JIRBDAI

Journal of Institutional Research, Big Data Analytics and Innovation

Comparison of Machine Learning Algorithms for Personalized Healthcare: A Data-Driven Approach to Predicting Heart Attack

Ismail Idowu Akuji*[1], Ronke Seyi Babatunde[2], Hamidat Ayodeji Arikewuyo[3], Muhammad Sharafadeen Jimoh[4], Jubril Alabi Ayodeji[5], and Idris Babatunde Adeyemi[6]

[1, 3, 4]Department of Computer Science, Abdulrasaq Abubakar Toyin University, Oke-Ogba, Kwara State, Nigeria
[2,5]Department of Computer Science, Kwara State University, Malete, Nigeria
[6]Department of Computer Science, University of Ilorin, Ilorin, Nigeria
Corresponding Author: akujiismaheel@gmail.com

ABSTRACT: Heart attack remains a leading cause of mortality worldwide; while traditional diagnostics like electrocardiography (ECG) and clinical evaluation have reduced death rates, they struggle with complex data and real-time insights, prompting machine learning (ML) for prediction and explainability. This study compares three ML algorithms, k-nearest neighbor (KNN), decision tree (DT), and random forest (RF), for heart-attack prediction. The dataset, combined from Cleveland, Hungarian, Switzerland, Long Beach VA, and Stalog sources (Kaggle), contains 918 instances and 12 features. Categorical variables were converted to numeric using label encoder. Recursive Feature Elimination (RFE), Mutual Information, and Random Forest Gini importance consensually identified ST_Slope, ChestPainType, ExerciseAngina, MaxHR, and Oldpeak as the top five features. GridSearchCV tuned hyperparameters, and SHAP provided model explainability. Results: RF (GridSearchCV) achieved accuracy 83.7 %, 84.0% precision, recall 83.7 %, F1-score 83.8 %; KNN (GridSearchCV) 83.3 % accuracy, 84.3 % precision, 83.3 % recall, 83.5 % F1-score; DT (GridSearchCV) 79.0 % accuracy, 81.0 % precision, 79.0 % recall, 79.2 % F1-score. ROC-AUC scores were RF 0.905, KNN 0.876, DT 0.770; McNemar tests (RF $p = 0.0008$) confirmed RF's edge with statistical similarity to KNN. SHAP waterfall highlighted ST_Slope and ChestPainType as strongest risk drivers. RF's ensemble nature explains its superiority and better explainability via SHAP, making it preferable for clinical decision support. Future work should involve multi-hospital cross-validation, additional variables (labs, medications), hybrid methods, and more diverse datasets.

KEYWORDS: Heart Attack, Heart Disease, Machine Learning, Decision Tree, K-Nearest Neighbor, Random Forest.

I. INTRODUCTION

In recent times, humans have lost their lives owing to heart attacks, as heart disease is considered one of the most prevalent causes of mortality globally. The World Health Organization reported that an estimated 12 million people died from heart diseases worldwide each year [1], representing 70% of all global deaths can be traced to hear disease [2]. Heart attack occurs as a result of the reduction or blockage of blood flow to the coronary artery for a period of time resulting in myocardial ischemia, leading to heart disease. Several factors contribute to the growth of heart

Volume 1, Issue 4, 2025
https://doi.org/10.5281/zenodo.17945075

disease, these includes the smoking, poor nutrition, high blood pressure, and sedentary lifestyles [3].

Traditionally, heart disease is diagnosed via clinical devices such as electrocardiography (ECG) and blood tests. Although this method is effective in identifying the vulnerable people to heart attacks. However, these mechanisms are time consuming, expensive, and demanding for both patients and clinicians. Recently, the healthcare system is drastically shifting from reactive responses to dealing with intermittent situations using an active methodology wherein patients are characterized by early detection, prevention, and better management in healthcare [4]. For this reason, the healthcare system and human wellbeing control become paramount and plausible for human fitness care. Machine learning (ML) offers a promising approach to enhancing the accuracy and efficiency of heart attack predictions by leveraging a wide array of risk factors and their interconnections by employing historical-related data. Medical organizations all around the world collect data on various health-related issues; these data can be exploited using various machine learning techniques to gain useful insights [5]. While several machine learning algorithms (such as K-Nearest Neighbor, Decision Tree, Random Forest among others) have shown promise in heart-attack prediction. However, identifying the most robust algorithm requires comparative study of these common models to determine which perform bests. This study, as a result, performs comparative to affirm the efficacy of the application of machine learning in this domain. However, there's need for justifying the most robust and promising algorithm through the conduct of comparative investigation of the commonly employed algorithms. Therefore, this study compares the three most performing algorithms (k-nearest neighbor, decision tree, and random forest) for heart-attack prediction on similar dataset and improves existing models by revealing the most important features for identifying vulnerable individuals.

II. LITERATURE REVIEW

The K-Nearest Neighbor (KNN), Random Forest (RF), and Decision Tree (DT) have been widely affirmed as the best performing algorithm across diverse studies. This is owing to their owing to their balance of interpretability, robustness to noise, and relatively low computational demands. The following sub-sections review relevant literatures on the three algorithms investigated highlighting their applications, performance metrics, and recent advancements in heart-attack prediction.

A. K-Nearest Neighbor (KNN): Evidence of Superior Performance in Recent Studies

Authors in [6] developed a system to predict heart attack using the UCI repository dataset. The study employed logistic regression and KNN and found that KNN achieved an accuracy score of 88.52% to outperform logistic regression, which achieved an accuracy score of 85%. The model based on the accuracy score is still open for further improvement. However, this study fails to perform feature selection which can offer better understanding of the major factors that contribute to heart attack. Authors in [7] employed machine learning techniques to predict heart disease using the K-Nearest Neighbor (KNN) algorithm. The model achieved the highest accuracy of 90.761% when KNN is configured with 7 neighbors. This underscores the criticality of careful distance metric selection in heart disease prediction models. While the studies show promising accuracy score and the efficacy of the KNN algorithm, the lack of important feature identification weaken their reliability and clinical relevance. Addressing those points would greatly improve credibility [8] developed machine learning-based model to detect heart disease early and adequately. A weighted *K*-nearest neighbor model using feature scores is used and the relief is employed for

Volume 1, Issue 4, 2025
https://doi.org/10.5281/zenodo.17945075

feature selection to determine the scores, which will be used as weights. This study is applied to five aggregated datasets from IEEE Data Port or Kaggle, including Hungarian, Cleveland, Long Beach VA, Switzerland, and Statlog (Heart). The result of the study shows that the weighted KNN has outperformed all other algorithms used in this study for comparison across all measures having achieved the best accuracy of 93.28%.

B. Decision Tree (DT): Evidence of Superior Performance in Recent Studies

[9] employed Classification and Regression Tree (CART) algorithm, a variant of decision tree, to predict heart disease and extract decision rules in clarifying relationships between input and output variables. The model achieved a 87% accuracy of the prediction, validating the model's reliability. Both studies present promising accuracy scores to affirm the performance of decision tree-based algorithms, however, the studies failed to perform feature selection which is vital for any clinical-decision support tool. [10] proposed a squirrel search-optimized Gradient Boosted Decision Tree (SS-GBDT) for the detection of heart disease. The results show that the model is highly efficient in medical diagnosis having achieved a 95%, 95.8%, 96.8%, and 96.3% accuracy, precision, recall and F1-score.

C. Random Forest (RF): Evidence of Superior Performance in Recent Studies

Authors in [11] presented a comparative analysis of machine learning techniques (namely Random Forest [RF], Logistic Regression [LR], Support Vector Machine [SVM], and Naïve Bayes [NB]) in the classification of cardiovascular disease. The study indicated that RF has proven to be the most accurate and reliable algorithm with an accuracy of 84.81% as compared to LR (83.83%), SVM using linear kernel function (74.05%), SVM using radial base kernel function (58.577%), and Naïve Bayes (54.08%). Authors in [12] proposed machine learning (ML) for risk factor analysis and survival prediction of heart failure (HF) patients using a survival dataset. Five supervised ML methods were applied, including XGBoost and Gradient Boosting (GB) algorithms. The authors compare the applied algorithms' performances based on accuracy, precision, recall, F-measure, and log loss value and show RF provides the highest accuracy of 97.78%. The inability of the boosting techniques employed here to outperform other algorithms might be due to the nature of the data. Authors in [13] applied decision tree (DT), random forest (RF), logistic regression (LR), Naïve Bayes (NB), and support vector machine (SVM) for the classification and prediction of cardiovascular (heart disease) patients in Pakistan. The RF algorithm had the highest accuracy of prediction, sensitivity, and recursive operative characteristic curve of 85.01%, 92.11%, and 87.73%, respectively, for CVD. These results indicated that the RF algorithm is the most appropriate algorithm for CVD classification and prediction. All the studies show the robustness of RF, though, lack of important features identification limit confidence and clinical usefulness.

D. Other Algorithms: Evidence of Superior Performance in Recent Studies

Authors in [14] proposed a novel model for detecting emergency readmission of heart disease patients using a robust Stacking Ensemble Learner (SEL) to maximize the detection performance. To ensure robustness and high accuracy in the prediction results across multiple runs, the study employed XGBoost as a meta-learner in the SEL model. The study's experimental results show that the stacking model provides high accuracy, recall, and F1 score compared to the baseline models such as logistic regression, k-nearest neighbor, decision tree, random forest, support vector machines, bagging, and boosting. The SEL model has achieved an accuracy of 88% in predicting

https://doi.org/10.5281/zenodo.17945075

emergency readmission of heart-disease patients, which is very promising for the production-ready model in clinical practice. Authors in [15] developed a model based on supervised learning techniques, RF, DT, and LR using an existing dataset from the UCI Cleveland database to predict heart disease in patients, containing 303 occurrences and 76 characteristics. Only 14 of these 76 characteristics are evaluated for testing, which is necessary to validate the performance of various methods. The findings show that logistic regression achieves the best accuracy score (92.10%). Rindhe et al. (2021) developed models to predict heart disease using ANN, RF, and SVM. The models were trained and tested with the maximum percent of the training set and resulted in 84.0%, 83.5%, and 80.0% for SVM, ANN, and RF, respectively. Hence, support vector machines obtained the best and optimal accuracy result. Authors in [16] carried out research on heart disease from a data analytics point of view. The study applied three data analytics techniques (neural networks, SVM, and KNN) to datasets of different sizes in order to study the accuracy and stability of each of them. The study found neural networks are easier to configure and obtain much better results (accuracy of 93%). [3] applied the Jordan University Hospital (JUH) Heart Dataset to develop heart disease predictive model using random forest, SVM, decision tree, naive Bayes, and K-nearest neighbours (KNN) and particle swarm optimization (PSO) for feature selection. This study shows that SVM with PSO achieved the best classification accuracy of 94.3%. The study contributes significantly to improving heart disease diagnosis in Jordan, where cardiovascular diseases are a leading cause of mortality.

III. METHODOLOGY

A. Framework for the Proposed Models

Figure 1 depicts a workflow diagram that guide the development of the proposed model. It starts with Data Acquisition & Preprocessing, exploratory analysis, feature selection, model development, and model evaluation.

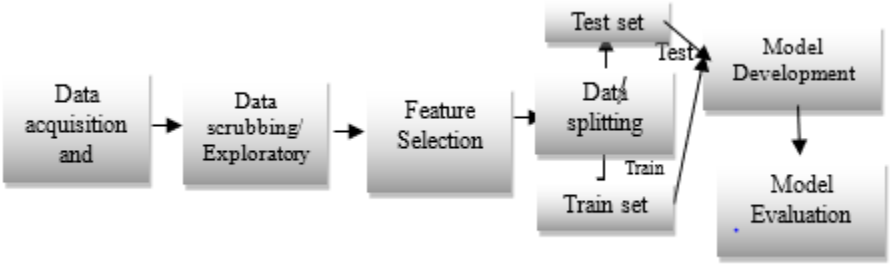


Figure 1: Framework of the Proposed Heart Disease Prediction Model

B. Data Acquisition and Preprocessing

The heart failure prediction dataset used in this paper was obtained from the Kaggle repository (https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction) and imported into the Jupyter Notebook environment for analysis. This dataset is made up of combined five heart disease datasets, namely Cleveland, Hungarian, Switzerland, Long Beach VA, and Stalog, resulting in 918 instances and 12 features relevant to cardiovascular disease prediction. Table 1 provides descriptions of the dataset features along with their respective clinical relevance.

Volume 1, Issue 4, 2025
https://doi.org/10.5281/zenodo.17945075

Table 1: Heart Failure Dataset Features and Medical Interpretation

Feature	Type	Description	Medical Interpretation
Age	Numeric (years)	Age of the patient	Older age significantly increases cardiovascular disease risk due to cumulative vascular damage.
Sex	Categorical	Male (M), Female (F)	Males typically exhibit higher heart disease rates; females' risk rises post-menopause.
ChestPainType	Categorical	Typical Angina (TA), Atypical Angina (ATA), Non-Anginal Pain (NAP), Asymptomatic (ASY)	Indicates ischemia severity; typical angina strongly correlates with coronary artery disease.
RestingBP	Numeric (mm Hg)	Resting blood pressure	Hypertension (>140/90 mmHg) strains the heart.
Cholesterol	Numeric (mg/dl)	Serum cholesterol level	Elevated levels (>200 mg/dl) promote atherosclerosis and plaque buildup in coronary arteries.
FastingBS	Binary	1 if fasting blood sugar > 120 mg/dl, else 0	High fasting blood sugar indicates diabetes or prediabetes, a cardiovascular risk factor
RestingECG	Categorical	Normal, ST-T abnormality (ST), Left ventricular hypertrophy (LVH)	Measures heart electrical function abnormalities
MaxHR	Numeric	Maximum heart rate achieved	Lower max heart rate can indicate impaired cardiovascular fitness
ExerciseAngina	Binary	Yes (Y), No (N)	Exercise-induced angina suggests myocardial ischemia
Oldpeak	Numeric	ST depression induced by exercise relative to rest (measured in mm)	Greater depression (>1 mm) correlates with multivessel coronary disease severity .
ST_Slope	Categorical	Up, Flat, Down	Slope of the ST segment during peak exercise relates to severity of coronary artery disease
HeartDisease	Binary (target)	1: presence of heart disease, 0: absence	Outcome variable for predictive modeling

Furthermore, this dataset is limited to specific hospitals but not population-wide. As a result, it may over-represent certain demographic and not capture all possible cardiovascular risk factors which could affect model generalizability. Lastly, ethical considerations regarding patient consent and anonymization in the source datasets are not detailed on Kaggle but were reportedly addressed by the dataset creators. The dataset contains 508 cases labeled as heart disease (55.6%) and 410 as normal (44.4%). Although, this distribution indicates a minor class imbalance favoring the positive class (heart disease (Figure 2), however, no resampling techniques (neither under sampling nor oversampling techniques) were applied during preprocessing, as the imbalance ratio ($1 \approx 1.24:1$) is moderate and

Volume 1, Issue 4, 2025
https://doi.org/10.5281/zenodo.17945075

unlikely to severely bias common classifiers like Random Forest. Instead, hyperparameter tuning via GridSearchCV with cross-validation mitigated mild imbalance effects on model evaluation by optimizing thresholds and parameters (e.g., class weights, n_estimators) to maximize balanced metrics like F1-score.

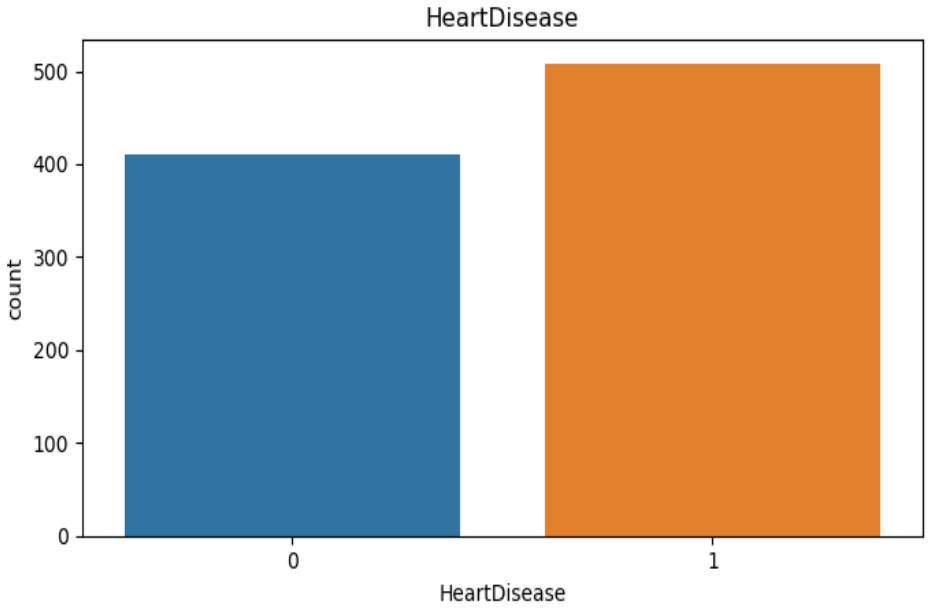


Figure 2: Class Distribution of Heart Disease Classes

During the preprocessing phase, missing values were checked, and no significant missing data were found. Categorical features were encoded and transformed to numerical values using label encoder (Figure 6). The features of the dataset were normalized using StandardScaler to ensure a uniform scale for model training. The dataset was split into training and testing sets using stratified sampling to maintain class distribution across subsets. These preprocessing steps enabled development and evaluation of the proposed heart disease prediction model.

C. Feature Selection Process

Feature selection using Recursive Feature Elimination (RFE) was performed by instantiating an RFE object with an estimator (Logistic Regression algorithm). Logistic Regression (LR) was chosen as the estimator for Recursive Feature Elimination (RFE) because it provides stable and interpretable coefficient-based rankings. LR-driven RFE effectively identifies features with strong linear relationships to the target, serving as a robust baseline for feature importance. This linear stability helps avoid overfitting common and ensures consistent elimination steps. Additionally, LR-RFE is computationally efficient and widely validated in domains like healthcare where some linear separability exists, making it a practical choice to guide initial feature selection before applying more complex models. This approach balances interpretability, efficiency, and generalizability across diverse classifiers. For consistency, tree-based feature importance (Random Forest Gini importance) and mutual information were also applied as complementary methods. This comparative analysis of the three feature selection techniques enabled the identification of the most important features across all the techniques. In addition, a bar chart visualized the feature importance scores and rankings across methods for intuitive comparison (Figures 8-10).

Volume 1, Issue 4, 2025
https://doi.org/10.5281/zenodo.17945075

D. Models Training and Validation Process

In this phase, data was split into 70 % training and 30 % testing with a fixed random state (42) for reproducibility. The parameters used to train the models are shown in Table 2, Tables 3 and 4.

Table 2: KNN Model's Parameters and Values

Parameter	Value	Description
n_neighbors	3	Number of neighbors to use for voting
Weights	"uniform"	All neighbors weighted equally
Algorithm	"auto"	Chooses best algorithm based on input data
P	2	Euclidean distance (Minkowski metric p=2)

The choice of setting *n_neighbors* = 3 strikes a balance between noise sensitivity (a low value of k) and excessive smoothing (high value of k); this is common practical choice for KNN. Using *'uniform'* for weights ensures that all neighbors contribute equally to the voting process. The *'auto'* algorithm choice allows the KNN model to select the most efficient method depending on the dataset's characteristics. Finally, the distance metric parameter p = 2 specifies the Euclidean distance, which is widely used and effective for continuous feature spaces, calculating straight line distance between pints. .

Table 3: Decision Tree Model's Parameters and Values

Parameter	Value	Description
Criterion	"gini"	Measures quality of split (default is "gini")
Splitter	"best"	Chooses the best split at each node
max_depth	None	Full growth allowed without restriction
min_samples_split	2	Minimum samples required to split a node
random_state	None	No fixed seed, so results can vary per run

For the Decision Tree model, default parameters were used to allow the tree to grow until all leaves are pure or contain fewer than the minimum samples split. This enables the model to freely adjust complexity. The use of default settings serves as a useful baseline to evaluate whether restricting tree depth or pruning is needed in subsequent experiments.

Volume 1, Issue 4, 2025
https://doi.org/10.5281/zenodo.17945075

Table 4: Random Forest Model's Parameters and Values

Parameter	Value	Description
n_estimators	100	Default number of trees in the forest
max_depth	2	Limits tree depth to control model complexity
random_state	0	Ensures reproducible results
Criterion	"gini"	Split quality measure
Bootstrap	True	Uses bootstrap samples for building trees

For the Random Forest model, the parameters used are listed in Table 4. The choice of *max_depth = 2* is to restrict the depth of each tree to reduce overfitting through the creation of simpler decision boundaries, which often improves generalization on on smaller datasets. Fixing random_state=0 ensures reproducibility of the ensemble construction. Other parameters such as *100 estimators* provide enough trees for stable bagging performance, *criterion = 'gini'* maintains consistency with the decision trees' split measure, and *boostrap=true* ensures each tree is trained on a different random sample of the dataset to enable increased diversity among trees.

E. Discussion of Hyperparameter Tuning with GridSearchCV

In this paper, grid search cross validation was applied across all the three algorithms to finetune their parameters accordingly. The chosen parameters for this process are depicted in Table 5, Table 6, and Table 7.

Table 5: Decision Tree Parameters Tuning

Parameter	Value	Best
param_grid	'max_depth': [3,5,7, None]	5
	'min_samples_split': [2,5,10]	1
	'min_samples_leaf': [1,2,4]	10
	'criterion': ['gini, entropy']	entropy
random_state	42	
cv	5	
scoring	accuracy	
n_jobs	1	

Table 5 indicates that entropy was selected as the split criterion instead of Gini, implying the model (decision tree) gained a slightly more discriminative information-gain metric. The other best hyperparameters found were max_depth = 5, min_samples_split = 1, and min_samples_leaf = 10. Additional settings: random_state = 42, 5-fold cross-validation, scoring on accuracy, and n_jobs = 1. This configuration balances fit and complexity.

Volume 1, Issue 4, 2025
https://doi.org/10.5281/zenodo.17945075

Table 6: KNN Parameters Tuning

Parameter	Value	Best
param_grid	'max_depth': [3,5,7, 9]	5
	'weights': ['uniform', 'distance']	Uniform
	'p': [1,2]	1
cv	10	
scoring	accuracy	
n_jobs	1	

The K-Nearest Neighbor tuning (Table 6) selected n_neighbors = 5, weights = 'uniform', and p = 1 (Manhattan distance) as the optimal configuration. Uniform weighting outperformed distance-weighted voting, indicating equal neighbor contribution works better here, while Manhattan distance (p = 1) fit the data slightly better than Euclidean. Choosing K = 5 strikes a balance between under-fitting and over-fitting for this dataset.

Table 7: Random Forest Parameters Tuning

Parameter	Value	Best
param_grid	'n_estimators': [50,100,200]	200
	max_depth: [2,4,6,None]	4
	'min_samples_split': [2,5]	2
	'min_samples_leaf':['1, '2]	5
cv	5	
scoring	accuracy	
n_jobs	1	

Table 7 indicates that Random Forest tuning chose n_estimators = 200, max_depth = 4, min_samples_split = 2, and min_samples_leaf = 5 as the optimal settings. With 200 trees and a depth of 4, the model balances complexity and generalization, while the split and leaf thresholds (2 and 5, respectively) ensure robust node sizes. These parameters together achieved the highest cross-validated accuracy for the Random Forest. Figure 3 represents summary of the best parameters chosen for each model.

```
BEST PARAMETERS FOR EACH MODEL:
--------------------------------------------------
DT Best Params: {'criterion': 'entropy', 'max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 10}
KNN Best Params: {'n_neighbors': 5, 'p': 1, 'weights': 'uniform'}
RF Best Params: {'max_depth': 4, 'min_samples_leaf': 2, 'min_samples_split': 5, 'n_estimators': 200}
```

Figure 3: *Best Parameters for each Model (KNN, Decision Tree, and Random Forest)*

F. Performance Evaluation Metrics

In this paper, four performance evaluation metrics suitable for classification tasks are employed: accuracy, precision, recall, and f1-score.

(a). **Accuracy** is the percentage of accurate predictions in proportion to the total predictions made by the model. Mathematically, accuracy is depicted as follows:

$$Accuracy = \frac{TN+TP}{TN+FP+TP+FN} \quad (1)$$

Volume 1, Issue 4, 2025
https://doi.org/10.5281/zenodo.17945075

(b). **Precision** is the number of true positive predictions divided by the overall number of positive results. Mathematically,

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

(c). **Recall** is the total number of samples that should have been identified as positive, divided by the number of true positive results. Recall is represented in the equation below:

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

(d). **The F1-Score** combines both precision and recall. It is represented as follows:

$$F1 - score = 2 * \frac{precision*recall}{precision+recall} \quad (4)$$

Where: TP is true positive, TN is true negative, FN is false negative, and FP is false positive.

G. Model Explainability Process

In this paper, SHAP (SHapley Additive exPlanations) with a Permutation Explainer was employed, which estimates feature contributions by randomly permuting each feature and measuring the impact on predictions. After computing the SHAP values for the test set, the SHAP plot waterfall was used to visualize the impact of each model on prediction, how each feature pushed the model's output up or down from the baseline (expected) value, giving a clear, additive breakdown of why the prediction turned out the way it did. This approach helps verify that the model aligns with domain knowledge, spot potential biases, and build trust with stakeholders by making the decision process transparent.

H. List of Libraries

The analysis was run on Python 3.10.15 (packaged by Anaconda). The accompanying libraries include: Pandas 2.2.3, NumPy 1.25.0, Scikit-learn 1.5.1, Matplotlib 3.9.2, Seaborn 0.13.2, SciPy 1.12.0, Statsmodels 0.14.2, and SHAP 0.47.2 (Figure 4).

```
Python version: 3.10.15 | packaged by Anaconda, Inc. | (main, Oct  3 2024, 07:22:19) [MSC v.1929 64 bit (AMD64)]
Pandas: 2.2.3
NumPy: 1.25.0
Scikit-learn: 1.5.1
Matplotlib: 3.9.2
Seaborn: 0.13.2
SciPy: 1.12.0
Statsmodels: 0.14.2
SHAP: 0.47.2
```

Figure 4: Versions of Libraries and Environment Details used in this Paper

Volume 1, Issue 4, 2025
https://doi.org/10.5281/zenodo.17945075

IV. RESULTS AND DISCUSSION

A. Sample Dataset

Figure 5 presents the first five rows of the heart-disease dataset (heart.head()), offering a snapshot of the feature structure including age, sex, chest pain type, resting blood pressure, cholesterol, fasting blood sugar, resting ECG, maximum heart rate, exercise-induced angina, ST depression, slope of peak exercise, number of major vessels, and the target label (presence/absence of disease).

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	40	M	ATA	140	289	0	Normal	172	N	0.0	Up	0
1	49	F	NAP	160	180	0	Normal	156	N	1.0	Flat	1
2	37	M	ATA	130	283	0	ST	98	N	0.0	Up	0
3	48	F	ASY	138	214	0	Normal	108	Y	1.5	Flat	1
4	54	M	NAP	150	195	0	Normal	122	N	0.0	Up	0

Figure 5: Sample Dataset before Encoding

The dataset was encoded, as shown in Figure 6, owing the presence of categorical features such as sex, chest pain type, resting ECG, Exercise Angina, ST Slope and Heart Disease target feature.

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	40	0	1	140	289	0	0	172	1	0.0	0	0
1	49	1	2	160	180	0	0	156	1	1.0	1	1
2	37	0	1	130	283	0	1	98	1	0.0	0	0
3	48	1	3	138	214	0	0	108	0	1.5	1	1
4	54	0	2	150	195	0	0	122	1	0.0	0	0

Figure 6: Sample Dataset after Encoding

B. Exploratory Data Analysis

Figure 7 represents the distribution of sex (0 denoting male and 1 denoting female) with heart disease to know which gender has heart attack most. The figure shows that more males have heart disease than females.

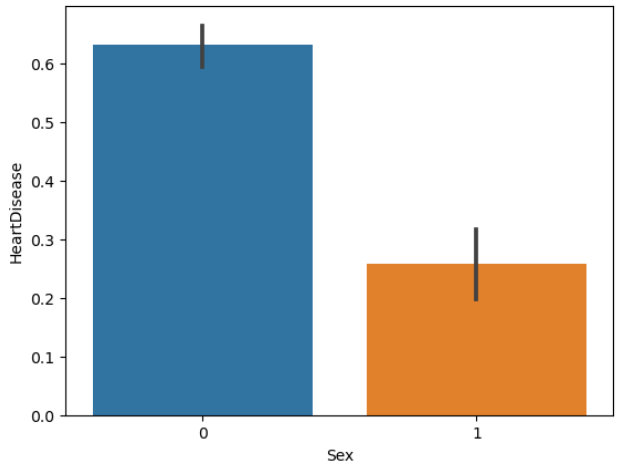


Figure 7: *Distribution of Sex with Heart Disease*

Volume 1, Issue 4, 2025
https://doi.org/10.5281/zenodo.17945075

Figure 8 reveals the age range of people with and without heart disease. Meanwhile, the age range with heart disease falls within 58–60, while the age range without heart disease falls within 53–57.

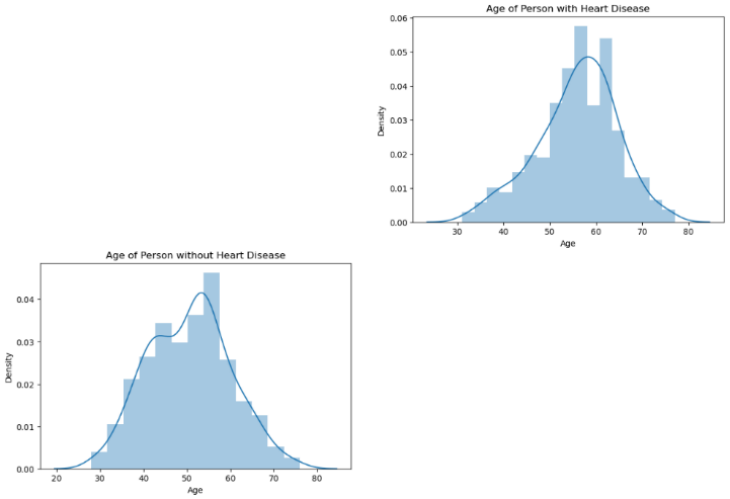
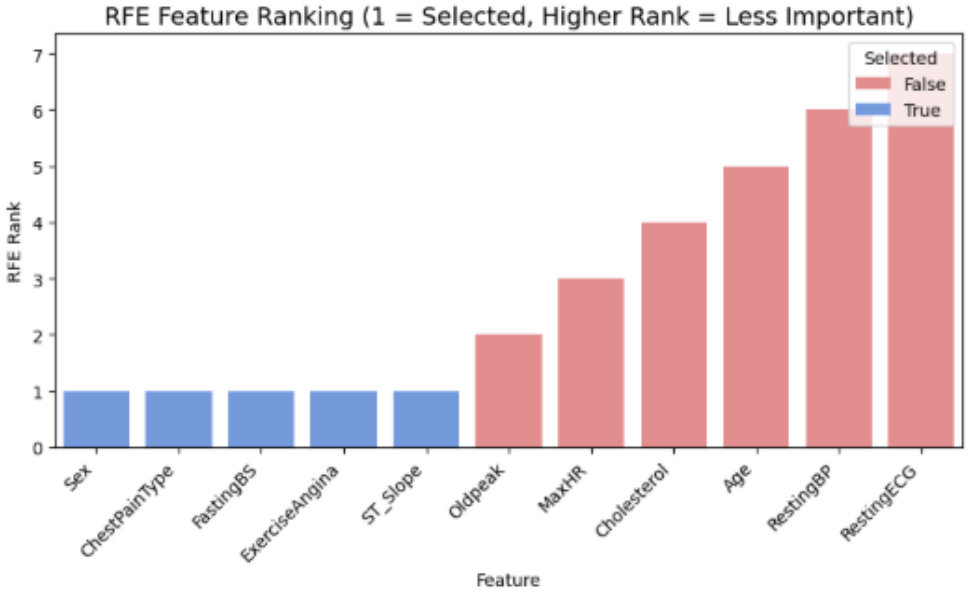


Figure 8: *Distribution of Age with Heart Disease*

C. Features Importance Visualization

(a). Recursive Feature Elimination: Figure 9 shows the results of Recursive Feature Elimination (RFE) applied to the heart-disease dataset. Each row lists a feature, its RFE-assigned rank (1 = most important, higher numbers = less important), and a Selected



flag (Blue chart ➔ True = kept, pink chart ➔ False = discarded).

Figure 9: Feature Importance by RFE Score

Volume 1, Issue 4, 2025
https://doi.org/10.5281/zenodo.17945075

The top 5 features indicated by RFE are Sex, ChestPainType, FastingBS, ExerciseAngina, and ST_Slope, indicating their strong predictive power over the target feature.

(b). Mutual Information: Mutual information analysis highlighted ST_Slope as the strongest signal, followed by ExerciseAngina, ChestPainType , Oldpeak, and MaxHR (Figure 10).

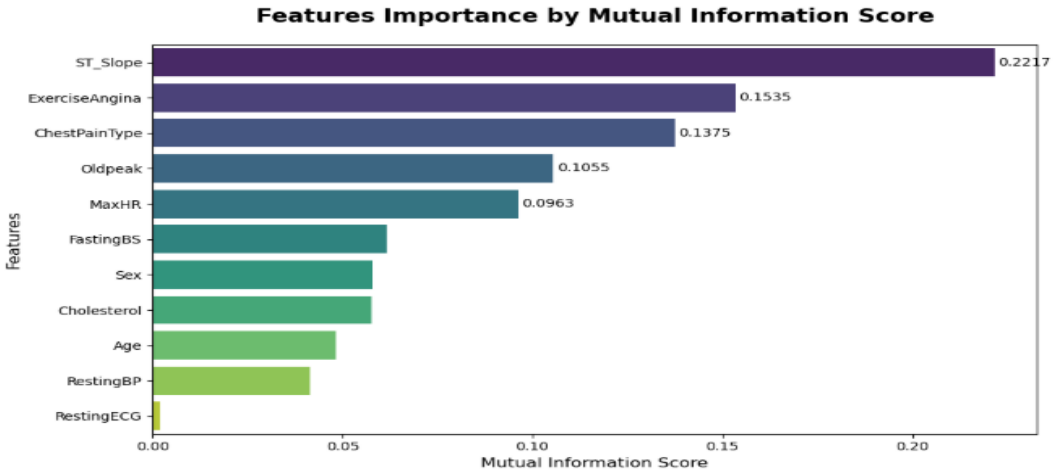


Figure 10: Features Importance by Mutual Information Score

These five attributes collectively capture the most relevant information about the target variable, suggesting that the slope of the ST segment, presence of exercise-induced angina, type of chest pain, degree of ST depression (Oldpeak), and maximum heart rate are the most discriminative factors for the model.

(c). Random Forest Gini Importance: The random forest gini importance identified ST_Slope as the most influential predictor, , followed by MaxHR, Cholesterol, Oldpeak, and ChestPainType (Figure 11).

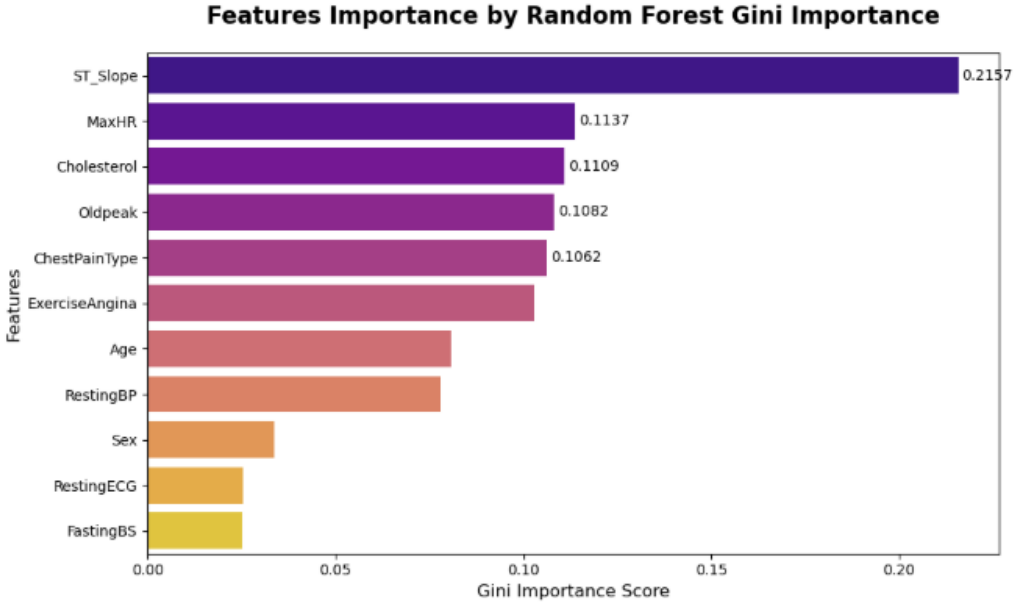


Figure 11: Features Importance by Random Forest Gini Importance Score

Volume 1, Issue 4, 2025
https://doi.org/10.5281/zenodo.17945075

Together, these five features indicate that slope of the ST segment, maximum heart rate achieved, cholesterol level, exercise-induced depression (Oldpeak), and the type of chest pain are the key drivers behind the model's decision-making process. Table 8 represents the selected top 5 features based on consensus.

Table 8: Recommended Features for Model Training

Feature	RFE	MI	RF Gini	Count	Selection Rationale
st_slope	☑	☑	☑	3/3	Top consensus for its appearance across all methods
chestpaintype	☑	☑	☑	3/3	Top consensus for its appearance across all methods
exerciseangina	☑	☑	✗	2/3	Strong agreement (RFE+MI)
maxhr	✗	☑	☑	2/3	Strong agreement (MI+RF)
oldpeak	✗	☑	☑	2/3	Strong agreement (MI+RF)
restingbs	☑	✗	✗	1/3	RFE-only
cholesterol	✗	✗	☑	1/3	RF-only
sex	☑	✗	✗	1/3	RFE-only

This study used these top 5 consensus features: st_slope, chestpaintype, exerciseangina, maxhr, and oldpeak because st_slope and chestpaintype appears across the three feature selection methods and 3 features, namely exerciseangina, maxhr, and oldpeak are with 2/3 agreement. The diversity of the feature selection methods considered: RFE (wrapper), MI (filter), RF (embedded) enhance stability of the model as features appearing in multiple methods generalize better.

D. Results of Models across Accuracy, Precision, Recall and F1-score

The performance of evaluation results of KNN, DT, and RF, without grid search CV, are discussed in this section. Figure 8 shows the results of the algorithms without GridSearch CV across accuracy, precision, recall and f1-score.

Metrics Table:

	Model	Accuracy	Precision	Recall	F1-Score
0	Decision Tree	0.757246	0.773975	0.757246	0.759418
1	K-Nearest Neighbor	0.811594	0.823306	0.811594	0.813199
2	Random Forest	0.829710	0.832234	0.829710	0.830434

Figure 11: Performance Evaluation Results of DT, KNN, and RF without GridSearch CV

Volume 1, Issue 4, 2025
https://doi.org/10.5281/zenodo.17945075

Figure 11 shows that decision tree achieved the following results: accuracy (75.7%), precision (77.4%), recall (75.7%), and f1-score (75.9%). KNN attained: accuracy (81.2%), precision (82.3%), recall (81.2%), and f1-score (81.3%). Random forest resulted to: accuracy (83.0%), precision (83.2%), recall (83.0%), and f1-score (83.0%). These results revealed that random forest outperformed KNN and DT across all metrics, demonstrating the robustness of ensemble method. These results are visualized in Figure 12 for better representation.

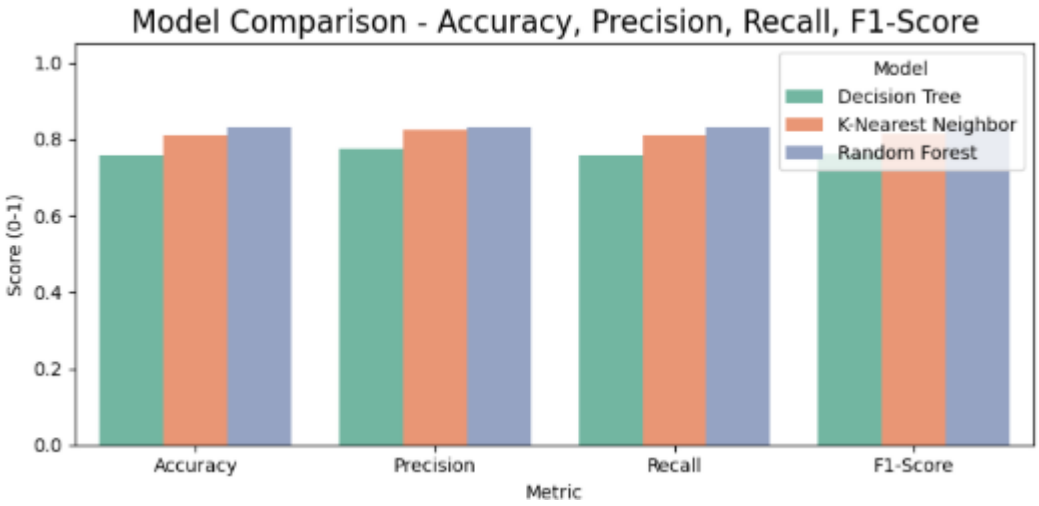


Figure 12: Performance Evaluation Results of KNN, DT, and RF Models without GridSearch CV

Furthermore, the models' parameters were fine-turned using grid search cross validation. After tuning, the performance evaluation results of the KNN, RF, and DT are depicted in Figure 13.

🏆 GRIDSEARCHCV RESULTS - BEST MODELS PERFORMANCE

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.8370	0.8404	0.8370	0.8378
K-Nearest Neighbor	0.8333	0.8431	0.8333	0.8347
Decision Tree	0.7899	0.8098	0.7899	0.7917

Figure 13: Performance Evaluation Results of DT, KNN, and RF with GridSearch CV

Figure 13 shows that RF achieved the following results: accuracy (83.7%), precision (84.0%), recall (83.7%), and f1-score (83.8%). KNN attained: accuracy (83.3%), precision (84.3%), recall (83.3%), and f1-score (83.5%). DT resulted to: accuracy (79.0%), precision (81.0%), recall (79.0%), and f1-score (79.2%). These results revealed that random forest outperformed KNN and DT across accuracy, recall, and f1-score while KNN slight defeats RF in precision (84.3% vs 84.0%). Nevertheless, the results reflect the robustness of ensemble method. These results are visualized in Figure 14 for better representation.

Volume 1, Issue 4, 2025
https://doi.org/10.5281/zenodo.17945075

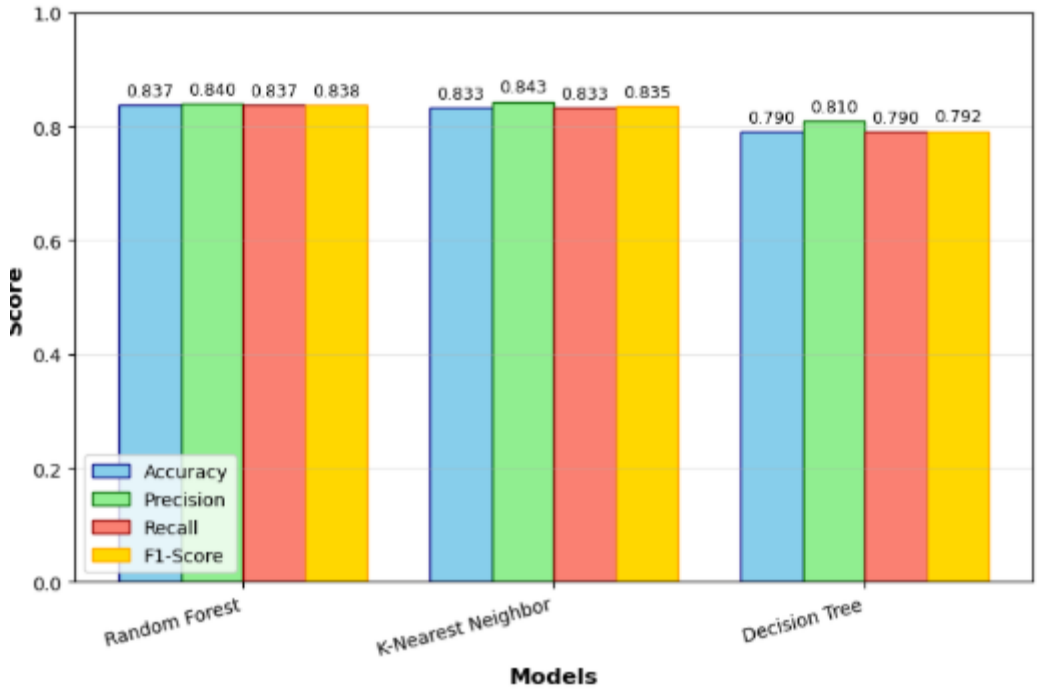


Figure 14: Performance Evaluation Results of KNN, DT, and RF Models with GridSearch *CV*

Table 9 presents the model performance evaluation comparison before vs after GridSearch Tuning.

Table 9: Performance Summary Table of RF, DT, and KNN

Model	Stage	Accuracy	Precision	Recall	F1-Score	Best Stage
Random Forest	Before	83.0%	83.2%	83.0%	83.0%	After
	After	83.7%	84.0%	83.7%	83.8%	
Decision Tree	Before	75.7%	77.3%	75.7%	75.9%	After
	After	79.0%	81.0%	79.0%	79.2%	
K-Nearest Neighbor	Before	81.2%	82.3%	81.2%	81.3%	After
	After	83.3%	84.3%	83.3%	83.5%	

Table 9 shows all three models improved after tuning. Overall, Random Forest remains the top performer post-tuning, followed by K-NN, followed by Decision Tree. Random Forest (RF) tops the performance table because it's an ensemble method that combines many decision trees, each built on a bootstrap sample of data and a random subset of features. This bagging approach reduces variance (over-fitting) while keeping low bias, so RF generalizes better than a single Decision Tree (DT). Unlike DT, which splits on the full feature set and can chase noise, RF's feature randomness decorrelates the trees, making the model more robust to outliers and noisy data. Additionally, the averaging of many trees smooths out errors, which translates into higher accuracy, precision, recall, and F1-score across both "before" and "after" stages. K-NN also improved, but it's a distance-based, non-ensemble algorithm, so it lacks the variance-reduction power of bagging and is more sensitive to irrelevant features or scaling.

Volume 1, Issue 4, 2025
https://doi.org/10.5281/zenodo.17945075

E. Visualization of Models' Results using Confusion Matrices

The confusion matrices shown in Figures 15 to 17 represent the performance breakdown of the models tuned with Grid Search CV, consisting of the true positive, false positive, true negative, and false negative counts. Decision Tree with Grid Search CV, as shown in Figure 15, has No disease = 91 TN, 21 FP; Disease = 46 FN, 118 TP.

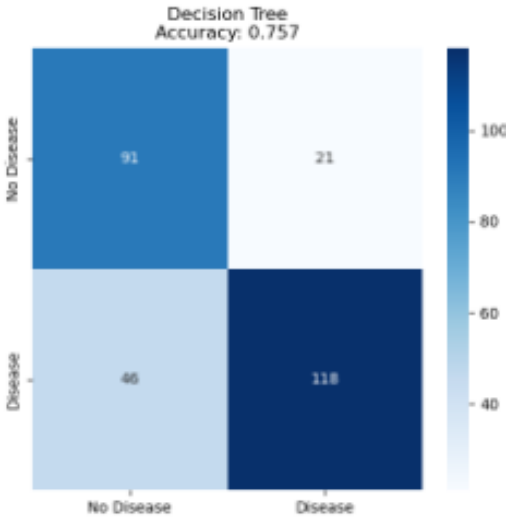


Figure 15: Confusion Matrix of the Decision Tree with Grid Search CV

The Figure revealed that the Decision Tree model (tuned with GridSearch CV) misclassified 67 instances (21 false positives + 46 false negatives). These errors indicate the model's tendency to miss 46 actual disease cases (false negatives), which is risky if the disease is severe and requires treatment. The relatively high false-negative rate suggests a need to adjust the decision threshold or re-weight the loss function to prioritize recall, especially in clinical contexts where missing a disease case is costlier than a false alarm. KNN with Grid Search CV, as shown in Figure 16, has No disease = 96 TN, 16 FP; Disease = 36 FN, 128.

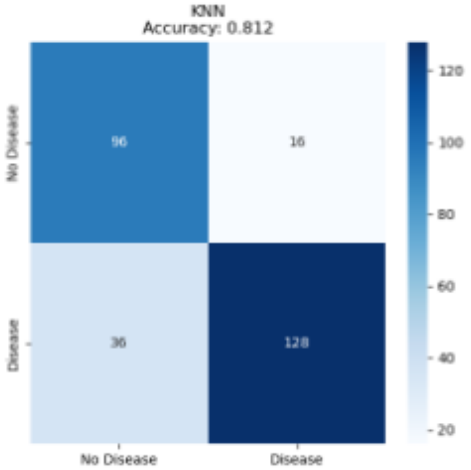


Figure 16: Confusion Matrix of the KNN with Grid Search CV

Volume 1, Issue 4, 2025
https://doi.org/10.5281/zenodo.17945075

K-Nearest Neighbors tuned with Grid Search CV (Figure 16) yields 52 misclassified (16 + 36) instances. Compared with the Decision Tree (67 misclassifications, 46 FN), KNN shows fewer

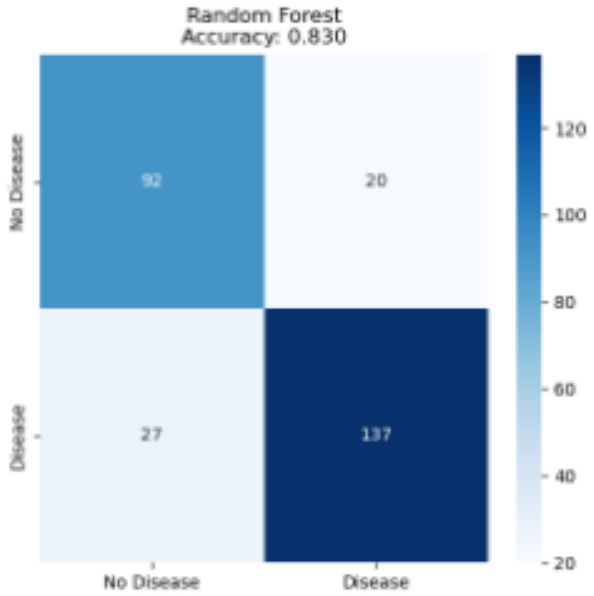


Figure 17: Confusion Matrix of the Random Forest with Grid Search CV

errors and a lower FN count, indicating GridSearch CV improved sensitivity (recall 0.78) while keeping decent precision (0.89). Random Forest with Grid Search CV, as shown in Figure 17, has No disease = 92 TN, 20 FP; Disease = 27 FN, 137. Random Forest tuned with Grid Search CV (Figure 17) shows 47 (20 + 27) misclassified instances. Compared with Decision Tree (46 FN, 67 errors) and K-NN (36 FN, 52 errors), Random Forest achieves the fewest false negatives and overall misclassifications, reinforcing its stronger ensemble robustness.

F. Visualization of Models' Results using ROC curves and AUC scores

The ROC curve results show all three models achieve excellent discrimination (Figure 18).

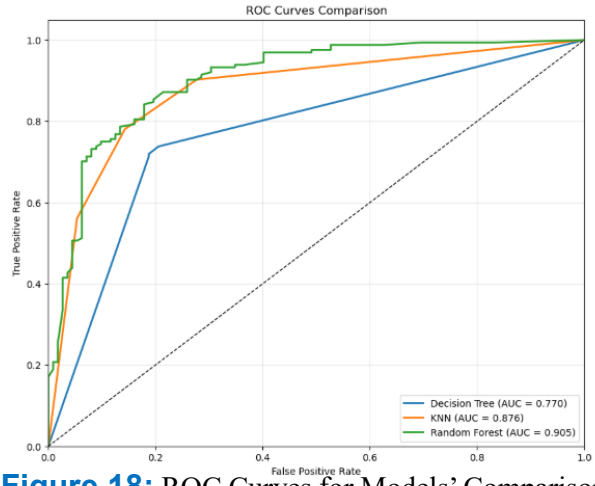


Figure 18: ROC Curves for Models' Comparison

Volume 1, Issue 4, 2025
https://doi.org/10.5281/zenodo.17945075

The ROC curve, shown in Figure 18, curve shows Random Forest (RF) clinging closest to the top-left corner, followed by K-Nearest Neighbors (KNN) and then Decision Tree (DT), reflecting their ability to discriminate between disease and no-disease cases. Quantitatively, the AUC values confirm this ordering: RF = 0.905, KNN = 0.876, DT = 0.770. A higher AUC means a better trade-off between sensitivity (true-positive rate) and specificity (1 – false-positive rate) across thresholds, so RF achieves the highest overall accuracy, KNN sits in the middle, and DT performs the weakest. Visually, the closer a curve hugs the top-left corner, the more the true positives it captures while keeping false positives low. Above all, RF is the preferred choice for classification reliability.

G. McNemar's Test or paired t-tests

McNemar's test results show statistically significant differences for Decision Tree vs. K-NN (McNemar stat = 11.0, p = 0.020 < 0.05) and Decision Tree vs. Random Forest (stat = 7.0, p = 0.0008 < 0.05), implying that the performance discrepancy between Decision Tree and the other two models is unlikely due to chance. Conversely, K-NN vs. Random Forest (stat = 9.0, p = 0.405 > 0.05) shows no significant difference, indicating their error structures are statistically comparable (Figure 19).

McNEMAR'S TEST (p < 0.05 = Significant Difference)

Comparison	McNemar Stat	p-value	Significant
Decision Tree vs KNN	11.0	0.020074	✅ YES
Decision Tree vs Random Forest	7.0	0.000821	✅ YES
KNN vs Random Forest	9.0	0.404873	❌ NO

Figure 19: ROC Curves for Models' Comparison

According to the Figure 19, since Decision Tree differs significantly from both K-NN and Random Forest, it is the weaker model here and should be de-prioritized. Meanwhile, K-NN and Random Forest can be considered interchangeable in terms of error consistency, reinforcing Random Forest's slight edge in metrics as the safer, top choice.

H. Confidence intervals (CI) for evaluation metrics.

The confidence-interval, Figure 20, shows Random Forest (RF) leading across accuracy and AUC, followed by K-NN and Decision Tree. The overlapping CIs indicate RF's slightly higher point estimates and tighter upper bounds and its ability to maintain a modest edge in both discrimination (AUC) and overall correctness (accuracy).

METRICS WITH 95% CONFIDENCE INTERVALS

Model	Accuracy	Accuracy CI	AUC	AUC CI
Decision Tree	0.757246	0.706-0.808	0.770361	0.723-0.820
KNN	0.811594	0.764-0.855	0.875544	0.834-0.914
Random Forest	0.829710	0.786-0.873	0.904535	0.863-0.937

Figure 20: ROC Curves for Models' Comparison

Volume 1, Issue 4, 2025
https://doi.org/10.5281/zenodo.17945075

Figure 20 implies that RF's ensemble nature gives it a reliability boost, making it the preferred model for deployment if you prioritize stable, higher-performing predictions; K-NN is a close alternative, while the Decision Tree lags behind, possibly due to higher variance.

I. Explainability of the Top Model (Random Forest with GridSearch CV)

(a). Interpretation of the SHAP Results

The SHAP waterfall (Figure 21) plot shows how the model's prediction for a single instance moved from the baseline (expected) value E[f(x)] = 0.52 to the final output F(x) = 1 (indicating a positive disease prediction). Starting at 0.52, each feature's SHAP value shifts the score up (red bars) or down (blue bars). ST_Slope = 0 → +0.32 (red) indicates the most influential factor, pushing probability upward. ChestPainType = 1 → +0.26 (red) is also strongly increasing risk. ExerciseAngina = 1 → –0.04 (blue) modestly reduces the predicted risk, which is counter-intuitive and may reflect dataset peculiarities or interaction effects. MaxHR = 179 → –0.03 (blue) which denotes high maximum heart rate slightly lowers risk. Oldpeak = 0 → –0.02 (blue) which indicates no ST-depression (oldpeak: 0) gives a small protective effect. The sum of these SHAP values (+0.32 + 0.26 – 0.04 – 0.03 – 0.02 = 0.49) added to the baseline 0.52 reaches the final score 1.01 (≈ 1 due to rounding), confirming the model's confidence of disease presence.

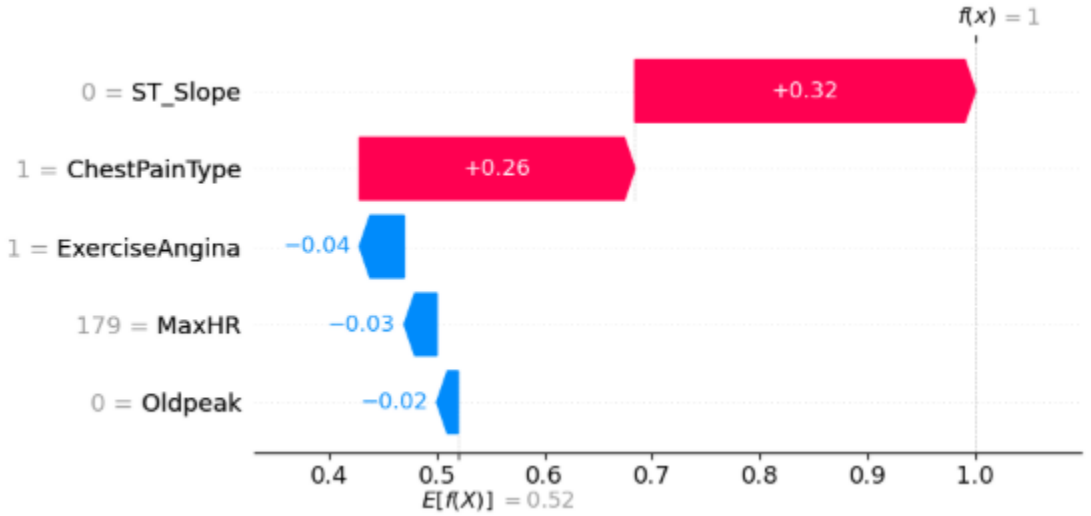


Figure 21: ROC Curves for Models' Comparison

(b). Clinical Interpretation of the Important Features

T Slope (ST_Slope = 0) implies a flat or down-sloping ST segment during exercise, a well-known marker of myocardial ischemia; its large positive SHAP value aligns with clinical risk stratification. Chest Pain Type (ChestPainType = 1) likely represents an atypical or non-anginal pain category that the model has learned to associate with higher risk, consistent with many risk scores that treat atypical pain as a red flag. Exercise-Induced Angina (ExerciseAngina = 1) is usually a danger signal, though it has a small negative weight in this prediction. This could arise from data sparsity. Max Heart Rate (MaxHR = 179) indicates a higher achieved heart rate (often seen in better fitness or younger age) which can be linked to lower cardiovascular risk in several

Volume 1, Issue 4, 2025
https://doi.org/10.5281/zenodo.17945075

scores. Oldpeak (ST-depression = 0) is an absence of ST depression (oldpeak 0) which is protective, matching physiological expectation.

(c). Implications for Real-Life Decision Support Systems

i. Transparency & Trust: Displaying the waterfall plot in a clinician's dashboard lets them see why the model flagged a patient (e.g., abnormal ST slope + chest-pain type). This supports trust and enables verification against clinical intuition.
ii. Risk Prioritization: Patients with large positive contributors (ST slope, chest pain) can be fast-tracked for further diagnostics (stress imaging, coronary CT, or invasive angiography).
iii. Safety Checks: The unexpected negative weight for exercise angina signals a need for model review or additional features, prompting clinicians to treat such cases with caution.
iv. Integration into EMR: Embedding SHAP explanations alongside patient vitals allows quick "what-if" analyses (e.g., simulate how a change in ST slope would alter risk).
v. Overall, the SHAP chart reflects the model's learned associations, not causality. Clinicians must still reconcile model output with history, comorbidities, and domain knowledge, especially when counter-intuitive contributions appear.
vi. The SHAP waterfall clarifies that ST slope and chest-pain type drive the Random Forest's high-risk prediction, while factors like high MaxHR and absent ST-depression modestly temper it. Leveraging this explainability in a decision-support tool can improve triage accuracy, foster clinician confidence, and highlight where the model may need refinement.

V. CONCLUSION

This study systematically compared k-nearest neighbor (KNN), decision tree (DT), and random forest (RF) models for heart-attack prediction on a single, modest-sized dataset. After encoding categorical variables (sex, chest pain type, resting ECG, exercise angina, ST slope, etc.) and performing exploratory analysis, feature importance was quantified through RFE, mutual information, and RF Gini impurity. A consensus set of five features, namely ST slope, chest pain type, exercise angina, max HR, and Oldpeak (ST-depression) emerged as the strongest signals across methods, aligning with established clinical risk factors. Performance metrics (accuracy, precision, recall, F1-score) improved for all models after GridSearch CV hyper-parameter tuning, with RF achieving the highest overall scores (accuracy 83.7 %, AUC 0.905) and a statistically significant edge over DT, while the difference between RF and KNN was not significant (McNemar's $p > 0.05$). SHAP waterfall analysis of the best RF model confirmed that abnormal ST slope and chest-pain type drive risk predictions, whereas high MaxHR and absent ST-depression modestly lower it; however, the counter-intuitive negative weight for exercise-induced angina warrants further investigation. This paper suggests the following areas for future work: cross-validation across multiple hospitals to verify robustness and reduce site-specific bias, a hybrid method is recommended to blend complementary strengths, incorporation of additional clinical variables (lab results, medication history) and more diverse dataset to improve generalizability, and development of a clinician-facing dashboard that displays SHAP explanations, confidence intervals, and decision thresholds adjustable for sensitivity vs. specificity trade-offs.

Volume 1, Issue 4, 2025
https://doi.org/10.5281/zenodo.17945075

REFERENCES

[1] A. Rajdhan, and A. Agarwal, M. Sai, D. Ravi, and D.P. Ghuli, "Heart disease prediction using machine learning," *Journal of Engineering Sciences*, *14*(4), 2023, 440-450.

[2] C.M. Bhatt, P. Patel, T. Ghetia, and P.L. Mazzeo, "Effective heart disease prediction using machine learning techniques," *Algorithms*, *16*(2), 2023.

[3] M. Alshraideh, N. Alshraideh, A. Alshraideh, Y. Alkayed, Y. Al Trabsheh, and B. Alshraideh, "Enhancing heart attack prediction with machine learning: A study at Jordan University Hospital," *Applied Computational Intelligence and Soft Computing*, *2024*(1), 2024.

[4] H. Nasaka, S. Sahejbir, and M. Murali, "Arduino Based Heart Rate Monitor and Heart Attack Detection System," *International Journal of Advanced Science and Technology*, *29*(7), 2020, 971–977.

[5] B.U. Rindhe, N. Ahire, R. Patil, S. Gagare, and M. Darade, "Heart disease prediction using machine learning," *Heart Disease*, *5*(1), 2021.

[6] H. Jindal, S. Agrawal, R. Khera, R. Jain, and P. Nagrath, "Heart disease prediction using machine learning algorithms," In *IOP conference series: materials science and engineering* (Vol. 1022, No. 1, p. 012072). 2021, IOP Publishing.

[7] S.Y. Prasetyo, A. Kurniawan, E.F.A. Sihotang, R. Puspita, ad K.E. Setiawan, "Heart disease risk prediction using K-nearest neighbor: A study of Euclidean and cosine distance metrics," In *2023 3rd International Conference on Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS)* (pp. 236-240), 2023. IEEE.

[8] K.A. Barry, Y. Manzali, M. Lamrini, F. Rachid, M. and Elfar, "Heart Disease Prediction Using Weighted K-Nearest Neighbor Algorithm". In *Operations Research Forum* (Vol. 5, No. 3, p. 76), 2024. Cham: Springer International Publishing.

[9] M. Ozcan, and S. Peker, "A classification and regression tree algorithm for heart disease modeling and prediction," 2023, *Healthcare Analytics*, *3*, 100130.

[10] K. Shaik, J.V.N. Ramesh, M. Mahdal, M.Z.U. Rahman, S. Khasim, and K. Kalita, "Big data analytics framework using squirrel search optimized gradient boosted decision tree for heart disease diagnosis," *Applied Sciences*, *13*(9), 2023.

[11] P.E. Rubini, C.A. Subasini, A.V. Katharine, V. Kumaresan, S.G. Kumar, and T.M. Nithya, "A cardiovascular disease prediction using machine learning algorithms," *Annals of the Romanian Society for Cell Biology*, 2021, 904-912.

[12] M.M. Ali, V.S. Al-Doori, N. Mirzah, A.A. Hemu, I. Mahmud, S. Azam, and M.A. Moni, "A machine learning approach for risk factors analysis and survival prediction of Heart Failure patients," *Healthcare Analytics*, *3*, 2023.

[13] A. Khan, M. Qureshi, M. Daniyal, and K. Tawiah, "A novel study on machine learning algorithm-based cardiovascular disease prediction," *Health & Social Care in the Community*, *2023*(1), 2023.

[14] A. Ghasemieh, A. Lloyed, P. Bahrami, P. Vajar, and R. Kashef, "A novel machine learning model with Stacking Ensemble Learner for predicting emergency readmission of heart-disease patients," Decision Analytics Journal, 7, 2023.

[15] M.A. Hossen, T. Tazin, S. Khan, E. Alam, H.A. Sojib, M. Monirujjaman Khan, and A. Alsufyani, "Supervised machine learning-based cardiovascular disease analysis and prediction," *Mathematical Problems in Engineering*, *2021*, 1-10.

[16] D.E. Salhi, A. Tari, and M.T. Kechadi, "Using machine learning for heart disease prediction," In *Advances in Computing Systems and Applications: Proceedings of the 4th Conference on Computing Systems and Applications,* 2021, (pp. 70-81). Springer International Publishing.

https://doi.org/10.5281/zenodo.17945075