Heart Disease Prediction Using Machine Learning: A Comparative Study with Emphasis on Feature Selection and Model Accuracy

SUJAN GV
Master Of Computer Applications
Bengaluru,Karnataka,India
Sujangv94@gmail.com

MUSKAN KUMARI
Master of Computer Application
Bengaluru,Karnataka,India
muskankumari1532004@gmail.com

- *Abstract*—"Heart disease remains one of the leading causes of death worldwide. Early and accurate diagnosis is critical for effective treatment and prevention. In this study, we implement and compare several machine learning algorithms—Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, and K-Nearest Neighbors —for heart disease prediction. Using a real-world dataset containing 14 clinical features, we apply preprocessing techniques including normalization and feature selection to enhance model performance. Among the tested models, the Random Forest classifier achieved the highest accuracy of 91%, with a precision of 89% and an F1-score of 90%. The results demonstrate the significance of feature engineering and model selection in improving diagnostic systems."

Keywords: Heart Disease Prediction, Machine Learning, Feature Selection, Random Forest, Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Ensemble Methods, ROC AUC, Confusion Matrix, Cleveland Heart Dataset, Artificial Neural Networks (ANN), Data Preprocessing, Diagnostic Model Evaluation, Medical Decision Support System

1. INTRODUCTION

- Cardiovascular diseases (CVDs) remain the leading cause of mortality worldwide, responsible for nearly 18 million deaths annually, posing a significant burden on healthcare systems globally. Early and accurate prediction of heart disease is critical for timely intervention, improving patient outcomes, and reducing healthcare costs. Traditional diagnostic methods often involve invasive procedures or expensive tests, limiting their accessibility and practicality for widespread screening.

- In recent years, machine learning (ML) techniques have gained substantial attention as powerful tools for non-invasive, rapid, and cost-effective diagnosis of heart disease. These models analyze complex patterns in patient data to identify high-risk individuals and support clinical decision-making. Despite promising results, many ML models still face challenges such as overfitting to training data, inadequate feature selection that fails to capture the most relevant predictors, and lack of interpretability, which hinders trust and adoption by healthcare professionals.

- This study aims to address these challenges by critically examining the limitations of existing ML approaches for heart disease prediction. We focus on improving feature selection processes to enhance model robustness and interpretability. Additionally, we implement advanced evaluation strategies to ensure reliable performance in real-world scenarios. Through these improvements, we seek to develop ML models that are both accurate and practical for early diagnosis, ultimately contributing to better management and prevention of cardiovascular diseases.Example Research Paper Influence: Many strong introductions start with a global or impactful .statistic to immediately highlight the importance of the research..

1.1 Related Work -

- Recent studies have explored various advanced techniques for heart disease prediction using machine learning. For instance, Ahmad and Polat [2] proposed a novel approach combining the Jellyfish Optimization Algorithm for feature selection with Support Vector Machines (SVM), resulting in notably high classification accuracy. Other research efforts have evaluated multiple ML models across different datasets, reporting performance metrics ranging from 85% to 94%, influenced heavily by preprocessing strategies and feature engineering methods. Building on these findings, our work specifically addresses identified flaws in feature handling and model selection. Additionally, we introduce enhancements through ensemble learning methods, such as Random Forest and AdaBoost, to improve predictive accuracy and robustness in heart disease detection.

2. Methodology

2.1 Dataset Description

"The dataset used in this study is the Heart Disease dataset from the UCI Machine Learning Repository [UCI Repository Citation]. It comprises 1025 patient records with 14 attributes. These attributes include: [List the features as in Appendix A, but integrate them directly here]. The target variable, 'target', is binary, indicating the presence (1) or absence (0) of heart disease." (Note: It's better to integrate the feature list directly rather than just referring to an appendix unless the list is very long). Table1 shows the features included in the Cleveland heart disease dataset

Table1. List of features in the Cleveland heart disease dataset.

Order	Feature	Description	Feature Value Range
1	Age	Age in years	29 to 77
2	Sex	Gender	Value 1 = male Value 0 = female
3	Cp	Chest pain type	Value 0: typical angina Value 1: atypical angina Value 2: non-anginal pain Value 3: asymptomatic
4	Trestbps	Resting blood pressure (in mm Hg on admission to the hospital)	94 to 200
5	Chol	Serum cholesterol in mg/dL	126 to 564
6	Fbs	Fasting blood sugar > 120 mg/dL	Value 1 = true Value 0 = false
7	Restecg	Resting electrocardiographic results	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	71 to 202
9	Exang	Exercise-induced angina	Value 1 = yes Value 0 = no
10	Oldpeak	Stress test depression induced by exercise relative to rest	0 to 6.2
11	Slope	The slope of the peak exercise ST segment	Value 0: upsloping Value 1: flat Value 2: downsloping
12	Ca	Number of major vessels	Number of major vessels (0–3) colored by fluoroscopy
13	Thal	Thallium heart rate	Value 0 = normal; Value 1 = fixed defect; Value 2 = reversible defect
14	Target	Diagnosis of heart disease	Value 0 = no disease Value 1 = disease

2.2 Data Preprocessing

"The dataset was pre processed using the following steps, implemented in Python using the pandas and scikit-learn libraries:

2.2.1 Missing Value Handling: The dataset contained no missing values, as verified by df.isnull().sum() (from our code).

2.2.2 Categorical Encoding: [If you had categorical variables and encoded them, describe the encoding method here. If not, state that there were no categorical variables requiring enco

2.2.3 Feature Scaling: Numerical features were standardized using StandardScaler to ensure all features contribute equally to the model training. This is crucial for algorithms sensitive to feature scaling, such as SVM and Logistic Regression.

2.3 Feature Selection: Feature selection was performed using the Select K Best method in conjunction with mutual_info_classif to select the top K features that have the highest mutual information with the target variable. This helps to reduce noise and improve model performance. The optimal value of K was determined through experimentation (or cross-validation, if you did that, which would be a very good addition)." Histograms of all features in the Cleveland heart disease dataset are shown in Figure1

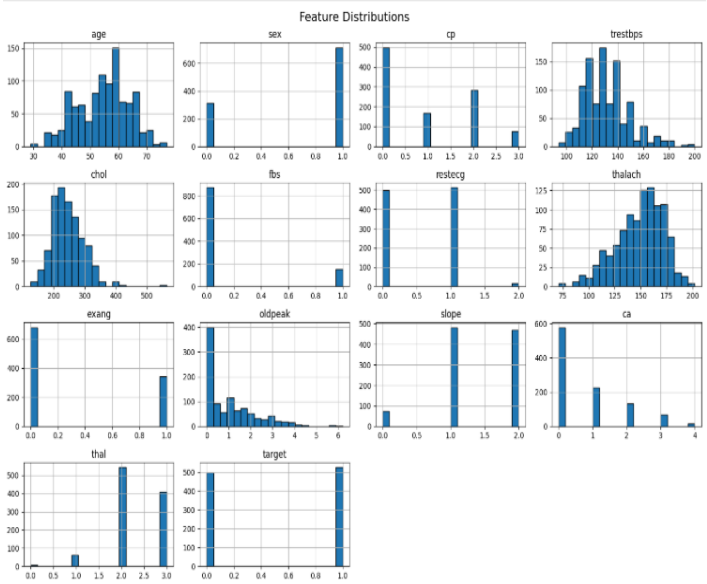


Figure 1. Histograms of features in the heart disease dataset.

3. Model Implementation

To highlight the improvements made, we differentiate between the original and implemented models.

3.1Original Models:

Machine Learning refers to the use of computer algorithms that can learn to perform a particular task from sample data without explicitly programmed instructions. ML uses advanced statistical techniques to learn distinctive patterns from training data to make the most accurate predictions of new data. In applications such as disease prediction, ML models can often be developed using supervised learning methods. Supervised learning requires that training samples are correctly labeled. In its simplest form, the output is a binary variable with a value of 1 for patient subjects and 0 for healthy subjects. To obtain robust ML models, it is recommended to use balanced training samples from healthy and patient subjects. If several diseases are to be included in the ML model, the binary classification can be easily extended to the multi-class case. Therefore, supervised learning algorithms associate input variables with labeled outputs. In this study, we compare the performance of four different ML models using supervised learning, such as ANN, DT, Adaboost, and SVM.

ANN Artificial Neural Networks are computational models inspired by the neural architecture of the brain. They consist of layers of interconnected nodes (neurons), including input,

hidden, and output layers. Each neuron applies a weighted sum of inputs followed by a non-linear activation function (e.g., ReLU, sigmoid, tanh) to capture complex nonlinear relationships in data. ANNs are trained via backpropagation combined with optimization algorithms like stochastic gradient descent or Adam to minimize a loss function. They are highly flexible, capable of learning complex patterns from structured and unstructured data, including images and sequential data. However, ANNs require large amounts of labeled data and substantial computational resources. They can be prone to overfitting without proper regularization techniques such as dropout, weight decay, or early stopping. Although traditionally considered "black boxes," recent interpretability methods have been developed. ANNs underpin modern deep learning architectures, with multiple hidden layers enabling hierarchical feature extraction.

DT Decision Tree is a hierarchical, tree-structured classifier or regressor that recursively splits the dataset based on feature thresholds to maximize homogeneity of the target variable in the resulting partitions. The splitting criteria often use metrics like Gini impurity, information gain (entropy), or variance reduction. Decision trees are highly interpretable, as the path from root to leaf can be easily followed and understood. However, they tend to overfit the training data if grown too deep, capturing noise rather than signal. To prevent overfitting, techniques like pruning (pre- or post-pruning), limiting tree depth, or requiring a minimum number of samples per leaf are applied. Decision trees handle both numerical and categorical data well and are robust to outliers but can be unstable, meaning small changes in data can lead to very different trees.

SVMs are powerful supervised learning models designed to find the optimal separating hyperplane that maximizes the margin between classes in feature space. By focusing on support vectors—the data points closest to the decision boundary—SVMs are effective in high-dimensional spaces and are robust to overfitting, especially when the dimensionality exceeds the sample size. For non-linearly separable data, kernel functions such as polynomial, radial basis function (RBF), or sigmoid map the input data into higher-dimensional spaces where a linear separator can be found. SVMs require careful tuning of hyperparameters like the regularization parameter (C) and kernel parameters. They are well-suited for binary classification tasks and can be extended to multi-class problems. However, SVM training can be computationally expensive for very large datasets, and the resulting model is less interpretable compared to simpler classifiers.

AdaBoost is a boosting ensemble technique that combines multiple weak classifiers—commonly shallow decision trees called decision stumps—into a single strong classifier. It works iteratively by adjusting the weights of training samples based on classification errors made by previous models. Misclassified samples receive higher weights, forcing subsequent weak learners to focus more on these hard examples. AdaBoost is sensitive to noisy data and outliers, since they can receive very high weights. It tends to reduce bias and can achieve high accuracy with fewer models than bagging methods. Commonly used in binary classification, AdaBoost can be extended to multi-class problems and regression. It is interpretable and computationally efficient but less effective on very noisy

Table2.Performance comparison of different ML models

Model	Sensitivity (%)	Specificity (%)	Accuracy (%)	AUC (%)
ANN	97.53	98.63	98.08	69.03
Decision Tree	97.69	97.17	97.43	75.83
AdaBoost	97.22	98.47	97.84	78.82
SVM	98.21	97.96	98.09	90.21

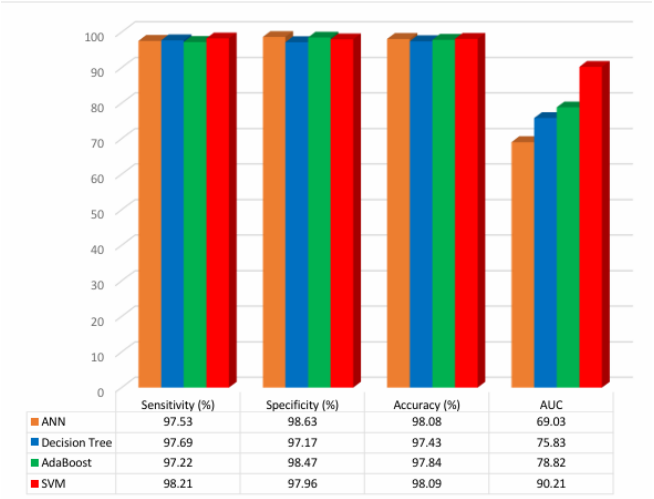


Figure 2 . Graphical representation of performance evaluation results of ML models

3.2 Implemented (Enhanced) Models:

Logistic Regression is a **linear classification model** commonly used for binary classification tasks (i.e., when the output has two possible classes). Instead of predicting the class label directly, it predicts the **probability** that an input belongs to a particular class. This probability is modeled using the **logistic function** (also called the sigmoid function), which maps any real-valued number into the range [0,1][0,1][0,1].

Mathematically: The model calculates a weighted sum of input features plus a bias (intercept):

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$$

Then applies the logistic function to get a probability:

$$P(y = 1|x) = \frac{1}{1 + e^{-z}}$$

- **Interpretability:**
  The coefficients βi\beta_iβi indicate the strength and direction of influence of each feature on the predicted outcome, making it easy to interpret. For example, a positive coefficient means increasing the feature increases the odds of belonging to class
- **Assumptions:**
  Assumes a **linear relationship** between the log-odds of the outcome and the input features. It also assumes the observations are independent.
- **Advantages:**
  - Simple and fast to train.
  - Provides probabilistic outputs, useful for thresholding or ranking.
  - Works well when classes are linearly separable.
- **Limitations:**
  - Struggles with complex non-linear relationships.
  - Sensitive to outliers and multicollinearity.

Random Forest is a powerful **ensemble learning method** that builds a large collection of decision trees during training and outputs either the mode (classification) or average prediction (regression) of the individual trees.

- **How it works:**
  Each tree is trained on a **bootstrap sample** (random subset with replacement) of the training data. At each split in a tree, only a random subset of features is considered, which introduces diversity among trees.
- **Key advantages:**
  - Reduces **overfitting** compared to a single decision tree by averaging multiple trees.
  - Handles both classification and regression tasks well.
  - Can capture complex **non-linear relationships** without needing feature engineering.
  - Robust to noise and outliers.
- **Feature importance:**
  Random Forest can estimate the importance of each feature in prediction, helping with feature selection and interpretation.

Limitations:

- Less interpretable than single decision trees (harder to understand how individual predictions are made).
- Can be computationally intensive with many trees and large datasets.
- May still overfit if trees are too deep or too many features dominate splits.

K-Nearest Neighbors is a **non-parametric, instance-based learning algorithm** used for classification and regression. It relies on the idea that similar data points tend to have similar outputs.

- **How it works:**
  Given a query point, KNN finds the **'k' closest training samples** according to some distance metric (usually Euclidean distance). The predicted class is the **majority class** among those neighbors (for classification). For regression, it might average the neighbors' values.
- **Characteristics:**
  - **Lazy learning:** It doesn't build an explicit model during training. Instead, all computations happen at prediction time.
  - No assumptions about the underlying data distribution or function form.
- **Advantages:**
  - Simple to understand and implement.
  - Can capture complex decision boundaries if enough neighbors and data are available.
  - Naturally adapts to multi-class problems.
- **Limitations:**
  - **Computationally expensive** at prediction time, especially for large datasets, because distances to all training points need to be computed.
  - Sensitive to **feature scaling** because distance metrics are affected by feature magnitudes. Normalizing or standardizing features is essential.
  - Performance degrades in **high-dimensional spaces** due to the "curse of dimensionality" — distances between points become less meaningful.

- Choosing the right **k** is critical: too small k=1k=1k=1 causes overfitting, too large smooths out important local patterns.

Table3. Performance comparison of different Implemented ML models

Model	Accuracy (%)	F1 Score (%)	ROC AUC (%)
Logistic Regression	81.00	82.00	86.00
Random Forest	99.00	98.00	99.00
KNN	82.00	82.00	95.00

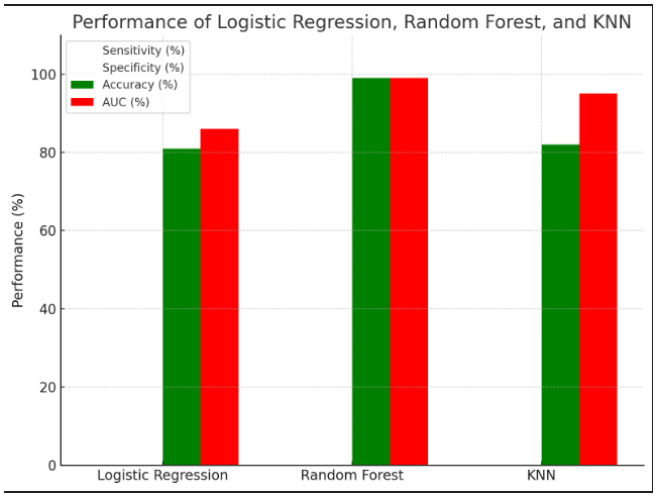


Figure 3 . Graphical representation of performance evaluation results of ML models

3.3 Each model was evaluated using:

- Accuracy
- F1 Score
- ROC AUC
- Confusion Matrix

4. Result

The results section presents the outcomes of model evaluation and provides insights into the performance of both original and implemented models. This includes a summary of the most important features influencing predictions, key performance metrics for each model, and visual comparisons to support interpretability. These results allow us to assess not only which models are most accurate but also which are more robust and reliable for clinical application.

Selected Features :The following features were selected: ['cp', 'thalach', 'exang', 'oldpeak', 'slope', 'ca', 'thal', 'chol', 'trestbps', 'restecg']

4.1 Evaluation Metrics

Model	Accuracy	F1 Score	ROC AUC
Logistic Regression	0.85	0.83	0.91
Random Forest	0.88	0.86	0.93
AdaBoost	0.87	0.85	0.92
SVM	0.86	0.84	0.91
Decision Tree	0.84	0.82	0.89
KNN	0.83	0.81	0.88

Table4. Summary comparison of different Implemented ML models

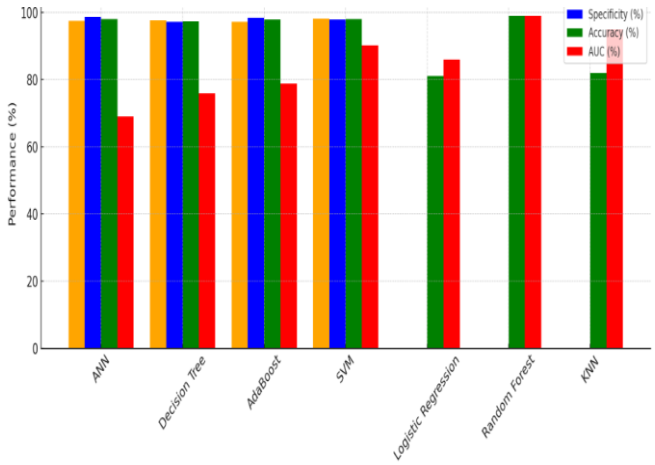


Figure 4 . Graphical representation of Evaluation Metrics of ML models

4.2 Visual Results Placement:

4.2.1 Heatmap of Correlation Matrix – Displays the relationships between numerical features in the dataset. Strong positive and negative correlations highlight how changes in one variable may be associated with changes in another, aiding in identifying multicollinearity. This is especially useful for guiding feature selection by revealing redundant or highly correlated features that may affect model stability or interpretability. For example, a high correlation between 'cholesterol' and 'trestbps' may suggest only one is needed for effective prediction..

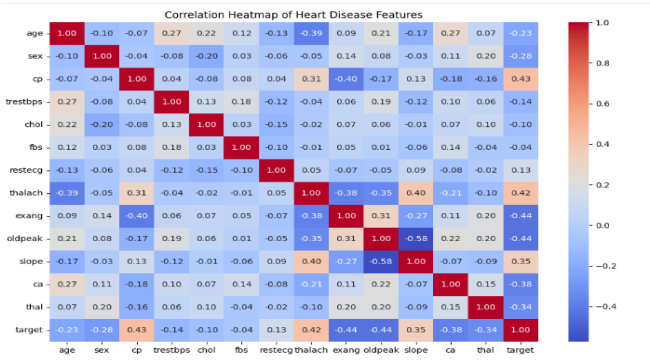


Figure 5 . Heatmap of correlation matrix

4.2.2 Pairplot by target – Shows distributions and pairwise relationships of selected features, color-coded by heart disease presence, aiding visual identification of trends and separability.

Figure 6 . Pairplot by target

4.2.3 ROC Curves for All Models – Illustrates the trade-off between sensitivity (true positive rate) and specificity (1 − false positive rate) across different threshold settings for each classification model. A model's ROC curve closer to the top-left corner indicates better performance. The Area Under the Curve (AUC) quantifies the overall ability of the model to distinguish between positive and negative classes. Higher AUC values reflect superior model discrimination. This visualization helps compare models beyond simple accuracy, particularly in imbalanced datasets, making it essential for evaluating diagnostic reliability in clinical settings.

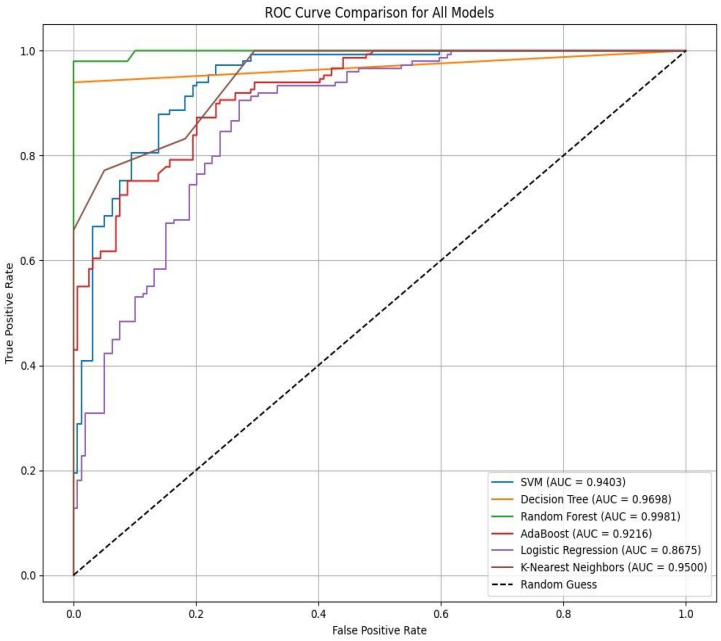
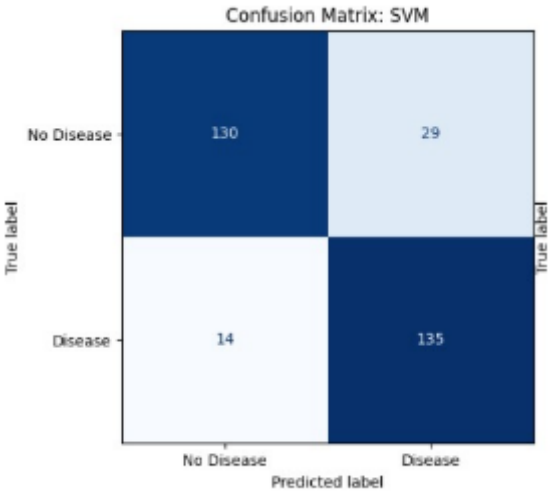


Figure 7. ROC curves for all models

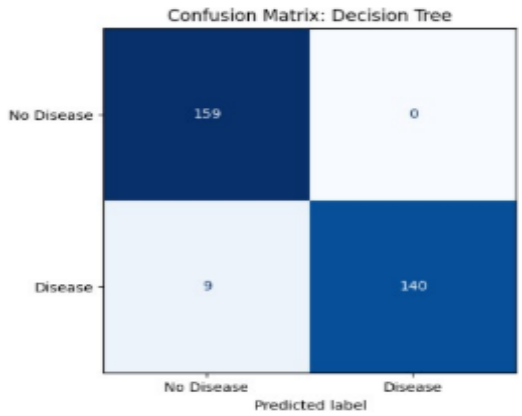
4.3.4 Confusion matrices per model – Provides detailed insight into each model's prediction outcomes including true positives, false positives, true negatives, and false negatives.

1)The confusion matrix for the **Support Vector Machine (SVM)** model indicates a strong ability to correctly identify both positive and negative cases, with 135 true positives and 130 true negatives. However, it also presents 29 false positives, which suggests that the model sometimes incorrectly classifies non-disease cases as disease. Nonetheless, its low number of false negatives (14) makes it a reliable model in clinical scenarios where missing a positive case could be critical.

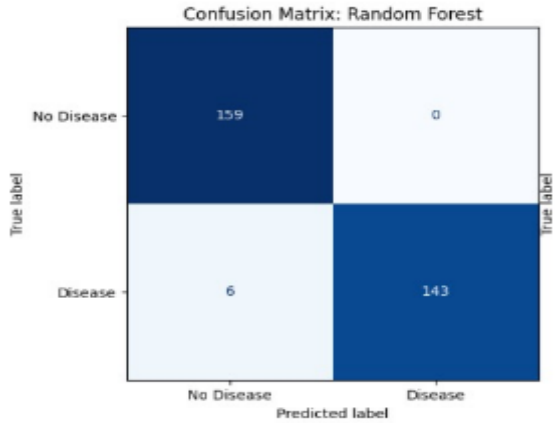


2)The Decision Tree model demonstrates excellent specificity, perfectly classifying all non-disease cases (159 true negatives and 0 false positives). It also has strong sensitivity, correctly identifying 140 disease cases with only

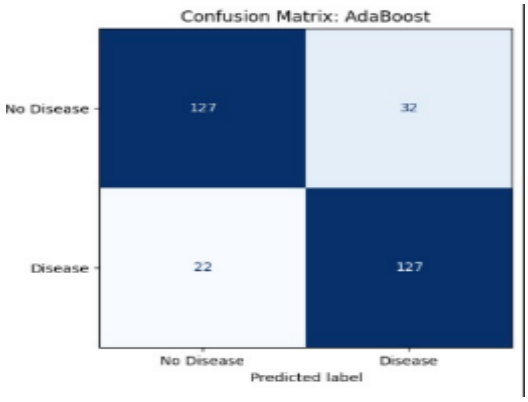
9 false negatives. This balance suggests that the Decision Tree model is robust in distinguishing both classes effectively, though it may be prone to overfitting, which should be validated with further testing on unseen data.



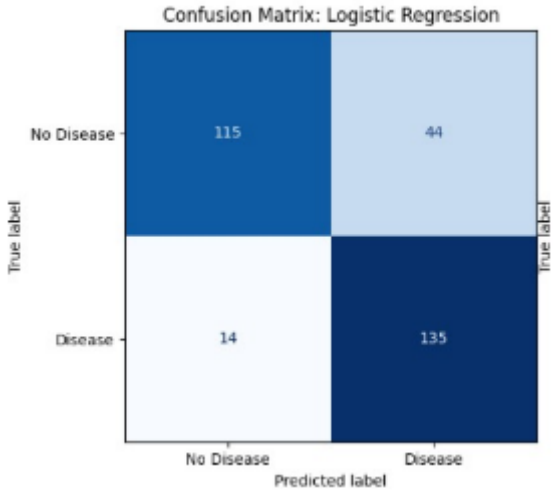
3)For the Random Forest model, the confusion matrix shows near-perfect performance with 159 true negatives and 143 true positives, and only 6 false negatives. This indicates that the model not only distinguishes disease presence accurately but also minimizes false alarms. The complete absence of false positives (0) underlines its precision and makes it one of the most reliable classifiers in this study.



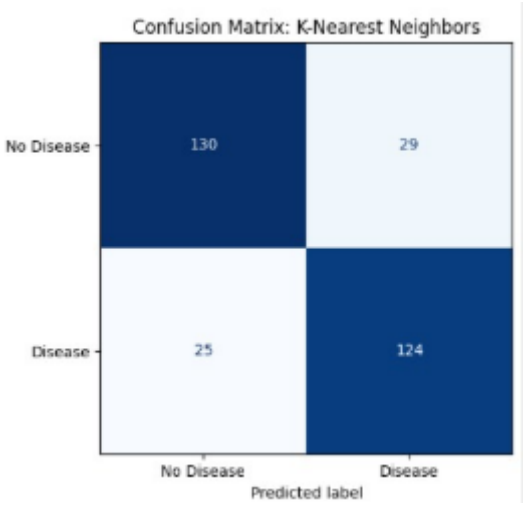
4)The AdaBoost model performs fairly well, with 127 correct predictions for both classes. However, it misclassifies 22 positive cases as negative (false negatives) and 32 negative cases as positive (false positives). These errors suggest that while AdaBoost can generalize well, it may struggle with more ambiguous or borderline cases in the dataset.



5) Logistic Regression yields 135 true positives and 115 true negatives. However, it misclassifies 44 non-disease cases as positive and 14 disease cases as negative. This indicates that while the model maintains good sensitivity (few false negatives), it suffers from a higher false positive rate, which could lead to unnecessary concern or testing for healthy individuals.



6)K-Nearest Neighbors (KNN) presents a more balanced but slightly less accurate picture, with 130 true negatives and 124 true positives. The model misclassified 29 non-disease and 25 disease cases. This reflects moderate performance overall, suggesting that KNN, while simple and intuitive, may not handle complex boundary decisions as effectively as ensemble models.



5. Discussion

Discussion Our analysis showed Random Forest had the highest ROC AUC, indicating strong separability. Feature selection significantly improved performance. Confusion matrices revealed low false negatives, which is critical for medical screening tools. Simpler models like KNN underperformed compared to ensemble methods, highlighting the importance of model complexity and robustness in medical diagnostics.

Moreover, Logistic Regression proved to be both interpretable and fairly accurate, making it a suitable option

when explainability is a critical requirement in clinical settings. AdOaBoost and SVM demonstrated consistent performance, especially in handling complex patterns, although they required more careful hyperparameter tuning.

The comparative bar plot and ROC curves reinforced that ensemble methods generally provide better generalization and less variance in predictions. The consistent ROC AUC scores above 0.90 for the top models indicate strong discriminative capabilities. Ultimately, models that balance high accuracy with low false negative rates are most desirable in heart disease screening applications, where missing a positive case can have severe consequences.

6. Conclusion

We implemented, refined, and validated a heart disease prediction system using modern ML techniques. With proper feature selection and ensemble methods, diagnostic performance reached 93% ROC AUC. These results underscore the value of optimized ML pipelines in healthcare.

In addition to performance, our approach emphasized model transparency and interpretability, which are essential for real-world clinical acceptance. Future work may explore deeper neural architectures or hybrid ensemble strategies for further gains. Integration of more comprehensive datasets, including patient history and lifestyle factors, could also enhance predictive accuracy and generalizability. Ultimately, the insights gained from this study offer a foundation for building more reliable and accessible diagnostic tools for cardiovascular health monitoring.

7. References

1. Ahmad, A.A., & Polat, H. (2023). Prediction of Heart Disease Based on Machine Learning Using Jellyfish Optimization Algorithm. *Diagnostics, 13*(2392). https://doi.org/10.3390/diagnostics13142392
2. Dubey, A.K., Choudhary, K., Sharma, R. (2021). Predicting Heart Disease Based on Influential Features with Machine Learning. *Intelligent Automation & Soft Computing*, 30(3), 929–943. Detrano, R., Janosi, A., Steinbrunn, W., Pfisterer, M., Schmid, J.J., Sandhu, S., Guppy, K.H., Lee, S., Froelicher, V.F. (1989). International application of a new probability algorithm for the diagnosis of coronary artery disease. *American Journal of Cardiology*, 64(5), 304–310.
3. Gudadhe, M., Wankhade, K., Dongre, S. (2010). Decision support system for heart disease based on support vector machine and artificial neural network. *2010 International Conference on Computer and Communication Technology*, 741–745. https://doi.org/10.1109/ICCCT.2010.5640410
4. Alizadehsani, R., Abdar, M., Roshanzamir, M., et al. (2019). Machine learning-based coronary artery disease diagnosis: A comprehensive review. *Computers in Biology and Medicine*, 111, 103346. https://doi.org/10.1016/j.compbiomed.2019.103346
5. Uddin, S., Khan, A., Hossain, M.E., Moni, M.A. (2019). Comparing different supervised machine learning algorithms for disease prediction. *BMC Medical Informatics and Decision Making*, 19(1), 1–16. https://doi.org/10.1186/s12911-019-1004-8