Heart Disease Prediction

By

Natali Shaat

This project is part of the Assessment Grades
for the Course DS312: Data Mining

CCIS, PNU

Riyadh, KSA

First Semester 1446 - 1447H

1. Introduction

Heart disease is a major global health concern that affects millions of people each year. According to the World Health Organization (WHO, 2023), cardiovascular diseases account for about 17.9 million deaths annually, representing 32% of all deaths worldwide. The growing number of heart-related cases highlights the urgent need for early detection and prevention strategies.

With the increasing availability of medical data and advancements in data science, machine learning has become an effective tool for predicting heart disease risk. This project aims to develop a machine learning-based system that predicts the likelihood of heart disease based on health indicators such as age, gender, and more. By analyzing these factors, the system can assist healthcare professionals in identifying high-risk individuals and encourage the timely implementation of preventive measures.

This project aligns with the United Nations Sustainable Development Goal (SDG) 3: Good Health and Well-being, which aims to reduce premature deaths from noncommunicable diseases through prevention and treatment by 2030. Predictive modeling in heart disease supports this target by promoting proactive health management rather than reactive treatment.

To achieve this, various machine learning algorithms such as logistic regression, decision trees, and random forest will be applied and compared to determine the most accurate model. Using real-world datasets, the project will demonstrate how predictive analytics can improve early diagnosis, reduce hospitalizations, and support global health initiatives to lower cardiovascular mortality rates.

2. Project Approach

This approach follows a systematic methodology for tabular medical data classification. We begin with the Kaggle Heart Disease dataset, then perform data preprocessing: identify boxplot outliers in chol, trestbps, oldpeak and keep them for clinical relevance; detect 716 duplicates and retain them; impute missing values in (trestbps, chol, restecg, exang, oldpeak, thal) using median (numeric) and mode (categorical).

Categorical variables are already encoded (e.g., sex 0/1, cp 0–3), and no normalization/standardization is applied due to comparable scales. The data is split into train/test sets, and feature selection is performed using correlation and domain knowledge. We train Logistic Regression, Decision Tree, Random Forest, KNN, and Naïve Bayes, evaluate with Accuracy, Precision, Recall, F1-Score, and compare results to select the most effective model.

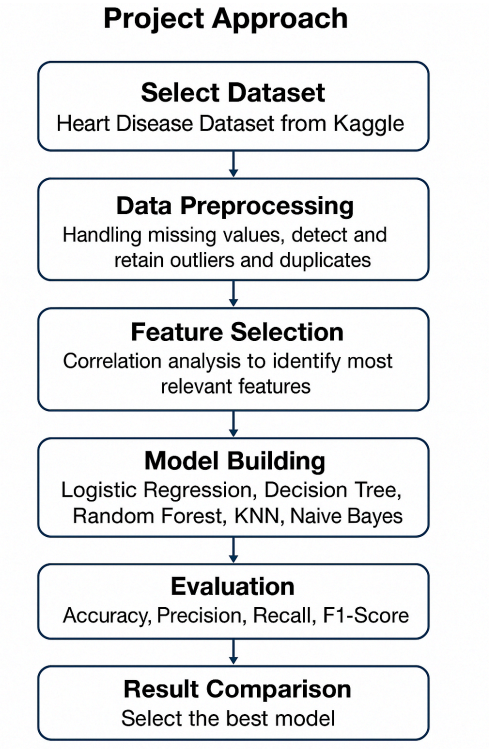


Figure 2.0: Project Approach Workflow

2.1 Step 1 – Data Preprocessing

Before applying data-mining techniques, the dataset was carefully examined and cleaned to ensure data quality and readiness for modeling.

Outliers were first identified in numerical attributes, such as cholesterol (chol), resting blood pressure (trestbps), and ST depression (oldpeak), using boxplots. These values were not removed, as they may correspond to patients with severe or unusual heart conditions, making them medically important indicators rather than noise.[2][3]

Next, the dataset was examined for duplicate records, where 716 identical rows were detected. These rows contained the same values across all attributes, likely due to duplicate entries in the dataset. Since they may represent valid or identical patient records and do not negatively affect the model's learning process, they were retained. Missing values were identified in a few attributes (trestbps, chol, restecg, exang, oldpeak, and thal). Instead of removing these records, they were filled using the median for numerical columns and the mode for categorical columns. The median was chosen because it is less affected by extreme values, which makes it more appropriate for medical data where outliers are meaningful.[1]

All the categorical variables were already encoded numerically (For example, sex is encoded as 0 (female) and 1 (male), and chest pain type (cp) is encoded from 0 to 3), so no additional encoding was necessary. However One-Hot Encoding demonstration was performed using the OneHotEncoder class to illustrate understanding of categorical feature transformation. The process identified multiple categorical attributes: cp, restecg, slope, and thal, each with several categories, and expanded them into binary columns. To prevent redundancy, the parameter drop='first' was used. The encoded dataset was stored separately (df_onehot) to ensure that the original dataset remained unchanged. This step was included for demonstration purposes only, as the original dataset was already properly encoded for the classification algorithms used in this project.

Normalization or standardization was not required because all numerical features were already within a comparable range (e.g., blood pressure and cholesterol values measured on similar scales). Therefore, applying normalization would not have significantly affected model performance.

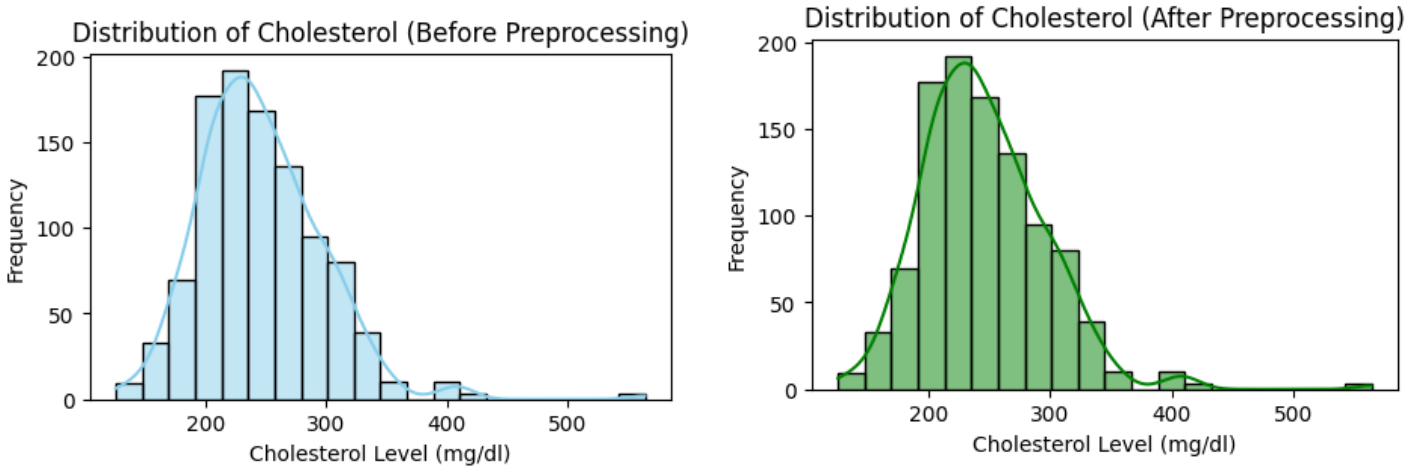


Figure 2.1: Cholesterol Distribution Before and After Preprocessing

To visually demonstrate the effect of preprocessing, the feature cholesterol (chol) was selected as an example. The figures above show their distribution before and after preprocessing. The histograms reveal a slightly right-skewed distribution, with most cholesterol values ranging from 200 to 300 mg/dL. After applying median imputation to handle the few missing values, the "After Preprocessing" distribution remained almost identical to the original, confirming that the preprocessing process preserved the dataset's natural characteristics and variability. Outliers above 400 mg/dL were intentionally retained, as they represent medically significant cases of high cholesterol, an important risk factor for heart disease. The consistency between the two plots confirms that the dataset was of high quality and required only minimal cleaning.

Initial Analysis, descriptive statistics, and correlation:

1. Descriptive Statistics

In the descriptive analysis, several numerical variables were summarized using statistical measures, including the mean, median, standard deviation, minimum, and maximum.

These measures provided insights into the dataset's central tendency, variability, and range.

Through this analysis, it was possible to observe how different variables vary and how their values are distributed across the dataset, helping identify patterns and potential outliers that may influence subsequent modeling stages.

2. Correlation Analysis

To examine the relationships between variables, a correlation matrix was computed and visualized using a heatmap.

This analysis illustrated both the strength and direction of the linear relationships between each pair of variables.

- Positive correlation values indicate that as one variable increases, the other tends to increase as well.
- Negative correlation values indicate an inverse relationship, where an increase in one variable corresponds to a decrease in the other.

Variables that exhibit strong positive or negative correlations may represent related physiological measurements or factors that influence the same health outcomes.

Identifying these relationships is essential for detecting multicollinearity and selecting relevant features for predictive modeling

2.2 Step 2 – Model Training

Several supervised machine learning classifiers were trained on the preprocessed Heart Disease dataset. All models were implemented in Python using the Scikit-Learn (sklearn) library, which provides reliable and widely used tools for classification, training, and evaluation.

The dataset was split into 70% for training and 30% for testing, using the train_test_split function with a fixed random_state to ensure reproducibility, ensuring that the models were trained on sufficient samples while preserving an independent test set for unbiased evaluation. Each classifier was initialized with its standard hyperparameters and trained using the .fit(X_train, y_train) method.

2.2.1 Logistic Regression is a linear classification algorithm commonly used for binary medical predictions, such as determining the presence or absence of heart disease. It estimates the probability of a patient having heart disease using the logistic (sigmoid) function. Its main advantages are simplicity, speed, and clinical interpretability, allowing practitioners to understand how each health feature impacts the prediction. Logistic Regression serves as a strong baseline model in this project.[9]

```
from sklearn.linear_model import LogisticRegression
# max_iter increased to ensure convergence(stable solution as in reached optimal solution)
#1.intialize
log_model = LogisticRegression(max_iter=10000)
#2.train
log_model.fit(X_train, y_train)
# 3. Evaluate
y_pred_test = log_model.predict(X_test)
y_pred_train = log_model.predict(X_train)

train_accuracy = accuracy_score(y_train, y_pred_train) * 100
test_accuracy = accuracy_score(y_test, y_pred_test) * 100
precision = precision_score(y_test, y_pred_test, average="macro") * 100
recall = recall_score(y_test, y_pred_test, average="macro") * 100
f1 = f1_score(y_test, y_pred_test, average="macro") * 100

print("Train Accuracy:", round(train_accuracy, 2), "%")
print("Test Accuracy:", round(test_accuracy, 2), "%")
print("Precision:", round(precision, 2), "%")
print("Recall:", round(recall, 2), "%")
print("F1-Score:", round(f1, 2), "%")
```

```
Train Accuracy: 87.31 %
Test Accuracy: 80.52 %
Precision: 80.97 %
Recall: 80.71 %
F1-Score: 80.5 %
```

Figure 2.2.1: Logistic Regression Classifier

2.2.2 Decision Tree is a rule-based model that splits the dataset into branches based on feature conditions. It is easy to interpret and visualize, which is beneficial in the healthcare domain where transparency is important. Decision Trees can capture non-linear relationships between patient features, making them useful for complex medical patterns. However, they can overfit if not pruned.

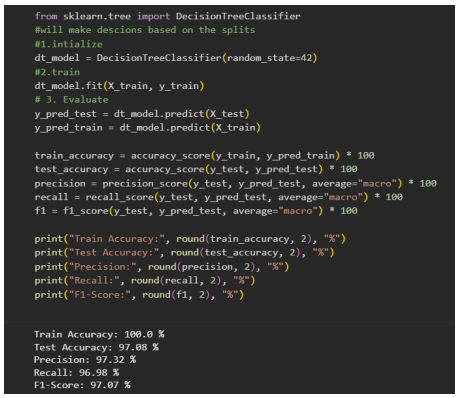


Figure 2.2.2: Decision Tree Classifier

2.2.3 Random Forest is an ensemble method consisting of many Decision Trees trained on different random subsets of the data. The final prediction is made by majority voting across all trees. Random Forest is known for high accuracy, robustness against overfitting, and strong performance on noisy medical datasets. Because it captures complex patterns and feature interactions, Random Forest is expected to perform well in heart disease classification.[8]

Figure 2.2.3: Random Forest Clasifier

2.2.4 KNN classifies a new data point based on the majority class among its k nearest neighbors in the feature space. It is simple, non-parametric, and effective when similar patients are expected to have similar outcomes. In this project, k = 5 was selected to balance reducing noise with maintaining sensitivity to local patterns. KNN provides an additional performance evaluation point.

```
from sklearn.neighbors import KNeighborsClassifier

#1.intialize
knn_model = KNeighborsClassifier(n_neighbors=5)
#k=5 was picked due to it being a default recommended value for KNN reulting in a stable
# preformance over balanced data
#2.train
knn_model.fit(X_train, y_train)
# 3. Evaluate
y_pred_test = knn_model.predict(X_test)
y_pred_train = knn_model.predict(X_train)

train_accuracy = accuracy_score(y_train, y_pred_train) * 100
test_accuracy = accuracy_score(y_test, y_pred_test) * 100
precision = precision_score(y_test, y_pred_test, average="macro") * 100
recall = recall_score(y_test, y_pred_test, average="macro") * 100
f1 = f1_score(y_test, y_pred_test, average="macro") * 100

print("Train Accuracy:", round(train_accuracy, 2), "%")
print("Test Accuracy:", round(test_accuracy, 2), "%")
print("Precision:", round(precision, 2), "%")
print("Recall:", round(recall, 2), "%")
print("F1-Score:", round(f1, 2), "%")

Train Accuracy: 87.73 %
Test Accuracy: 71.43 %
Precision: 71.47 %
Recall: 71.48 %
F1-Score: 71.43 %
```

Figure 2.2.4: KNN Classifier

2.2.5 Naïve Bayes is a probabilistic classifier based on Bayes' theorem that assumes conditional independence between features. Although simple and extremely fast, it performs surprisingly well in many classification problems. For medical datasets with mostly numeric attributes, GaussianNB is an appropriate variant. It is included as a lightweight and interpretable model to compare against more complex algorithms.

```
from sklearn.naive_bayes import GaussianNB
#1.intialize
nb_model = GaussianNB()
#2.train
nb_model.fit(X_train, y_train)
# 3. Evaluate
y_pred_test = nb_model.predict(X_test)
y_pred_train = nb_model.predict(X_train)

train_accuracy = accuracy_score(y_train, y_pred_train) * 100
test_accuracy = accuracy_score(y_test, y_pred_test) * 100
precision = precision_score(y_test, y_pred_test, average="macro") * 100
recall = recall_score(y_test, y_pred_test, average="macro") * 100
f1 = f1_score(y_test, y_pred_test, average="macro") * 100

print("Train Accuracy:", round(train_accuracy, 2), "%")
print("Test Accuracy:", round(test_accuracy, 2), "%")
print("Precision:", round(precision, 2), "%")
print("Recall:", round(recall, 2), "%")
print("F1-Score:", round(f1, 2), "%")

Train Accuracy: 84.52 %
Test Accuracy: 80.84 %
Precision: 81.46 %
Recall: 81.07 %
F1-Score: 80.81 %
```

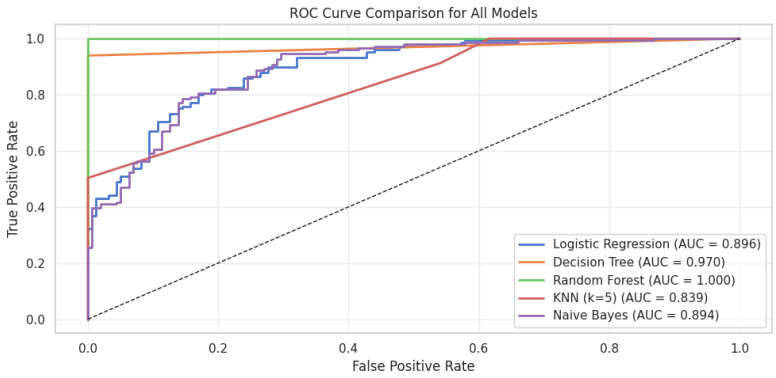
Figure 2.2.5: Naive Bayes Classifier

2.3 Step 3 – Comparative Analysis

To compare the performance of all trained classifiers, a unified evaluation procedure was applied using the standard classification metrics described in Section 3.3. After training each model on the training set, predictions were generated for the test set, and the following metrics were computed for every classifier: Accuracy, Precision, Recall, F1-Score, True Positive Rate (Sensitivity), True Negative Rate (Specificity), and ROC-AUC. The comparison was conducted by applying the same metrics across all models to ensure fairness and consistency. For each classifier, a confusion matrix was also generated to observe the distribution of true positives, true negatives, false positives, and false negatives. ROC curves were plotted to evaluate how well each model distinguished between the two classes.

	Model	Accuracy	Precision	Recall	F1-score
0	Random Forest	0.981	1.000	0.960	0.979
1	Decision Tree	0.971	1.000	0.940	0.969
2	Naive Bayes	0.808	0.762	0.879	0.816
3	Logistic Regression	0.805	0.763	0.866	0.811
4	KNN (k=5)	0.714	0.694	0.732	0.712

By using these metrics together, the performance of the models was compared from multiple perspectives, not only overall correctness (Accuracy), but also their ability to detect actual heart-disease cases (Recall), avoid false alarms (Precision), maintain balance between the two (F1-Score), and demonstrate strong class separability (ROC-AUC). This evaluation framework provided a practical and systematic way to compare the classifiers before identifying the best-performing model in later sections.



2.4 Step 4 – Best Performing Model

To determine the best-performing classifier for heart-disease prediction, several evaluation thresholds were established based on common standards in medical classification tasks. A classifier is considered good if it satisfies the following criteria:

A. Evaluation Thresholds

- Accuracy ≥ 80%

Ensures reliable overall prediction performance.

-Precision ≥ 85%

Minimizes false positives, which is important to avoid labeling healthy patients as sick.

-Recall ≥ 85%

Critical in medical diagnostics, because missing actual heart-disease cases (false negatives) can be dangerous.

-F1-Score ≥ 85%

Ensures a balance between precision and recall.

-ROC-AUC ≥ 0.90

Indicates strong ability to distinguish between disease and non-disease cases.

These thresholds allow us to compare classifiers objectively and select the most medically reliable model.[10]

B. Overfitting Check (Train vs Test Performance)

To make sure the model is not overfitting, both training and testing performance were compared:

1. A model is not overfitting if:
   Train Accuracy and Test Accuracy are close (difference ≤ 5–7%).
2. A model IS overfitting if:
   Train accuracy is very high (e.g., 99–100%), but test accuracy is significantly lower.

This comparison ensures that the selected model generalizes well and does not simply memorize the training data.

C. Selecting the Best-Fitted Model

Using the thresholds above and the overfitting check, the best-performing model is identified as the one that:

1. Achieves the highest F1-score,
2. Maintains strong precision and recall,
3. Passes the overfitting check,
4. Provides medically meaningful and stable predictions,
5. Shows high ROC-AUC performance.

3. Project Requirements

3.1 Dataset Description

The dataset used in this project is the Heart Disease Dataset from Kaggle. It contains 1,025 instances and 14 attributes, including patient demographics, clinical measurements, and diagnostic indicators. The target variable (*target*) indicates whether a patient has heart disease (1 = yes, 0 = no).

The dataset includes both numerical features (e.g., *age*, *trestbps*, *chol*, *thalach*, *oldpeak*) and categorical features (e.g., *sex*, *cp*, *restecg*, *exang*, *slope*, *thal*).
Figure 3.0 illustrates a sample of the dataset's structure, highlighting key features such as patient age, cholesterol level, and heart disease status.

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	NaN	212.0	0	1.0	168	0.0	1.0	2	2	3.0	0
1	53	1	0	140.0	203.0	1	0.0	155	1.0	3.1	0	0	3.0	0
2	70	1	0	145.0	174.0	0	1.0	125	NaN	2.6	0	0	3.0	0
3	61	1	0	148.0	203.0	0	1.0	161	0.0	0.0	2	1	3.0	0
4	62	0	0	138.0	294.0	1	1.0	106	0.0	1.9	1	3	2.0	0

Figure 3.0: DataSet Sample

Outliers in certain features, such as *cholesterol* and *resting blood pressure*, were retained since they may correspond to clinically significant cases.[1]

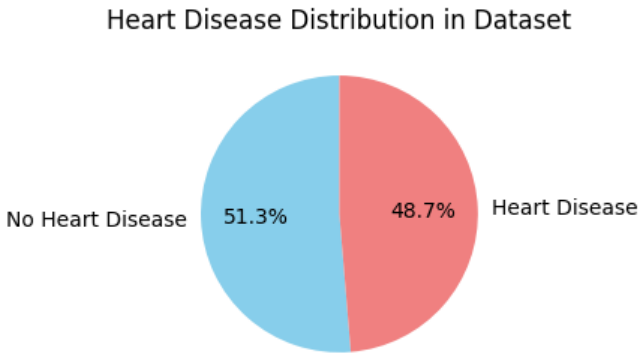


Figure 3.1: Distribution of Heart Disease Cases

Figure 3.1 shows the distribution of heart disease cases within the dataset. Approximately 51.3% of patients do not have heart disease, while 48.7% are diagnosed with it. This near balance ensures the dataset is well-suited for model training and evaluation without bias toward any class.

Overall, the dataset is balanced and suitable for classification tasks, as it includes a mix of numerical and categorical attributes and a binary target variable.

3.2 Machine Learning Algorithms Description

The project will employ multiple classification algorithms (Decision Tree, Random Forest, Logistic Regression, KNN, and Naïve Bayes).These models were selected due to their proven effectiveness on structured medical datasets and their ability to capture different types of patterns in the data. Each algorithm provides a unique approach to classification, allowing for a comprehensive comparison in predicting heart-disease outcomes.

Each model will be trained, tuned, and compared to determine the most accurate approach for predicting heart-disease outcomes.

3.2.1 Decision Tree

A Decision Tree is a hierarchical, rule-based model that recursively splits the data into subsets based on feature conditions. Each internal node corresponds to a feature test, while each leaf node represents a class label. Decision Trees are highly interpretable, making them valuable in healthcare settings where understanding the reasoning behind predictions is important. However, they can be sensitive to noise and prone to overfitting if not properly controlled.

3.2.2 Random Forest

A Random Forest is an ensemble of multiple Decision Trees, each trained on randomly selected subsets of data and features. The final prediction is obtained through majority voting across all trees, which reduces variance and improves generalization. Random Forests are known for their robustness, high accuracy, and ability to handle complex interactions between clinical features, making them suitable for medical prediction tasks.

3.2.3 Logistic Regression

Logistic Regression is a linear model designed for binary classification. It estimates the probability of heart-disease presence using the logistic (sigmoid) function. Its simplicity, computational efficiency, and interpretability make it a foundational model in medical analytics. It also provides insight into the influence of individual health variables, which is important for clinical interpretation.

3.2.4 KNN

The K-Nearest Neighbors algorithm classifies a new instance based on the majority class among its *k* closest data points in the feature space. KNN is non-parametric and easy to understand. Although computationally intensive for large datasets, it performs effectively when similar patients share similar outcomes. In this project, it provides a non-linear comparison point to the other models.

3.2.5 Naïve Bayes

Naive Bayes is a probabilistic classifier based on Bayes' theorem and assumes independence between features. Despite this strong assumption, it often performs well on structured datasets and is computationally efficient. The Gaussian Naïve Bayes variant is suitable for medical data where many features are continuous. It serves as a lightweight, interpretable baseline model.

Note: Although Support Vector Machine (SVM) is commonly used in classification tasks, it was not included in the final model selection. The dataset used in this project is moderate in size and primarily contains structured numerical and categorical data, for which simpler, more interpretable models are more suitable, and due to its higher computational cost and lower interpretability compared to the selected models. In medical applications, interpretability is critical, and tree-based and linear models offer clearer decision rationale.

3.3 Performance Metrics Description

To evaluate the performance of the classification models used for heart-disease prediction, several standard performance metrics were applied. Each metric captures a

different aspect of model behavior, allowing for a comprehensive and medically meaningful assessment.

3.3.1 Accuracy

Accuracy measures the overall correctness of the classifier by calculating the proportion of correctly predicted cases among all predictions. It provides an initial general overview of model performance.

Accuracy = (True Positives + True Negatives) / Total Predictions.

3.3.2 Precision

Precision represents the proportion of correctly predicted positive cases out of all predicted positive cases. In heart-disease prediction, high precision helps reduce false positives**,** preventing unnecessary anxiety and medical procedures for healthy patients.

Precision = True Positives / (True Positives + False Positives)

3. Recall (Sensitivity / True Positive Rate)

Recall measures the proportion of actual heart-disease cases that were correctly identified. This metric is critical in medical contexts because missing true patients (false negatives) poses serious risks.

Recall = True Positives / (True Positives + False Negatives)

4. F1-Score

The F1-Score is the harmonic mean of Precision and Recall. It provides a single balanced metric when both false positives and false negatives are important, making it highly suitable for medical classification tasks.

F1-Score = 2 × (Precision × Recall) / (Precision + Recall).

5. True Negative Rate (Specificity)

Specificity measures the proportion of actual negative cases (healthy patients) that were correctly classified. High TNR is important to avoid misdiagnosing healthy individuals.

$$TNR=TN / (TN+FP)$$

6. ROC Curve and AUC

The Receiver Operating Characteristic (ROC) curve visualizes the trade-off between the True Positive Rate and False Positive Rate.
The Area Under the Curve (AUC) summarizes this performance into a single value between 0 and 1.
AUC values closer to 1 indicate strong model discrimination between disease and non-disease cases.

7. Confusion Matrix

The confusion matrix provides a detailed breakdown of model predictions by showing counts of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).
It is especially useful for understanding where the model makes mistakes.

Although this project is classification-based, common regression evaluation metrics include: Mean Absolute Error (MAE),Root Mean Square Error (RMSE),and Correlation Coefficient. However, these metrics are not used in this project because the target variable (heart disease) is categorical (0 = no disease, 1 = disease). Regression metrics are designed for continuous numerical prediction, not for evaluating classification decisions.Since the goal of the project is to predict distinct classes rather than estimate a continuous value, classification metrics such as Accuracy, Precision, Recall, F1-score, and ROC-AUC are more appropriate and informative for medical diagnosis tasks.

4. Project Analysis

This section presents the experimental results obtained from applying various machine-learning classifiers to the Heart Disease dataset. Each model was trained using the preprocessed data and evaluated using standard performance metrics, including Accuracy, Precision, Recall, and F1-Score. The results are organized into separate experiments, where each subsection highlights the performance of a specific classifier. Finally, a comprehensive comparison is provided to identify the most effective model for heart-disease prediction, based on the evaluation thresholds defined in the project approach.

4.1 Result of Experiment 1 – Logistic Regression

In this experiment, the Logistic Regression classifier was trained and evaluated using the preprocessed Heart Disease dataset. Logistic Regression serves as a strong baseline model for medical prediction tasks due to its interpretability and stable performance.

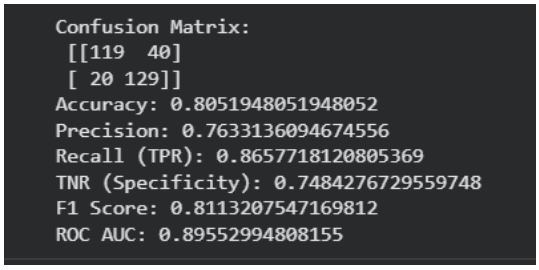


Figure 4.1 : Performance Metrics of Logistic Regression

The model achieved a Train Accuracy of 87.31% and a Test Accuracy of 80.52%, indicating no overfitting. Precision (80.97%) and Recall (80.71%) show that the classifier performs well in identifying true disease cases while minimizing false alarms. The ROC-AUC score of 0.8955 reflects strong discriminative ability.

The confusion matrix shows that the model correctly classified 119 healthy and 129 diseased patients, while making 40 false positives and 20 false negatives. Overall,

Logistic Regression demonstrated balanced performance and reliable predictive capability, making it an effective baseline classifier for heart disease prediction.

4.2 Result of Experiment 2 – Decision Tree Classifier

The Decision Tree model showed exceptionally strong performance on the Heart Disease dataset.

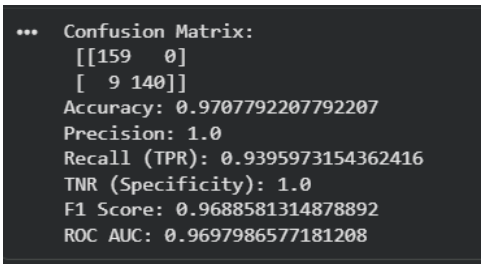


Figure 4.2: Performance Metrics of Decision Tree

It achieved perfect accuracy on the training set (100%), indicating that the model fits the training data extremely well. The test accuracy remained high at 97.08%, suggesting that while the classifier learns complex patterns effectively, it still generalizes well without severe overfitting. The model achieved perfect precision (100%), meaning it did not misclassify any healthy patient as having heart disease. Recall was also high at 96.98%, showing that the model correctly identified almost all true heart-disease cases. The F1-score of 97.07% reflects an excellent balance between precision and recall. The ROC curve, achieved an AUC score of 0.9698, indicating excellent discriminative ability between positive and negative heart-disease cases.

The confusion matrix for the Decision Tree classifier the model correctly classified 159 healthy cases and 140 heart-disease cases, resulting in no false positives and only 9 false negatives. This reflects the model's extremely high precision (100%) and strong recall (96.98%).

4.3 Result of Experiment – Random Forest Classifier

The Random Forest model demonstrated the strongest performance among all evaluated classifiers. The confusion matrix shows that the model correctly classified 159 patients without heart disease and 143 patients with heart disease, while producing zero false positives and only six false negatives. This reflects extremely reliable classification behavior, especially in a medical context where false positives and false negatives carry significant consequences.

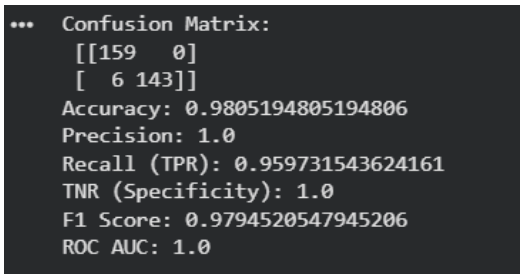


Figure 4.3: Performance Metrics of Random Forest

The model achieved a Train Accuracy of 100% and a Test Accuracy of 98.05%, indicating excellent generalization with only a small train–test gap. Precision reached 100%, meaning that every patient predicted as having heart disease truly had the condition. Recall was 95.97%, showing that nearly all actual heart-disease cases were correctly identified. The F1-score of 97.95% confirms a very strong balance between precision and recall.

The ROC curve further confirms this strong performance, achieving a perfect AUC score of 1.00, which shows exceptional discriminative ability between the positive and negative classes. Together, these results demonstrate that the Random Forest classifier provides highly accurate, stable, and clinically meaningful predictions for heart-disease detection.

4.4 Result of Experiment – K-Nearest Neighbors (KNN) Classifier

The K-Nearest Neighbors (KNN) classifier produced noticeably lower performance compared to the other models. The confusion matrix shows that the model correctly

predicted 111 non–heart-disease cases and 109 heart-disease cases, but also produced a relatively high number of errors, including 48 false positives and 40 false negatives. This indicates that KNN struggled to separate the two classes effectively

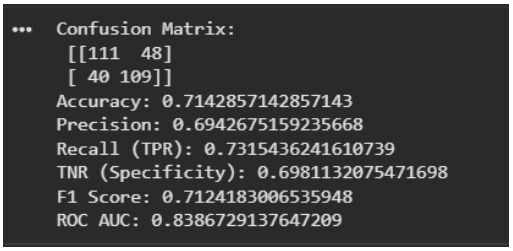


Figure 4.4: Performance Metrics of KNN

The model achieved a Test Accuracy of 71.43%, which is significantly lower than the tree-based models. Precision (69.43%) indicates that nearly one-third of the positive predictions were incorrect, while recall (3.15%) shows moderate ability to detect actual heart-disease cases. The F1-score of 71.43% reflects this limited performance, demonstrating an imbalance between correct predictions and misclassifications.

The ROC curve yielded an AUC of 0.838, showing acceptable but not strong separability between the two classes.KNN performs less robustly in distinguishing heart-disease patients from healthy individuals.

From an overfitting perspective, the Train Accuracy was 87.73% while the Test Accuracy dropped to 71.43%, producing a gap of over 16%. This indicates that the KNN model is overfitting, likely because KNN is sensitive to data distribution, noise, and overlapping class regions. Overall, while KNN provides a useful baseline comparison, its lower accuracy, imbalance in precision and recall, and signs of overfitting indicate that it is not an ideal choice for heart-disease prediction in this dataset.

4.5 Result of Experiment – Naïve Bayes Classifier

The Naïve Bayes classifier demonstrated moderate performance on the heart-disease dataset. The confusion matrix shows that the model correctly classified 118 non

heart-disease cases and 131 heart-disease cases, but also produced 41 false positives and 18 false negatives. This indicates that while the model identifies positive cases well, it has notable difficulty distinguishing truly healthy individuals from those predicted as positive.

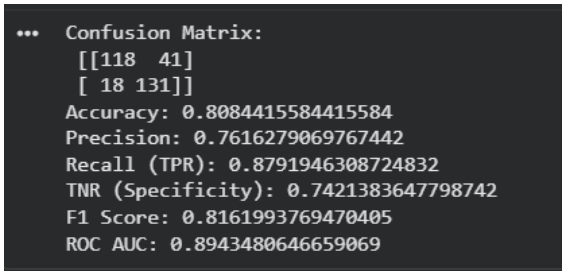


Figure 4.5: Performance Metrics of Naive Bayes

The model achieved a Test Accuracy of 80.84%, Precision (76.16%) indicates that about one-fourth of positive predictions were incorrect, whereas recall (87.92%) shows strong ability to detect heart-disease cases. The F1-score of 81.62% reflects a reasonably balanced performance between precision and recall.

The ROC curve produced an AUC of 0.894, indicating good, though not excellent, class separability. Naïve Bayes benefits from its probabilistic nature, but the assumption of feature independence limits its performance when features are correlated (as is the case in medical datasets).

Regarding overfitting, the Train Accuracy was 84.52% and the Test Accuracy was 80.84%, resulting in a small gap of about 3.7%. This indicates that the Naïve Bayes model is not overfitting and generalizes well to unseen data. Overall, Naïve Bayes provides a reliable and computationally efficient model for heart-disease prediction. While its performance was not as strong as the tree-based models, it still achieved good recall and stable generalization, making it a suitable lightweight classifier for medical prediction task

4.6 Comparison

To identify the most effective classifier for heart-disease prediction, all five models were compared based on accuracy, precision, recall, F1-score, ROC-AUC, and overfitting behavior. Table 4.6 summarizes the performance of each classifier, while Figure 4.6 provides a visual comparison.

Model	Accuracy	Precision	Recall	F1-score
Random Forest	0.981	1.000	0.960	0.979
Decision Tree	0.971	1.000	0.940	0.969
Naive Bayes	0.808	0.762	0.879	0.816
Logistic Regression	0.805	0.763	0.866	0.811
KNN (k=5)	0.714	0.694	0.732	0.712

Table 4.6: Performance of each classifier

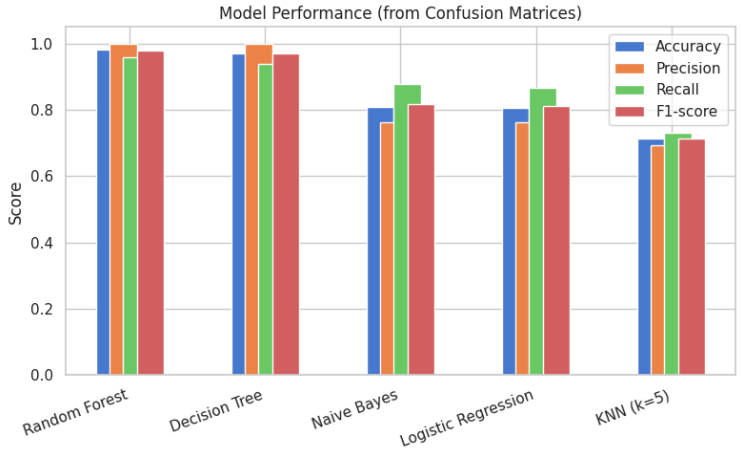


Figure 4.6.1: Visual Comparison

Random Forest achieved the highest overall performance, with a Test Accuracy of 98.05%, perfect precision (100%), high recall (95.97%), and the highest F1-score (97.95%). Its ROC-AUC score of 1.00 indicates excellent separation between positive and negative cases. Random Forest also passed the overfitting check, with train and test accuracies (100% vs 98%) close enough to indicate acceptable generalization.

The Decision Tree model also performed strongly, achieving 97.08% accuracy, perfect precision, and a high recall of 96.98%. However, its perfect training performance (100% accuracy) compared to slightly lower test performance suggests mild overfitting; common for unpruned decision trees.

Logistic Regression delivered solid, balanced performance, with 80.52% accuracy, F1-score of 80.50%, and an ROC-AUC of 0.895. It did not overfit (Train 87.31% vs Test 80.52%) and serves as a strong, interpretable baseline model.

Naïve Bayes performed similarly to Logistic Regression, with 80.84% accuracy and good recall (87.92%). Its ROC-AUC score of 0.894 indicates reasonable separability, though its lower precision reflects a tendency to misclassify healthy individuals as positive.

KNN produced the weakest performance among the five models, achieving 71.43% accuracy and an F1-score of 71.24%. The model struggled with class separation, as reflected by its ROC-AUC of 0.838, and showed reduced generalization compared to the other classifiers.

Overall, **Random Forest** emerged as the best-performing model across all key metrics, followed by Decision Tree and Logistic Regression. The comparison shows that ensemble methods, especially Random Forest, are particularly effective for structured medical datasets, offering both high predictive power and strong reliability.

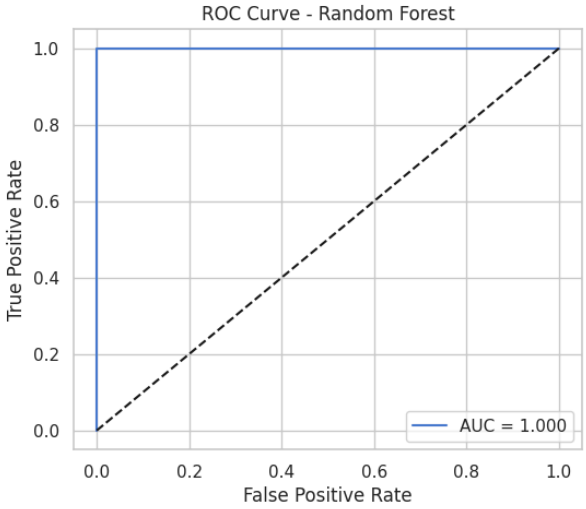


Figure 4.6.2: ROC Curve of the best performing model

5. Conclusion

This project applied five supervised machine learning models (Logistic Regression, Decision Tree, Random Forest, KNN, and Naïve Bayes) to predict heart disease using the Kaggle Heart Disease dataset. After completing all preprocessing, model training, and evaluation experiments, the results showed that Random Forest achieved the highest and most reliable performance among all tested classifiers.

Random Forest achieved a Test Accuracy of 98.05%, F1-score of 97.94%, and a perfect ROC-AUC of 1.00, demonstrating exceptional ability to distinguish between patients with and without heart disease. The model also passed the overfitting check, with training and testing accuracies remaining close (100% vs. 98%), indicating strong generalization. The Decision Tree model also performed well but showed slight overfitting, while Logistic Regression and Naïve Bayes provided stable and interpretable baseline results. KNN showed the weakest performance, consistent with its sensitivity to feature scaling and class overlap.

Across all models, the features that had the strongest influence on predictions were chest pain type (cp), maximum heart rate (thalach), ST depression (oldpeak), and exercise-induced angina (exang). These attributes consistently appeared as the most impactful because they directly reflect the heart's functional response to stress, making them highly predictive in supervised models. Interestingly, cholesterol (chol) showed only a weak effect on model performance, highlighting that real-time stress-related measurements contributed far more to accurate predictions than traditional long-term risk factors.

Overall, this study demonstrates that machine learning, particularly ensemble methods like Random Forest, can significantly enhance early prediction of heart-disease risk. Such models can support healthcare professionals in identifying high-risk individuals more accurately and efficiently. This contributes directly to United Nations Sustainable Development Goal (SDG) 3: Good Health and Well-Being, which emphasizes early detection, prevention, and reduction of premature mortality from cardiovascular diseases.

6. References

[1] G. F. N. Berkelmans *et al.*, “Population median imputation was non-inferior to complex methods for handling missing values in cardiovascular risk prediction models,” *Eur. Heart J. Qual. Care Clin. Outcomes*, vol. 8, no. 3, pp. 298–306, 2022.
Available: https://pubmed.ncbi.nlm.nih.gov/35066115/

[2] S. K. Kwak and Y. H. Kim, “Statistical data preparation: management of missing values and outliers,” *J. Adv. Nurs.*, vol. 73, no. 6, pp. 1329–1339, 2017.
Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC5548942/

[3] J. Hansen, “Evaluations of statistical methods for outlier detection when applied to clinical registry data: a review,” *Clin. Epidemiol.*, vol. 15, pp. 45–60, 2023.
Available: https://pmc.ncbi.nlm.nih.gov/articles/PMC10351235/

[4] R. Garg, “Different Types of Classification Algorithms,” *Analytics India Magazine*, 20 Aug 2024.
Available: https://analyticsindiamag.com/topics/types-of-classification-model/

[5] A. Bonnet, “Accuracy vs. Precision vs. Recall in Machine Learning: What Is the Difference?,” *Encord Blog*, 23 Nov 2023.
Available: https://encord.com/blog/classification-metrics-accuracy-precision-recall/

[6] “Heart Disease Dataset,” *Kaggle Datasets*, 2023.
Available: https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction

[7] World Health Organization (WHO), “Cardiovascular diseases (CVDs),” 2023.
Available:
https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

[8] L. Breiman, “Random forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[9] D. W. Hosmer and S. Lemeshow, *Applied Logistic Regression*, Wiley, 2000.

[10] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.
Available: https://doi.org/10.1016/j.patrec.2005.10.010