

FINAL PROJECT

DA46

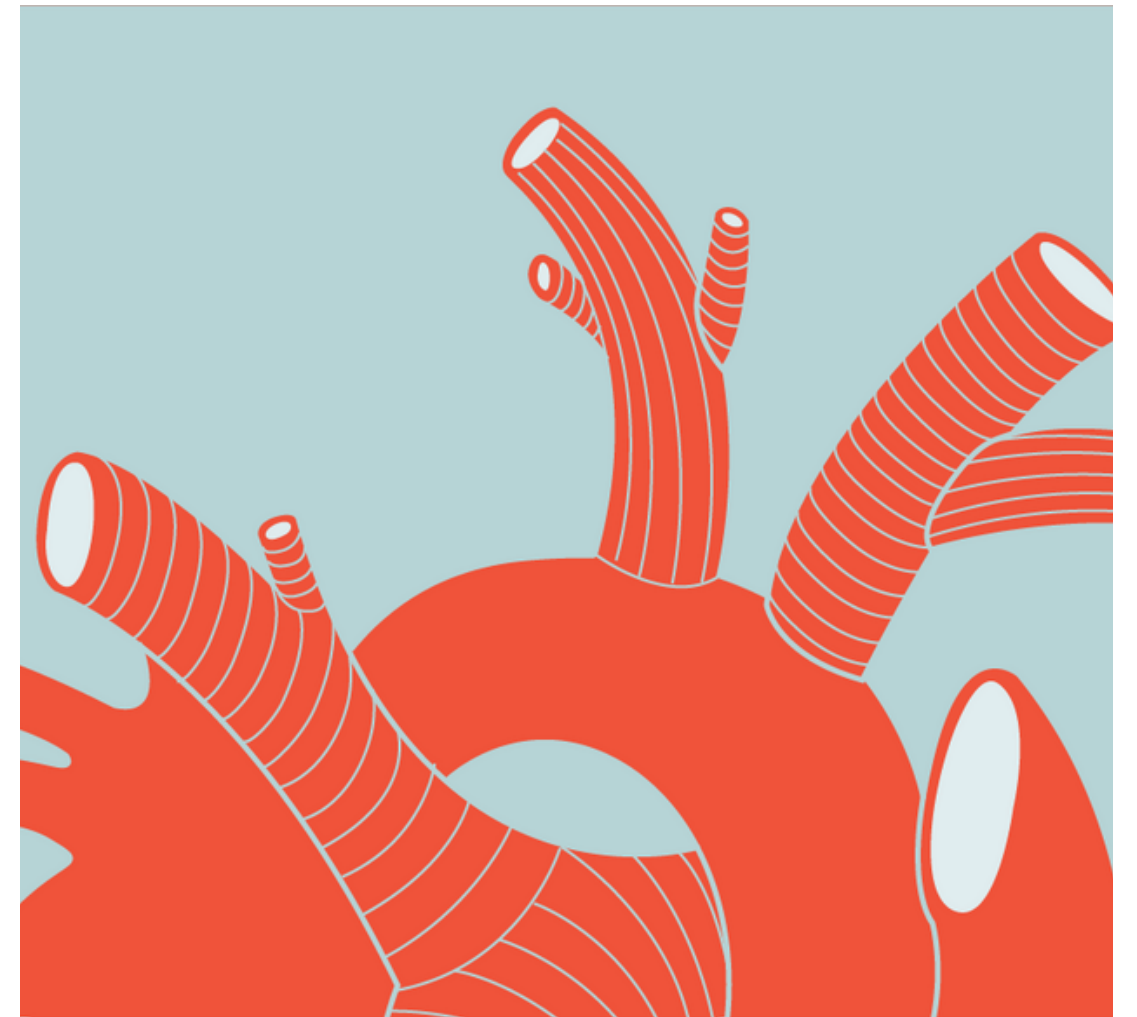
# DỰ ĐOÁN BỆNH TIM MẠCH

TRAN THI THUY TIEN  
LY HIEU SIENG

# NỘI DUNG

---

- Tổng quan
- EDA
- Xây dựng mô hình
- Đánh giá mô hình
- Kết luận



# TỔNG QUAN

---

- Bệnh tim mạch là do các rối loạn của tim và mạch máu.
  - Bệnh mạch vành (nhồi máu cơ tim)
  - Tai biến mạch máu não (đột quỵ)
  - Tăng huyết áp tăng (cao huyết áp)
  - ...
- Các nguyên nhân chính:
  - Sử dụng thuốc lá
  - Thiếu hoạt động thể lực
  - Chế độ ăn uống không lành mạnh
  - Sử dụng rượu, bia ở mức độ nguy hại
- Có thể phòng ngừa được bằng cách giải quyết các yếu tố nguy cơ hành vi trên



***Việc chẩn đoán sớm có thể giúp bệnh nhân gia tăng cơ hội sống.***

***Việc chẩn đoán có thể được thực hiện bằng học máy.***

# EDA

- Tập dữ liệu được sử dụng là một phần của tập dữ liệu Heart Disease từ UCI machine learning repository.
- Đây là kết quả xét nghiệm lâm sàng của 303 bệnh nhân tại Phòng khám Cleveland ở Cleveland, Ohio, Mỹ. Được dùng cho việc dự đoán bệnh mạch vành.
- Tập dữ liệu gồm 14 thuộc tính bao gồm cả biến mục tiêu.

```
df.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          299 non-null    float64   
12  thal        301 non-null    float64   
13  num         303 non-null    int64    
dtypes: float64(3), int64(11)  
memory usage: 33.3 KB
```

```
df = df.rename(columns={'num': 'target'})
```

```
df.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	1	145	233	1	2	150	0	2.3	3	0.0	6.0	0
1	67	1	4	160	286	0	2	108	1	1.5	2	3.0	3.0	2
2	67	1	4	120	229	0	2	129	1	2.6	2	2.0	7.0	1
3	37	1	3	130	250	0	0	187	0	3.5	3	0.0	3.0	0
4	41	0	2	130	204	0	2	172	0	1.4	1	0.0	3.0	0

```
print(df['target'].unique())
```

```
[0 2 1 3 4]
```

- “target” đề cập đến sự hiện diện của bệnh tim ở bệnh nhân. Có giá trị từ 0 đến 4.
- Các thử nghiệm với tập dữ liệu này tập trung vào việc phân biệt sự hiện diện (giá trị 1,2,3,4) với sự vắng mặt (giá trị 0) nên sẽ tiến hành xử lý các giá trị 1,2,3,4 thành 1.

```
df['target'] = df['target'].replace([1, 2, 3, 4], 1)
```

```
print(df['target'].unique())
```

```
[0 1]
```

```
df.head()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	1	145	233	1	2	150	0	2.3	3	0.0	6.0	0
1	67	1	4	160	286	0	2	108	1	1.5	2	3.0	3.0	1
2	67	1	4	120	229	0	2	129	1	2.6	2	2.0	7.0	1
3	37	1	3	130	250	0	0	187	0	3.5	3	0.0	3.0	0
4	41	0	2	130	204	0	2	172	0	1.4	1	0.0	3.0	0

```
categorical_data = ['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', 'thal']
```

```
numerical_data = ['age', 'trestbps', 'chol', 'thalach', 'oldpeak']
```

# EDA

- Kiểm tra và xử lý giá trị null

```
print("missing_values_count:")
print(df.isnull().sum())
```

```
missing_values_count:
age          0
sex          0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalach     0
exang       0
oldpeak     0
slope       0
ca          4
thal        2
target      0
dtype: int64
```

```
print(df['ca'].unique())
print(df['thal'].unique())
```

```
[ 0.  3.  2.  1. nan]
[ 6.  3.  7. nan]
```

```
# Thay thế giá trị null bằng giá trị phổ biến nhất
print(df['ca'].mode()[0])
df['ca'] = df['ca'].fillna(df['ca'].mode()[0])
print(df['ca'].unique())
```

```
print(df['thal'].mode()[0])
df['thal'] = df['thal'].fillna(df['thal'].mode()[0])
print(df['thal'].unique())
```

```
3.0
[6.  3.  7.]
```

```
df.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   float64
12  thal        303 non-null   float64
13  target      303 non-null   int64
dtypes: float64(3), int64(11)
memory usage: 33.3 KB
```

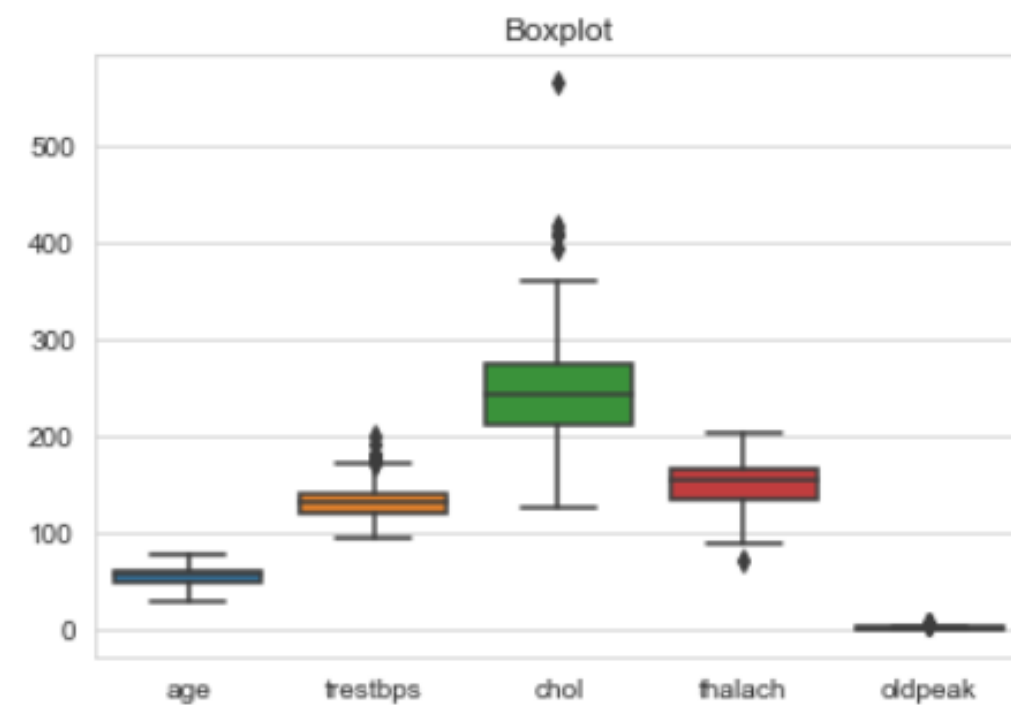


# EDA

```
df[numerical_data].describe().T
```

	count	mean	std	min	25%	50%	75%	max
age	303.0	54.438944	9.038662	29.0	48.0	56.0	61.0	77.0
trestbps	303.0	131.689769	17.599748	94.0	120.0	130.0	140.0	200.0
chol	303.0	246.693069	51.776918	126.0	211.0	241.0	275.0	564.0
thalach	303.0	149.607261	22.875003	71.0	133.5	153.0	166.0	202.0
oldpeak	303.0	1.039604	1.161075	0.0	0.0	0.8	1.6	6.2

```
sns.boxplot(df[numerical_data])  
plt.title('Boxplot')  
plt.show()
```



```
# the distributaion of Target variable.  
sns.set_style('whitegrid')  
sns.countplot(x='target',data=df,palette='PuRd')  
plt.title('Target Distribution');
```

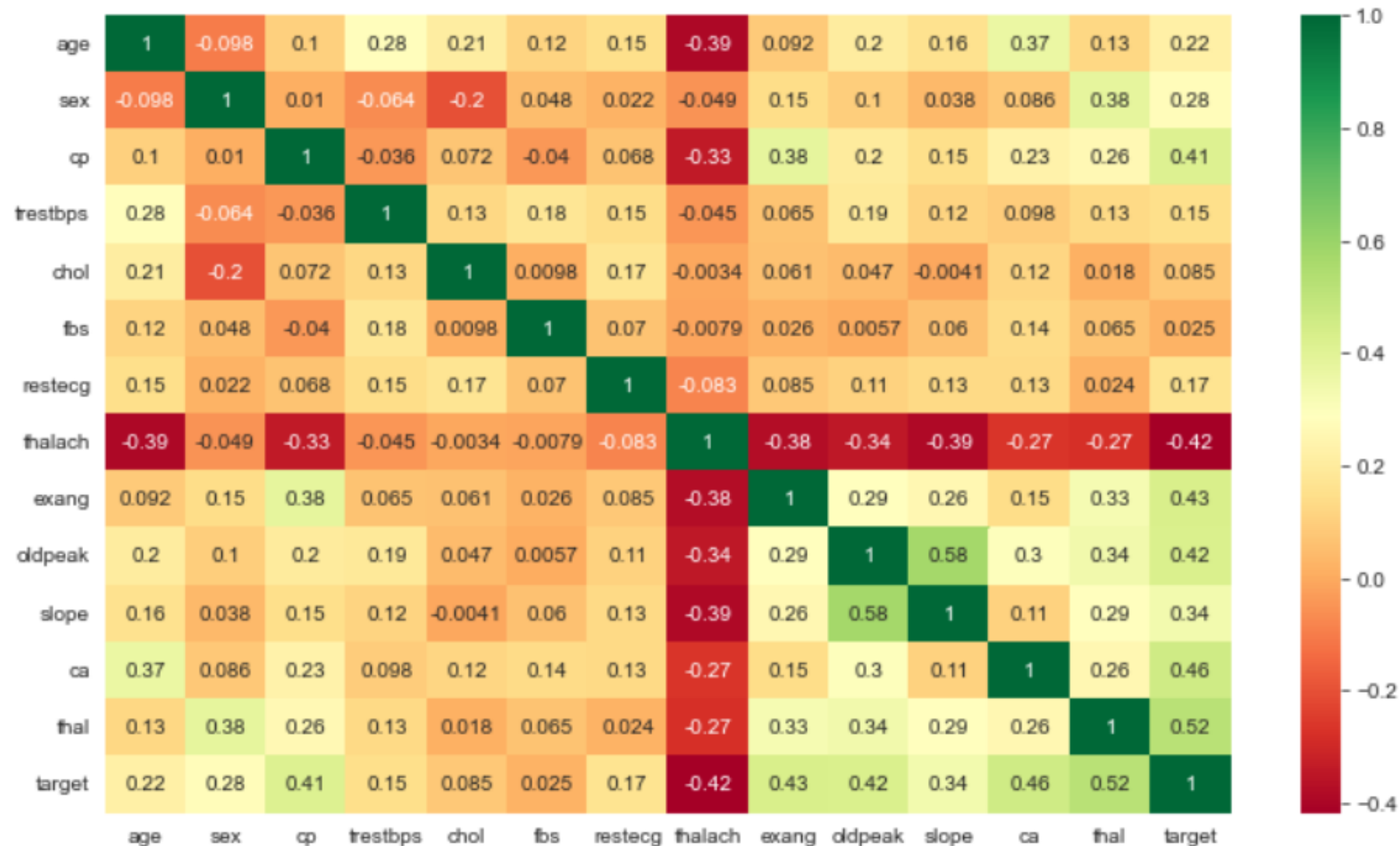


```
# Calculate and print the percentage of people with and without Heart Disease  
print(f"The percentage of people with Heart Disease: {df['target'].value_counts(normalize=True)[1] * 100:.2f}%")  
print(f"The percentage of people without Heart Disease: {df['target'].value_counts(normalize=True)[0] * 100:.2f}%")
```

The percentage of people with Heart Disease: 45.87%  
The percentage of people without Heart Disease: 54.13%

# EDA

```
corrmat = df.corr()
top_corr_features = corrmat.index
plt.figure(figsize=(12,7))
g=sns.heatmap(df[top_corr_features].corr(),annot=True,cmap="RdYlGn")
```

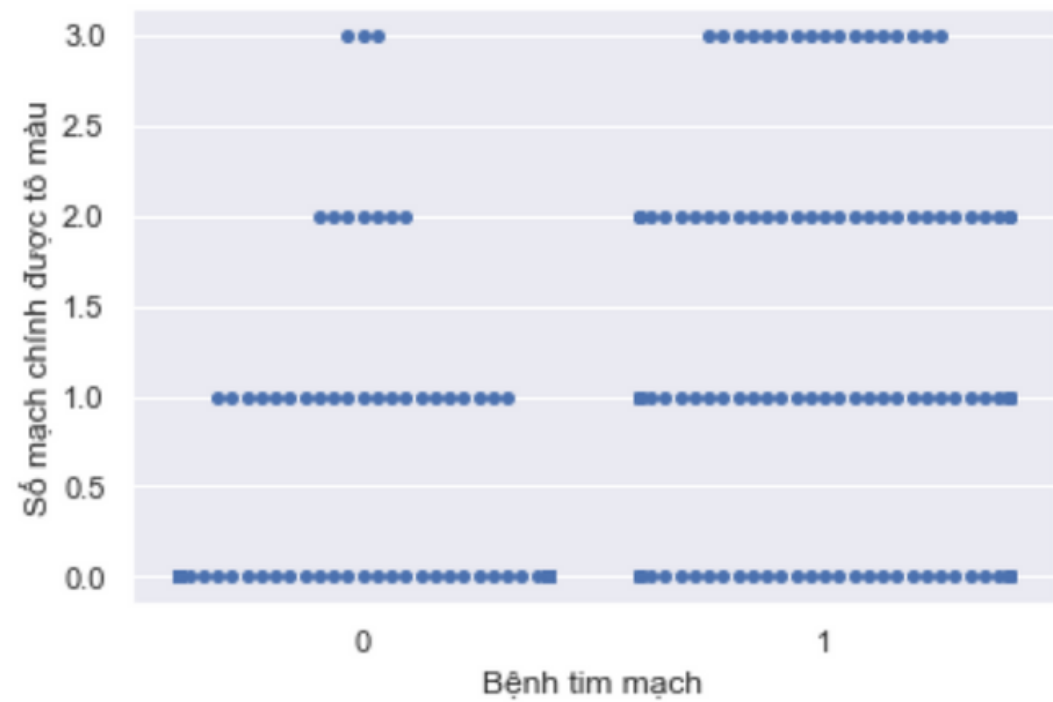


- Tương quan giữa biến **mục tiêu** và biến **độc lập** cao hơn 0.4:
  - thal
  - ca
  - exang
  - oldpeak
  - cp
- Tương quan giữa các biến **độc lập** cao hơn 0.4:
  - oldpeak - slope: chỉ số trong điện tâm đồ (ECG)

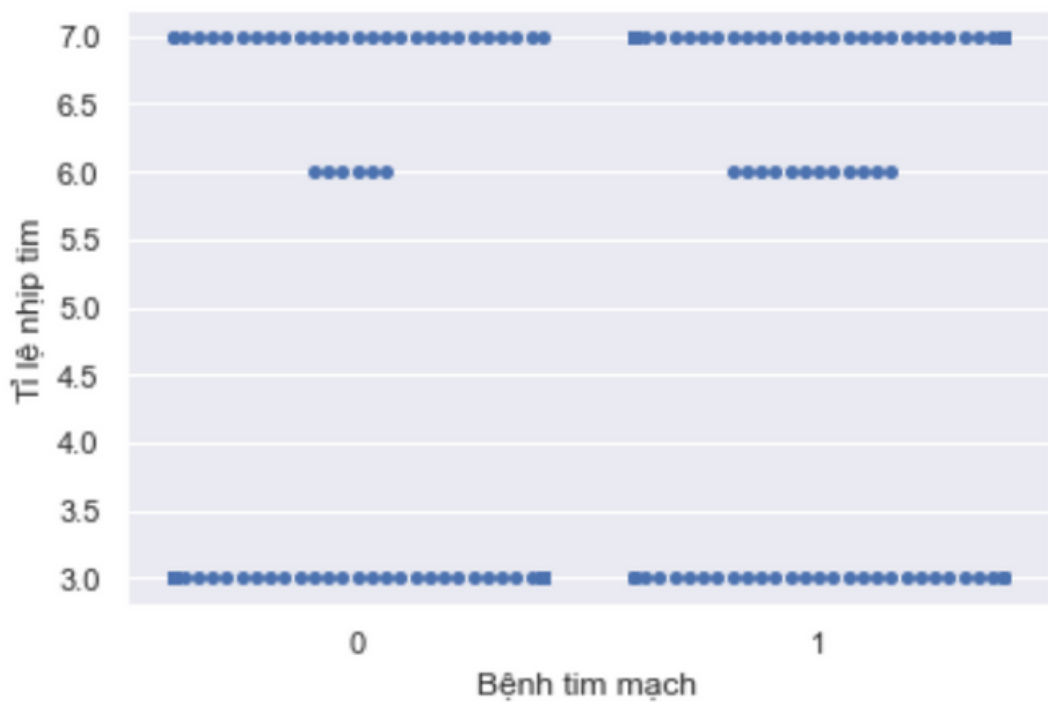


# EDA

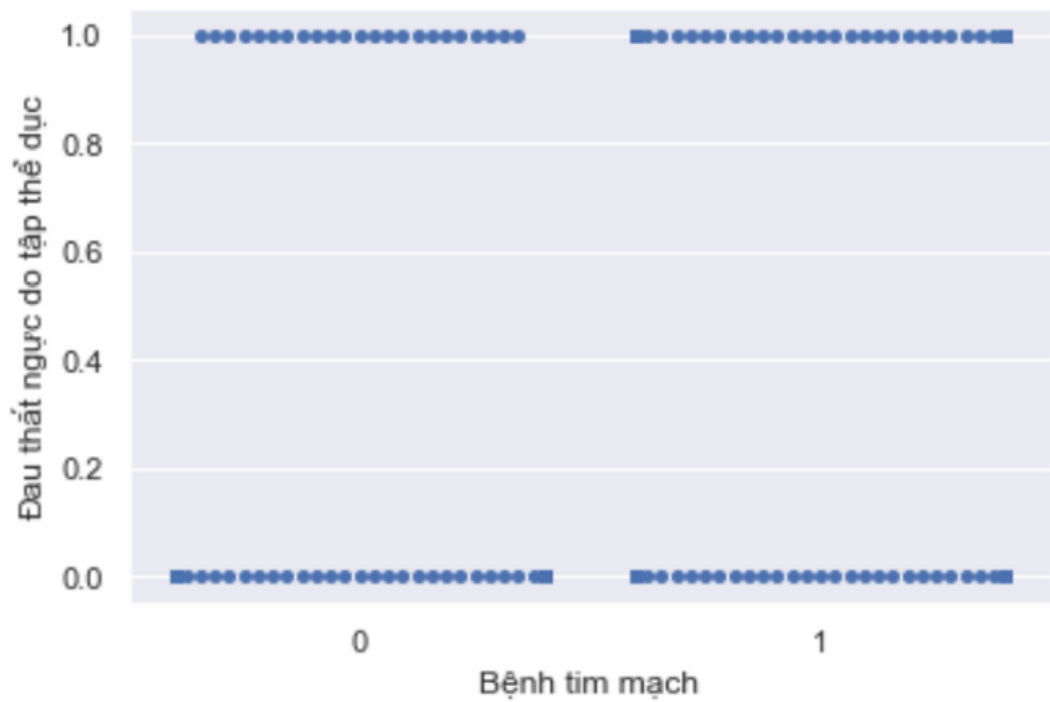
thal



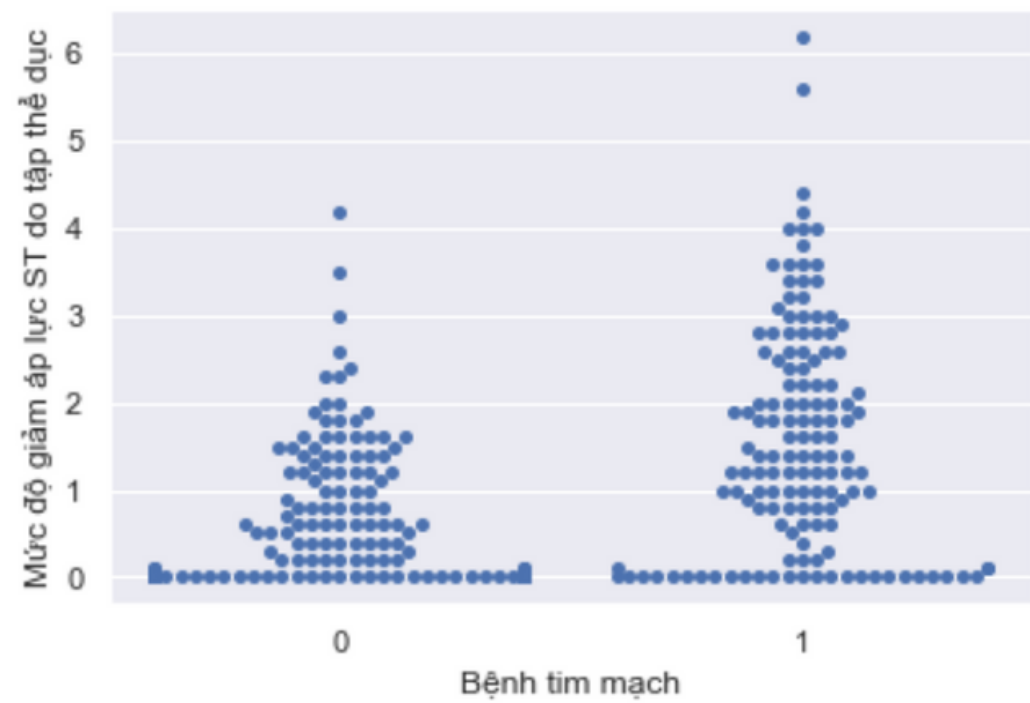
ca



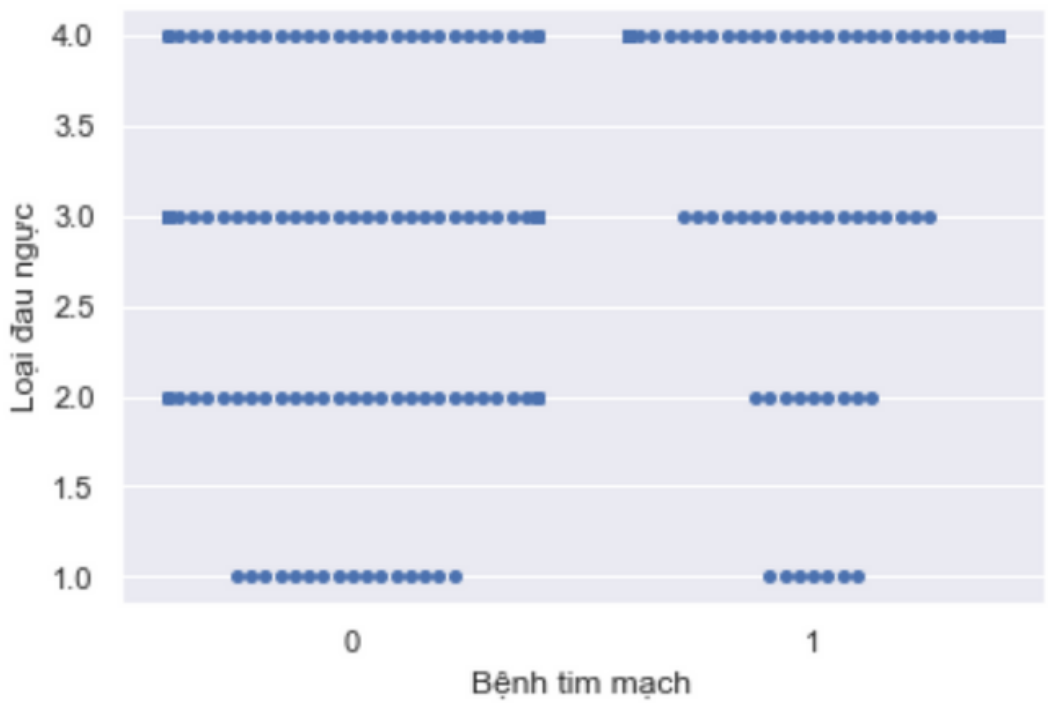
exang



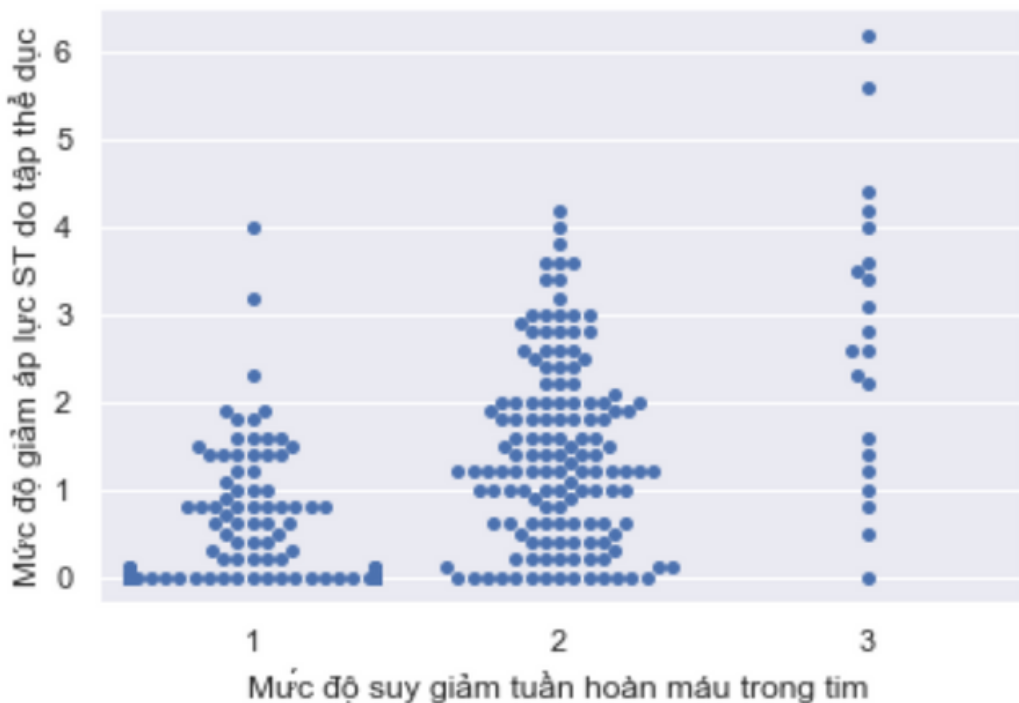
oldpeak



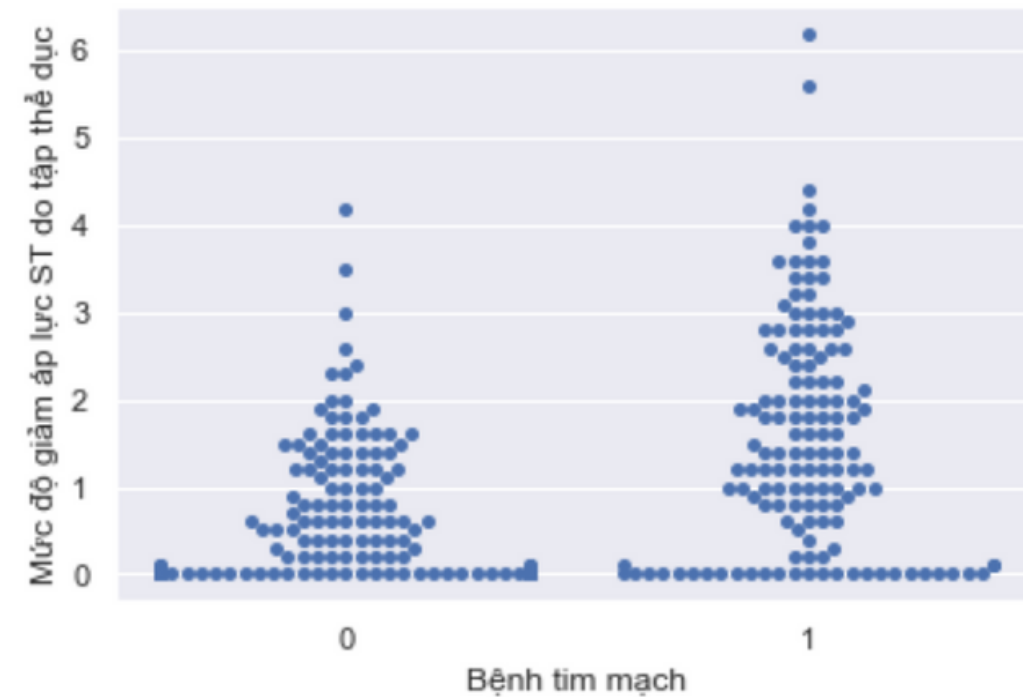
cp



oldpeak - slope

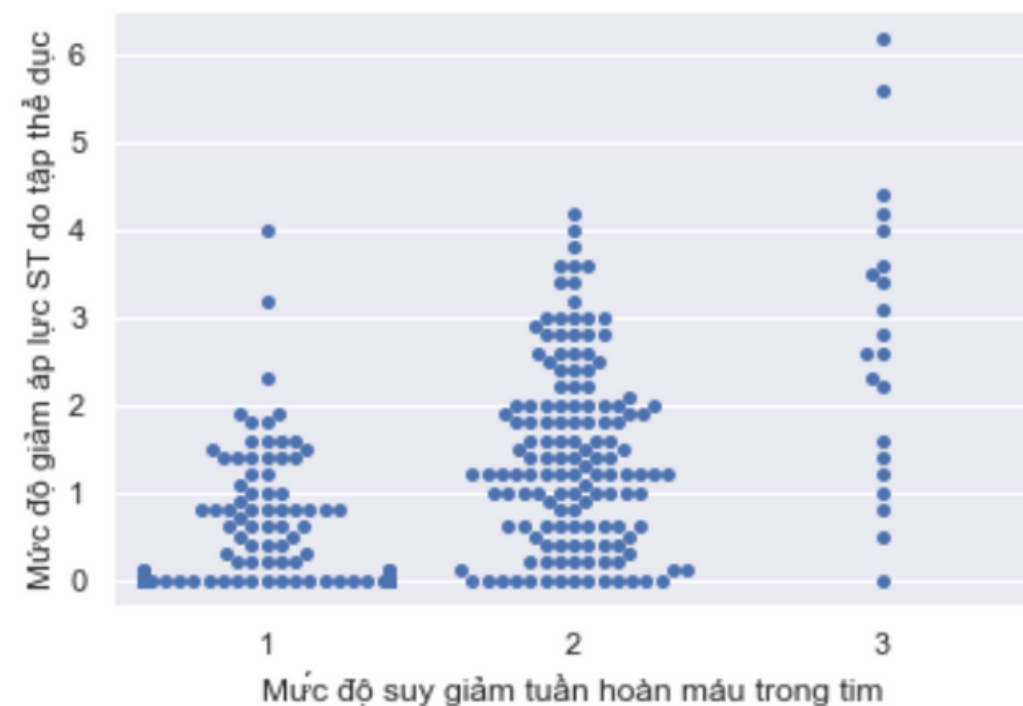


oldpeak

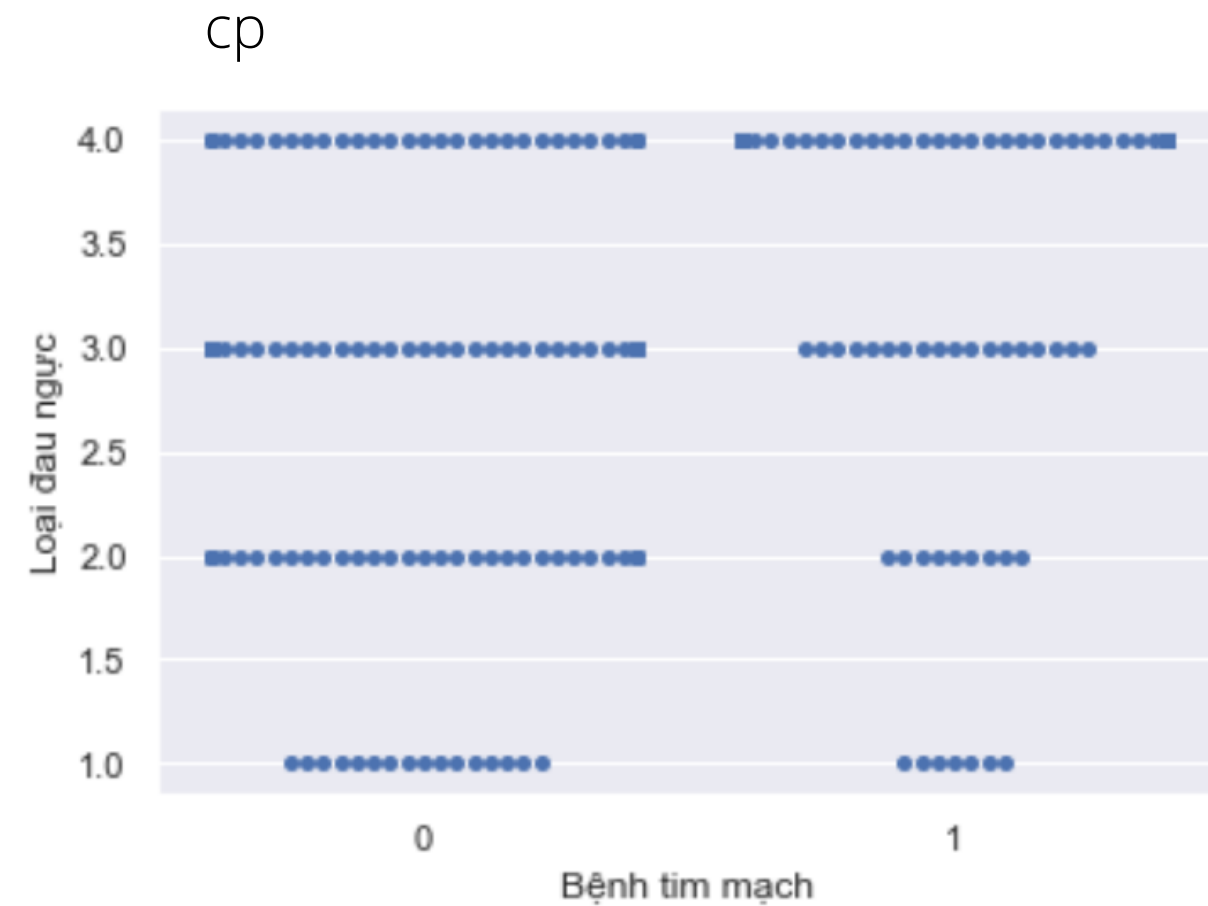
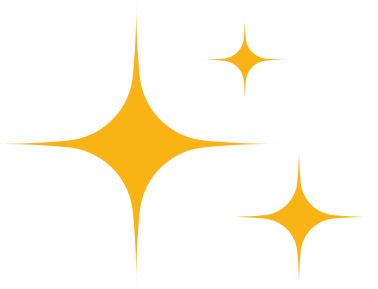


- Oldpeak đánh giá phản ứng của tim mạch trong quá trình tăng cường vận động.
- Sự tăng cao của oldpeak có thể liên quan đến sự suy giảm của tuần hoàn máu trong tim, và thường được coi là một dấu hiệu bất thường trong quá trình cung cấp oxy cho tim mạch, có thể liên quan đến bệnh tim mạch.

oldpeak - slope



- Slope đề cập đến độ dốc của đoạn đỉnh của đường ST trên điện tâm đồ (ECG) sau quá trình tập thể dục
  - **Giá trị 1: Dốc lên (upsloping):** phản ứng bình thường
  - **Giá trị 2: Phẳng (flat):** phản ứng không bình thường của tim mạch, có thể liên quan đến sự suy giảm tuần hoàn máu trong tim hoặc các vấn đề khác về sức khỏe tim mạch
  - **Giá trị 3: Dốc xuống (downsloping):** dấu hiệu không bình thường và có thể liên quan đến các vấn đề nghiêm trọng về sức khỏe tim mạch, như cung cấp oxy kém cho cơ tim, bất thường về các mạch của tim, hoặc các vấn đề về điện thế tim



- Cp đề cập đến loại đau ngực mà bệnh nhân ghi nhận
  - **Giá trị 1: Đau ngực điển hình (typical angina)** - Đây là loại đau ngực được xem là đặc trưng của bệnh tim mạch.
  - **Giá trị 2: Đau ngực không điển hình (atypical angina)** - Đau ngực không theo mô típ của đau ngực điển hình, có thể gây nhầm lẫn trong việc chẩn đoán bệnh tim mạch.
  - **Giá trị 3: Đau ngực không phải do cơ tim (non-anginal pain)** - Loại đau ngực không liên quan đến sự co thắt của động mạch cơ tim, có thể là do các vấn đề khác
  - **Giá trị 4: Không có triệu chứng (asymptomatic)** - Bệnh nhân không ghi nhận bất kỳ triệu chứng nào liên quan đến đau ngực hoặc vấn đề tim mạch.

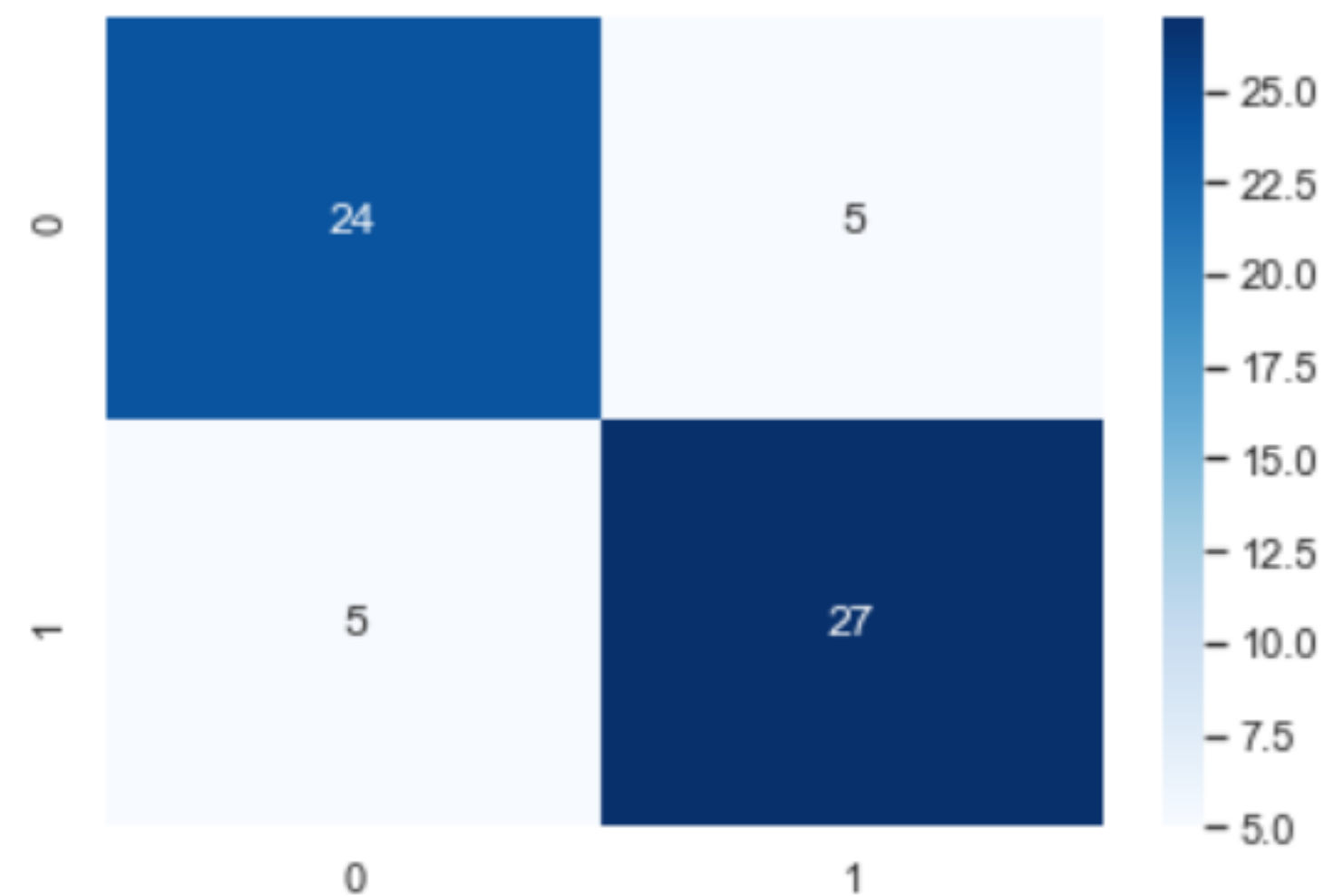
# XÂY DỰNG MÔ HÌNH

---

1. Logistic Regression
2. SVC
3. GaussianNB
4. Random Forest
5. Decision Tree
6. XGBoost
7. Extreme Gradient Boost

# XÂY DỰNG MÔ HÌNH

## Logistic Regression



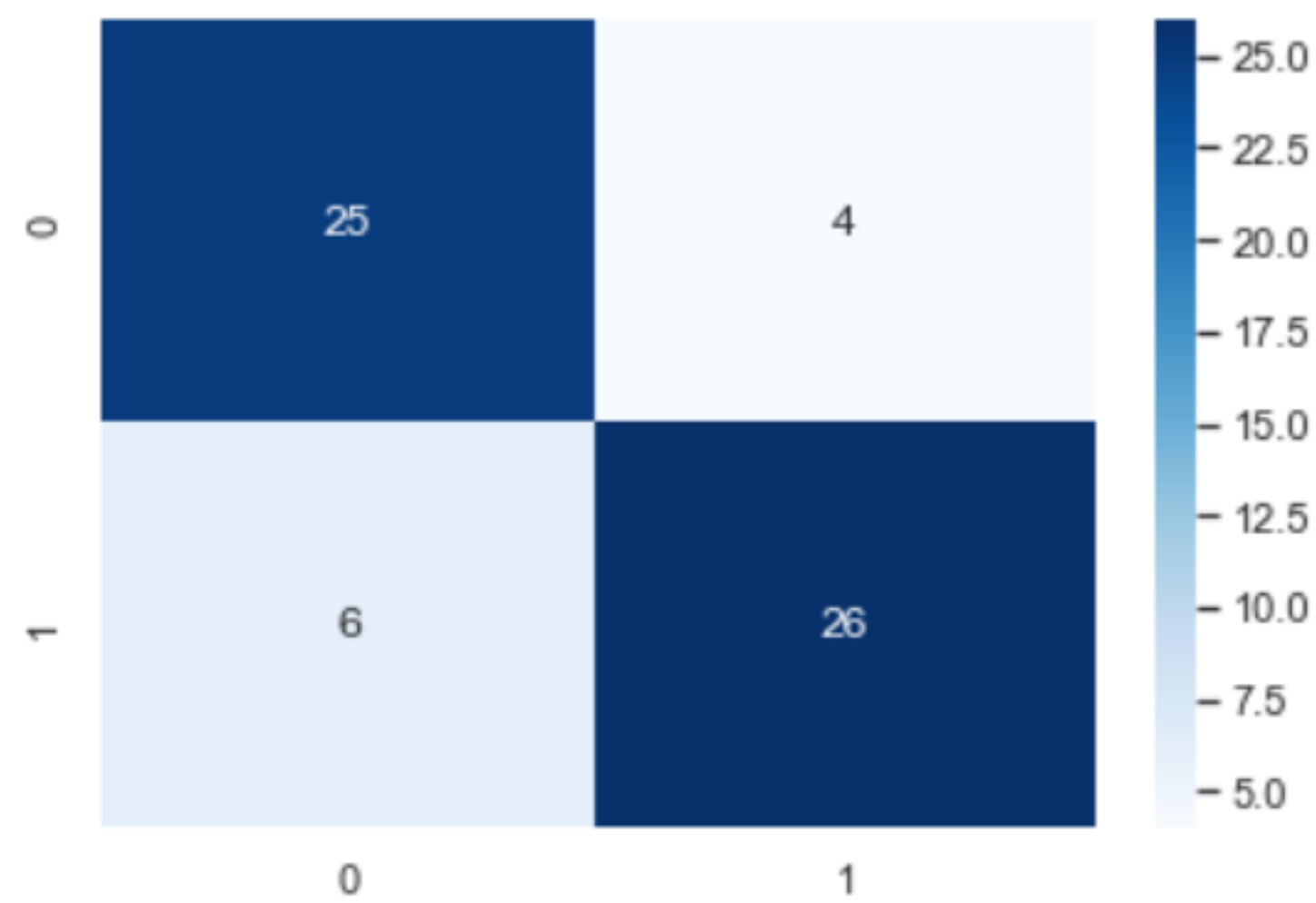
Accuracy of Logistic Regression: 83.60655737704919

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

ROC AUC: 0.9191810344827586

# XÂY DỰNG MÔ HÌNH

SVC



Accuracy of SVC: 83.60655737704919

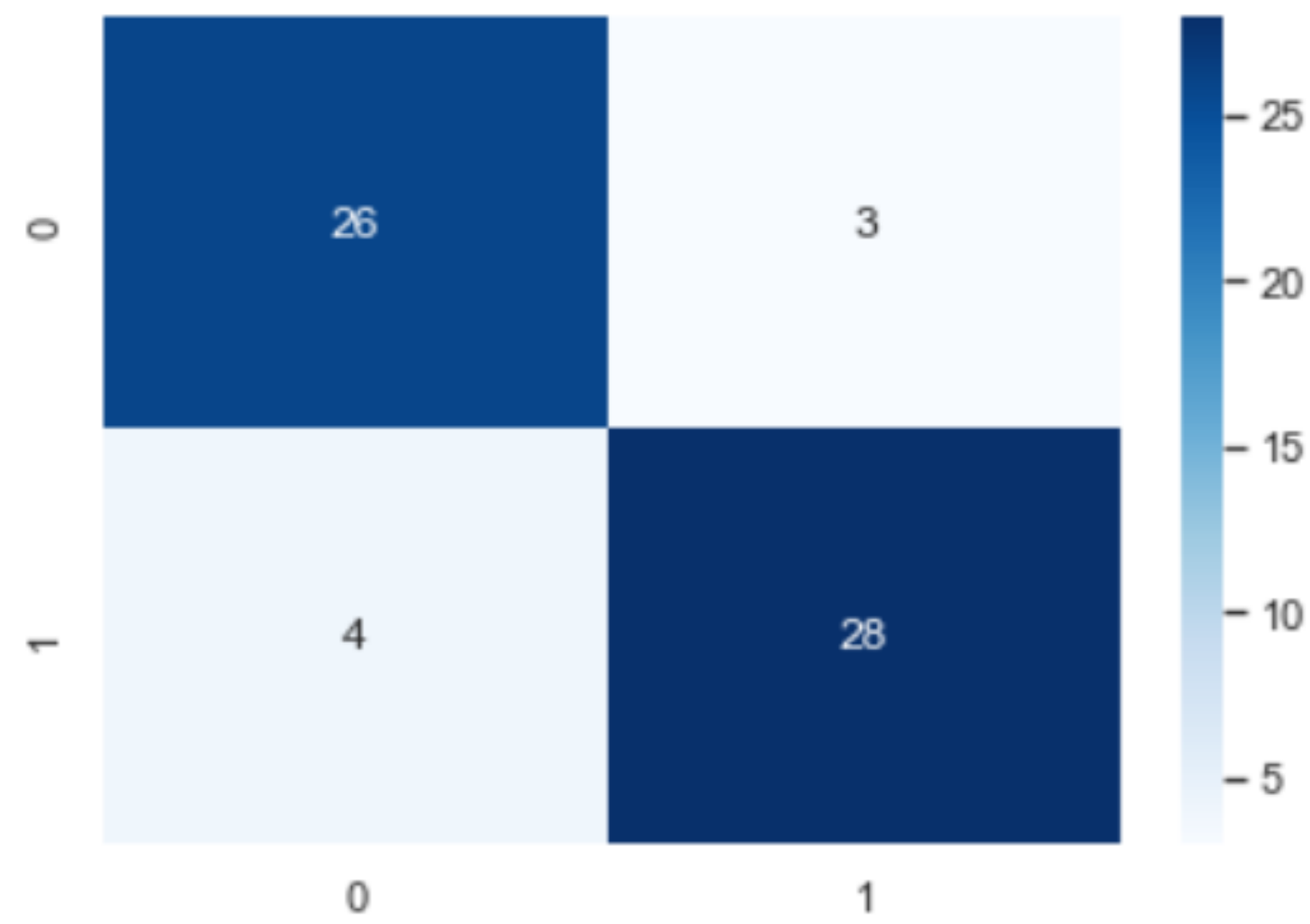
	precision	recall	f1-score	support
0	0.81	0.86	0.83	29
1	0.87	0.81	0.84	32
accuracy			0.84	61
macro avg	0.84	0.84	0.84	61
weighted avg	0.84	0.84	0.84	61

ROC AUC: 0.9191810344827585



# XÂY DỰNG MÔ HÌNH

## GaussianNB



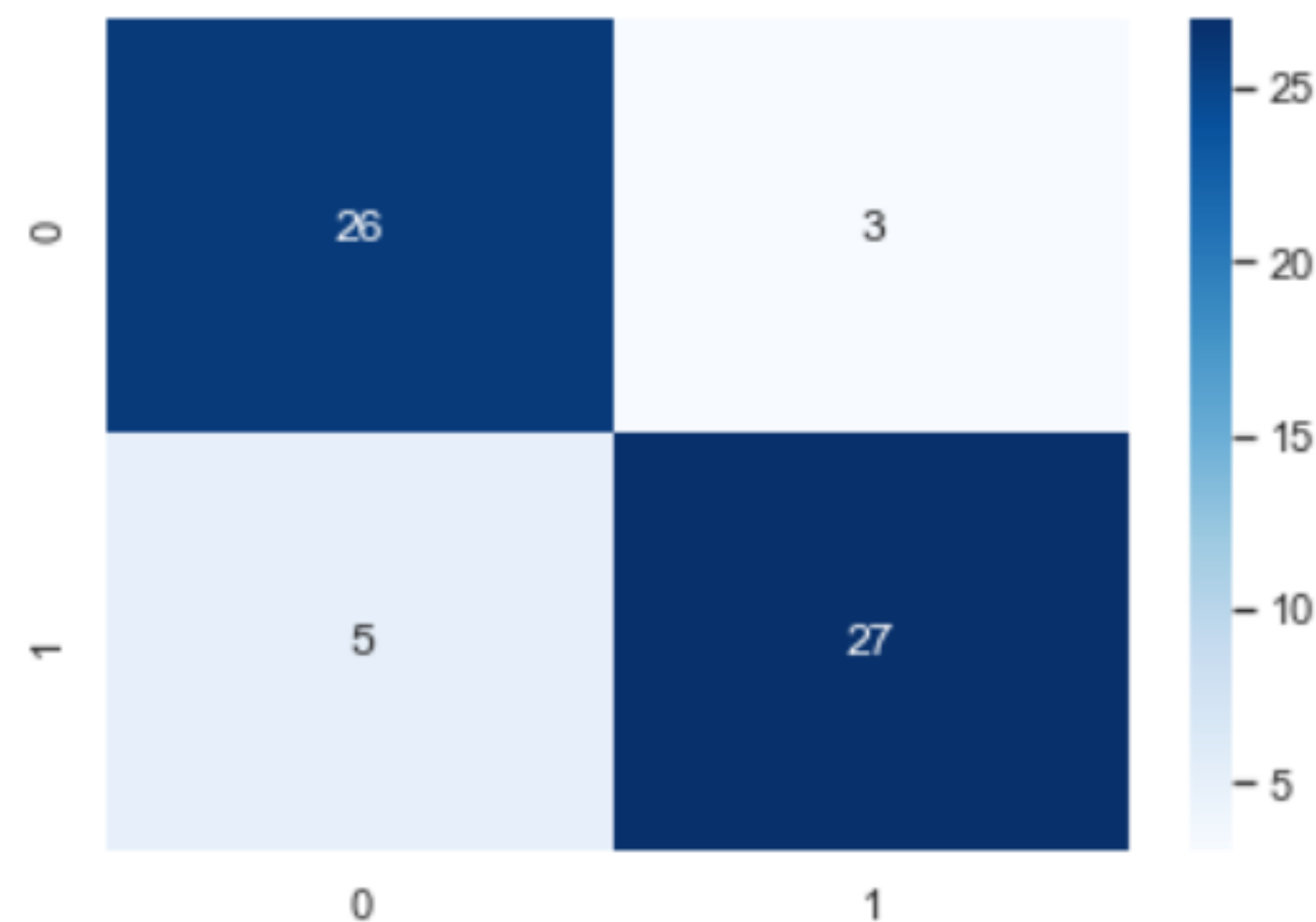
Accuracy of GaussianNB: 88.52459016393442

	precision	recall	f1-score	support
0	0.87	0.90	0.88	29
1	0.90	0.88	0.89	32
accuracy			0.89	61
macro avg	0.88	0.89	0.89	61
weighted avg	0.89	0.89	0.89	61

ROC AUC: 0.9137931034482758

# XÂY DỰNG MÔ HÌNH

## Random Forest



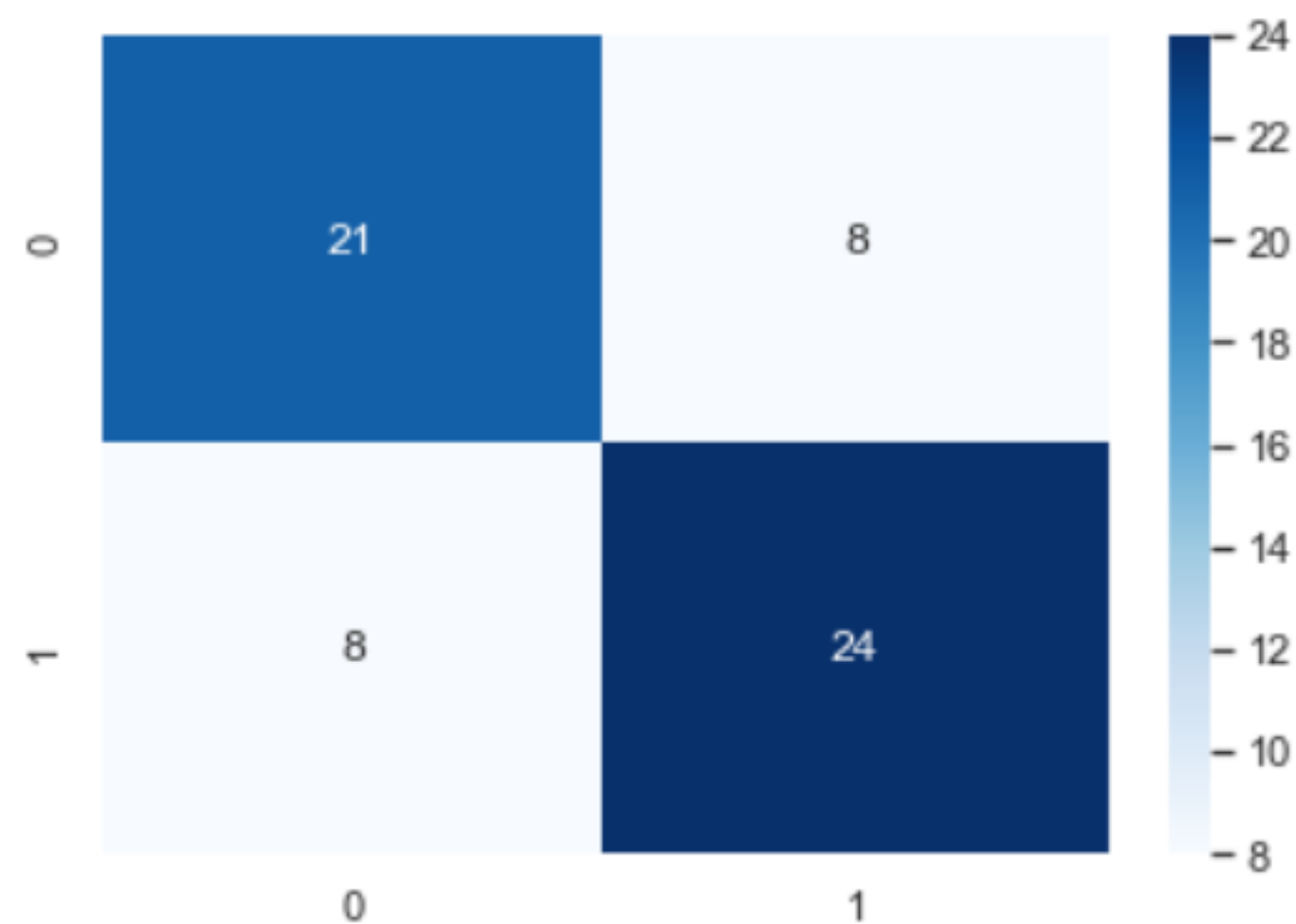
Accuracy of Random Forest: 86.88524590163934

	precision	recall	f1-score	support
0	0.84	0.90	0.87	29
1	0.90	0.84	0.87	32
accuracy			0.87	61
macro avg	0.87	0.87	0.87	61
weighted avg	0.87	0.87	0.87	61

ROC AUC: 0.9450431034482758

# XÂY DỰNG MÔ HÌNH

## Decision Tree



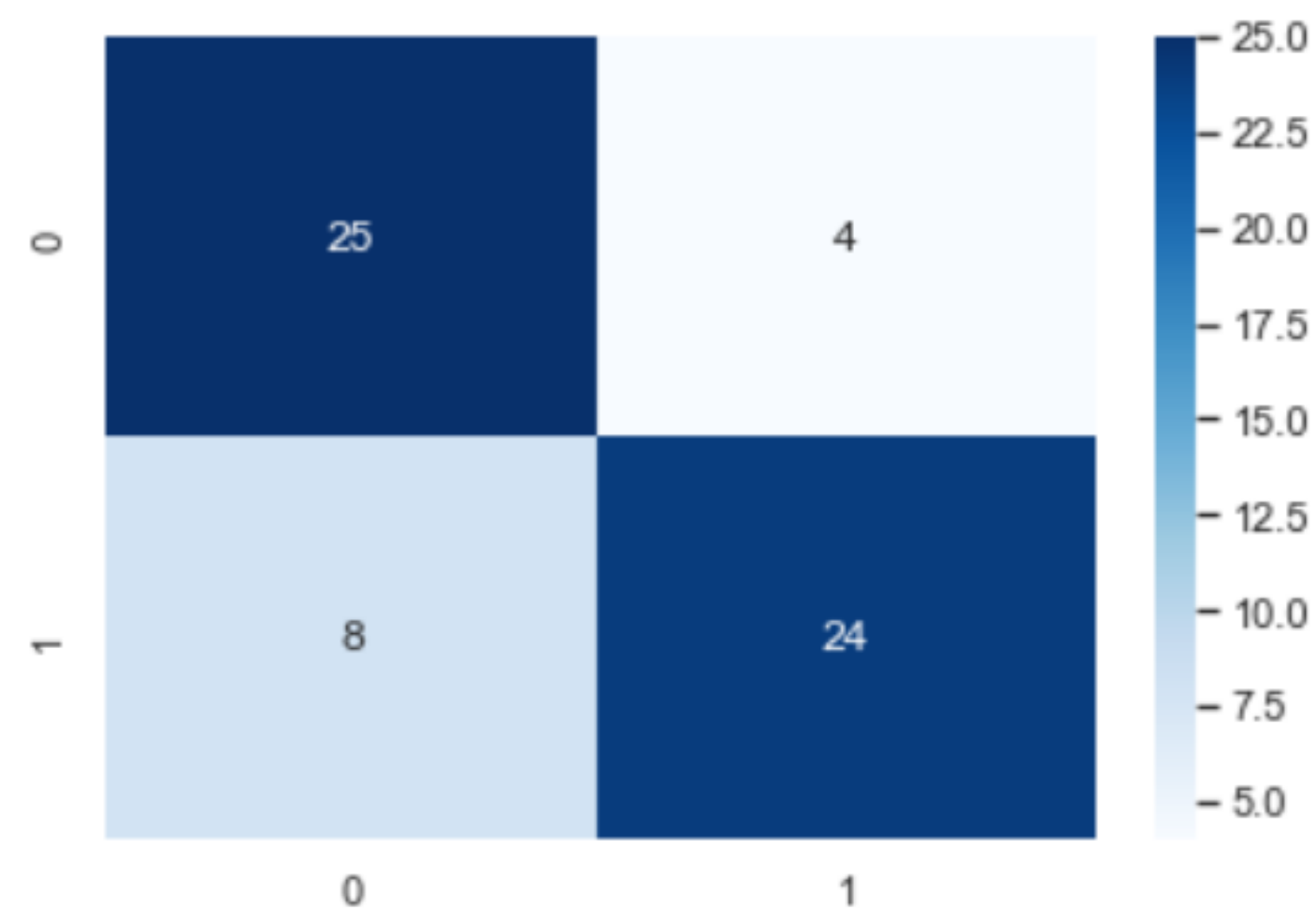
Accuracy of Decision Tree: 73.77049180327869

	precision	recall	f1-score	support
0	0.72	0.72	0.72	29
1	0.75	0.75	0.75	32
accuracy			0.74	61
macro avg	0.74	0.74	0.74	61
weighted avg	0.74	0.74	0.74	61

ROC AUC: 0.7516163793103448

# XÂY DỰNG MÔ HÌNH

## XGBoost



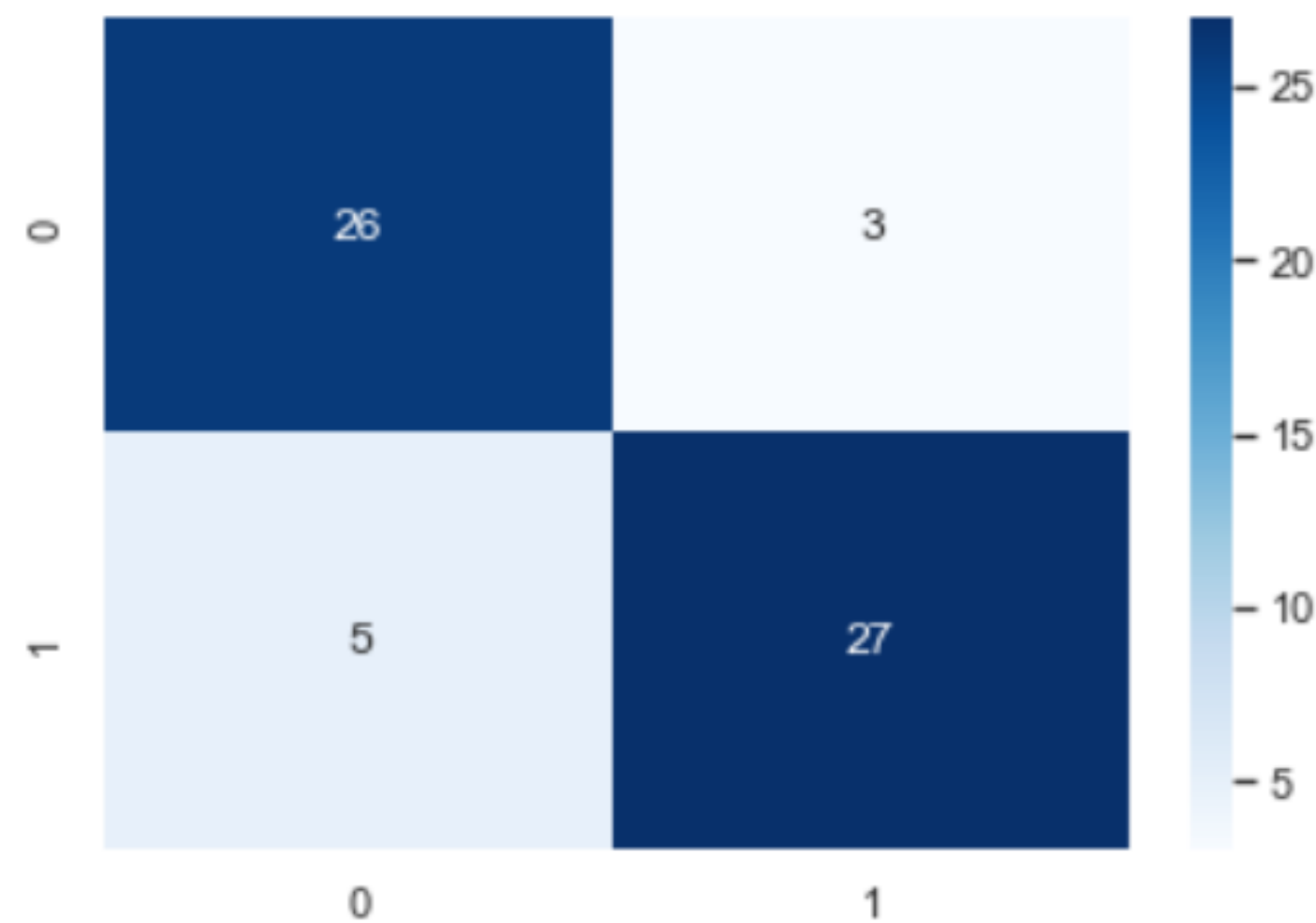
Accuracy of XGBoost: 80.32786885245902

	precision	recall	f1-score	support
0	0.76	0.86	0.81	29
1	0.86	0.75	0.80	32
accuracy			0.80	61
macro avg	0.81	0.81	0.80	61
weighted avg	0.81	0.80	0.80	61

ROC AUC: 0.875

# XÂY DỰNG MÔ HÌNH

## Extreme Gradient Boost

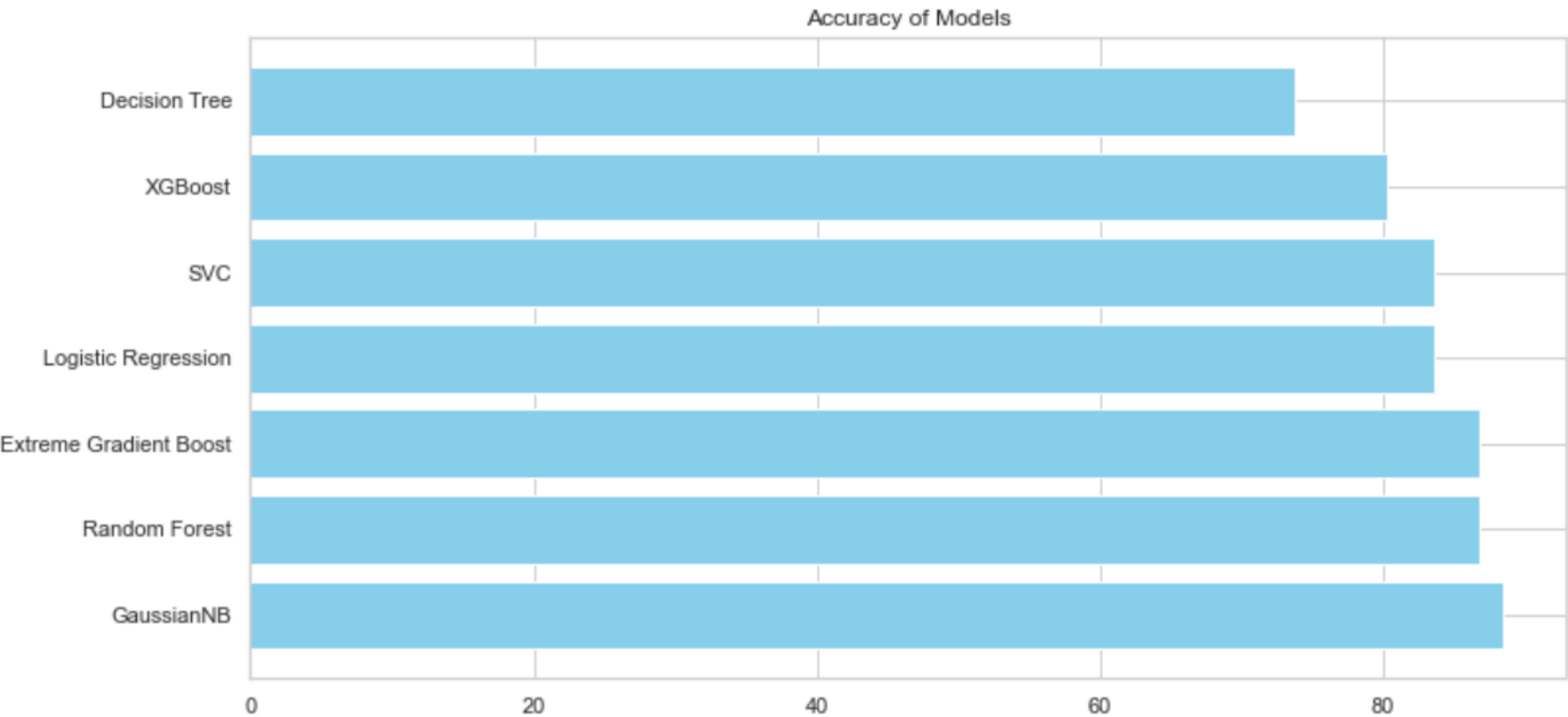


Accuracy of Extreme Gradient Boost: 86.88524590163934

	precision	recall	f1-score	support
0	0.84	0.90	0.87	29
1	0.90	0.84	0.87	32
accuracy			0.87	61
macro avg	0.87	0.87	0.87	61
weighted avg	0.87	0.87	0.87	61

ROC AUC: 0.9170258620689655

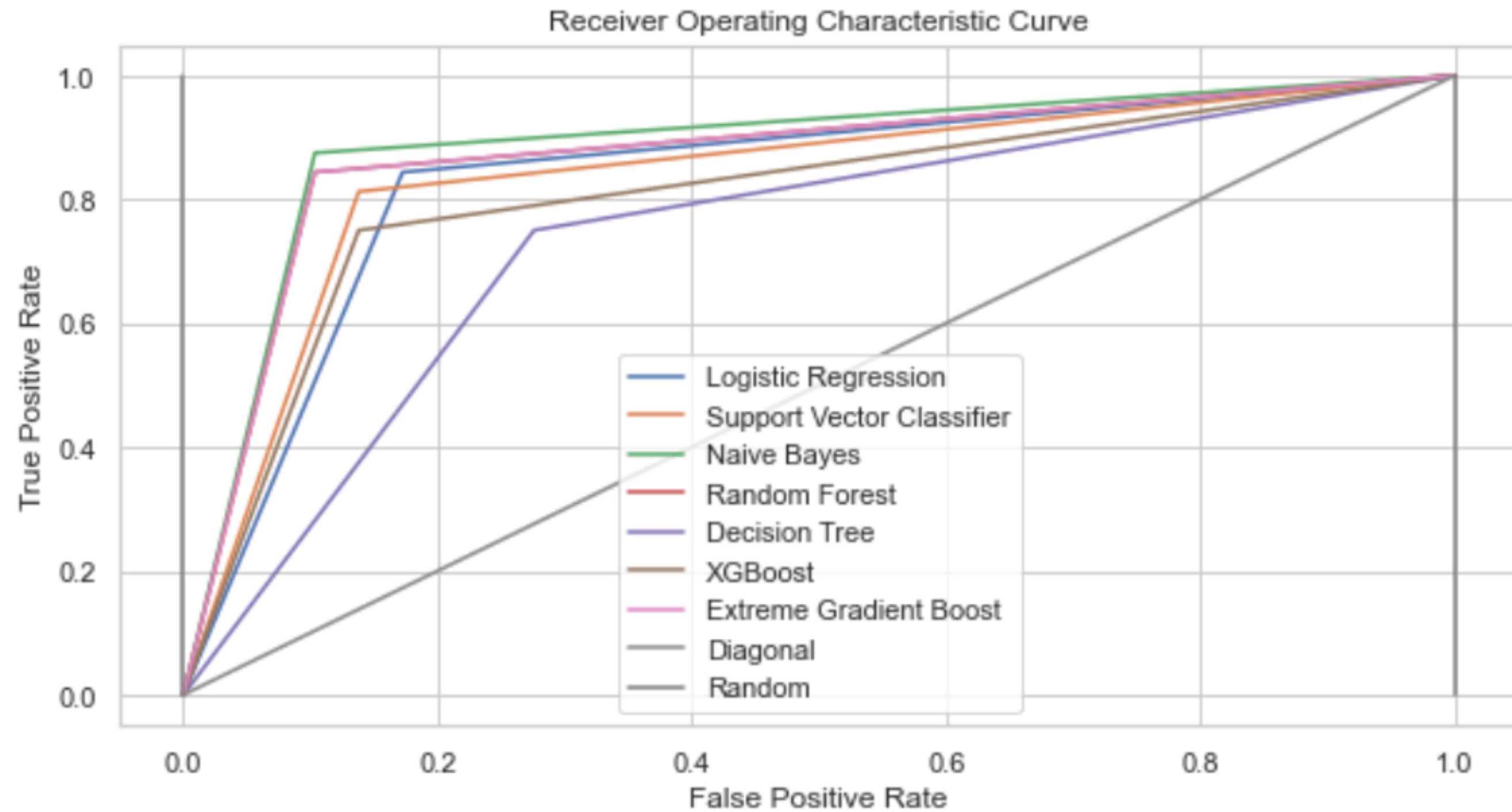
# ĐÁNH GIÁ MÔ HÌNH



	Models	Accuracy
2	GaussianNB	88.524590
3	Random Forest	86.885246
6	Extreme Gradient Boost	86.885246
0	Logistic Regression	83.606557
1	SVC	83.606557
5	XGBoost	80.327869
4	Decision Tree	73.770492



# ĐÁNH GIÁ MÔ HÌNH



# KẾT LUẬN

---

Xác suất bệnh mạch vành bắt nguồn từ các hàm phân biệt là đáng tin cậy và hữu ích về mặt lâm sàng khi áp dụng cho những bệnh nhân mắc hội chứng đau ngực và tỷ lệ mắc bệnh ở mức độ trung bình.

# TÀI LIỆU THAM KHẢO



**Bệnh tim mạch (CVD) ở Việt Nam**

Bệnh tim mạch (CVD)

 who.int



**Heart Disease Dataset**

Public Health Dataset

 kaggle.com



**UCI Machine Learning Repository**

Discover datasets around the world!

 ics.uci.edu

# PHỤ LỤC

No.	Column name	Description	Value
1	age	Age	[29:77]
2	sex	Sex	(1 = male; 0 = female)
3	cp	Chest pain type	4 values -- Value 1: typical angina -- Value 2: atypical angina -- Value 3: non-anginal pain -- Value 4: asymptomatic
4	trestbps	Resting blood pressure	in mm Hg
5	chol	Serum cholesterol	in mg/dl
6	fbs	Fasting blood sugar	> 120 mg/dl (0 - normal/false; 1 - abnormal/true)
7	restecg	Resting electrocardiographic results	3 values -- Value 0: normal -- Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV) -- Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria
8	thalach	Maximum heart rate achieved	in bpm
9	exang	Exerciseinduced angina	Yes/No
10	oldpeak	ST depression induced by exercise relative to rest	
11	slope	Slope of the peak exercise ST segment	3 values -- Value 1: upsloping -- Value 2: flat -- Value 3: downsloping
12	ca	Number of major vessels colored by fluoroscopy	
13	thal	Heart rate	4 values
14	target	Heart disease in the patient	Yes/No



**THANK YOU**

