FPT UNIVERSITY

# Final Report in Course AIL303m

## Heart Disease Prediction Using Machine Learning

FPT University
Ho Chi Minh city

June 21, 2025

***Lecturer:***

MsC Luu Giang Nam

***Student:***

| | | |
|---|---|---|
| Nguyen Thanh Phat | : | SE192617 |
| Trinh Vy Kiet | : | SE192636 |
| Pham Hoang Phuc | : | SE192874 |
| Nguyen Xuan Trung | : | SE193716 |
| Phan Anh Minh | : | SE193522 |

# Contents

# Abstract

This study investigates the application of machine learning techniques for the early prediction of heart disease, a leading cause of mortality worldwide. Using a dataset of 918 patients with 12 clinical attributes, we conducted thorough data preprocessing and exploratory analysis to identify relevant predictors such as *Oldpeak*, *MaxHR*, and *Age*. Three supervised models **Logistic Regression**, **Random Forest**, and **XGBoost** were trained and evaluated using four key performance metrics: **Accuracy**, **F1 Score**, **Recall**, and **ROC-AUC**.

While all models demonstrated competitive performance, **Random Forest** emerged as the most effective, achieving the highest scores across nearly all metrics: 88.0% accuracy, 89.9% F1 score, 89.0% recall, and an AUC of 0.8782. These results indicate not only strong overall performance but also the models superior ability to minimize false negatives a critical factor in clinical contexts where missed diagnoses carry serious risks.

Our findings suggest that Random Forest is the most robust and reliable model for heart disease prediction among those tested. This research highlights the potential of data-driven approaches to assist in early diagnosis and supports the integration of machine learning into clinical decision-making processes to improve patient outcomes.

# 1 Introduction

Cardiovascular diseases (CVDs) is the leading cause of death globally, responsible for millions of fatalities each year and significant socio-economic burden. Early detection is essential for prompt intervention and reducing mortality. The rise of the Fourth Industrial Revolution has driven significant advancements in Artificial Intelligence (AI) and Data Science, transforming healthcare by enabling the analysis of complex medical data and creating predictive models that often outperform traditional methods.

This report explores the application of Machine Learning in predicting the risk of cardiovascular disease. The focus is on comparing the performance of three models:

- Logistic Regression
- Random Fores
- XGBoost

By evaluating metrics such as Accuracy, Recall, Specificity, and Area Under the ROC Curve (AUC), this study aims to identify the most effective model for predicting the risk of cardiovascular disease, contributing valuable insights to improve early diagnosis and patient outcomes.

## 1.1 Motivation

Early detection of heart disease has been proven to significantly improve patient prognosis and survival rates through timely medical interventions. However, in many developing countries, including Vietnam, medical resources are often limited, including both medical personnel and diagnostic equipment. In this context, leveraging machine learning for automated analysis of clinical data promises to deliver highly feasible, efficient, and cost-effective decision support solutions.

Moreover, for us as students, this project not only represents an invaluable practical application of theoretical machine learning knowledge but also an opportunity to directly confront real-world healthcare challenges. Through this, we hope to contribute a small part to improving treatment outcomes, and potentially even saving lives.

## 1.2 Research Objeactive

The primary objectives of this research are:

- To develop and rigorously evaluate multiple machine learning models for predicting heart disease based on standard clinical health metrics.

- To systematically compare the predictive performance of these models using well-established evaluation metrics such as accuracy, F1-score, and ROC-AUC.
- To identify the most effective machine learning model capable of facilitating early detection and clinical decision-making for heart disease.

## 1.3 Member Task Responsibilities

| Member | Task |
|---|---|
| Phan Anh Minh | • Write the Introduction and research objectives<br>• Conclude and summarize the findings<br>• Compose the Abstract |
| Nguyen Thanh Phat | • Develop and evaluate models<br>• Build and tune the XGBoost model<br>• Build and evaluate Logistic Regression |
| Trinh Vy Kiet | • Compare the result of three models<br>• Evaluate based on performance metrics<br>• Select the best-performing model |
| Nguyen Xuan Trung | • Understand and preprocess data<br>• Handle missing values and encoding<br>• Prepare dataset for modeling |
| Pham Hoang Phuc | • Perform Exploratory Data Analysis (EDA)<br>• Visualize data by charts<br>• Extract insights from visualizations |

# 2 Data Understanding and Preprocessing

## 2.1 Data Understanding

The dataset was obtained from Kaggle: heart-failure-prediction.

With 918 samples and 12 attributes, including characteristics such as Age, Sex, ChestPainType, RestingBP, Cholesterol, FastingBS, RestingECG, MaxHR, ExerciseAngina, Oldpeak, ST_Slope, HeartDisease. (Figure 1) describes the summary statistics of the numeric variables.

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| Age | 918.0 | 53.510893 | 9.432617 | 28.0 | 47.00 | 54.0 | 60.0 | 77.0 |
| RestingBP | 918.0 | 132.396514 | 18.514154 | 0.0 | 120.00 | 130.0 | 140.0 | 200.0 |
| Cholesterol | 918.0 | 198.799564 | 109.384145 | 0.0 | 173.25 | 223.0 | 267.0 | 603.0 |
| FastingBS | 918.0 | 0.233115 | 0.423046 | 0.0 | 0.00 | 0.0 | 0.0 | 1.0 |
| MaxHR | 918.0 | 136.809368 | 25.460334 | 60.0 | 120.00 | 138.0 | 156.0 | 202.0 |
| Oldpeak | 918.0 | 0.887364 | 1.066570 | -2.6 | 0.00 | 0.6 | 1.5 | 6.2 |
| HeartDisease | 918.0 | 0.553377 | 0.497414 | 0.0 | 0.00 | 1.0 | 1.0 | 1.0 |

Figure 1: Dataset describe

| Tên | Ý nghĩa | Tác dụng |
|---|---|---|
| Age | Tuổi của bệnh nhân | Đánh giá nguy cơ tim mạch tăng theo độ tuổi |
| Sex | Giới tính (Nam/Nữ) | Xác định ảnh hưởng của giới tính đến bệnh tim |
| ChestPainType | Loại đau ngực (typical, atypical, non-anginal, asymptomatic) | Phân biệt các dạng đau ngực giúp chẩn đoán |
| RestingBP | Huyết áp lúc nghỉ (mm Hg) | Phát hiện huyết áp cao, yếu tố nguy cơ tim mạch |
| Cholesterol | Mức cholesterol huyết thanh (mg/dl) | Xác định tình trạng mỡ máu, ảnh hưởng đến bệnh tim |
| FastingBS | Đường huyết lúc đói > 120 mg/dl (1: có, 0: không) | Phát hiện tiền tiểu đường/tiểu đường – nguy cơ tim mạch |
| RestingECG | Kết quả điện tâm đồ khi nghỉ | Phát hiện bất thường điện tim ban đầu |
| MaxHR | Nhịp tim tối đa khi gắng sức | Đánh giá khả năng chịu đựng tim và chức năng tim |
| ExerciseAngina | Có đau thắt ngực khi gắng sức không (Y/N) | Đánh giá mức độ ảnh hưởng của tim khi hoạt động |
| Oldpeak | Mức độ trũng ST so với nghỉ (mm) | Chỉ số điện tim phản ánh thiếu máu cơ tim |
| ST_Slope | Độ dốc của đoạn ST sau khi vận động | Giúp chẩn đoán thiếu máu cơ tim (up, flat, down) |
| HeartDisease | Kết quả chẩn đoán bệnh tim (1: có, 0: không) | Mục tiêu để mô hình học và dự đoán |

Figure 2: Dataset overview

The table in (Figure 2) presents the variables used in the heart disease dataset, their meanings, and their corresponding effects or purposes in the context of heart disease prediction. These variables are essential for evaluating the risk factors associated with heart disease and its diagnosis.

### 2.2 Data Pre-processing

- Handling missing values: replacing zeroes in `RestingBP` and `Cholesterol` with mean.
- Check and handle duplicated values.
- Encoding categorical variables use One-hot-encoding (e.g., `ChestPainType`, `Sex`).

## 3 Exploratory Data

In this section, we will conduct an in-depth exploratory data analysis (EDA) to uncover key insights from the dataset. Using a variety of visual tools, such as heat maps, pair plots, and histograms,... we will systematically examine the relationships between variables, identify patterns, and detect potential outliers. This process will allow us to gain a deeper understanding of the underlying trends and distributions in the data, helping us to make informed decisions as we proceed with the analysis.

### 3.1 Bar plots

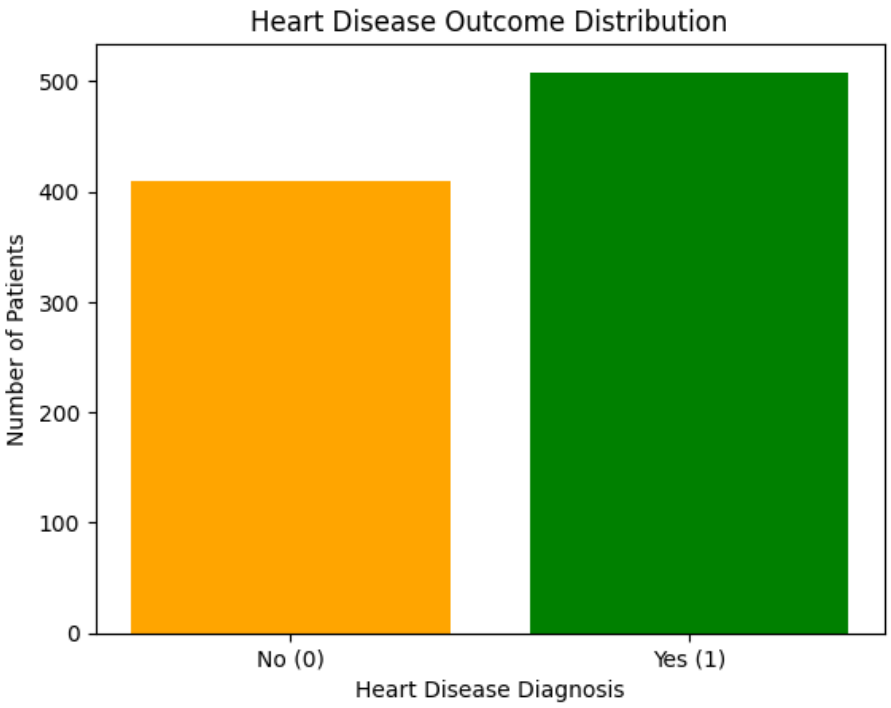


Figure 3: Heart disease diagnosis

From the chart in (Figure 3), we can draw the following conclusions:

- The dataset contains information on a total of 918 patients (410 without heart disease and 508 with heart disease).
- The number of patients with heart disease (508) exceeds the number of patients without heart disease (410).
- The dataset exhibits a slight class imbalance, with the 'Heart Disease' group being the majority. This is an important factor to consider when using this dataset for machine learning model development, as it may impact the model's performance.

## 3.2 Heatmap

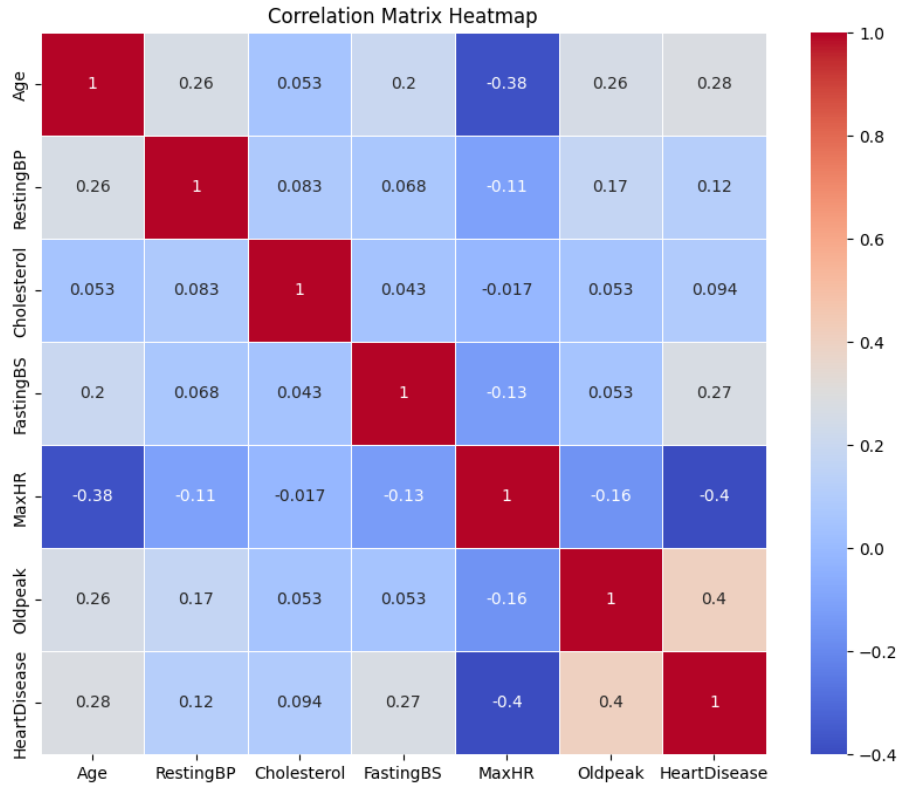


Figure 4: Correlations between features

From the heatmap (Figure 4), we can conclude:

### 3.2.1 Factors Strongly Positively Correlated with Heart Disease

- **Oldpeak** (Coefficient: 0.4):

  - Meaning: This shows the strongest positive correlation with **HeartDisease**. **Oldpeak** typically measures the ST segment depression on an ECG during exercise compared to rest.
  - Conclusion: Higher **Oldpeak** values are strong indicators of a higher likelihood of heart disease. This is a key predictor.
- **Age** (Coefficient: 0.28):
  - Meaning: A moderate positive correlation.
  - Conclusion: Higher fasting blood sugar levels are associated with increased heart disease risk.
- **FastingBS** (Coefficient: 0.27):
  - Meaning: A positive correlation.
  - Conclusion: Higher fasting blood sugar levels are associated with increased heart disease risk.

#### 3.2.2 Factors Strongly Negatively Correlated with Heart Disease

- **MaxHR** (Coefficient: -0.4):
  - Meaning: This shows the strongest negative correlation. **MaxHR** is the highest heart rate a person can achieve during intense exercise.
  - Conclusion: A lower **MaxHR** indicates a higher risk of heart disease. In other words, a healthier heart is capable of achieving a higher maximum heart rate during exertion. This is a critical predictor.

#### 3.2.3 Weakly Correlated Factors with Heart Disease

- **Cholesterol** (Coefficient: 0.094) and **RestingBP** (Coefficient: 0.12) :
  - Meaning: These coefficients are close to zero.
  - Conclusion: **Cholesterol** and **RestingBP** alone are not strong predictors of heart disease in this dataset. Their relationship with heart disease may be more complex and should be considered alongside other factors.

#### 3.2.4 Summary

- **Risk factors** (increase in these factors increases the likelihood of disease): Oldpeak (strongest), Age, and FastingBS.
- **Protective factor** (increase in this factor reduces the likelihood of disease): MaxHR (higher maximum heart rate is better).
- Factors such as Cholesterol and RestingBP seem to have less direct influence when considered in isolation.

## 3.3 Histogram

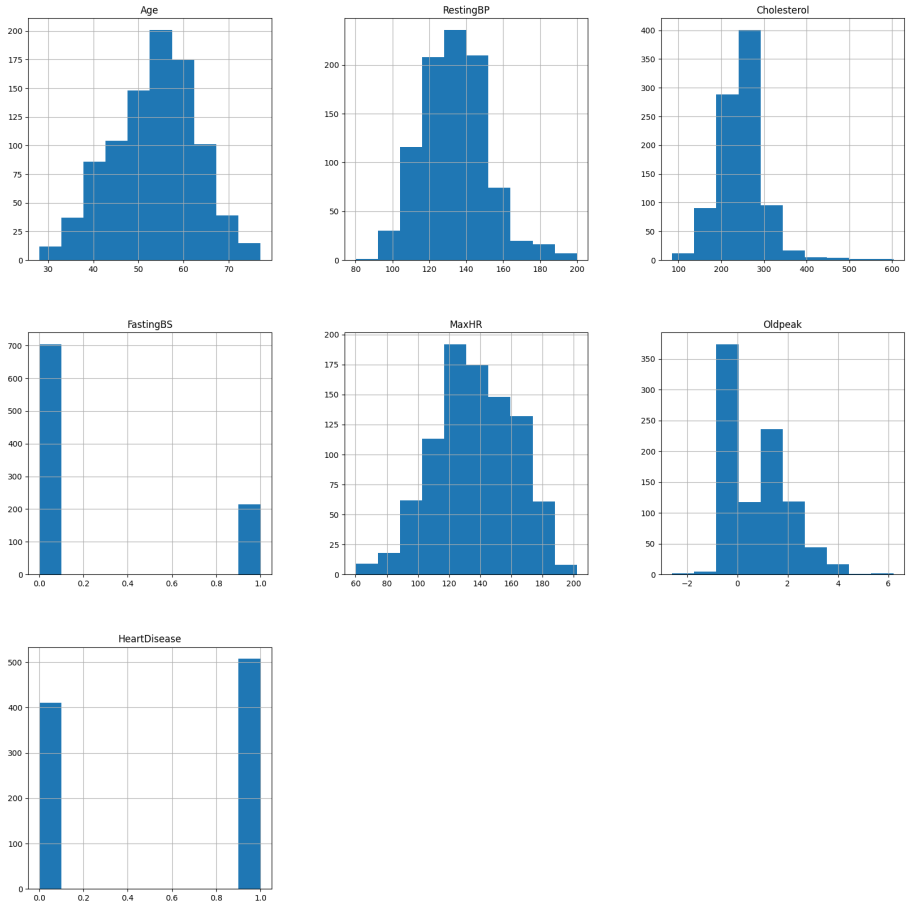


Figure 5: Histograms of each attribute

### 3.3.1 Analysis and Interpretation of Histogram Charts

The histograms 5 illustrate the distribution of each attribute in the dataset, helping us to gain a deeper understanding of their characteristics.

- **Age:**
  - **Distribution:** The histogram resembles a normal distribution (bell-shaped curve), with a fairly symmetrical shape.
  - **Interpretation:** The majority of patients in this dataset are middle-aged, with ages ranging from 47 to 63 years. The peak of the histogram is around 55-60 years.
- **RestingBP:**
  - **Distribution:** It follows a pattern similar to a normal distribution, with the peak around 120-140 mmHg.

  - **Interpretation:** Most patients have resting blood pressure within the normal to slightly elevated range.
- **Cholesterol:**
  - **Distribution:** The distribution is right-skewed.
  - **Interpretation:** Most patients have cholesterol levels between 180 to 300 mg/dl. The right-skew indicates a larger number of patients with low to moderate cholesterol levels, with a small group having very high cholesterol levels extending towards the right. The highest bar in the lowest value range suggests a significant portion of patients with lower cholesterol levels.
- **FastingBS:**
  - **Distribution:** This is a categorical variable coded as a number with only two values: 0 and 1.
  - **Interpretation:** A 0 likely represents normal blood sugar, while 1 indicates high blood sugar. The chart shows that normal blood sugar (0) is predominant in the dataset.
- **MaxHR:**
  - **Distribution:** The distribution resembles a bell-shaped curve but is slightly left-skewed. The peak lies between 130-160 beats per minute.
  - **Interpretation:** Most patients achieve a maximum heart rate in this range. The left skew suggests some cases have lower-than-average maximum heart rates.
- **Oldpeak:**
  - **Distribution:** The histogram is strongly right-skewed.
  - **Interpretation:** Most Oldpeak values are clustered near 0, indicating normal results. Only a few patients exhibit high Oldpeak values.
- **HeartDisease:**
  - **Distribution:** This is a binary variable, which is our target variable.
  - **Interpretation:** The chart shows the distribution of patients between the two groups: 0 (No disease) and 1 (Heart disease). The 1 group slightly outnumbers the 0 group, confirming the slight class imbalance in the data.

## 3.4 Pairplot

### 3.4.1 How to read pariplot

- **Diagonal cells:** (from top left to bottom right). These are KDE plots showing the distribution of each variable, colored by the target variable HeartDisease. They display the distribution of each attribute for the "Heart Disease" group (orange) and the "No Heart Disease" group (blue).

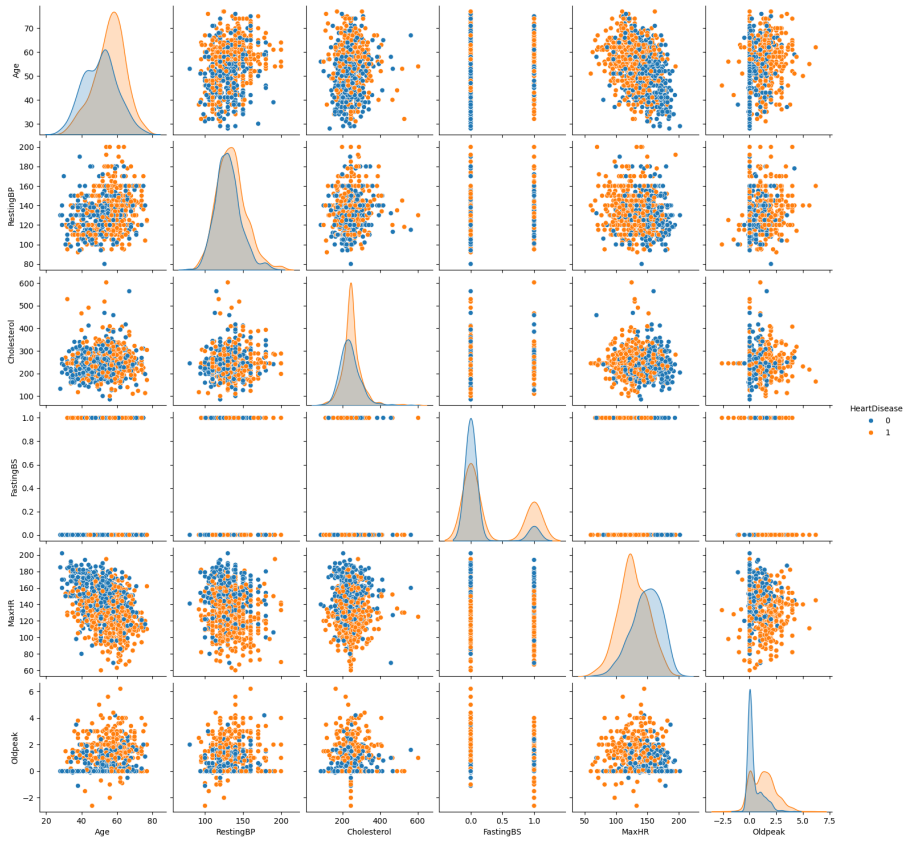


Figure 6: Pairplots

- **Other cells:** (off the diagonal). These are scatter plots, showing the relationships between two different variables. Each data point is colored according to HeartDisease.

The main objective of analyzing this pairplot is to identify which attributes or pairs of attributes best separate the "Heart Disease" and "No Heart Disease" groups.

### 3.4.2 Analysis on the Diagonal (KDE Distribution Plots)

This is where we assess the predictive power of **individual attributes**. The clearer the separation between the orange and blue curves, the better the attribute is as a predictor.

- **Oldpeak:**
  - **Distribution:** The two curves (orange and blue) are distinctly separated. The "No Disease" group (0 - blue) peaks at 0, while the "Heart Disease" group (1 - orange) peaks at a higher value.
  - **Interpretation:** This is a strong indicator of a very good predictor.

- **MaxHR(Maximum Heart Rate):**
  - **Distribution:** There is a clear separation. The peak of the "No Disease" group (blue) is at a higher heart rate compared to the "Heart Disease" group (orange).
  - **Interpretation:** MaxHR is a good predictor. A lower maximum heart rate is associated with a higher risk of heart disease.
- **Age:**
  - **Distribution:** Moderate separation. The peak for the orange group (Heart Disease) is slightly shifted to older ages compared to the blue group.
  - **Interpretation:** Age is a moderate predictor.
- **RestingBP (Resting Blood Pressure) and Cholesterol:**
  - **Distribution:** The two curves nearly overlap completely.
  - **Interpretation:** It is difficult to distinguish between the two groups based solely on these attributes. These are weak predictors when considered individually.

### 3.4.3 Analysis Off the Diagonal (Scatter Plots)

This is where we assess the predictive power when combining two attributes. We look for cells where the orange and blue points form distinct clusters.

- **Strongest Pair** (MaxHR and Oldpeak):
  - **Analysis:** In the cell between Oldpeak and MaxHR, there is a clear trend: orange points (1) tend to concentrate in the area with lower MaxHR and higher Oldpeak, while blue points (0) cluster in the area with higher MaxHR and lower Oldpeak.
  - **Interpretation:** This separation shows that combining these two variables is very effective for classification.
- **Fairly Good Pair** (Age and MaxHR):
  - **Analysis:** The cell between Age and MaxHR shows that orange points (Heart Disease) tend to concentrate in older age groups with lower MaxHR.
  - **Interpretation:** This combination also has value, reinforcing what we observed in the distribution plots.
- **Weakest Pair** (RestingBP and Cholesterol):
  - **Analysis:** In the cell between these two variables, the points for both groups (orange and blue) mix together without any clear pattern.
  - **Interpretation:** Combining these two variables does not offer much separation, showing that they are not useful predictors either alone or together.

### 3.4.4 Summary of Key Conclusions from Pairplot

- **Best individual predictors:** Oldpeak and MaxHR are the most effective in differentiating between the two patient groups.
- **Strength of combining attributes:** Combining attributes, particularly Oldpeak and MaxHR, provides much better separation between the groups than using a single attribute alone.
- **Least informative attributes:** RestingBP and Cholesterol show significant overlap between the groups and offer little predictive value, both individually and when combined.
- **Direction for modeling:** This analysis confirms that machine learning models should focus on attributes like Oldpeak, MaxHR, and Age for optimal performance.

## 3.5 Count plots

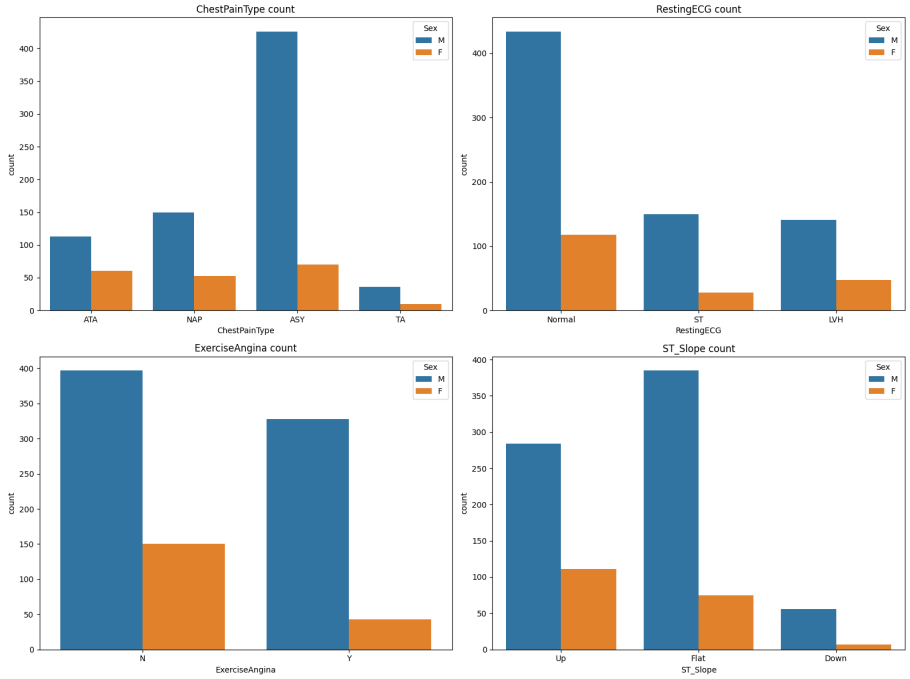


Figure 7: Count object columns by sex

These plots in Figure 7 are called Count Plots. They display the frequency of each category within a categorical attribute. In this case, each plot is further separated by Sex, with blue representing males (M) and orange representing females (F).The main goal is to understand the distribution characteristics of these categorical attributes and to observe any differences in quantity between the two sexes.

### 3.5.1 Detailed Analysis of Each Plot

- **ChestPainType** (Type of Chest Pain):
  - **Analysis:** Among the four types of chest pain, ASY (Asymptomatic - no symptoms) is the most dominant, especially among males. The number of males (M) is higher than females (F) across all chest pain types.
  - **Interpretation:** The majority of patients in this dataset, particularly males, are recorded with asymptomatic chest pain. This could be an important feature when analyzing the causes of heart disease.
- **RestingECG** (Resting Electrocardiogram):
  - **Analysis:** The Normal (Normal) result is the most common. The ST (ST-T wave abnormalities) and LVH (Left Ventricular Hypertrophy) types are less frequent. Similarly, the number of males exceeds the number of females in all three categories.
  - **Interpretation:** Most patients in the dataset have a normal resting electrocardiogram result.
- **ExerciseAngina** (Exercise-Induced Angina):
  - **Analysis:** This plot has two categories: N (No) and Y (Yes). The number of patients who do not experience exercise-induced angina (N) is significantly higher than those who do.
  - **Interpretation:** Exercise-induced angina is not a common symptom among the surveyed group. In both groups (with and without symptoms), males represent the majority.
- **ST Slope** (ST Segment Slope):
  - **Analysis:** Among the three slope types, Flat (Horizontal) is the most common, followed by Up (Upward). The Down (Downward) type is very rare.
  - **Interpretation:** The characteristic of a flat ST segment slope is the most frequent in this dataset. This is an important indicator on stress electrocardiograms and is often related to coronary artery disease.

### 3.5.2 Summary of Key Conclusions

- **Sex Imbalance:** The most prominent and important observation across all four plots is that the number of male patients (M) consistently exceeds the number of female patients (F) in nearly all categories. This indicates a significant gender imbalance in the dataset, which could affect model results.
- **Characteristics of the Majority Group:** The 'typical patient' in this data set tends to be a male, with the following characteristics:

asymptomatic chest pain (ASY), normal resting ECG (normal), no exercise-induced angina (N), and a flat slope of the ST segment (flat).

- **Guidance for Modeling:** Due to the large gender imbalance, the sex variable is likely to be an important predictive feature. Additionally, the prevalence of certain types (such as ASY or Flat) indicates that these features should be given particular attention when building the model.

# 4 Modeling and Evaluation

Before training and evaluating the three models, we first separated our dataset into input features (X) and the target label (y), and then split the data into training and testing sets using a 70-30 ratio, with a fixed random state for reproducibility.

```
# Split dataset into features and target variable
X = df.drop('HeartDisease', axis=1)
y = df['HeartDisease']

# Split into training and testing sets (70% train, 30% test)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42)
```

## 4.1 XGBoost (Extreme Gradient Boosting)

XGBoost is a machine learning algorithm based on decision trees, using a technique called boosting. Boosting is a method that combines many weak learners (models that perform slightly better than random guessing) to create a strong model. In XGBoost, each new tree is built sequentially, learning from the mistakes of the previous tree. This algorithm is very powerful and has great optimization capabilities thanks to techniques like regularization (which reduces overfitting) and tree pruning (which optimizes tree structure).

### 4.1.1 Hyperparameter Tuning for XGBoost

The following code demonstrates how we performed hyperparameter optimization for the XGBoost classifier using `RandomizedSearchCV` with cross-validation:

```
                    from xgboost import XGBClassifier
                    model1 = XGBClassifier(
                        max_depth=4,
```

```
    learning_rate=0.01,
    n_estimators=300,
    objective='binary:logistic',
    eval_metric='logloss'
)

param_dist1 = {
    'max_depth': [3,4,5,6,7,8,9],
    'learning_rate': [0.01, 0.05, 0.1, 0.3],
    'n_estimators': [100, 200, 500],
    'subsample': [0.6, 0.7, 0.9, 1.0],
    'colsample_bytree': [0.6, 0.7, 0.9, 1.0],
    'gamma': [0, 0.1, 0.2, 0.5],
    'reg_alpha': [0, 0.1, 1]
}

random_search1 = RandomizedSearchCV(
    model1,
    param_distributions=param_dist1,
    scoring='f1',
    n_iter=100,
    cv=3,
    verbose=1,
    n_jobs=-1
)

random_search1.fit(X_train, y_train)
print("Best parameters:", random_search1.best_params_)
```

The best XGBoost model was fine-tuned using a set of optimized hyperparameters obtained through randomized search. The selected configuration is as follows:

- **n_estimators**: 100
- **max_depth**: 8
- **learning_rate**: 0.1
- **subsample**: 1.0
- **colsample_bytree**: 0.6
- **gamma**: 0
- **reg_alpha (L1 regularization)**: 1

This configuration balances model complexity and generalization by limiting tree depth, controlling feature sampling per tree, and applying L1 regularization to prevent overfitting.

### 4.1.2 Results of XGBoost

```
Best accuracy: 0.8659
Best f1 score: 0.8840
Best recall score: 0.8598
Best roc-auc score: 0.8674
Classification report:
              precision    recall  f1-score   support

           0       0.81      0.88      0.84       112
           1       0.91      0.86      0.88       164

    accuracy                           0.87       276
   macro avg       0.86      0.87      0.86       276
weighted avg       0.87      0.87      0.87       276
```

Figure 8: Results of XGBoost model

The best-performing model achieved an accuracy of **86.6%**, with an **F1-score of 88.0%**, **recall of 85.9%**, and a **ROC-AUC score of 86.7%**. As shown in the classification report, the model performed consistently across both classes, with slightly higher precision and F1-score for the positive class (`class 1`). These evaluation metrics demonstrate strong and balanced predictive ability, providing a solid foundation before further assessment with the ROC curve and confusion matrix.

### 4.1.3 ROC curve and AUC score of XGBoost

Figure 9 show the ROC curve and AUC score of XGBoost

- **ROC Curve (Receiver Operating Characteristic Curve)**
  - **Meaning:** The ROC curve illustrates the model's ability to distinguish between the two classes  patients with heart disease and those without. It is plotted using the **True Positive Rate (TPR)** against the **False Positive Rate (FPR)** at various classification thresholds.
  - **Analysis:**
    - The ROC curve of the model (in blue) lies significantly closer to the **top-left corner** of the plot compared to the diagonal baseline (black), which represents random guessing.

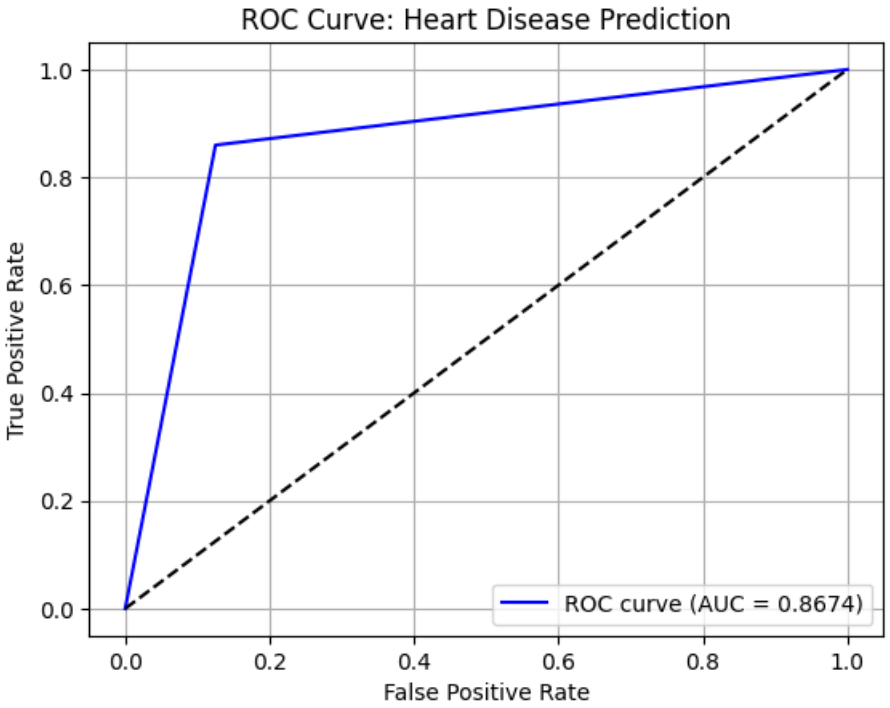


Figure 9: ROC curve of XGBoost model

  - This indicates that the model performs **much better than chance** in class separation.

- **AUC Score (Area Under the Curve)**
  - **Value:** 0.8674
  - **Conclusion:**
    - The AUC score quantifies the overall ability of the model to distinguish between the positive and negative classes across all thresholds.
    - An **AUC of 0.5** implies no discriminative power (equivalent to random guessing).
    - An **AUC of 1.0** indicates perfect classification.
    - With an AUC of **0.8674**, the XGBoost model demonstrates a **very strong discriminative capability**, effectively identifying patients with and without heart disease.
    - This is a **highly promising result**.

### 4.1.4 Confusion matrix of XGBoost

- **Understanding the Confusion Matrix:** Figure 10

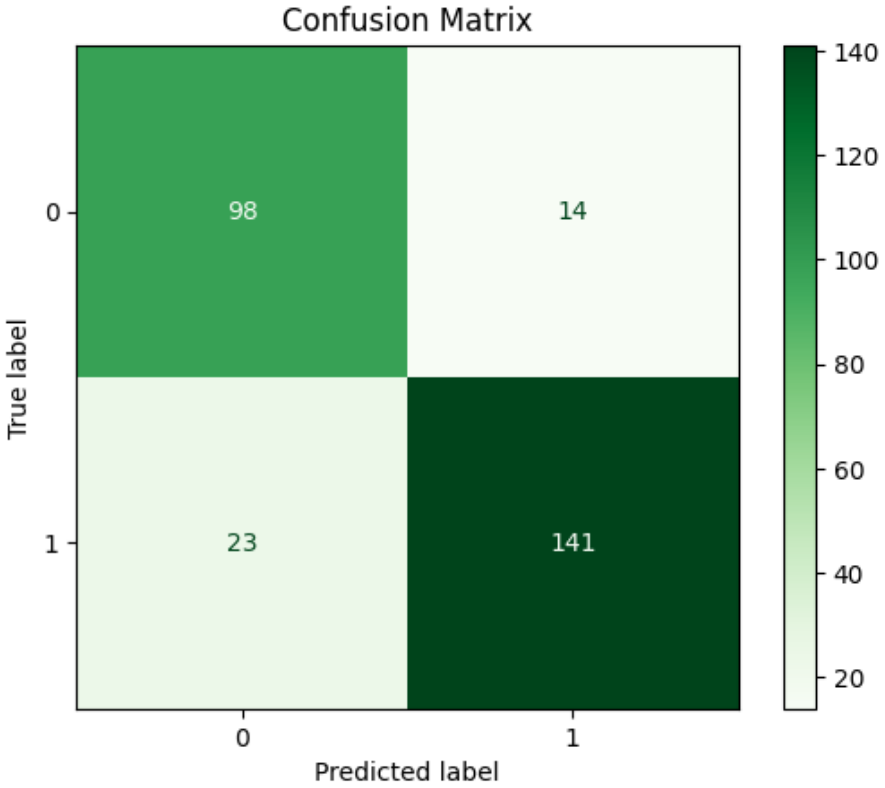


Figure 10: Confusion matrix of XGBoost model

  - The confusion matrix provides a detailed view of the models correct and incorrect predictions. In this case, class `0` represents *No disease*, and class `1` represents *Heart disease*.
  - **True Negative (TN): 98**
    - The model correctly predicted **98** patients as not having heart disease.
  - **False Positive (FP): 14**
    - The model incorrectly predicted **14** patients as having heart disease, while they were actually healthy (Type I error).
  - **False Negative (FN): 23**
    - The model incorrectly predicted **23** patients as not having heart disease, while they actually had it (Type II error  missed diagnosis).
  - **True Positive (TP): 141**
    - The model correctly predicted **141** patients as having heart disease.
- **Detailed Performance Evaluation:**
  - **Accuracy:** The proportion of correct predictions over the total cases.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{141 + 98}{276} \approx \mathbf{86.6\%}$$

- **Recall (Sensitivity):** The ability to detect actual positive cases (patients with heart disease).

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{141}{141 + 23} \approx \mathbf{86.0\%}$$

- **Specificity:** The ability to correctly identify negative cases (healthy individuals).

$$\text{Specificity} = \frac{TN}{TN + FP} = \frac{98}{98 + 14} \approx \mathbf{87.5\%}$$

- **Important trade-off:** The model exhibits a higher number of **False Negatives (23)** than **False Positives (14)**. This indicates a tendency to *miss* patients with the disease rather than falsely labeling healthy individuals. In medical applications, such omissions are critical and should be addressed, as missing true cases can pose serious risks.

## 4.2 Random Forest

Random Forest is a powerful ensemble learning algorithm that operates by constructing a multitude of decision trees at training time to improve predictive accuracy and control overfitting. Its core strength lies in two layers of randomness: first, it uses a technique called bagging (Bootstrap Aggregating), where each tree is trained on a different random data sample drawn with replacement from the original dataset. Second, when splitting each node, the algorithm considers only a random subset of features rather than all of them, which decorrelates the individual trees. To make a final prediction, the Random Forest aggregates the results from all treesby taking the majority vote for classification tasks or the average for regression tasks. This collective decision-making process cancels out individual errors, leading to a highly accurate and robust model that generalizes well to new data.

### 4.2.1 Build and train model of Random Forest

The following code snippet shows the configuration and training of a **Random Forest Classifier** with manually defined hyperparameters.

```
1  from sklearn.ensemble import
   ↪ RandomForestClassifier
2  from scipy.stats import randint
3
4  model2 = RandomForestClassifier(
5      n_estimators=300,
```

```
                    max_depth=15,
                    min_samples_split=5,
                    min_samples_leaf=2,
                    max_features='sqrt',
                    bootstrap=True,
                    random_state=42,
                    n_jobs=-1
                )

                model2.fit(X_train, y_train)
                y_pred2 = model2.predict(X_test)
```

### 4.2.2 Results of Random Forest

```
Best accuracy: 0.8804
Best f1 score: 0.8985
Best recall score: 0.8902
Best roc-auc score: 0.8782
Classification report:
              precision    recall  f1-score   support

           0       0.84      0.87      0.85       112
           1       0.91      0.89      0.90       164

    accuracy                           0.88       276
   macro avg       0.88      0.88      0.88       276
weighted avg       0.88      0.88      0.88       276
```

Figure 11: Results of Random forest model

The Random Forest model (Figure 11) achieved an overall accuracy of **88.0%**, with an impressive **F1-score of 89.9%** and a **recall of 89.0%**, indicating high effectiveness in detecting heart disease cases. Class-wise performance is well-balanced, with precision and recall values consistently strong for both classes. These results confirm the model's robustness and justify further evaluation with the ROC curve and confusion matrix.

### 4.2.3 ROC curve and AUC score of Random Forest

- **ROC Curve (Receiver Operating Characteristic Curve):**
  - **Analysis:** The ROC curve for the Random Forest model lies significantly high and leans toward the **top-left corner**, indicating excel-

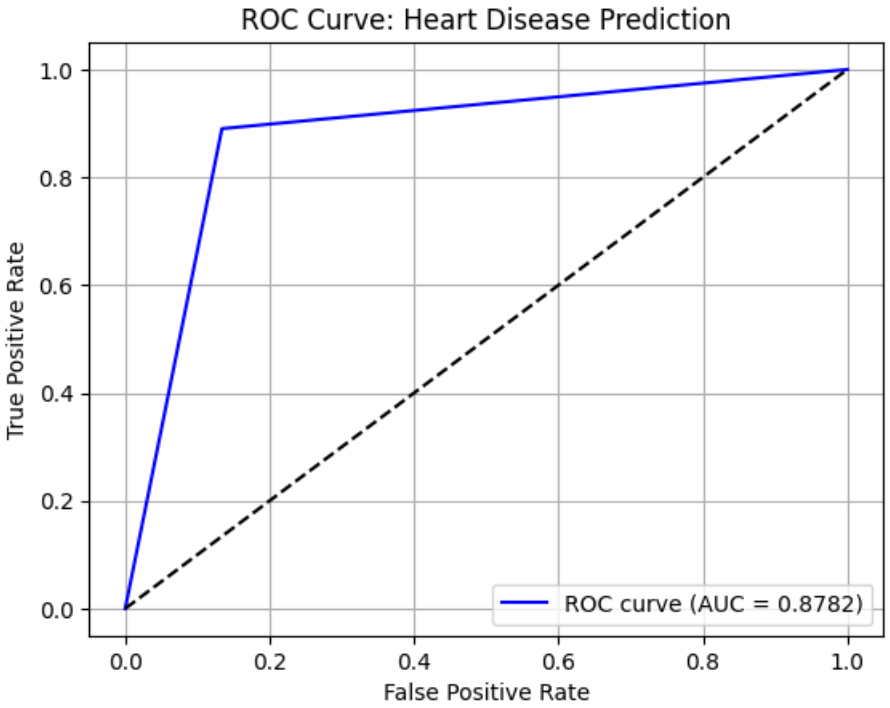


Figure 12: ROC curve of Random forest model

lent classification performance and a strong ability to separate the two classes.

- **AUC Score (Area Under the Curve):**
  - **Value:** 0.8782
  - **Conclusion:**
    - An AUC score of **0.8782** is considered **excellent** in binary classification tasks.
    - Compared to the XGBoost model (AUC = 0.8674), the Random Forest model shows a **slightly better discriminative ability**.

### 4.2.4 Confusion matrix of Random Forest

- **Understanding the Confusion Matrix:**
  - This matrix shows the detailed breakdown of correct and incorrect predictions made by the Random Forest model.
  - **True Negative (TN): 97**
    - The model correctly predicted **97** individuals as not having heart disease.
  - **False Positive (FP): 15**

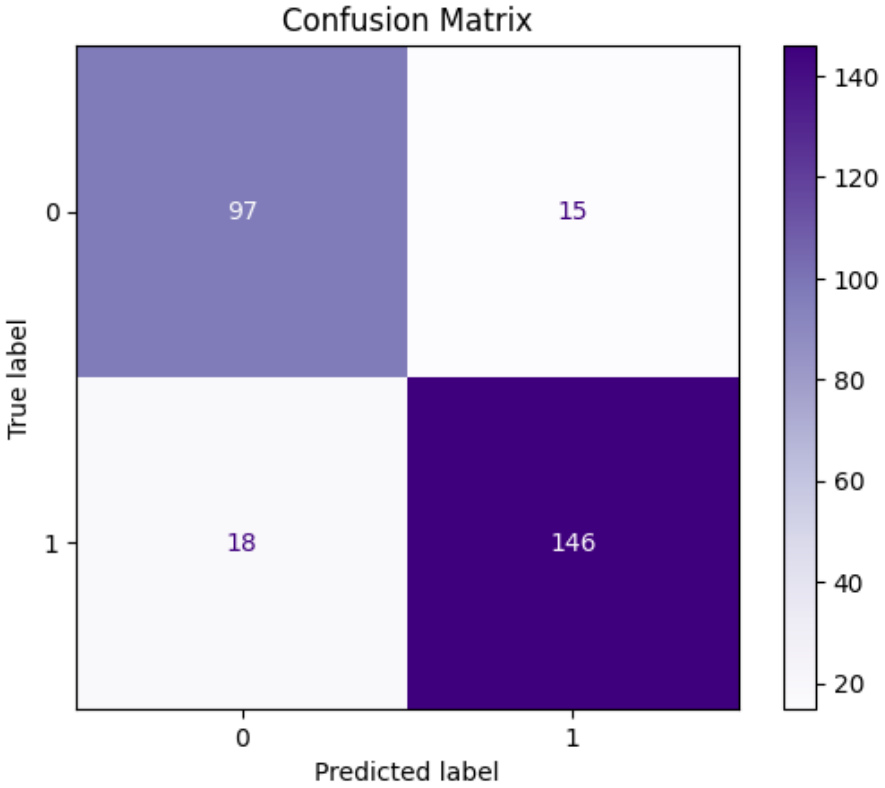


Figure 13: Confusion matrix of Random forest model

  - The model incorrectly predicted **15** individuals as having heart disease, while they were actually healthy.
- **False Negative (FN): 18**
  - The model missed **18** individuals who actually had heart disease (Type II error).
- **True Positive (TP): 146**
  - The model correctly identified **146** individuals as having heart disease.

- **Detailed Performance Evaluation:**
  - **Accuracy:**

$$\frac{TP + TN}{TP + TN + FP + FN} = \frac{146 + 97}{276} = \frac{243}{276} \approx \mathbf{88.0\%}$$

    This is higher than the accuracy of XGBoost (86.6%).
  - **Recall (Sensitivity):**

$$\frac{TP}{TP + FN} = \frac{146}{146 + 18} \approx \mathbf{89.0\%}$$

This represents a significant improvement over XGBoost (86.0%).

– **Most Important Improvement:** The number of missed positive cases (False Negatives) decreased from **23 (XGBoost)** to **18 (Random Forest)**. This is a notable advancement in the context of medical diagnosis, as it helps reduce the risk of undetected heart disease in patients.

## 4.3 Logistic Regression

Logistic Regression is a fundamental statistical model widely used for binary classification tasks. It estimates the probability that a given input belongs to a particular class by applying the logistic (sigmoid) function to a linear combination of input features. The model outputs values between 0 and 1, which are interpreted as class probabilities. Its simplicity, interpretability, and efficiency make it a strong baseline for many classification problems, particularly when relationships between variables are approximately linear. Despite its limitations in capturing complex patterns, Logistic Regression remains a robust choice for datasets with well-separated classes and low dimensionality.

### 4.3.1 Build and train model of Logistic Regression

The following code snippet shows the process of scaling input features and training a Logistic Regression model using Scikit-learn:

```
from sklearn.linear_model import
↪ LogisticRegression
from scipy.stats import loguniform
from sklearn.preprocessing import
↪ StandardScaler

# Feature scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Logistic Regression training
model3 = LogisticRegression()
model3.fit(X_train_scaled, y_train)
y_pred3 = model3.predict(X_test_scaled)
```

### 4.3.2 Results of Logistic Regression

The Logistic Regression model achieved an accuracy of **86.6%**, an **F1-score of 88.3%**, and a **recall of 84.8%**, showing reliable performance in detecting

```
Best accuracy: 0.8659
Best f1 score: 0.8825
Best recall score: 0.8476
Best roc-auc score: 0.8702
Classification report:
              precision    recall  f1-score   support

           0       0.80      0.89      0.84       112
           1       0.92      0.85      0.88       164

    accuracy                           0.87       276
   macro avg       0.86      0.87      0.86       276
weighted avg       0.87      0.87      0.87       276
```

Figure 14: Results of Logistic Regression model

heart disease cases. Notably, class 1 (patients with heart disease) had higher precision (**0.92**) and a strong F1-score (**0.88**), indicating the model's strong ability to correctly identify positive cases. Overall, the results are well-balanced, and the model offers competitive performance when compared to more complex models.

### 4.3.3 ROC curve and AUC score of Logistic Regression

Figure 15 show the result of ROC curve and AUC score

- **ROC Curve (Receiver Operating Characteristic Curve):**
  - **Analysis:** The ROC curve of the Logistic Regression model also demonstrates strong performance. It lies high and curves toward the **top-left corner**, indicating a clear ability to distinguish between positive and negative classes better than random guessing.
- **AUC Score (Area Under the Curve):**
  - **Value:** 0.8702
  - **Conclusion:**
    - This AUC score is considered **good**.
    - It is slightly higher than XGBoost (0.8674), but slightly lower than Random Forest (0.8782).
    - These results show that all three models perform **competitively** in distinguishing between the two classes.

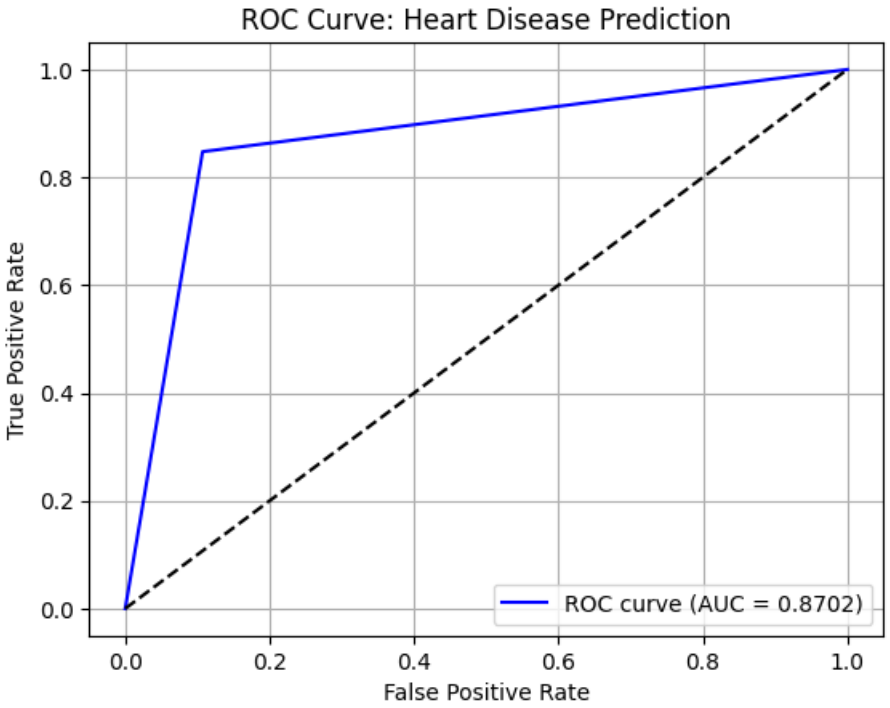


Figure 15: ROC curve Logistic Regression model

### 4.3.4 Confusion matrix of Logistic Regression

- **Understanding the Confusion Matrix:**
  - This matrix provides detailed information on the correct and incorrect predictions made by the Logistic Regression model.
  - **True Negative (TN): 100**
    - The model correctly predicted **100** individuals as not having heart disease.
  - **False Positive (FP): 12**
    - The model incorrectly predicted **12** individuals as having heart disease, while they were actually healthy.
  - **False Negative (FN): 25**
    - The model missed **25** individuals who actually had heart disease (Type II error).
  - **True Positive (TP): 139**
    - The model correctly identified **139** individuals as having heart disease.
- **Detailed Performance Evaluation:**

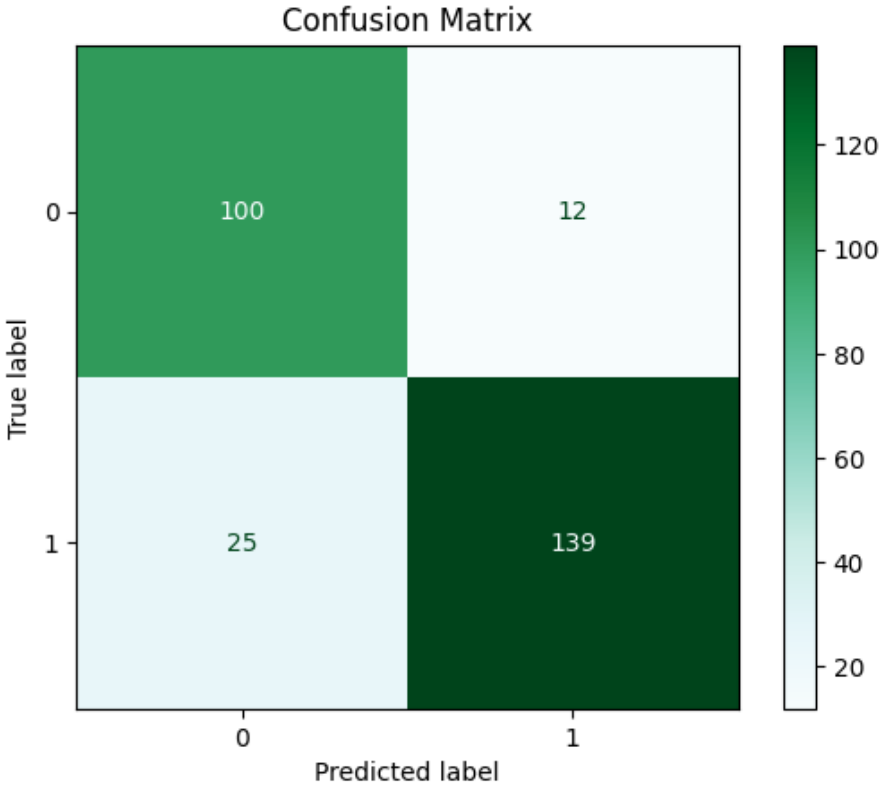


Figure 16: Confusion matrix Logistic Regression model

- **Accuracy:**

$$\frac{TP+TN}{TP+TN+FP+FN} = \frac{139+100}{276} = \frac{239}{276} \approx \mathbf{86.6\%}$$

  This is on par with XGBoost and slightly lower than Random Forest.
- **Recall (Sensitivity):**

$$\frac{TP}{TP+FN} = \frac{139}{139+25} \approx \mathbf{84.8\%}$$

  This is the **lowest recall** among the three evaluated models.
- **Weakness:** The model has the highest number of **False Negatives (25)**, indicating a notable risk in missing actual disease cases, which is critical in medical settings.
- **Strength:** It produces the lowest number of **False Positives (12)**, meaning it is relatively conservative and less likely to wrongly classify healthy individuals as having the disease.

### 4.4 Model Tuning Decision for both Random Forest and Logistic Regression

For both the **Random Forest** and **Logistic Regression** models, hyperparameter tuning was initially applied using `RandomizedSearchCV` to optimize for performance metrics such as F1-score and recall. However, after rigorous evaluation, we observed that the tuned models actually performed worse than their default or manually specified configurations, particularly in terms of accuracy.

In the case of **Random Forest**, tuning introduced deeper trees or overly aggressive sampling rates (e.g., low `subsample` or `max_features`) which may have led to overfitting on the training data, resulting in decreased generalization on the test set.

For **Logistic Regression**, the use of certain regularization strengths (e.g., extreme values of the inverse regularization parameter `C`) may have either underfit or overfit the data, causing a drop in predictive performance.

**Conclusion:** To maintain robustness and avoid unnecessary complexity, we reverted to the default or empirically better-performing configurations for these two models. Only the XGBoost model retained its tuned parameters, as it demonstrated clear performance gains from optimization.

## 5 Results Comparison


Figure 17: Performance Comparison of Models

The bar chart in Figure 17 provides a comprehensive visual comparison of the three models (XGBoost, Random Forest, and Logistic Regression) across four key evaluation metrics, aiming to identify the most suitable model for heart disease prediction.

## Analysis of Each Metric

- **Accuracy:**
  - All models achieved high accuracy scores, with values above 86%.
  - **Random Forest** (orange) had the highest accuracy, slightly ahead of XGBoost and Logistic Regression.
- **F1 Score:**
  - This metric balances both precision and recall.
  - **Random Forest** achieved the highest F1 Score (close to 0.90), followed closely by XGBoost and Logistic Regression.
- **Recall (Sensitivity):**
  - Recall is particularly important in medical prediction, where missing actual positive cases (false negatives) can be critical.
  - **Random Forest** stands out with the highest recall (approximately 0.89), indicating strong ability to identify patients with heart disease.
  - XGBoost performed moderately well, while Logistic Regression had the lowest recall among the three.
- **ROC AUC:**
  - All three models achieved competitive ROC AUC scores, closely clustered around 0.870.88.
  - This suggests that all models are capable of effectively distinguishing between positive and negative cases overall.

## Conclusion and Model Selection

- Although all models performed well, **Random Forest** consistently led across all key metrics.
- **Reason for selection:** It ranks first in **Accuracy**, **F1 Score**, and especially **Recall** the most important metric in medical diagnostics.
- **Final Decision:** Based on this comparison, **Random Forest is identified as the most suitable and effective model** for heart disease prediction in this study.

# 6 Conclusion

In this study, we evaluated and compared the performance of three machine learning models XGBoost, Random Forest, and Logistic Regression for heart disease prediction based on key metrics including Accuracy, F1 Score, Recall, and ROC AUC.

While all models demonstrated strong and competitive results, **Random Forest** consistently outperformed the others, particularly in **Recall** and **F1 Score**. This indicates that Random Forest is more effective in identifying patients with heart disease, while maintaining a good balance between sensitivity and precision.

Therefore, **Random Forest is recommended as the most suitable and reliable model** for this classification task, especially considering the importance of minimizing false negatives in medical diagnosis.

# References

[1] UCI Machine Learning Repository, Heart Disease Dataset.

[2] T. Chen, C. Guestrin, "XGBoost: A Scalable Tree Boosting System", arXiv:1603.02754

[3] Scikit-learn Documentation, https://scikit-learn.org

[4] Seaborn Visualization Library, https://seaborn.pydata.org