phân loại được mã hóa. Chuẩn hóa không là bắt buộc, nhưng thường là thực hành tốt. StandardScaler trong sklearn giúp biến đổi dữ liệu sao cho giá trị trung bình là 0 và độ lệch chuẩn là 1.

```
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

# Chuẩn hóa dữ liệu: Normalize the data
scaler = StandardScaler() # Tạo đối tượng StandardScaler để chuẩn hóa dữ liệu
data_scaled = scaler.fit_transform(data.drop('target', axis=1)) # Fit và transform dữ liệu, loại bỏ cột 'target'

# Xác định số cụm tối ưu bằng phương pháp Elbow: Determine the optimal number of clusters using the Elbow method
wcss = [] # Khởi tạo một danh sách rỗng để lưu trữ WCSS (within-cluster sum of squares)
for i in range(1, 11): # Lặp qua các số từ 1 đến 10 đại diện cho số lượng cụm tiềm năng
    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=0)
    # Tạo đối tượng KMeans với số lượng cụm khác nhau và các tham số khác được chỉ định

    kmeans.fit(data_scaled) # Fit mô hình KMeans vào dữ liệu đã chuẩn hóa

    wcss.append(kmeans.inertia_)
    # Thêm giá trị WCSS (inertia) cho mỗi số lượng cụm vào danh sách wcss

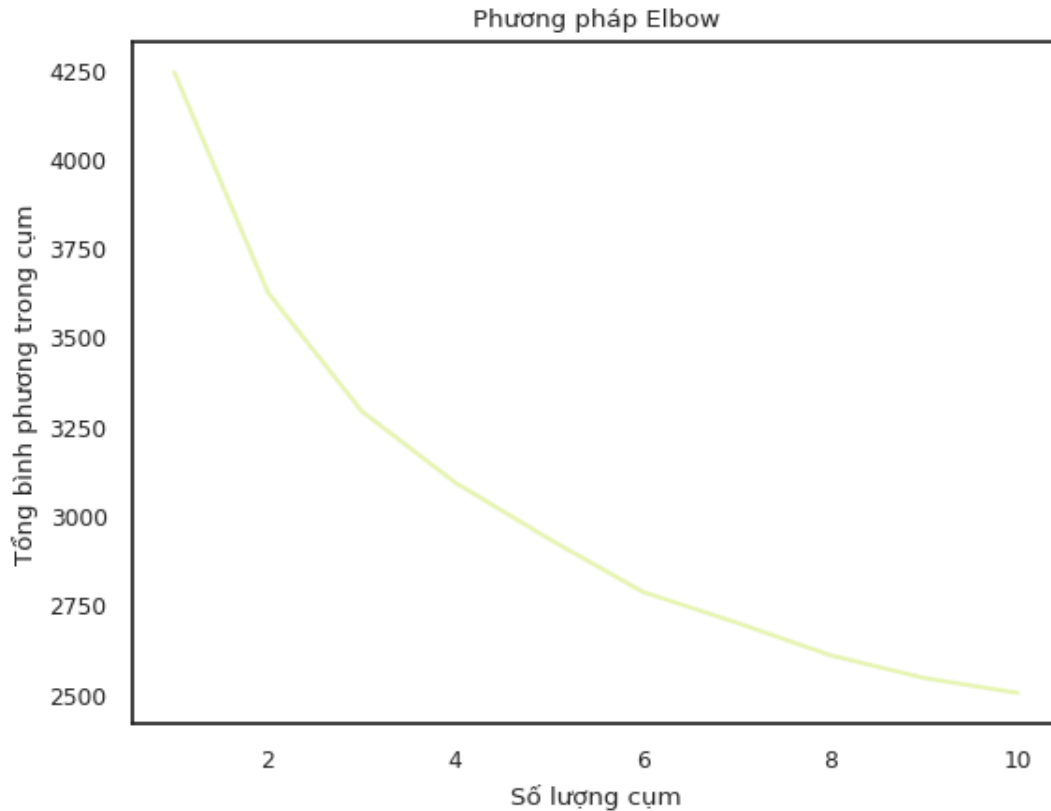
plt.plot(range(1,11), wcss)
# Vẽ đồ thị với trục x là số lượng cụm và trục y là giá trị WCSS tương ứng

plt.title('Phương pháp Elbow')
# Đặt tiêu đề cho đồ thị là 'Phương pháp Elbow'

plt.xlabel('Số lượng cụm')
# Đặt nhãn trục x là 'Số lượng cụm'

plt.ylabel('Tổng bình phương trong cụm')
# Đặt nhãn trục y là 'Tổng bình phương trong cụm'

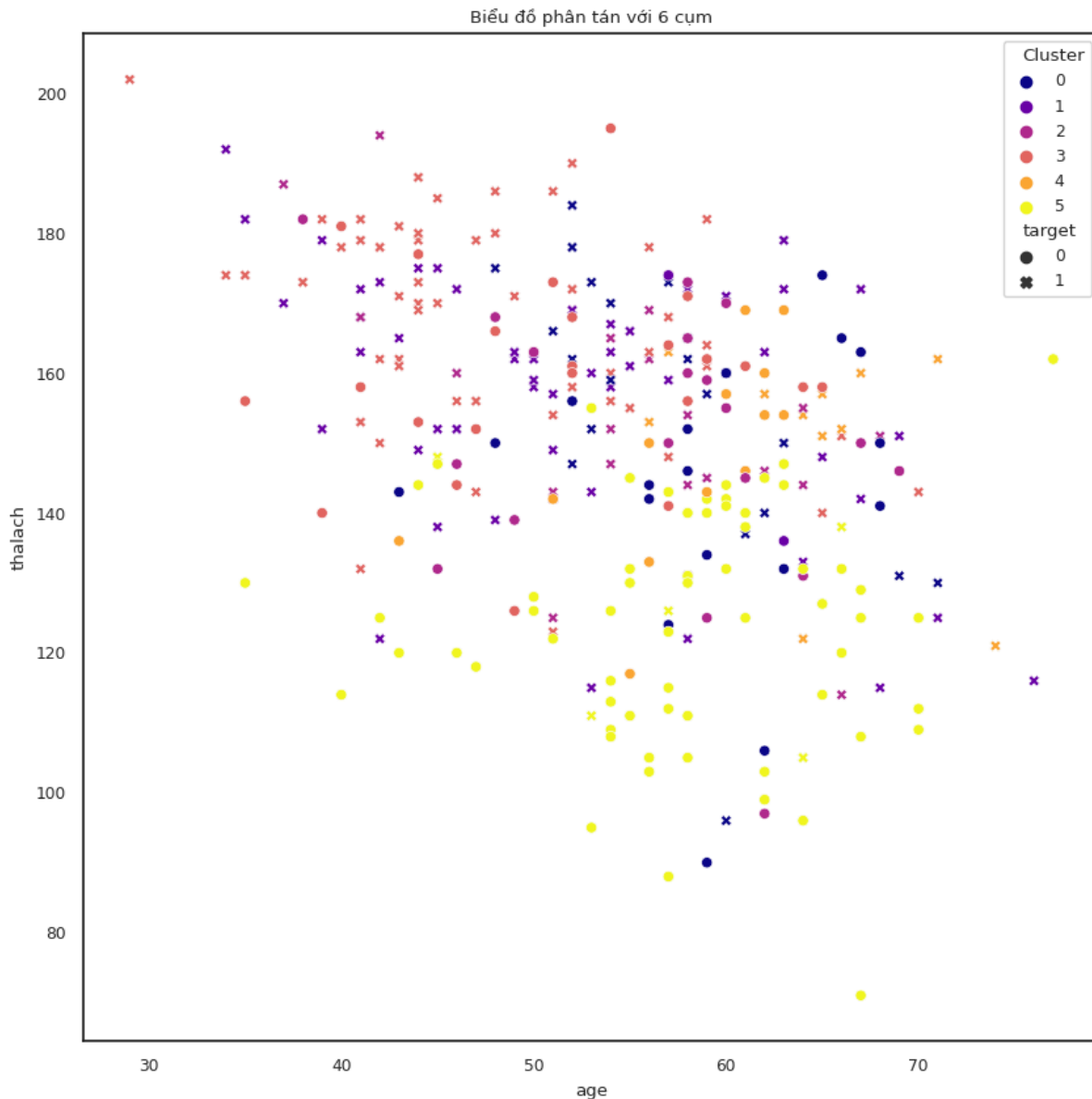
plt.show()
# Hiển thị đồ thị biểu diễn phương pháp Elbow để xác định số lượng cụm tối ưu
```



***Hình 4.1. Biểu đồ Elbow***

Biểu đồ Elbow là công cụ quan trọng để xác định số lượng cụm tối ưu trong phân cụm K-means. Trên biểu đồ, trục x thể hiện số lượng cụm, trong khi trục y biểu diễn Tổng bình phương trong cụm (WCSS) - một đánh giá về độ nhóm của dữ liệu.

'Elbow' trên biểu đồ là điểm mà thêm cụm không cải thiện đáng kể WCSS. Nhìn chung, số cụm tối ưu cho dữ liệu là 6, được xác định dựa trên phương pháp Elbow.



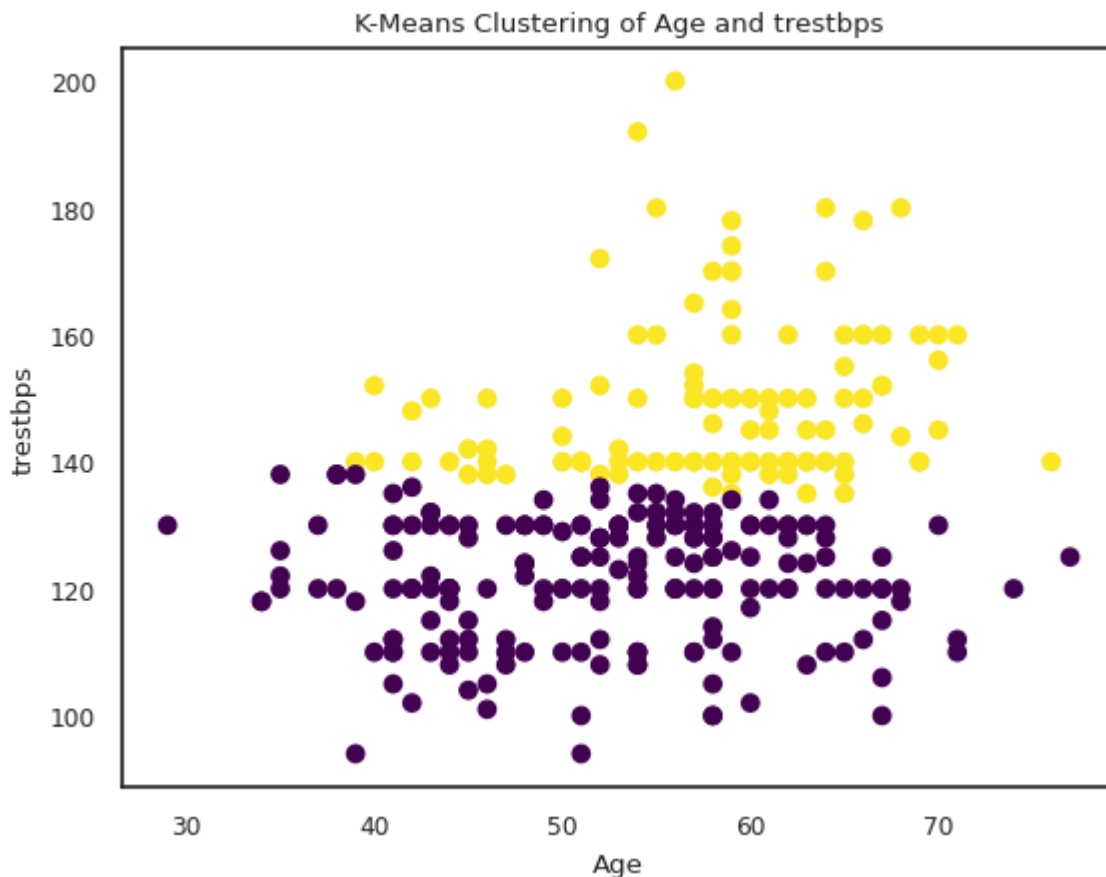
*Hình 4.2. Biểu đồ phân tán trực quan hóa mối quan hệ giữa Age và Thalach*

Biểu đồ phân tán trực quan hóa mối quan hệ giữa Age và Thalach, hai tính năng quan trọng trong tập dữ liệu. Mỗi điểm trên biểu đồ đại diện cho một cá nhân trong tập dữ liệu và màu của điểm biểu thị cụm mà cá nhân đó thuộc về dựa trên phân tích phân cụm K-means:

- Age: Tính năng này được thể hiện trên trục x. Các cá nhân có mức độ Thalach khác nhau.
- Thalach: Tính năng này được thể hiện trên trục y. Các cá nhân có nhiều giá trị Age khác nhau.

Clusters: Các màu khác nhau trên biểu đồ đại diện cho các cụm khác nhau. Dùng thuật toán phân cụm K-mean đã nhóm các cá nhân thành các cụm riêng biệt dựa trên Age và Trestbps.

## 2. Gom cụm



*Hình 4.3. Biểu đồ gom cụm giữa 'Age' và 'Trestbps'*

- Nhìn vào biểu đồ phân tán giữa trestbps và age, có thể thấy rằng các cụm thường phân bố theo đường chéo từ góc dưới bên trái lên góc trên bên phải. Điều này cho thấy rằng mối quan hệ giữa trestbps và age là tuyến tính, tức là khi tuổi tác tăng lên, huyết áp lúc nghỉ cũng có xu hướng tăng lên.
- Tuy nhiên, cũng có một số cụm nằm ngoài đường chéo này. Điều này cho thấy rằng có một số người có huyết áp lúc nghỉ cao hơn hoặc thấp hơn so với mức trung bình của nhóm tuổi của họ
- Nhìn chung, biểu đồ phân tán cho thấy rằng có mối tương quan dương giữa trestbps và age. Điều này có nghĩa là khi tuổi tác tăng lên, huyết

áp lúc nghỉ cũng có xu hướng tăng lên. Ngoài ra, biểu đồ phân tán cũng cho thấy rằng có một số người có huyết áp lúc nghỉ cao hơn hoặc thấp hơn so với mức trung bình của nhóm tuổi của họ.

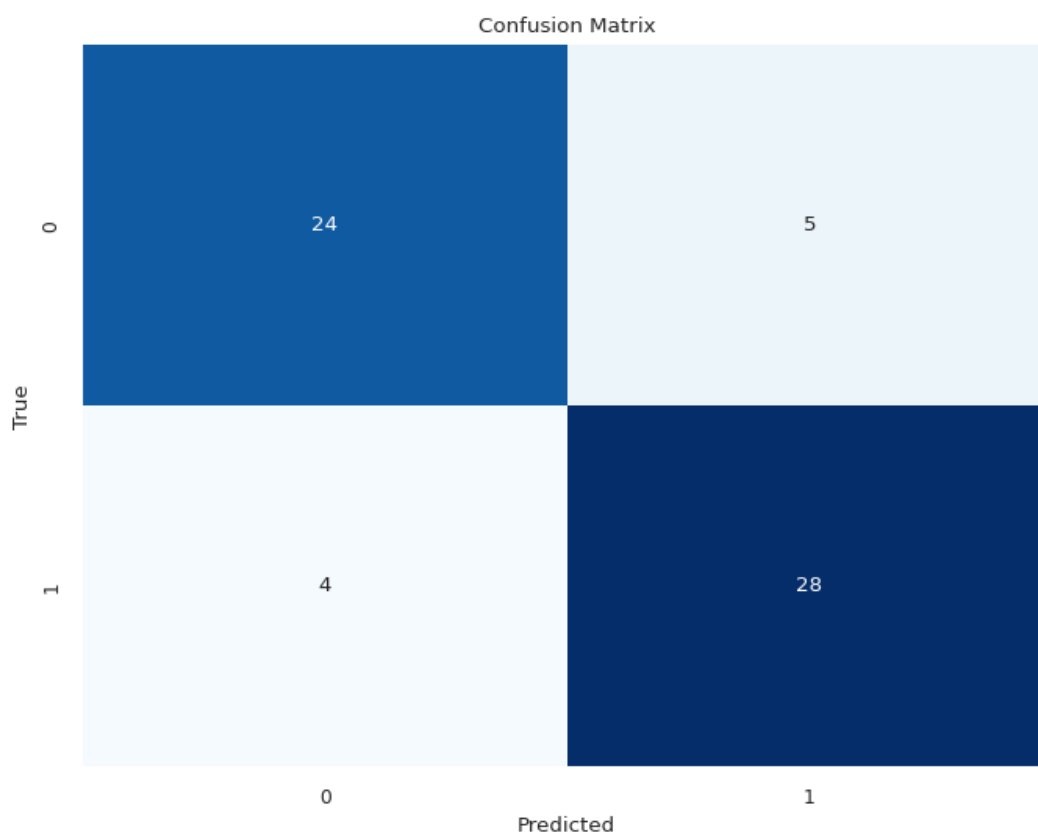
## V. THUẬT TOÁN PHÂN LOẠI

### ● Confusion Matrix

```
y_pred = rf_best.predict(X_test)

cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', cbar=False)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()
```



*Hình 5.1. Biểu đồ Ma trận hỗn loạn*



Biểu đồ trên là một Confusion Matrix, được sử dụng để đánh giá hiệu suất của mô hình phân loại trong việc dự đoán nguy cơ nhồi máu cơ tim. Confusion Matrix này bao gồm 4 ô, mỗi ô đại diện cho một kết quả dự đoán. Số 0 trên cả hai trục đại diện cho kết quả dự đoán không có nguy cơ, trong khi số 1 trên cả hai trục đại diện cho kết quả dự đoán có nguy cơ.

Cụ thể, Confusion Matrix này có các thông số sau:

- **True Negative (TN):** Có 24 trường hợp được dự đoán chính xác là không có nguy cơ.
- **False Positive (FP):** Có 5 trường hợp thực tế không có nguy cơ nhưng được dự đoán là có nguy cơ.
- **False Negative (FN):** Có 4 trường hợp thực tế có nguy cơ nhưng được dự đoán là không có nguy cơ.
- **True Positive (TP):** Có 28 trường hợp được dự đoán chính xác là có nguy cơ.

## • Features Importance

```
# Huấn luyện mô hình RandomForestRegressor đã được tối ưu hóa (rf_best) trên dữ liệu đào tạo (X) và nhãn (y).
rf_best.fit(X, y)

# Lấy độ quan trọng của từng đặc trưng từ mô hình đã được huấn luyện.
feature_importances_ = rf_best.feature_importances_

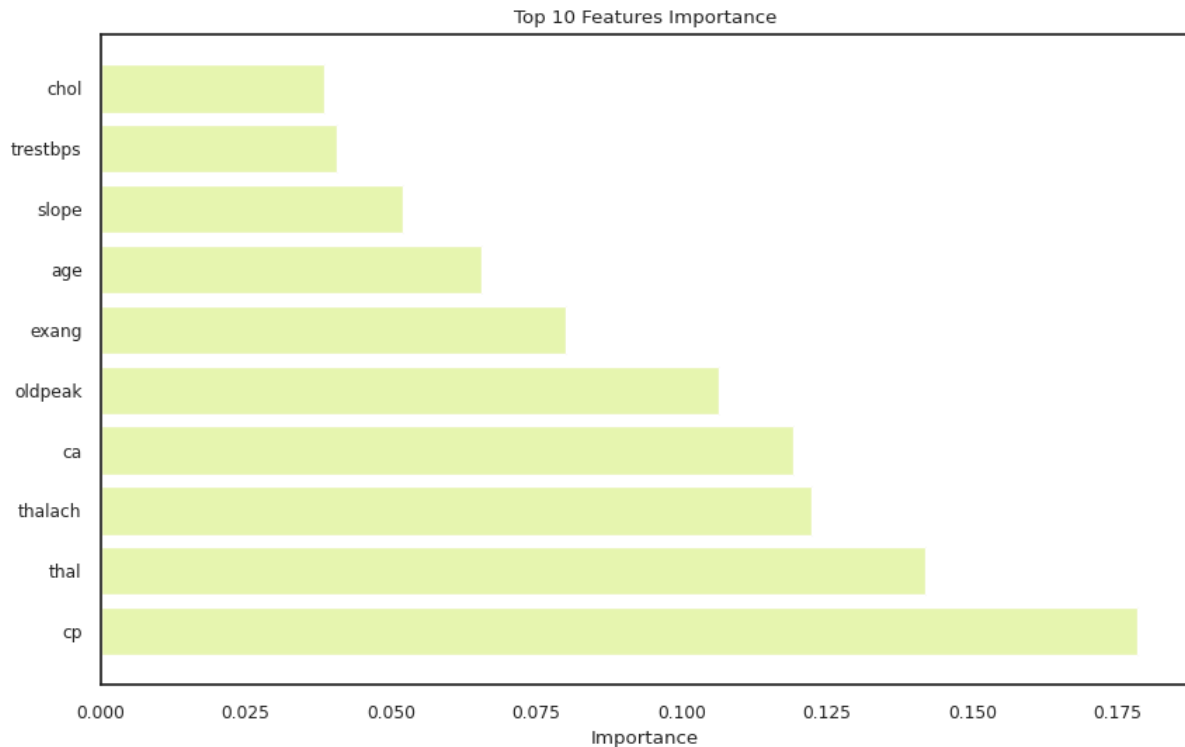
# Tạo DataFrame chứa thông tin về độ quan trọng của từng đặc trưng.
feature_importance_df = pd.DataFrame({
    'Feature': X.columns,
    'Importance': feature_importances_
})

# Sắp xếp DataFrame theo độ quan trọng giảm dần.
feature_importance_df = feature_importance_df.sort_values(by='Importance', ascending=False)

# In ra màn hình
print("Feature Importance:")
print(feature_importance_df.head())

# Vẽ biểu đồ cột ngang hiển thị độ quan trọng của 10 đặc trưng hàng đầu.
plt.figure(figsize=(10, 6))
plt.barh(feature_importance_df['Feature'][:10], feature_importance_df['Importance'][:10])
plt.xlabel('Importance')
plt.title('Top 10 Features Importance')
plt.show()
```

```
Feature Importance:
   Feature  Importance
2        cp    0.178424
12       thal    0.141988
7    thalach    0.122274
11        ca    0.119214
9    oldpeak    0.106469
```



*Hình 5.2. Biểu đồ Top 10 thuộc tính quan trọng*

## VI. CÂY QUYẾT ĐỊNH

### 1. Mô tả lại thuộc tính

Nhìn vào thông tin về các yếu tố nguy cơ mắc bệnh tim, nhóm em nhận ra một số điều sau: **cholesterol cao, huyết áp cao, tiểu đường, cân nặng, tiền sử gia đình và hút thuốc**. Theo một nguồn tin khác, những yếu tố chính không thể thay đổi là: **tuổi tác ngày càng tăng, giới tính nam và yếu tố di truyền**. Lưu ý rằng **bệnh thalassemia**, một trong những biến số trong bộ dữ liệu này, là tính di truyền. Các yếu tố chính có thể được sửa đổi là: **Hút thuốc, cholesterol cao, huyết áp cao, ít hoạt động thể chất, thừa cân và mắc bệnh tiểu đường**. Các yếu tố khác bao gồm căng thẳng, rượu và chế độ ăn/dinh dưỡng kém.

Với những điều trên, nhóm em đưa ra giả thuyết rằng, nếu mô hình có khả năng dự đoán nào đó, chúng ta sẽ thấy những yếu tố này nổi bật là quan trọng nhất.

Nhóm em đã thay đổi tên cột cho rõ ràng hơn:

```
[15] data.columns = ['age', 'sex', 'chest_pain_type', 'resting_blood_pressure', 'cholesterol', 'fasting_blood_sugar', 'rest_ecg', 'max_heart_rate_achieved', 'exercise_induced_angina', 'st_depression', 'st_slope', 'num_major_vessels', 'thalassemia', 'target']
```

Đối với các biến phân loại, chúng ta cần tạo các biến giả.

Và thay đổi giá trị các biến phân loại để cải thiện việc diễn giải sau này:

```
data['sex'][data['sex'] == 0] = 'female'
data['sex'][data['sex'] == 1] = 'male'

data['chest_pain_type'][data['chest_pain_type'] == 1] = 'typical angina'
data['chest_pain_type'][data['chest_pain_type'] == 2] = 'atypical angina'
data['chest_pain_type'][data['chest_pain_type'] == 3] = 'non-anginal pain'
data['chest_pain_type'][data['chest_pain_type'] == 4] = 'asymptomatic'

data['fasting_blood_sugar'][data['fasting_blood_sugar'] == 0] = 'lower than 120mg/ml'
data['fasting_blood_sugar'][data['fasting_blood_sugar'] == 1] = 'greater than 120mg/ml'

data['rest_ecg'][data['rest_ecg'] == 0] = 'normal'
data['rest_ecg'][data['rest_ecg'] == 1] = 'ST-T wave abnormality'
data['rest_ecg'][data['rest_ecg'] == 2] = 'left ventricular hypertrophy'

data['exercise_induced_angina'][data['exercise_induced_angina'] == 0] = 'no'
data['exercise_induced_angina'][data['exercise_induced_angina'] == 1] = 'yes'

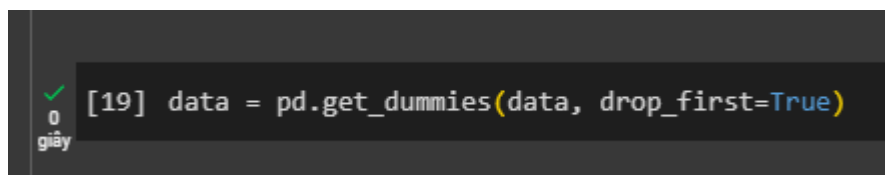
data['st_slope'][data['st_slope'] == 1] = 'upsloping'
data['st_slope'][data['st_slope'] == 2] = 'flat'
data['st_slope'][data['st_slope'] == 3] = 'downsloping'

data['thalassemia'][data['thalassemia'] == 1] = 'normal'
data['thalassemia'][data['thalassemia'] == 2] = 'fixed defect'
data['thalassemia'][data['thalassemia'] == 3] = 'reversible defect'
```

- sex: sex
  - + Female (0): nữ
  - + Male (1): nam
- chest pain type: cp
  - + typical angina (1): đau thắt ngực điển hình
  - + atypical angina (2): đau thắt ngực không điển hình
  - + non-anginal pain (3): đau không đau thắt ngực
  - + asymptomatic (4): không có triệu chứng
- fasting blood sugar: fbs
  - + lower than 120mg/ml (0): thấp hơn 120 mg/ml
  - + greater than 120mg/ml (1): lớn hơn 120mg/ml

- rest ecg: restecg
  - + normal (0): bình thường
  - + ST-T wave abnormality (1): sóng ST-T bất thường
  - + left ventricular hypertrophy (2): phì đại thất trái
- exercise induced angina: exang
  - + No (0)
  - + Yes (1)
- st slope: slope
  - + upsloping (1): dốc lên
  - + flat (2): phẳng
  - + downsloping (3): dốc xuống
- thalassemia: thal
  - + normal (1): bình thường
  - + fixed defect (2): khiếm khuyết cố định
  - + eversible defect (3): khiếm khuyết vĩnh viễn

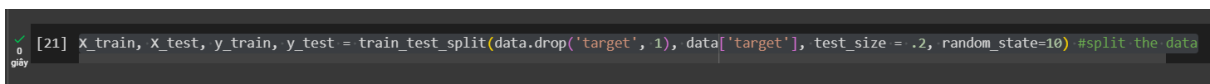
## 2. Tạo biến ảo



```
[19] data = pd.get_dummies(data, drop_first=True)
```

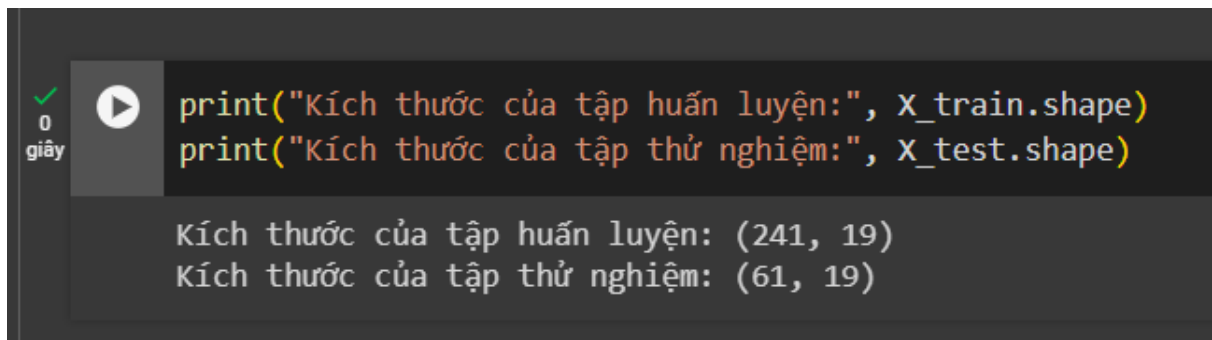
Tạo các biến ảo từ các biến phân loại giúp chuẩn bị dữ liệu cho mô hình hóa, loại bỏ cạm bẫy biến giả, mô hình hóa biến phân loại và phân tích khả năng ảnh hưởng của từng nhóm giá trị. Việc này mở ra nhiều cơ hội để nắm bắt thông tin và tương quan giữa các nhóm giá trị, tăng khả năng phân tích và hiểu sâu hơn về dữ liệu.

## 3. Tách dữ liệu thành tập huấn luyện và tập thử nghiệm



```
[21] X_train, X_test, y_train, y_test = train_test_split(data.drop('target', 1), data['target'], test_size = .2, random_state=10) #split the data
```

In kích thước của các tập dữ liệu

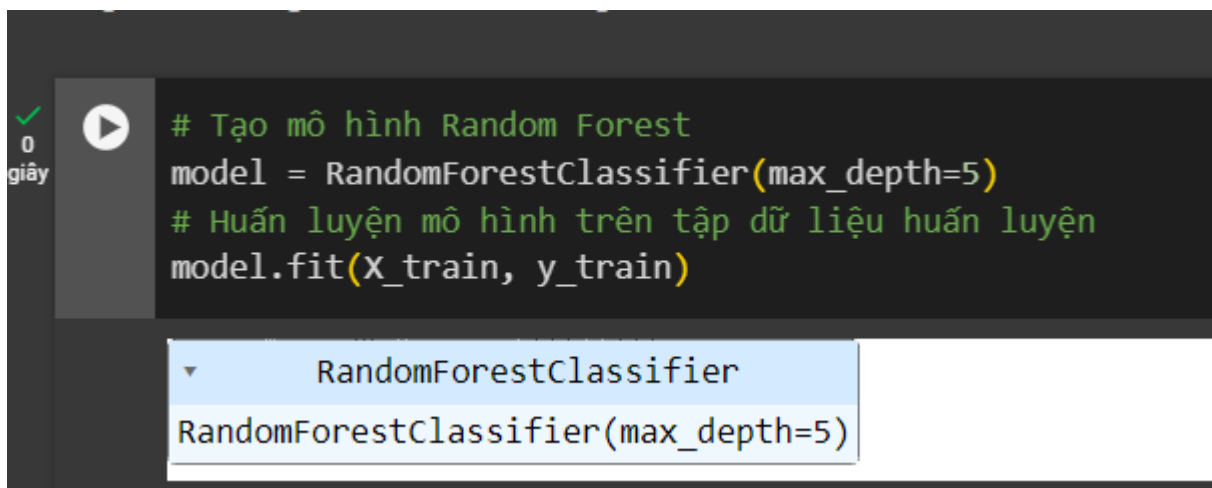


```
print("Kích thước của tập huấn luyện:", X_train.shape)
print("Kích thước của tập thử nghiệm:", X_test.shape)
```

Kích thước của tập huấn luyện: (241, 19)  
Kích thước của tập thử nghiệm: (61, 19)

**Nhận xét:** Kích thước của tập huấn luyện là 241 mẫu và kích thước của tập thử nghiệm là 61 mẫu, điều này cho thấy dữ liệu đã được chia thành tập huấn luyện và tập thử nghiệm với tỷ lệ khoảng 80% - 20%. Đây là một tỷ lệ chia dữ liệu phổ biến trong học máy. Tỷ lệ này đảm bảo rằng mô hình được huấn luyện trên một lượng dữ liệu đủ lớn để học được các đặc điểm của dữ liệu và tránh tình trạng quá khớp (overfitting). Đồng thời, tập thử nghiệm cũng đủ lớn để đánh giá hiệu suất của mô hình một cách đáng tin cậy.

#### 4. Mô hình cuối cùng



```
# Tạo mô hình Random Forest
model = RandomForestClassifier(max_depth=5)
# Huấn luyện mô hình trên tập dữ liệu huấn luyện
model.fit(X_train, y_train)
```

▼ RandomForestClassifier  
RandomForestClassifier(max\_depth=5)

## 5. Dự đoán mẫu dữ liệu mới

```
✓ 0 giây [30] # Dự đoán nhãn cho nhiều mẫu dữ liệu mới
new_samples = [[10, 150, 120, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0],
               [20, 200, 140, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]]
predictions = model.predict(new_samples)

# In kết quả dự đoán
print("Nhãn dự đoán:", predictions)

Nhãn dự đoán: [0 1]
```

### Nhận xét:

- Nhãn 0 và 1 tương ứng với các lớp "không bị bệnh tim" và "bị bệnh tim". Do đó, kết quả dự đoán có nghĩa là:
- Mẫu dữ liệu đầu tiên được dự đoán là không bị bệnh tim. Mẫu dữ liệu thứ hai được dự đoán là bị bệnh tim.

## 6. Trực quan hóa cây quyết định

```
[23] # Lấy bộ ước tính thứ hai trong rừng
estimator = model.estimators_[1]
# Lấy tên các đặc trưng
feature_names = [i for i in X_train.columns]

# Chuyển nhãn huấn luyện sang kiểu chuỗi
y_train_str = y_train.astype('str')

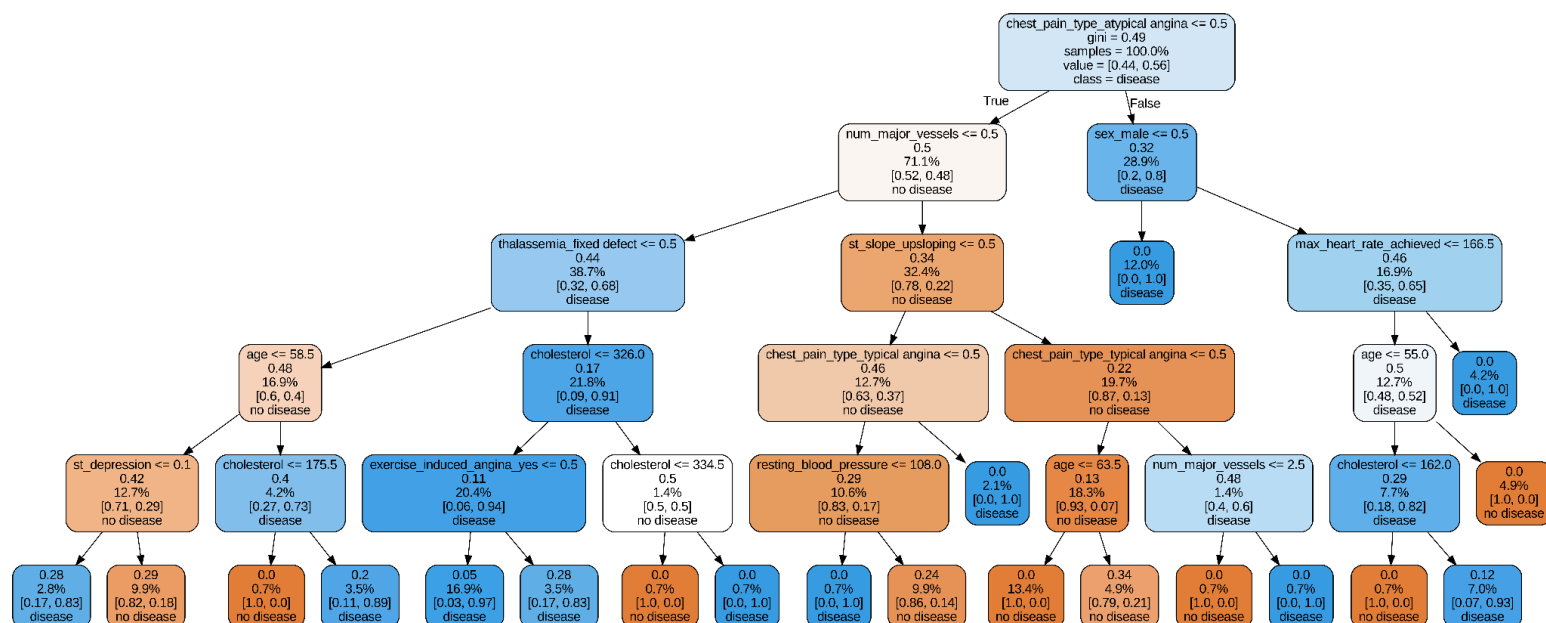
# Thay thế '0' bằng 'no disease' và '1' bằng 'disease'
y_train_str[y_train_str == '0'] = 'no disease'
y_train_str[y_train_str == '1'] = 'disease'

# Chuyển nhãn huấn luyện sang kiểu mảng
y_train_str = y_train_str.values
```

```
✓ 11 giây export_graphviz(estimator, out_file='tree.dot',
                        feature_names = feature_names,
                        class_names = y_train_str,
                        rounded = True, proportion = True,
                        label='root',
                        precision = 2, filled = True)

from subprocess import call
call(['dot', '-Tpng', 'tree.dot', '-o', 'tree.png', '-Gdpi=600'])

from IPython.display import Image
Image(filename = 'tree.png')
```



Hình 6.1. Biểu đồ Cây quyết định

### Nhận xét:

Dựa vào cây này, chúng ta có thể khẳng định chest\_pain\_type\_atypical angina là nhân tố quyết định đánh giá chất lượng của việc chẩn đoán bệnh tim.

Các tính năng được sử dụng để dự đoán là: giới tính, đau ngực, lượng đường trong máu, điện tâm đồ lúc nghỉ, đau thắt ngực do gắng sức, độ dốc ST, thiếu máu và tình trạng mạch máu.

Cây quyết định hoạt động bằng cách đặt một loạt các câu hỏi về các tính năng này và sau đó đưa ra dự đoán có bệnh hay không (disease/no disease) dựa trên câu trả lời.

Ví dụ:

- Chuẩn đoán dựa vào chest\_pain\_type\_atypical:
  - + Nếu chest pain type  $\leq 0,5$  nghĩa là typical angina, cây quyết định sẽ tiếp tục kiểm tra đặc trưng khác.
  - + Nếu num\_majob\_vessels  $\leq 0,5$  nghĩa là số mạch chính chủ = 0 sẽ lại tiếp tục đưa ra những đặc trưng khác tiếp tục...

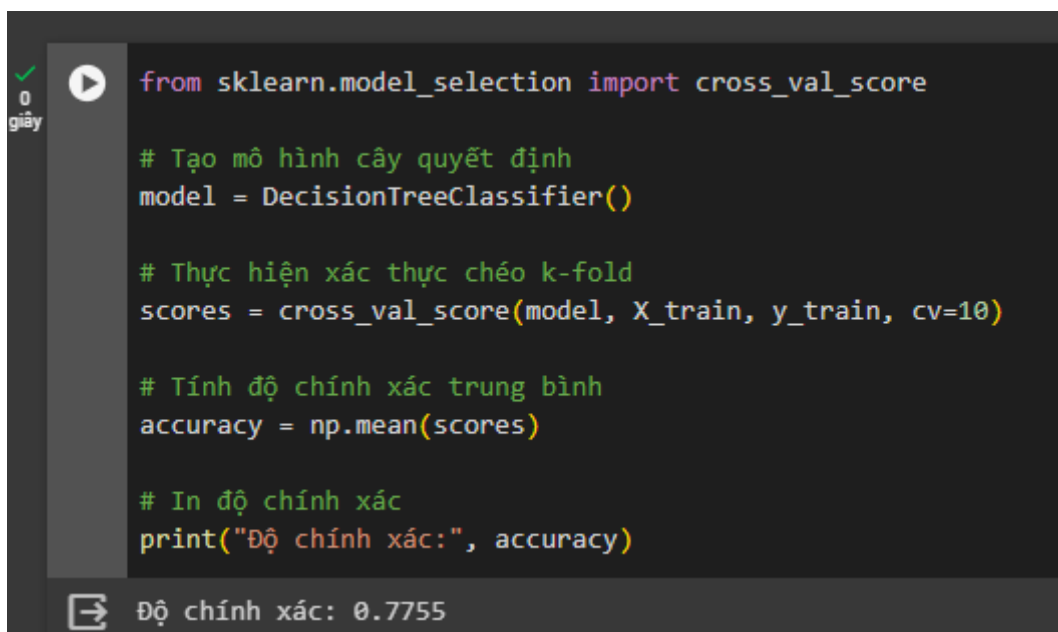


- + Nếu `num_majob_vessels > 0,5` nghĩa là số mạch chính chủ = 1, 2, 3 sẽ lại tiếp tục đưa ra những đặc trưng khác tiếp tục ...
- + Tương tự nếu `chest pain type > 0,5` cây quyết định sẽ tiếp tục kiểm tra đặc trưng khác....
- Cây quyết định sẽ tiếp tục đặt các câu hỏi cho đến khi đưa ra dự đoán. Dự đoán có thể là bệnh nhân có bị bệnh tim hay không. (disease/ no disease)

## 7. Độ chính xác của cây quyết định

Để biết độ chính xác của cây quyết định, ta có thể sử dụng phương pháp xác thực chéo (cross-validation). Xác thực chéo là một kỹ thuật đánh giá hiệu suất của mô hình học máy trên dữ liệu mới.

Có nhiều phương pháp xác thực chéo khác nhau, nhưng một phương pháp phổ biến là xác thực chéo k-fold. Trong xác thực chéo k-fold, dữ liệu được chia thành k phần bằng nhau. Sau đó, mô hình được huấn luyện trên k-1 phần và thử nghiệm trên phần còn lại. Quá trình này được lặp lại k lần, mỗi lần sử dụng một phần khác nhau làm tập thử nghiệm.



```

from sklearn.model_selection import cross_val_score

# Tạo mô hình cây quyết định
model = DecisionTreeClassifier()

# Thực hiện xác thực chéo k-fold
scores = cross_val_score(model, X_train, y_train, cv=10)

# Tính độ chính xác trung bình
accuracy = np.mean(scores)

# In độ chính xác
print("Độ chính xác:", accuracy)

```

Độ chính xác: 0.7755

### **Nhận xét:**

Độ chính xác của mô hình được tính bằng cách tính trung bình độ chính xác của mô hình trên  $k$  lần lặp lại. Trong trường hợp này, độ chính xác của cây quyết định là 78%, điều này có nghĩa là cây quyết định có thể dự đoán đúng bệnh tim ở 78% bệnh nhân.

## **VII. TỔNG KẾT**

### **1. Kết quả thu được**

- Thu thập và làm sạch dữ liệu để đảm bảo tính chính xác và tin cậy của dữ liệu trong quá trình phân tích.
- Phân tích dữ liệu khám phá để hiểu rõ hơn về đặc điểm và phân phối các biến quan trọng trong tập dữ liệu.
- Xác định được các yếu tố ảnh hưởng và mối tương quan giữa các biến.
- Phân cụm bệnh nhân tim mạch dựa trên các đặc trưng liên quan đến sức khỏe, qua đó nhóm bệnh nhân có thể được phân loại và nhận được thông tin về tình trạng sức khỏe của từng nhóm.
- Sử dụng thuật toán cây quyết định để mang lại khả năng dự đoán và phân loại cho các bệnh nhân có nguy cơ mắc bệnh đau tim.

### **2. Hạn chế**

- Tập dữ liệu còn hạn chế nên ảnh hưởng tới độ tin cậy một ít.
- Có thể tồn tại các phương pháp khác hoặc mở rộng để khám phá sâu hơn và tìm hiểu rõ hơn.

### **3. Ứng dụng**

- Xây dựng được mô hình khả năng dự đoán và phân loại cho các bệnh nhân có nguy cơ mắc bệnh đau tim. Điều này có thể hỗ trợ cho các chuyên gia y tế và nhà quản lý sức khỏe trong việc đánh giá và xác định nguy cơ mắc bệnh đau tim cho từng cá nhân dựa vào các yếu tố sức khỏe. Ngoài ra còn cung cấp thông tin quan

trọng để phát triển các chương trình phòng ngừa và quản lý bệnh tim mạch. Cuối cùng, có thể làm căn cứ để nghiên cứu và tiếp tục phát triển trong lĩnh vực này, nhằm nâng cao hiểu biết và chăm sóc sức khỏe của những người có nguy cơ mắc bệnh đau tim.