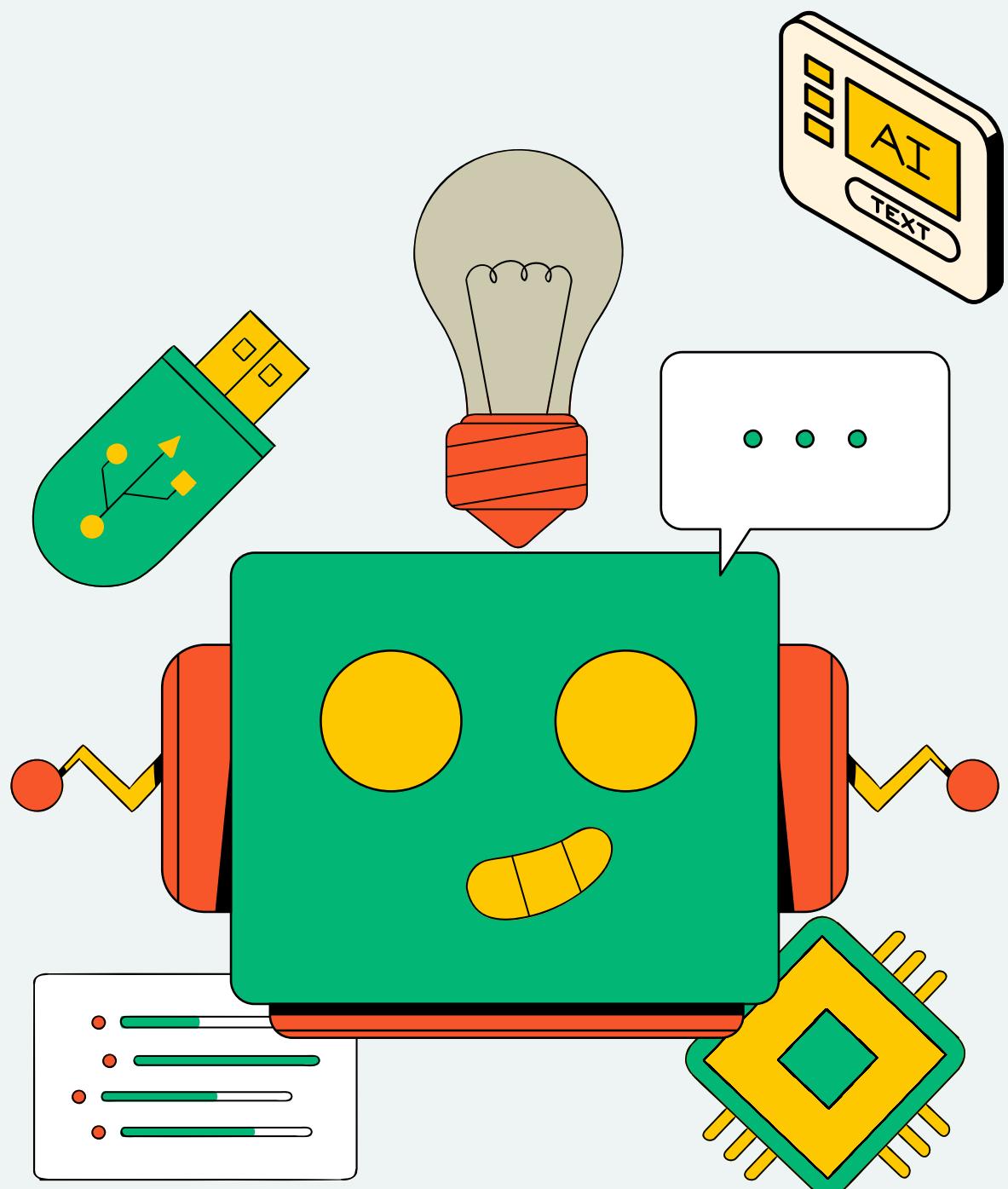
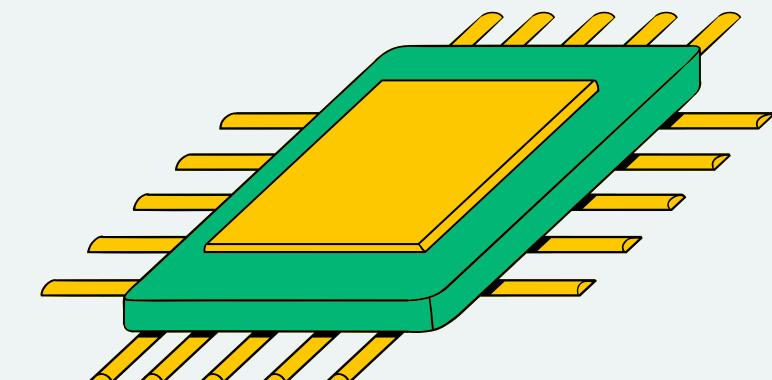


THYNK UNLIMITED
WE LEARN FOR THE FUTURE



DỰ ĐOÁN BỆNH TIM BẰNG CÁC MÔ HÌNH HỌC MÁY

NHÓM 9



THÀNH VIÊN

Nguyễn Thành Công

Vũ Minh Đức

Nguyễn Ngọc Hải

Nguyễn Hoàng Nguyên

Nguyễn Trọng Vỹ



PRESENTATION OUTLINE

- Giới thiệu bài toán
- Khám phá dữ liệu
- Logistic Regression
- Extra Tree Classifier
- Random Forest Classifier
- Decision Tree Classifier
- Neural Network
- Kết luận và hướng phát triển



GIỚI THIỆU BÀI TOÁN

Mục tiêu chính của bộ dữ liệu là xác định sự hiện diện của bệnh tim ở bệnh nhân. Thuộc tính mục tiêu này là một giá trị số nguyên, trong đó:

- 0: không có bệnh tim.
- 1: có bệnh tim.



Nguồn dữ liệu: <https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset>

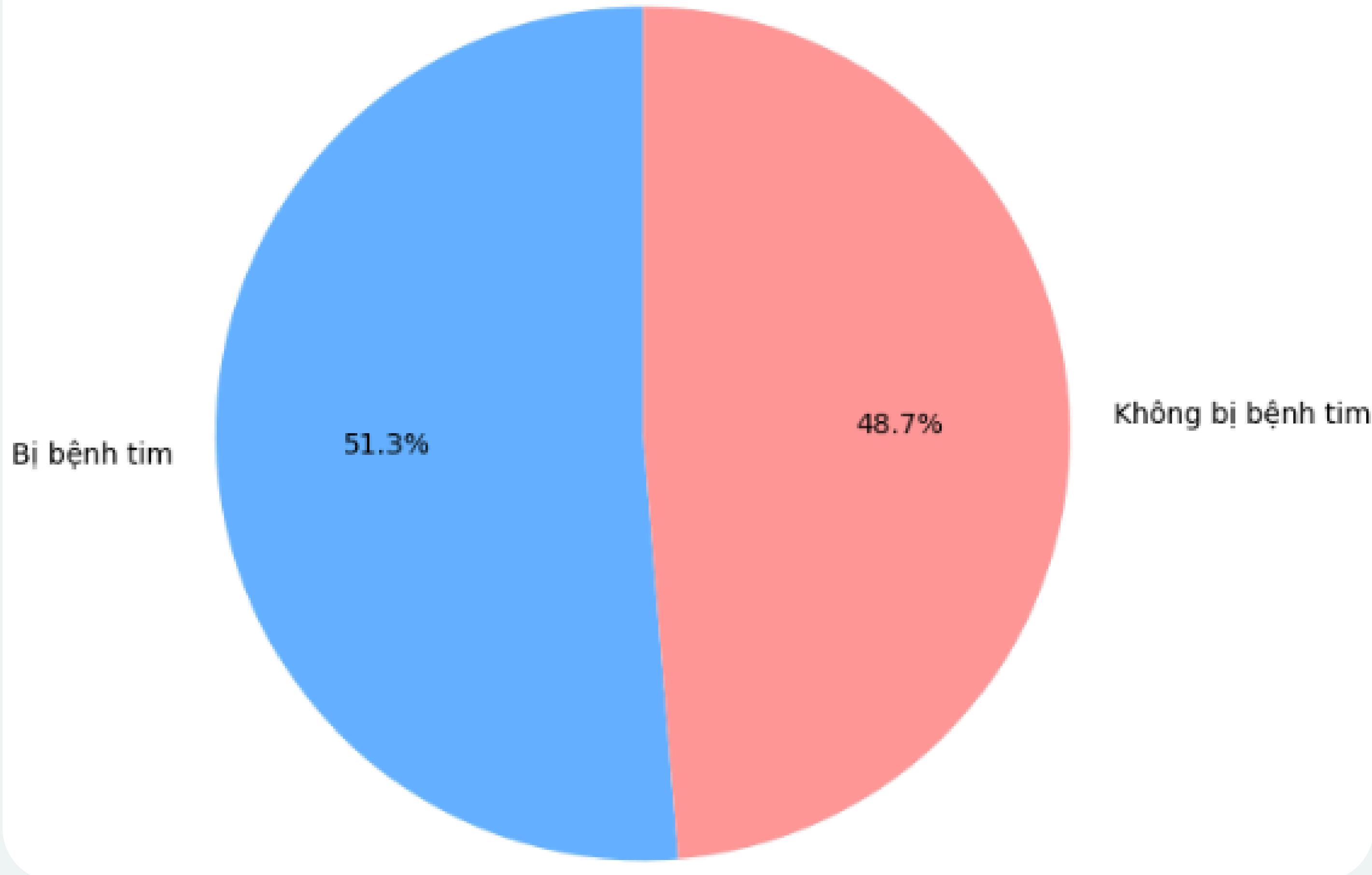


KHÁM PHÁ DỮ LIỆU

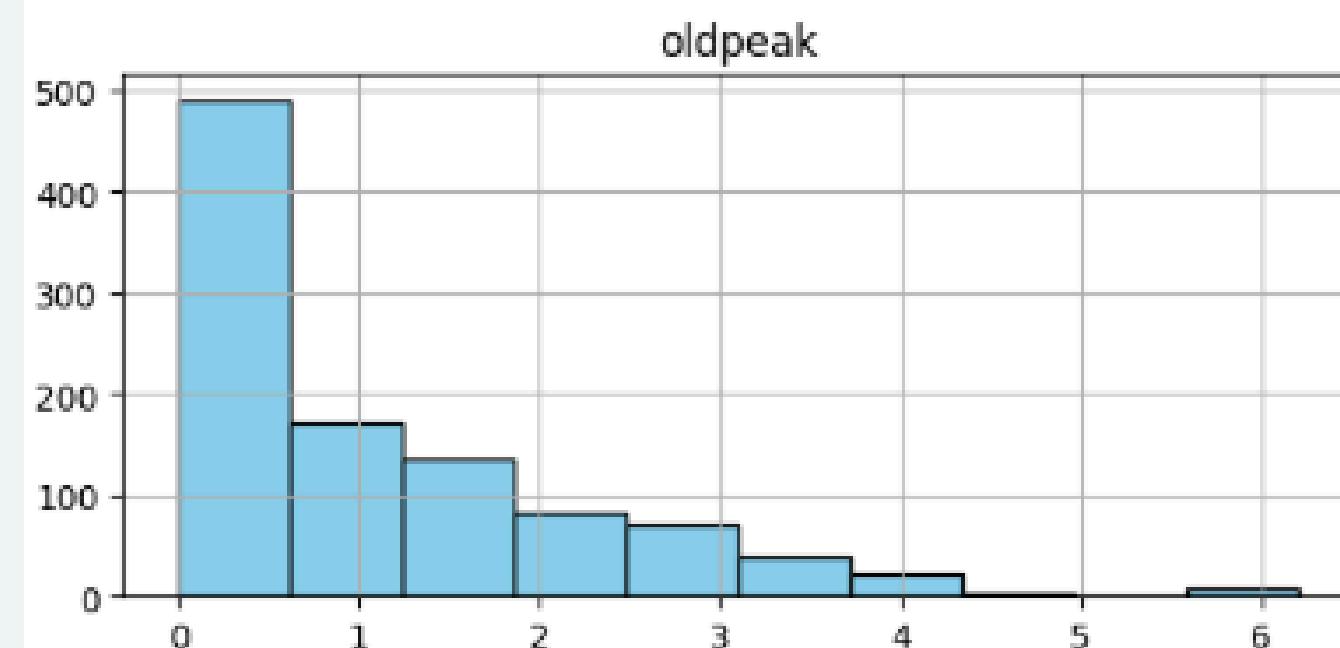
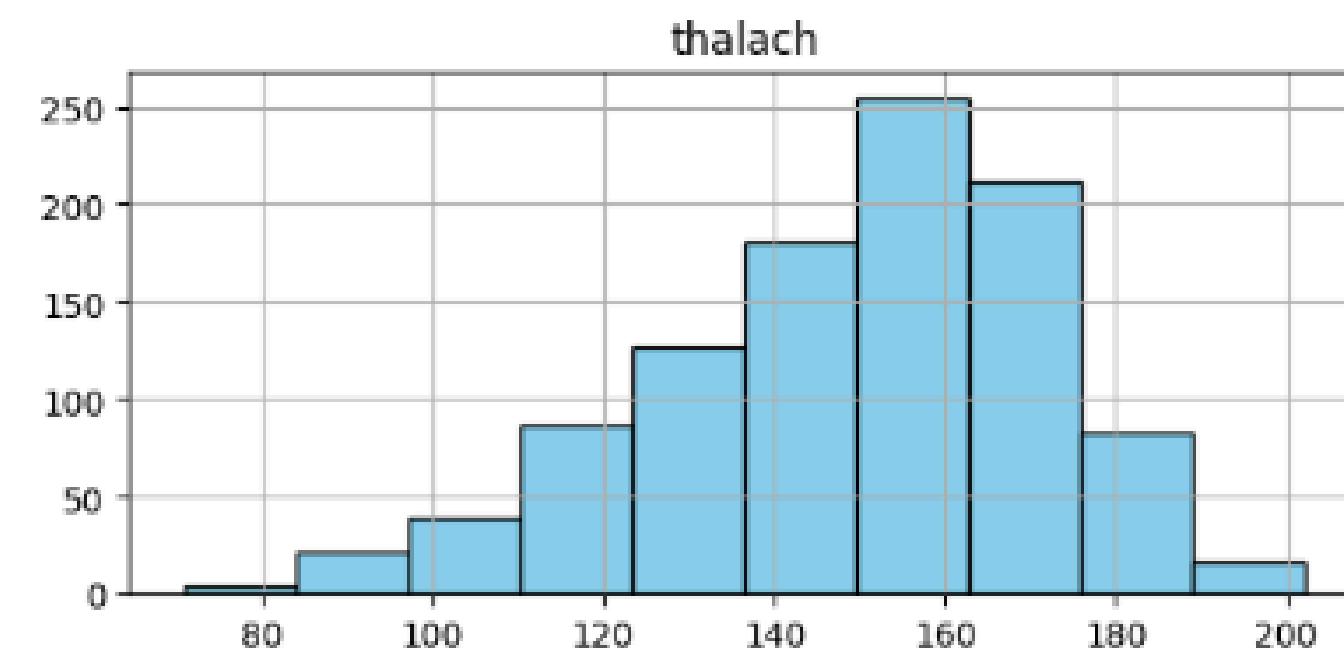
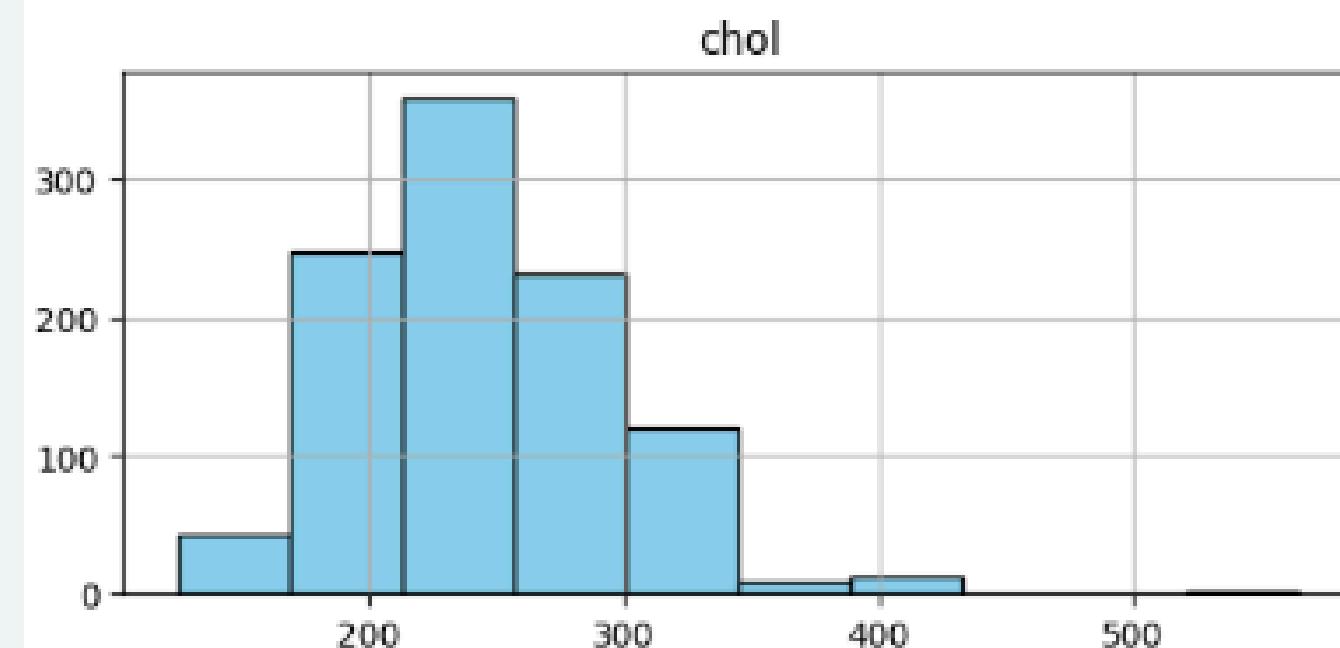
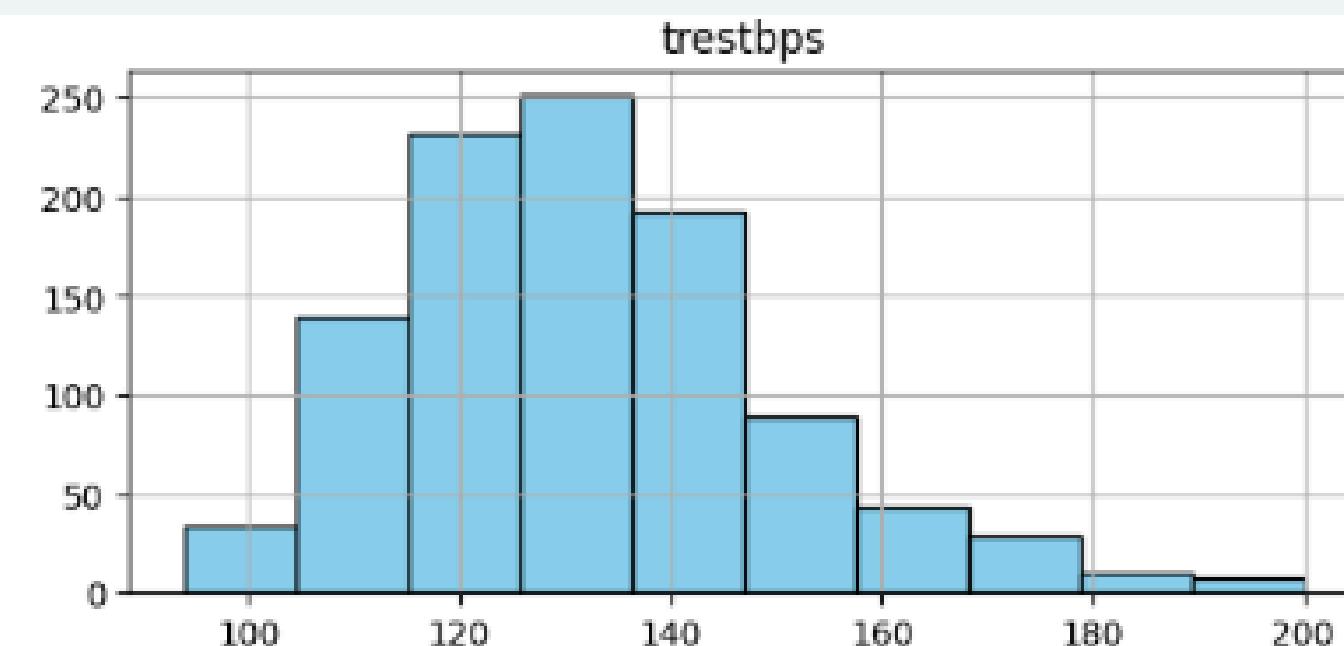
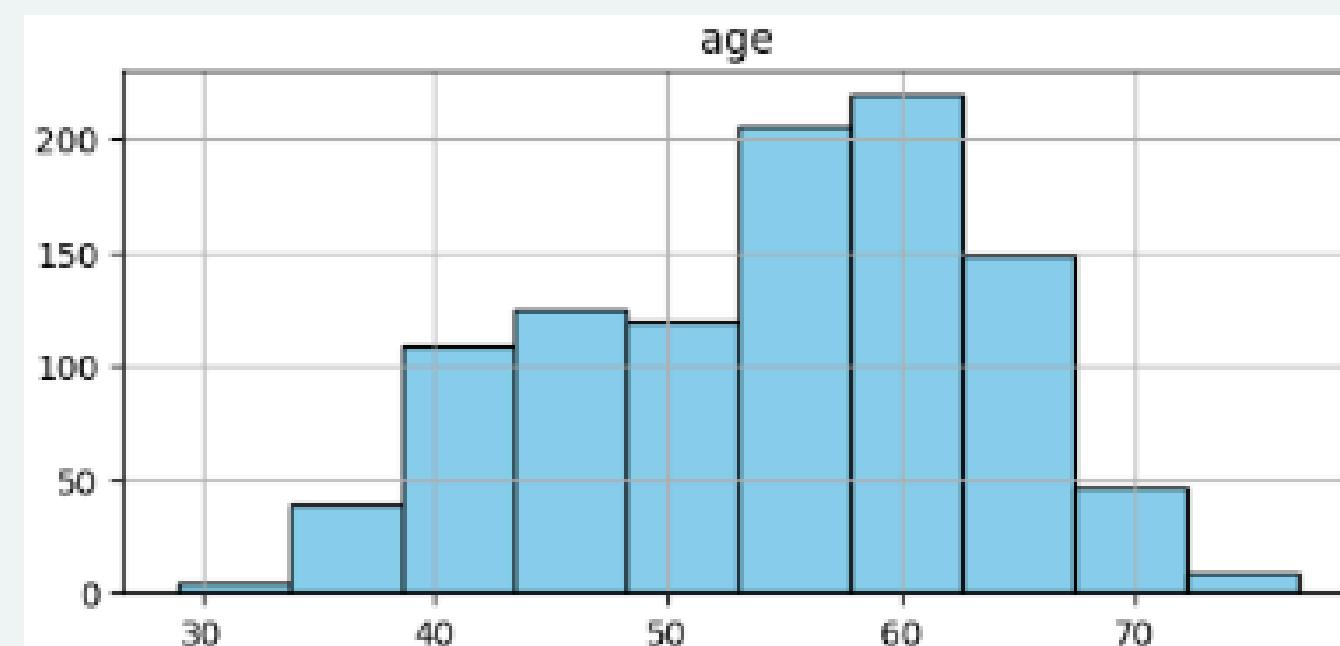
- age – Tuổi của bệnh nhân (tính theo năm).
- sex – Giới tính của bệnh nhân.
- cp – Loại đau ngực.
- trestbps – Thời gian ngủ mỗi đêm (giờ)
- chol – Lượng cholesterol trong huyết thanh.
- fbs – Đường huyết khi đói.
- restecg – Kết quả điện tâm đồ khi nghỉ.
- thalach – Nhịp tim tối đa đạt được trong khi gắng sức.
- exang – Có đau thắt ngực khi gắng sức không.
- oldpeak – Độ suy giảm ST do gắng sức so với lúc nghỉ.
- slope – Độ dốc của đoạn ST khi gắng sức.
- ca – Số lượng mạch máu lớn.
- thal – Tình trạng "thalassemia".

#	Column	Non-Null Count	Dtype
0	age	1025 non-null	int64
1	sex	1025 non-null	int64
2	cp	1025 non-null	int64
3	trestbps	1025 non-null	int64
4	chol	1025 non-null	int64
5	fbs	1025 non-null	int64
6	restecg	1025 non-null	int64
7	thalach	1025 non-null	int64
8	exang	1025 non-null	int64
9	oldpeak	1025 non-null	float64
10	slope	1025 non-null	int64
11	ca	1025 non-null	int64
12	thal	1025 non-null	int64
13	target	1025 non-null	int64

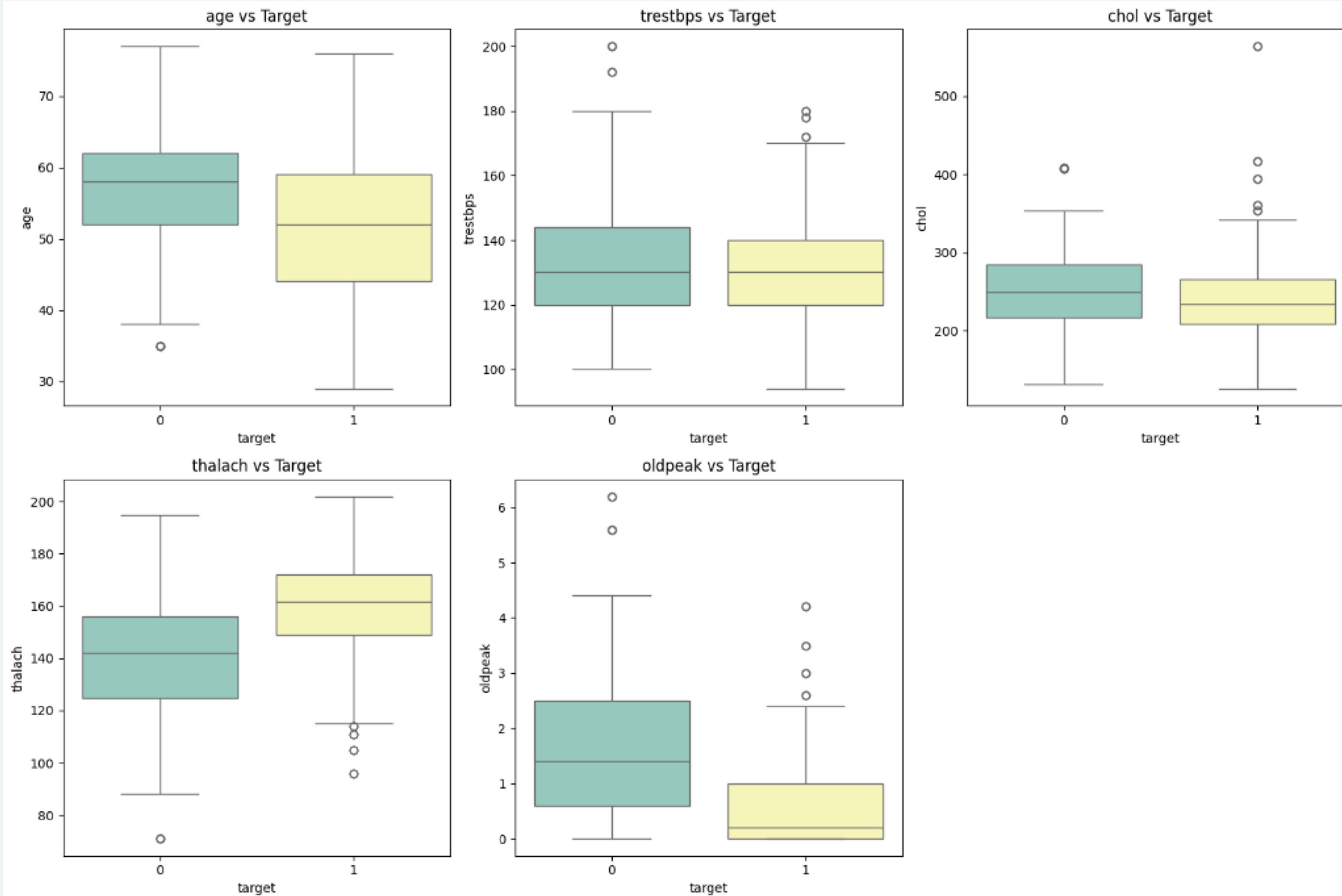
Tỷ lệ người bị bệnh tim và không bị bệnh tim



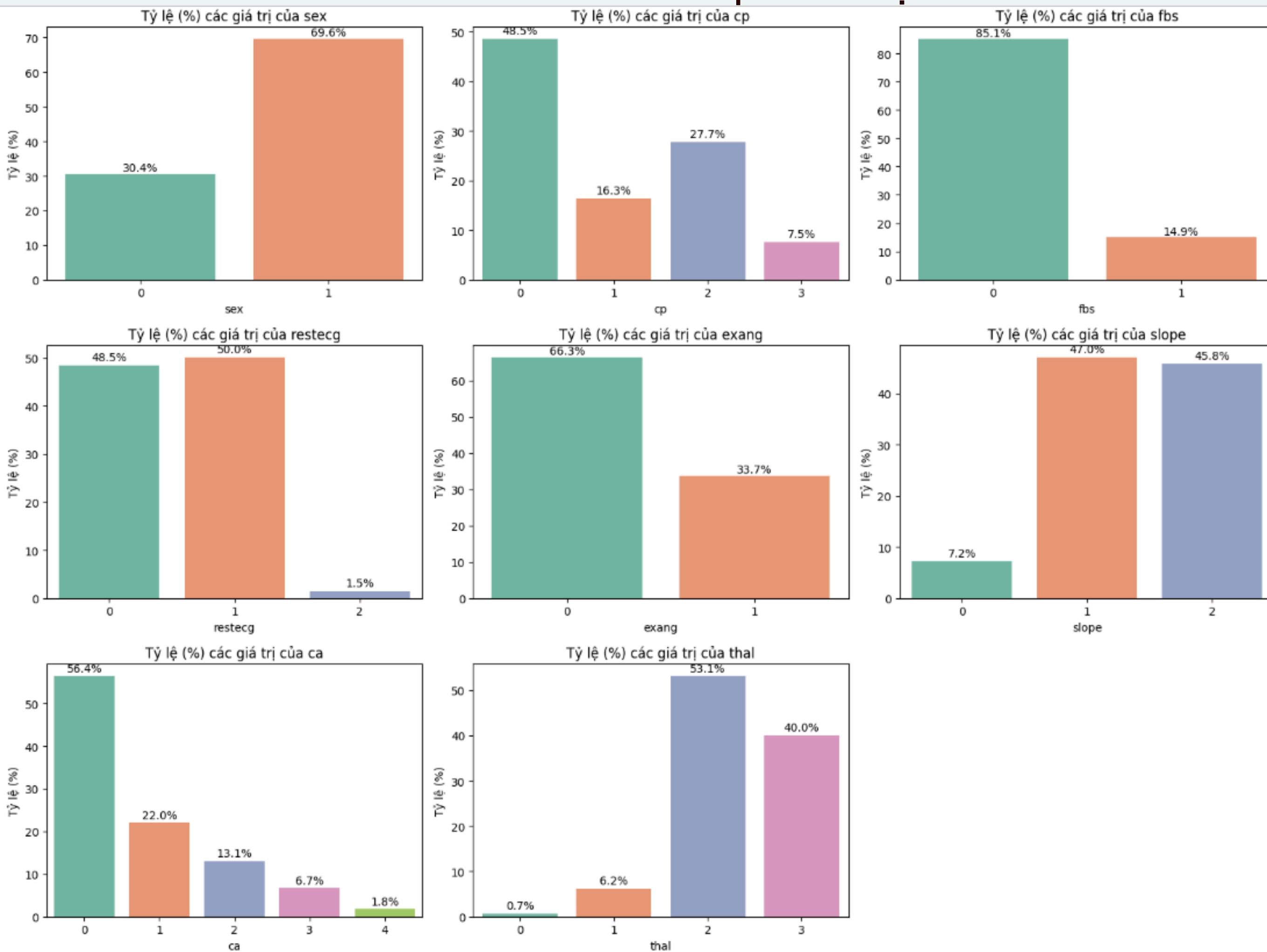
Histogram các biến liên tục



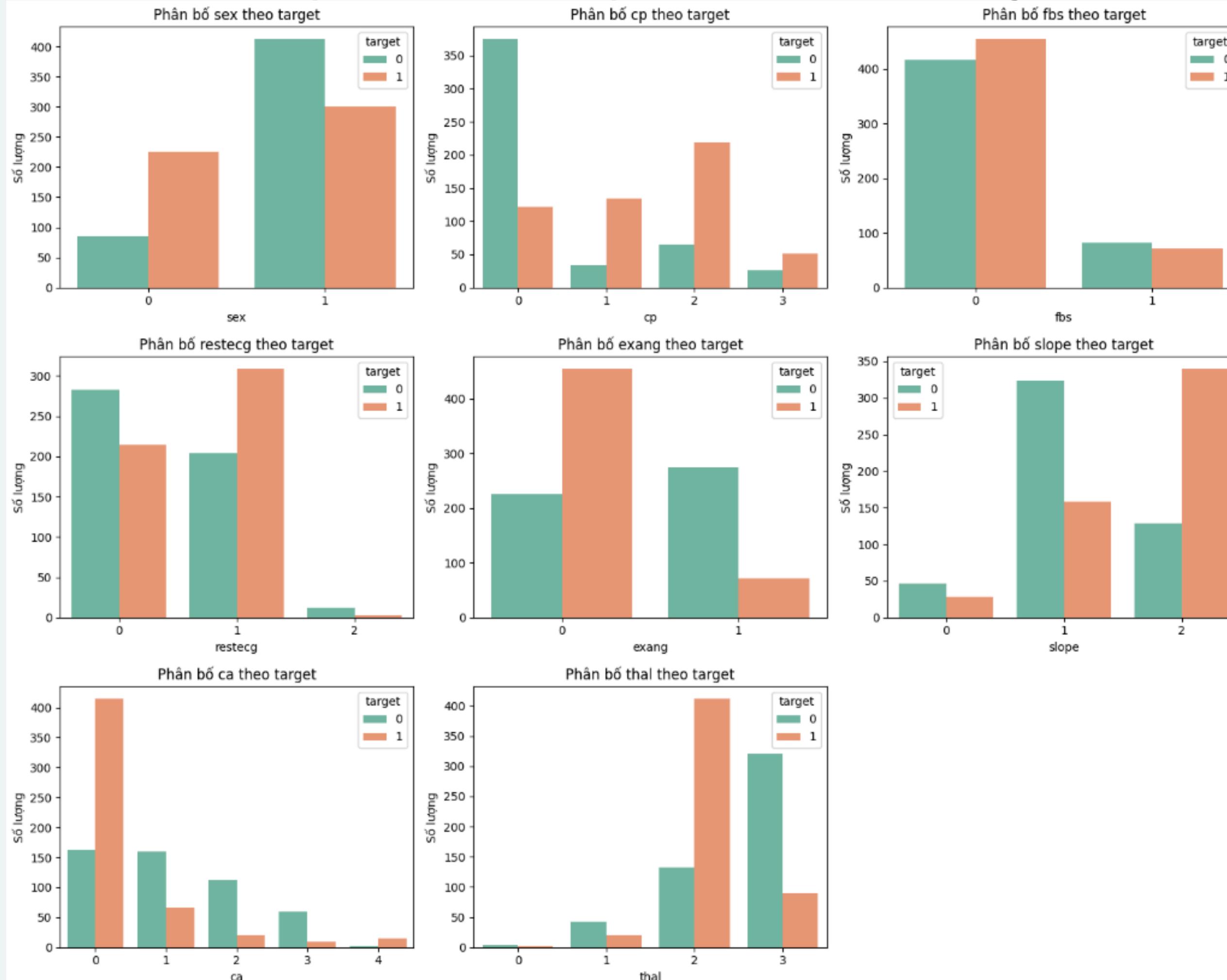
Boxplot các biến liên tục



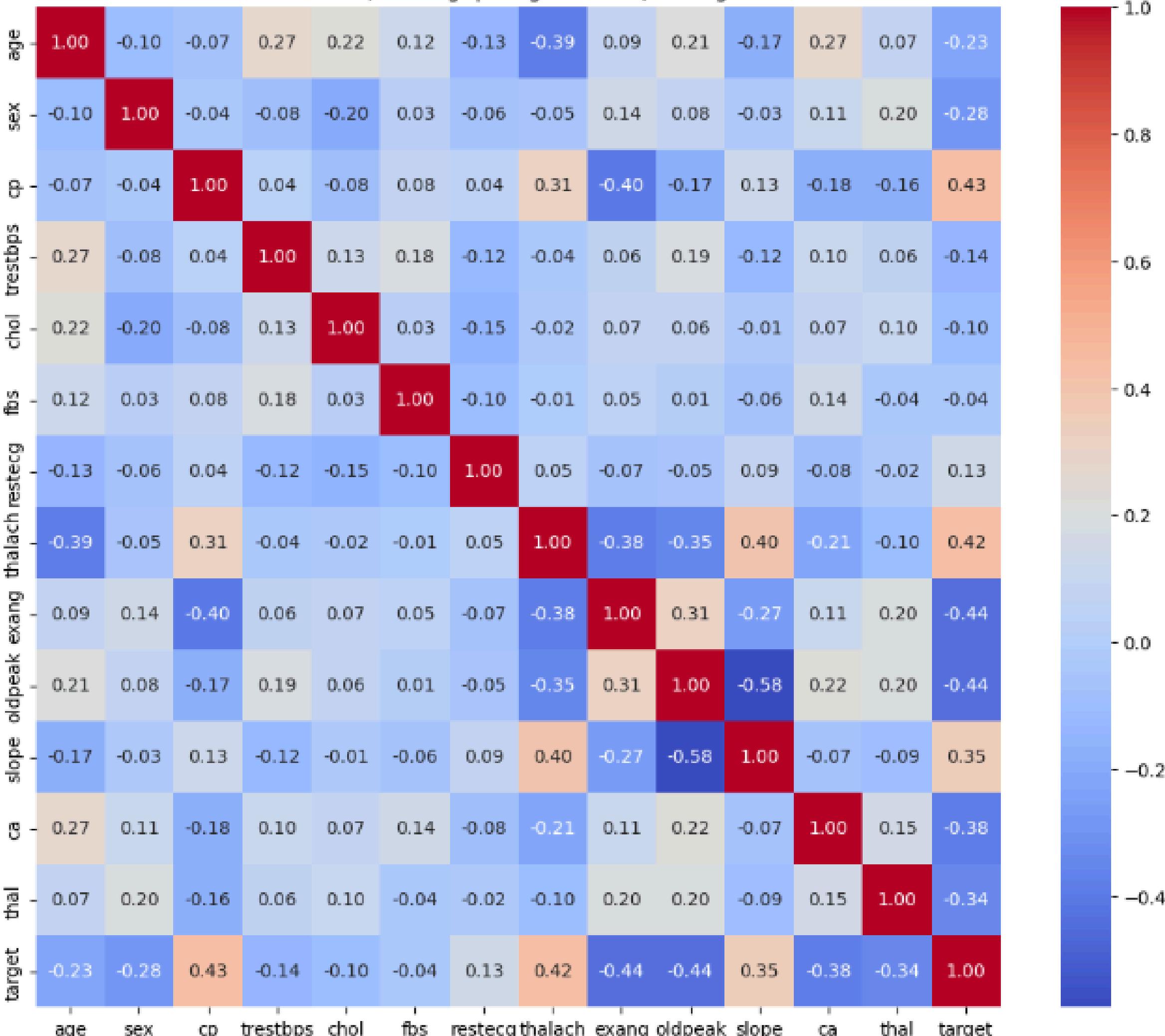
Phân bố các biến phân loại



Countplot các biến phân loại theo target

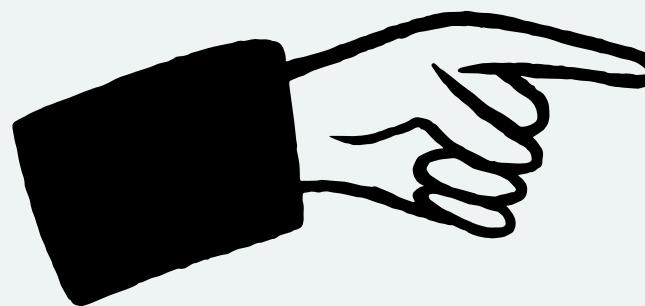


Ma trận tương quan giữa các đặc trưng



TIỀN XỬ LÝ DỮ LIỆU

Loại bỏ outliers các cột: 'oldpeak', 'trestbps', 'chol'



Số dòng nguyên bản: 1025
Số dòng sau khi loại bỏ outliers: 972



	target
1	508
0	464



TIỀN XỬ LÝ DỮ LIỆU

Loại bỏ cột fbs do ảnh hưởng
ít tới biến target

Chuẩn hoá dữ liệu

```
x_train = preprocessing.StandardScaler().fit_transform(x_train)  
x_test = preprocessing.StandardScaler().transform(x_test)
```

```
x train shape: (777, 12)  
y train shape: (777, )  
x test shape: (195, 12)  
y test shape: (195, )
```

```
x = df.drop(["target", "fbs"], axis=1)  
y = df["target"]  
print('X shape', x.shape)  
print('y shape', y.shape)
```

```
X shape (972, 12)  
y shape (972, )
```



Mô hình Logistic Regression (Nguyễn Thành Công)



Lý do chọn mô hình

Bài toán phân loại nhị phân

Đơn giản, dễ tiếp cận

Dữ liệu biến mục tiêu cân bằng



Kết quả huấn luyện

Training Accuracy: 0.8623
Testing Accuracy: 0.8462

Jaccard score: 0.7656

Log loss: 0.375

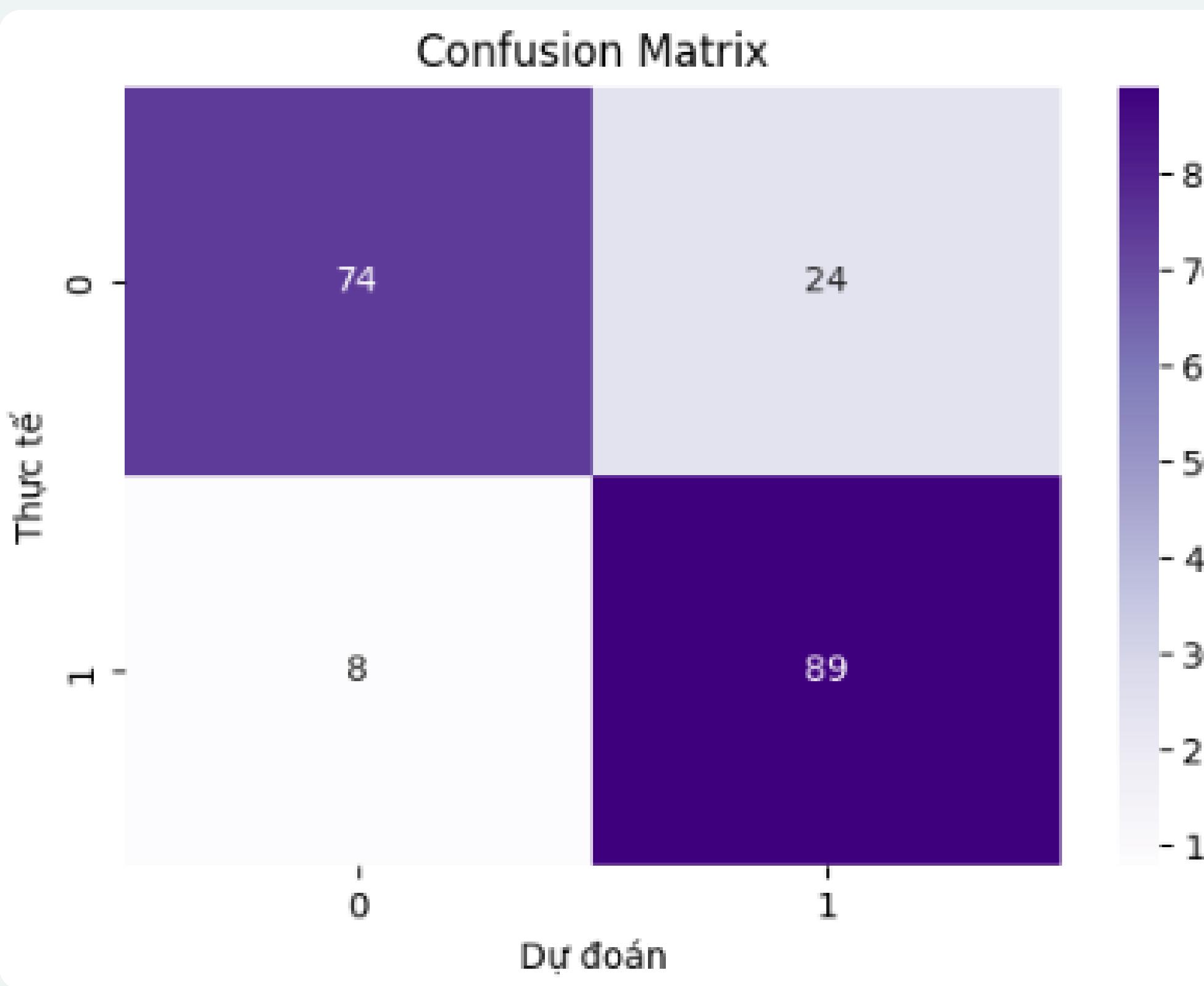
```
model = LogisticRegression(random_state=42)
model.fit(X_train, y_train)
train_score = model.score(X_train, y_train)
test_score = model.score(X_test, y_test)
y_pred = model.predict(X_test)
```

```
jaccard_score(y_test, y_pred)
```

```
y_prob = model.predict_proba(X_test)
log_loss(y_test, y_prob)
```



Confusion Matrix



Classification Report:

	precision	recall	f1-score	support
0	0.8701	0.7701	0.8171	87
1	0.8305	0.9074	0.8673	108
accuracy			0.8462	195
macro avg	0.8503	0.8388	0.8422	195
weighted avg	0.8482	0.8462	0.8449	195



Đánh giá

Kết quả ổn định và đáng tin cậy trên tập dữ liệu dự đoán bệnh tim

- Accuracy = 84.6% → Đúng phần lớn các trường hợp trong tập kiểm tra.
- F1-score (class 1) = 0.8673 → Cân bằng tốt giữa precision và recall đối với người có bệnh

Với lớp 1 (có bệnh):

- Phát hiện gần như đầy đủ bệnh nhân thực sự
- Trong số những người được dự đoán là "có bệnh", khoảng 83% là đúng thực sự.

Với lớp 0 (không có bệnh):

- Khi mô hình dự đoán là "không bệnh", thì phần lớn là đúng.
- Khoảng 23% người không bệnh thật bị dự đoán nhầm là có bệnh
→ Chấp nhận được, nhưng có thể gây cảnh báo sai.

 Kết luận: Logistic Regression là lựa chọn phù hợp cho bài toán phân loại nhị phân này, cho hiệu suất khá tốt, dễ diễn giải và dự đoán xác suất đáng tin cậy.

DECISION TREE CLASSIFIER

```
from sklearn.tree import DecisionTreeClassifier, plot_tree
X_train_DTC, X_test_DTC, y_train_DTC, y_test_DTC = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
```

```
tree = DecisionTreeClassifier(
    criterion="entropy",
    max_depth=10,
    min_samples_leaf=2,
    random_state=42
)
tree.fit(X_train_DTC, y_train_DTC)
y_pred_DTC = tree.predict(X_test_DTC)

print(classification_report(y_test_DTC, y_pred_DTC))

conf_matrix = confusion_matrix(y_test_DTC, y_pred_DTC)
conf_df = pd.DataFrame(conf_matrix, index=["0", "1"], columns=["0", "1"])

plt.figure(figsize=(6, 4))
sns.heatmap(conf_df, annot=True, cmap="Purples", fmt="d")
plt.title("Confusion Matrix")
plt.ylabel("Thực tế")
plt.xlabel("Dự đoán")
plt.show()
```

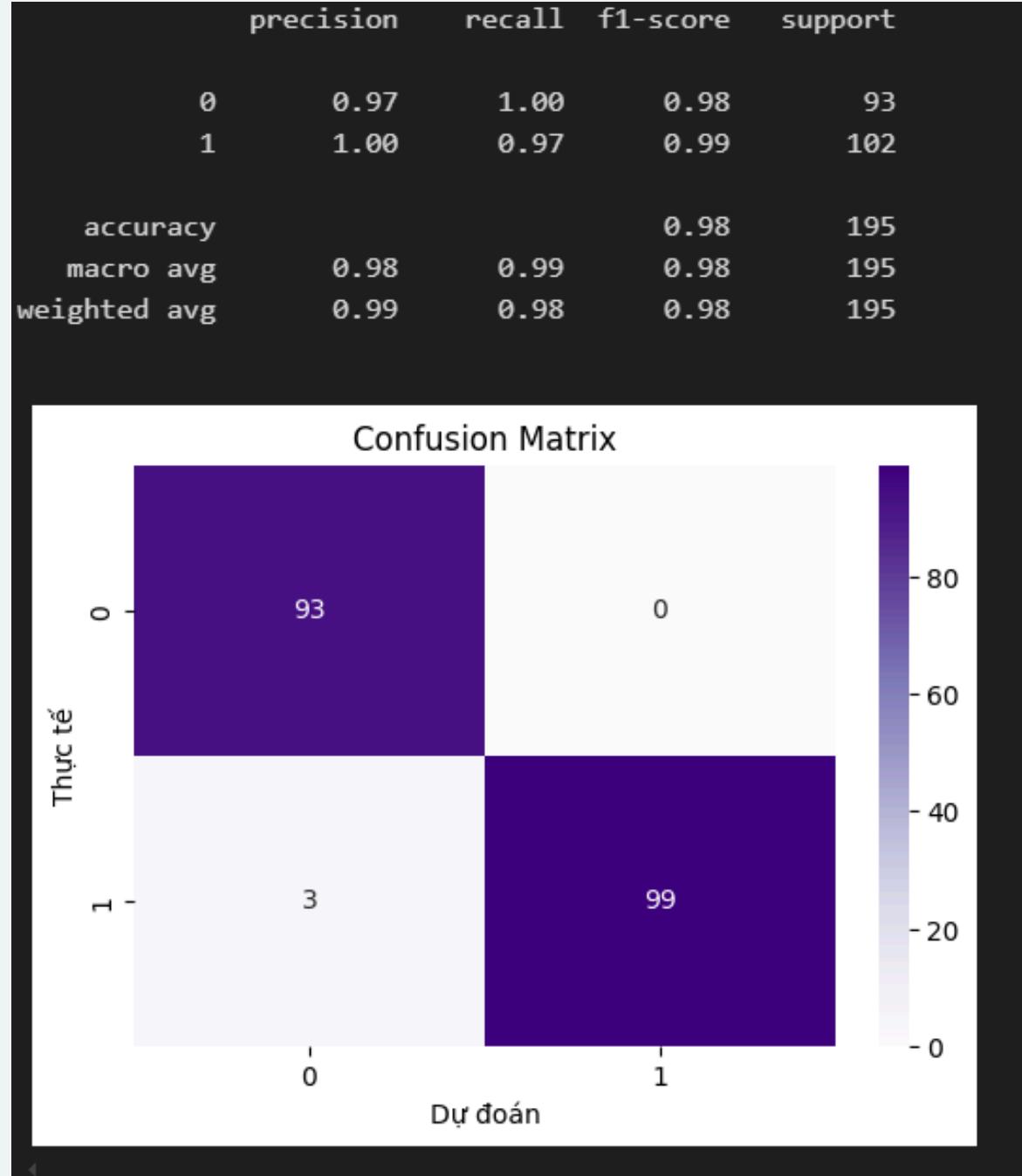
Xây dựng cây với tiêu chí chia entropy để tăng độ chính xác, maxdepth =10 và min samples leaf =2.

Thực hiện huấn luyện trên tập train và dự đoán với tập test.

In ra các chỉ số đánh giá độ chính xác và ma trận nhầm lẫn để đánh giá mô hình.



DECISION TREE CLASSIFIER



Mô hình Decision Tree Classifier hoạt động rất hiệu quả cho bài toán phân loại bệnh tim, với độ chính xác tổng thể là 98%, và các chỉ số precision, recall, F1-score đều rất cao ở cả hai lớp.

Ưu điểm:

- Phân loại chính xác cả hai lớp (bệnh/không bệnh).
- Không có False Positive (không dự đoán nhầm người khỏe là bệnh).
- Chỉ 3 lỗi False Negative, điều này vẫn cần được xem xét kỹ nếu bài toán nhạy cảm (mất bệnh nhân thật).

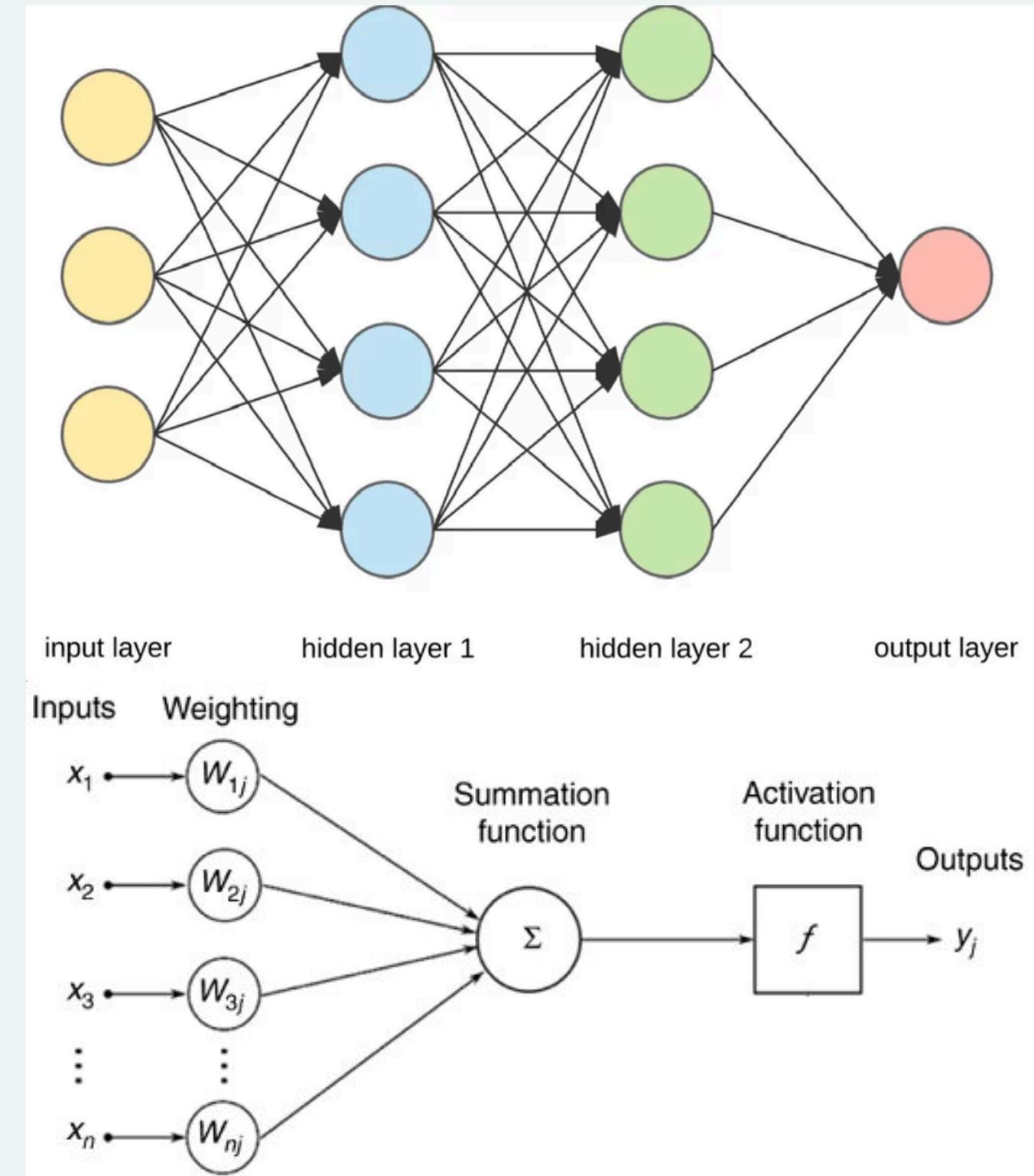
Nhận định về thực tế:

- Với bài toán y tế, recall ở lớp 1 (bệnh nhân thật) rất quan trọng.
- Recall lớp 1 = 97% → Mô hình rất ít bỏ sót bệnh nhân thật.



NEURAL NETWORK MLPCLASSIFIER

- Là mô hình học máy mô phỏng hoạt động của não người, dùng để dự đoán, phân loại, hoặc nhận dạng mẫu từ dữ liệu.
- Mạng Neural bao gồm: tầng đầu vào, 1 hoặc nhiều tầng ẩn và tầng đầu ra.
- Mỗi tầng ẩn chứa nhiều neuron, dùng để thực hiện phép tính và neuron ở mỗi tầng được kết nối với tất cả neuron ở tầng tiếp theo
- Các trọng số (weights) và bias sẽ được điều chỉnh khi huấn luyện để giảm sai số.



NEURAL NETWORK MLPCLASSIFIER

Nhận dữ liệu lớp đầu vào: Các đặc trưng (feature) được đưa vào mạng.

Xử lý qua các lớp ẩn:

Dữ liệu được tính toán qua 2 lớp ẩn, lần lượt có 64 nơ-ron và 32 nơ-ron.

Sử dụng hàm kích hoạt ‘tanh’

Thuật toán tối ưu hóa ‘adam’

Trả kết quả ở lớp đầu ra: Mạng tạo ra dự đoán cuối cùng là mắc bệnh hay không mắc bệnh

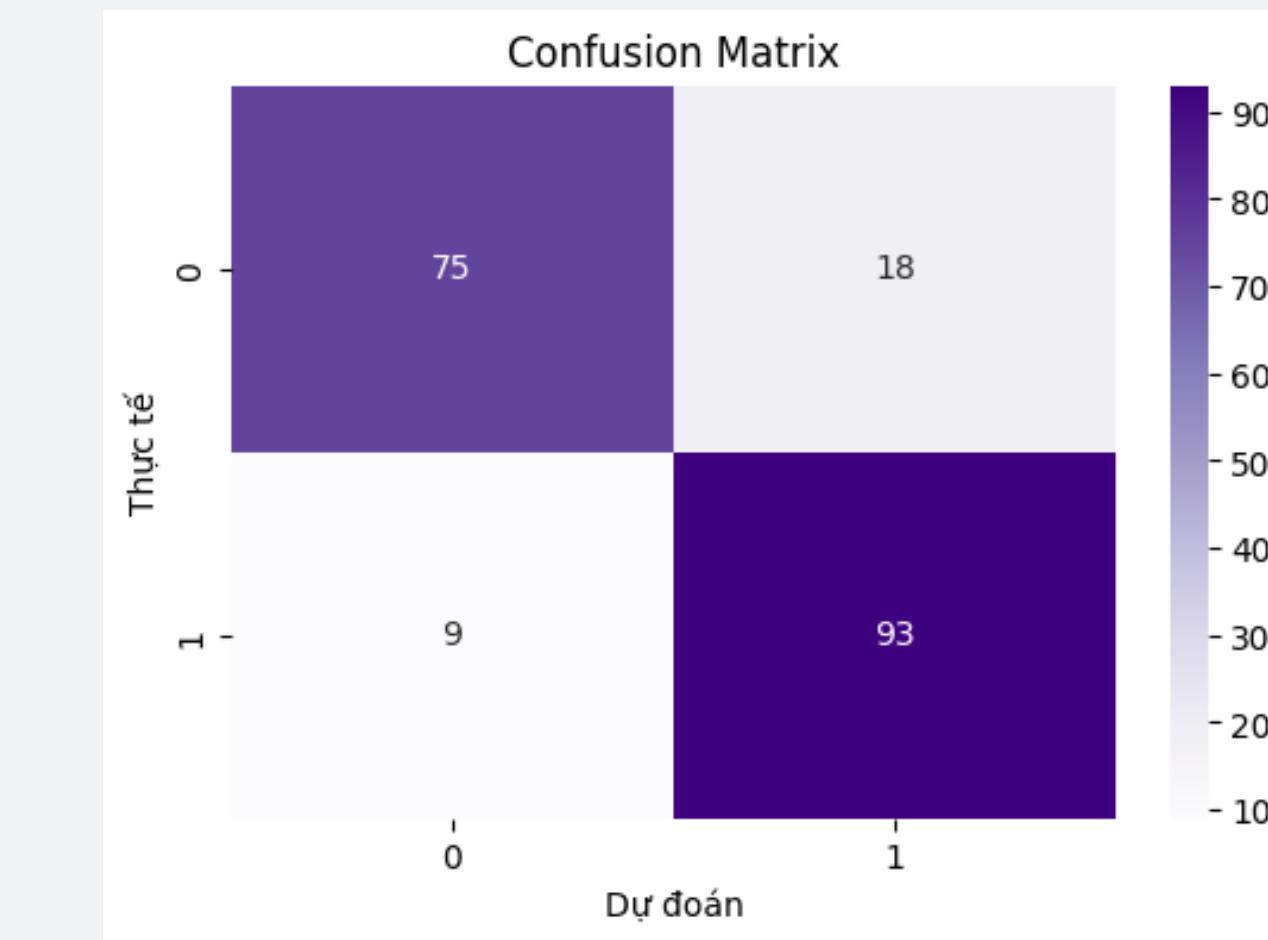
```
# Huấn luyện mô hình Neural Network
from sklearn.neural_network import MLPClassifier

model = MLPClassifier(hidden_layer_sizes=(64, 32),
                      activation='tanh', solver='adam',
                      max_iter=1000, early_stopping=True,
                      random_state=42)

model.fit(x_train, y_train)
```

NEURAL NETWORK MLPCLASSIFIER

Training Accuracy: 0.8546																														
Testing Accuracy: 0.8615																														
Jaccard Score: 0.775																														
Classification Report:																														
<table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>0</td><td>0.8929</td><td>0.8065</td><td>0.8475</td><td>93</td></tr><tr><td>1</td><td>0.8378</td><td>0.9118</td><td>0.8732</td><td>102</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.8615</td><td>195</td></tr><tr><td>macro avg</td><td>0.8653</td><td>0.8591</td><td>0.8603</td><td>195</td></tr><tr><td>weighted avg</td><td>0.8641</td><td>0.8615</td><td>0.8609</td><td>195</td></tr></tbody></table>		precision	recall	f1-score	support	0	0.8929	0.8065	0.8475	93	1	0.8378	0.9118	0.8732	102	accuracy			0.8615	195	macro avg	0.8653	0.8591	0.8603	195	weighted avg	0.8641	0.8615	0.8609	195
	precision	recall	f1-score	support																										
0	0.8929	0.8065	0.8475	93																										
1	0.8378	0.9118	0.8732	102																										
accuracy			0.8615	195																										
macro avg	0.8653	0.8591	0.8603	195																										
weighted avg	0.8641	0.8615	0.8609	195																										



- Mô hình Neural Network MLPClassifier sau khi huấn luyện cho kết quả tốt với độ chính xác 86,15%, cho thấy khả năng phân loại ổn định.
- Mô hình rất hiệu quả trong việc nhận diện đúng các trường hợp dương tính. F1-score của cả hai lớp đều cao (84~87%), thể hiện mô hình cân bằng giữa precision và recall.
- Mô hình học máy có hiệu suất tốt, không bị overfitting, và phù hợp để áp dụng trong thực tế.

RANDOM FOREST

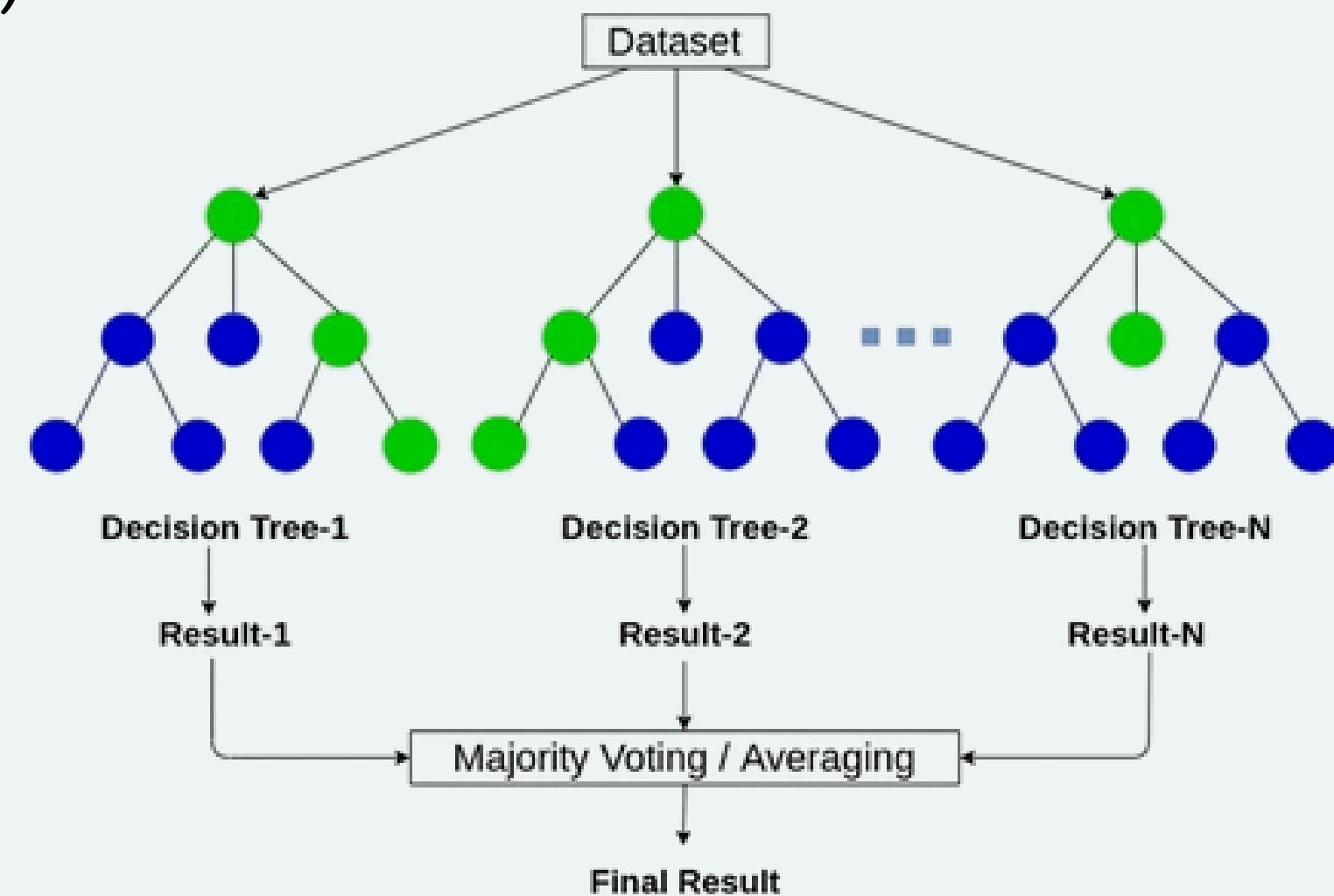
Random Forest là một mô hình học máy dùng nhiều cây quyết định để dự đoán kết quả.

Mỗi cây học từ dữ liệu khác nhau, và kết quả cuối cùng được quyết định dựa trên đa số phiếu (majority voting).

CÁCH HOẠT ĐỘNG

1. TẬP DỮ LIỆU GỐC ĐƯỢC CHIA NHỎ
→ HUẤN LUYỆN NHIỀU CÂY QUYẾT
ĐỊNH.
2. MỖI CÂY ĐƯA RA MỘT KẾT QUẢ
RIÊNG BIỆT.
3. CÁC KẾT QUẢ ĐƯỢC BỎ PHIẾU ĐA
SỐ → CHO RA KẾT QUẢ CUỐI CÙNG.

Random Forest



RANDOM FOREST

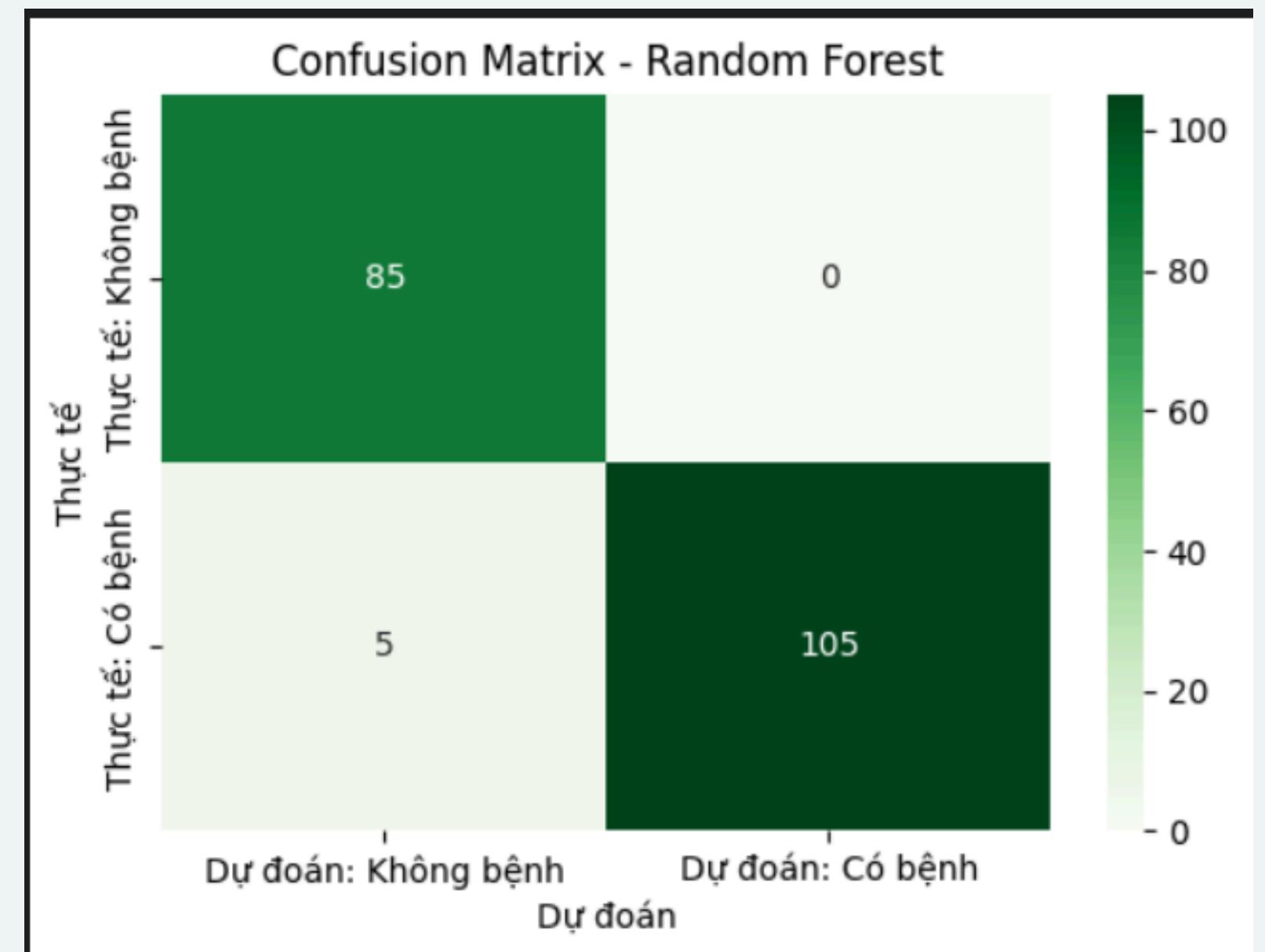
	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0

- **Phù hợp với dữ liệu y tế dạng bảng:** gồm nhiều biến số liên quan đến tuổi tác, huyết áp, cholesterol...
- **Mô hình hóa tốt quan hệ phi tuyến** giữa các chỉ số sức khỏe và nguy cơ bệnh.
- **Không cần chuẩn hóa dữ liệu** → tiết kiệm thời gian tiền xử lý.
- **Ít bị ảnh hưởng bởi giá trị ngoại lai (outliers).**
- **Dễ giải thích** bằng biểu đồ mức độ quan trọng của các đặc trưng.

RANDOMFOREST

Classification Report:

	precision	recall	f1-score	support
0	0.9444	1.0000	0.9714	85
1	1.0000	0.9545	0.9767	110
accuracy			0.9744	195
macro avg	0.9722	0.9773	0.9741	195
weighted avg	0.9758	0.9744	0.9744	195



Hiệu quả mô hình Random Forest

- Accuracy: 94.44% → Mô hình đáng tin cậy trong chẩn đoán bệnh tim
- Recall (phát hiện đúng người bệnh): 97.14%
- Precision (dự đoán đúng không sai): 100%
- F1-score: 97.67% → Cân bằng giữa độ chính xác & độ bao phủ

Điểm nổi bật

- Không có báo động giả (False Positive = 0)
- Chỉ bỏ sót 5/103 ca bệnh → Rất thấp
- Phân loại tốt cả 2 nhóm: "Có bệnh" & "Không bệnh"

RANDOM FOREST LÀ MÔ HÌNH HIỆU QUẢ, CHÍNH XÁC VÀ ĐÁNG TIN CẬY TRONG DỰ ĐOÁN BỆNH TIM. VỚI KHẢ NĂNG PHÁT HIỆN ĐÚNG CA BỆNH VÀ KHÔNG TẠO BÁO ĐỘNG GIẢ, MÔ HÌNH RẤT PHÙ HỢP ĐỂ HỖ TRỢ CHẨN ĐOÁN Y TẾ THỰC TIỄN.

EXTRATREECLASSIFIER

Chọn model

```
cft = LazyClassifier(verbose=0, ignore_warnings=True, custom_metric=None)
models,predictions = cft.fit(X_train, X_test, y_train, y_test)
m = pd.DataFrame(models)
```

Model	Accuracy	Balanced Accuracy	ROC AUC	F1 Score	recall_metric	Time Taken
LGBMClassifier	0.98	0.98	0.98	0.98	1.00	0.22
ExtraTreeClassifier	0.96	0.97	0.97	0.96	1.00	0.02
XGBClassifier	0.98	0.98	0.98	0.98	1.00	0.15
LabelSpreading	0.96	0.97	0.97	0.96	1.00	0.09
BaggingClassifier	0.96	0.97	0.97	0.96	1.00	0.06
DummyClassifier	0.46	0.50	0.50	0.29	1.00	0.02



EXTRATREECLASSIFIER

Chọn tham số huấn luyện model

```
param_grid = {  
    'n_estimators': [100, 200, 300],          # Số lượng cây  
    'max_depth': [None, 10, 20, 30],          # Độ sâu của cây  
    'min_samples_split': [2, 5, 10],          # Số mẫu tối thiểu để chia nút  
    'min_samples_leaf': [1, 2, 4],            # Số mẫu tối thiểu trong lá  
    'max_features': ['sqrt', 'log2', None]    # Cách chọn số feature tại mỗi split  
}  
clf = ExtraTreesClassifier(random_state=42)  
  
grid_search = GridSearchCV(estimator=clf,  
                           param_grid=param_grid,  
                           cv=5,  
                           scoring='recall',  
                           n_jobs=-1,  
                           verbose=1)  
  
grid_search.fit(x_train, y_train)  
print("Best parameters from GridSearch:", grid_search.best_params_)
```



```
Fitting 5 folds for each of 324 candidates, totalling 1620 fits  
Best parameters from GridSearch: {'max_depth': None, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100}
```



EXTRATREECLASSIFIER

Khái niệm ExtraTreesClassifier

EXTRATREESCLASSIFIER (CÒN GỌI LÀ EXTREMELY RANDOMIZED TREES) LÀ MỘT THUẬT TOÁN HỌC MÁY THUỘC HỌ ENSEMBLE LEARNING, ĐƯỢC XÂY DỰNG Dựa TRÊN TẬP HỢP NHIỀU CÂY QUYẾT ĐỊNH (DECISION TREES) ĐỂ GIẢI QUYẾT BÀI TOÁN PHÂN LOẠI.

- CHỌN NGẪU NHIÊN CÁC NGƯỠNG CHIA (THRESHOLDS) THAY VÌ TÌM NGƯỠNG TỐT NHẤT.
- LỰA CHỌN NGẪU NHIÊN TẬP THUỘC TÍNH TẠI MỖI NODE ĐỂ PHÂN CHIA.

CHÍNH NHỜ SỰ NGẪU NHIÊN CAO NÀY, EXTRATREES GIÚP:

- GIẢM PHƯƠNG SAI (VARIANCE), TRÁNH OVERFITTING.
- TĂNG TỐC ĐỘ HUẤN LUYỆN, VÌ KHÔNG CẦN TÌM NGƯỠNG CHIA TỐT NHẤT NHƯ RANDOM FOREST.
- GIỮ ĐƯỢC ĐỘ CHÍNH XÁC CAO, NHẤT LÀ VỚI CÁC BÀI TOÁN CÓ DỮ LIỆU PHỨC TẠP HOẶC NHIỄU.

Huấn luyện model

```
grid_search.fit(x_train, y_train)
```

EXTRATREECLASSIFIER

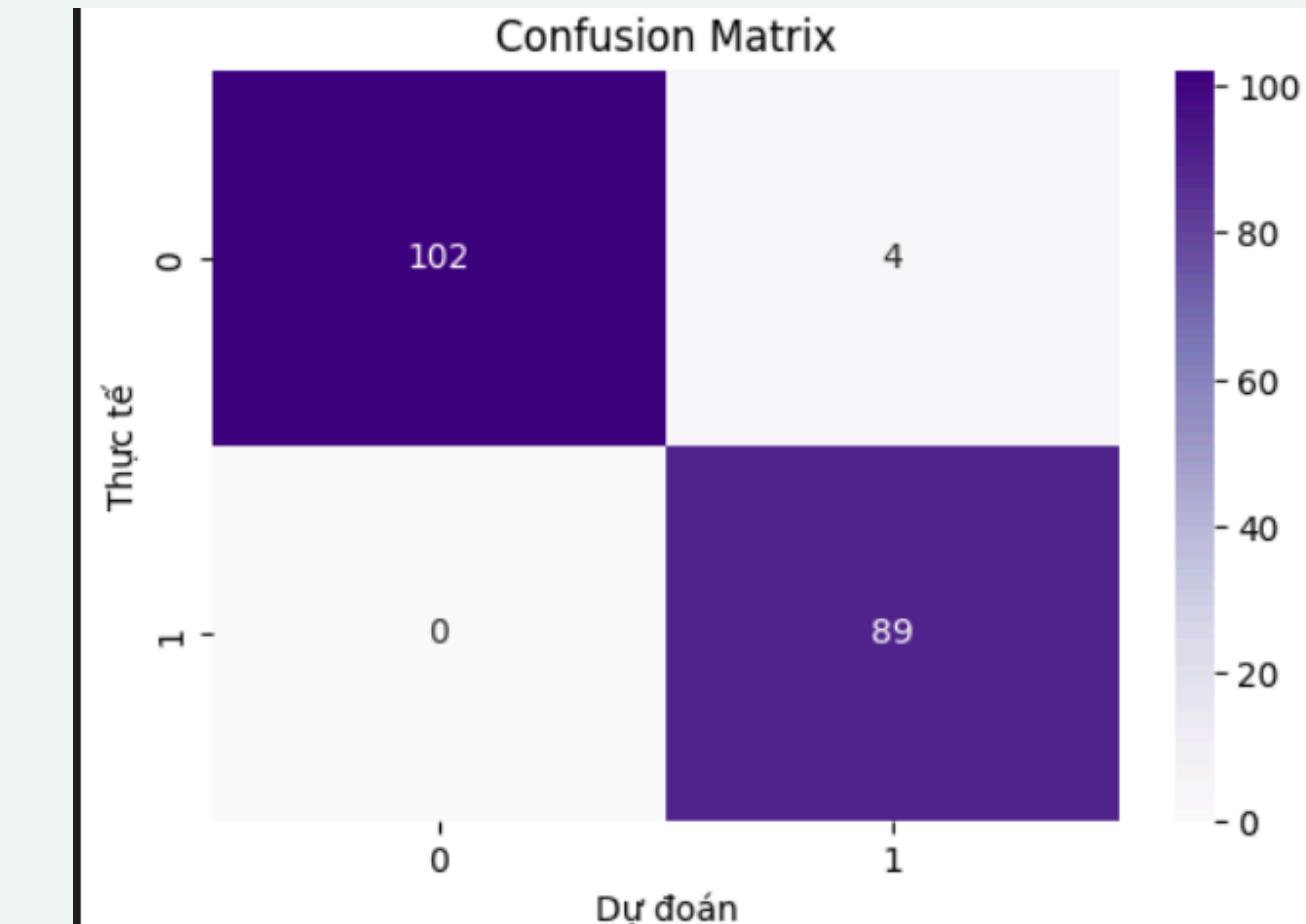
Đánh giá

```
print("\nFull classification report:")
print(classification_report(y_test, y_pred))

Full classification report:
      precision    recall  f1-score   support

          0       1.00     0.96    0.98    106
          1       0.96     1.00    0.98     89

      accuracy                           0.98    195
    macro avg       0.98     0.98    0.98    195
weighted avg       0.98     0.98    0.98    195
```



ĐÁNH GIÁ:

PRECISION (ĐỘ CHÍNH XÁC):

- PRECISION CHO CLASS 0: 1.00 → 100% DỰ ĐOÁN LÀ 0 ĐỀU ĐÚNG.
- PRECISION CHO CLASS 1: 0.96 → CÓ 4% DỰ ĐOÁN LÀ 1 NHƯNG THỰC TẾ LÀ 0 (FALSE POSITIVE).

RECALL (ĐỘ BAO PHỦ):

- RECALL CHO CLASS 0: 0.96 → 96% CÁC MẪU THẬT LÀ 0 ĐƯỢC MÔ HÌNH DỰ ĐOÁN ĐÚNG.
- RECALL CHO CLASS 1: 1.00 → TẤT CẢ CÁC MẪU THẬT LÀ 1 ĐỀU ĐƯỢC DỰ ĐOÁN ĐÚNG.

F1-SCORE (TRUNG BÌNH HÀI HÒA GIỮA PRECISION VÀ RECALL):

- CLASS 0: 0.98
- CLASS 1: 0.98 → CAO VÀ RẤT CÂN BẰNG → MÔ HÌNH TỐT VỚI CẢ HAI LỚP.



DECISION TREE CLASSIFIER

🔎 KẾT LUẬN:

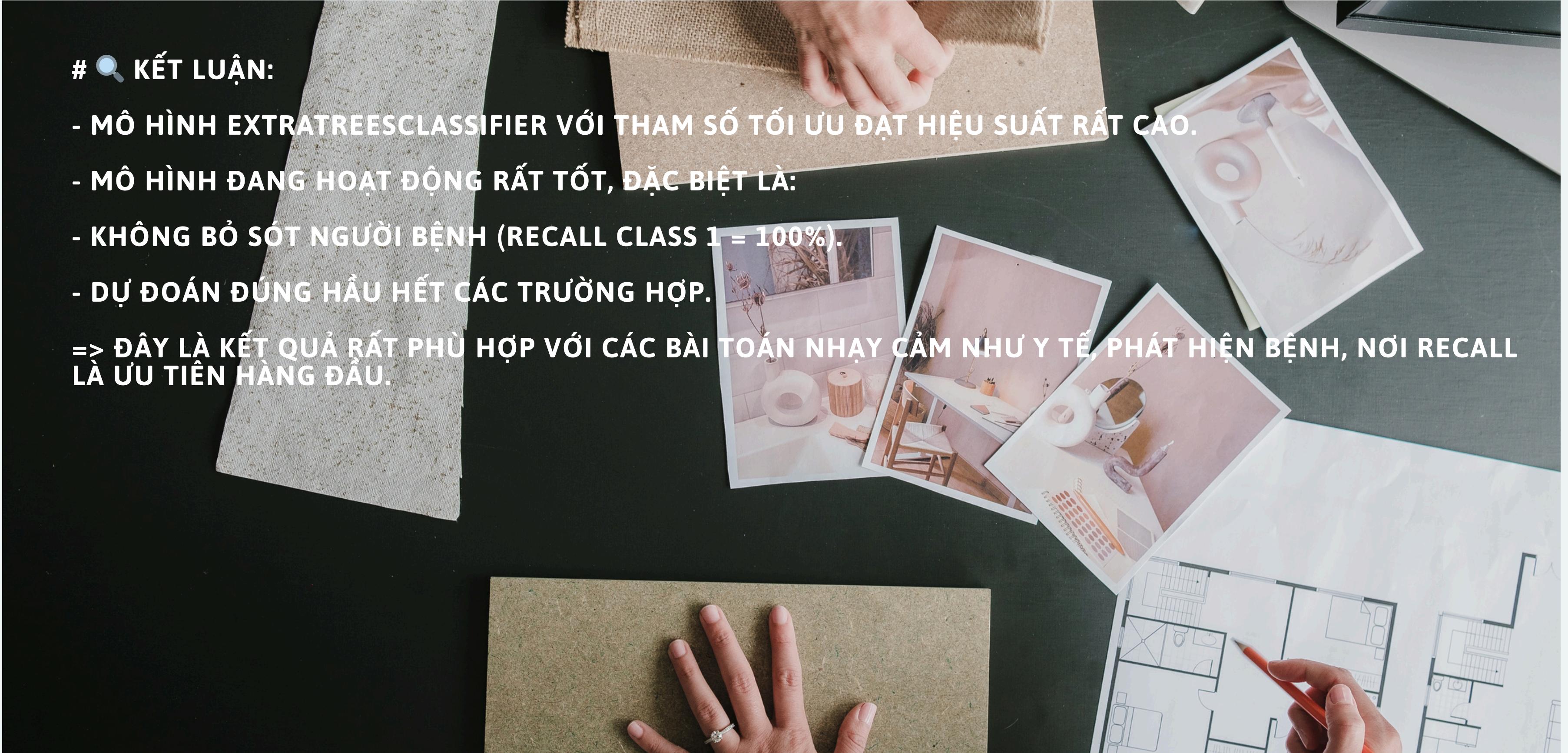
- MÔ HÌNH EXTRATREESCLASSIFIER VỚI THAM SỐ TỐI ƯU ĐẠT HIỆU SUẤT RẤT CAO.

- MÔ HÌNH ĐANG HOẠT ĐỘNG RẤT TỐT, ĐẶC BIỆT LÀ:

- KHÔNG BỎ SÓT NGƯỜI BỆNH (RECALL CLASS 1 = 100%).

- DỰ ĐOÁN ĐÚNG HẦU HẾT CÁC TRƯỜNG HỢP.

=> ĐÂY LÀ KẾT QUẢ RẤT PHÙ HỢP VỚI CÁC BÀI TOÁN NHẠY CẢM NHƯ Y TẾ, PHÁT HIỆN BỆNH, NƠI RECALL LÀ ƯU TIÊN HÀNG ĐẦU.



KẾT LUẬN VÀ HƯỚNG CẢI THIỆN PHÁT TRIỂN MODEL

Bảng tổng hợp đánh giá các mô hình

Mô hình	Lớp	Precision	Recall	F1-score
---	---	---	---	---
Logistic Regression	0	0.8701	0.7701	0.8171
	1	0.8305	0.9074	0.8673
Extra Trees	0	1.0000	0.9600	0.9800
	1	0.9600	1.0000	0.9800
Decision Tree	0	0.9700	1.0000	0.9800
	1	1.0000	0.9700	0.9900
Neural Network	0	0.8546	0.8065	0.8475
	1	0.8615	0.9118	0.8732
Random Forest	0	0.9444	1.0000	0.9714
	1	1.0000	0.9545	0.9767

TỐT NHẤT VỀ F1-SCORE TRUNG BÌNH:

- DECISION TREE: 0.980 (CLASS 0), 0.990 (CLASS 1) → CAO NHẤT.
- EXTRATREES, RANDOM FOREST: RẤT SÁT NHAU, F1 ≈ 0.97–0.98.

ỔN ĐỊNH NHẤT (PRECISION ≈ RECALL):

- RANDOM FOREST VÀ NEURAL NETWORK CÓ ĐỘ CÂN BẰNG CAO GIỮA PRECISION VÀ RECALL → KHÔNG BỊ LỆCH VỀ BÁO ĐỘNG SAI HAY BỎ SỐT.

LOGISTIC REGRESSION LÀ MÔ HÌNH ĐƠN GIẢN NHẤT:

- CHO KẾT QUẢ KHÁ ỔN, NHƯNG THẤP HƠN CÁC MÔ HÌNH PHỨC TẠP HƠN.
- PHÙ HỢP NẾU BẠN CẦN MÔ HÌNH DỄ HIỂU, DIỄN GIẢI

KẾT LUẬN VÀ HƯỚNG CẢI THIỆN PHÁT TRIỂN MODEL

Kết luận

DỰA VÀO KẾT QUẢ CỦA PRECISION, RECALL, F1-SCORE TA THẤY CÓ LOGISTIC REGRESSION LÀ HIỆU SUẤT THẤP HƠN HẲN CÒN 4 MODEL KIA THÌ CŨNG NHƯ NHAU NHƯNG CỦA MODEL LOGISTIC REGRESSION VÀ RANDOM FOREST CÓ DỰ PHẦN RECALL BÊN 1 THẤP HƠN MÀ TRONG KHI CHUẨN ĐOÁN NẾU BỊ BỆNH MÀ CHUẨN ĐOÁN KHÔNG BỆNH RẤT NGUY HIỂM CÒN BÊN EXTRA TREES VÀ NEURAL NETWORK THÌ RECALL Ở BÊN 0 THẤP HƠN THÌ NGƯỜI KHÔNG BỆNH CHUẨN ĐOÁN BỊ BỆNH ÍT RA THÌ ĐỖ NGUY HIỂM HƠN.

=> CHỌN MODEL EXTRA TREES HOẶC NEURAL NETWORK TUY NHIÊN TA CHỌN EXTRA TREES DO:

- TỐC ĐỘ NHANH
- GIẢM OVERRFITTING TỐT
- KHÔNG CẦN XỬ LÝ DỮ LIỆU CẦU KÌ
- HUẤN LUYỆN XONG CÒN BIẾT ĐƯỢC ĐỘ QUAN TRỌNG CỦA MỖI FEATURE
- PHÙ HỢP VỚI DỮ LIỆU BÀI TOÁN DƯỚI 10000 DÒNG DỮ LIỆU

→ LÀ LỰA CHỌN TOÀN DIỆN, ĐƠN GIẢN NHƯNG MẠNH MẼ, ĐẶC BIỆT HIỆU QUẢ VỚI DỮ LIỆU DẠNG BẢNG NHƯ DỰ ĐOÁN BỆNH TIM.

Hướng phát triển

- TỐI ƯU SIÊU THAM SỐ NÂNG CAO HƠN
- XỬ LÝ DỮ LIỆU TỐT HƠN(THÊM BỚT 1 SỐ FEATURE)
- KẾT HỢP MÔ HÌNH (ENSEMBLE)
- PHÂN TÍCH LỒI SÂU HƠN