Fuzzy Systems and Soft Computing
ISSN : 1819-4362

A REVIEW ON CORONARY ARTERY DISEASE PREDICTION USING ENSEMBLE LEARNING APPROACHES

Yesha Patel Ph.D Scholar , Gujarat Technological University, Ahmedabad, yeshashah.it@gmail.com.
Dr. Dushyantsinh Rathod Professor & HoD, Department of Computer Engineering, Ahmedabad Institute of Technology, Ahmedabad : dushyantsinh.rathod@gmail.com

Abstract

In this paper, the five machine learning models of Extra Tree Classifier , ADABoost,Logistic Regression, XGBoost, Bagging classifier and K-Nearest Neighbors are carefully compared and evaluated for the prediction of heart disease based on a clinical dataset. The objective is to describe the accuracy, strengths, weaknesses, and applicability of each model with emphasis on which one among the algorithms works best for heart disease diagnosis. We mainly evaluated them in terms of accuracy: Extra Tree Classifier gives with an accuracy of 81%. It showed good precision and recall abilities owing to being based on ensemble learning. Closely followed was XGBoost with an accuracy of 82% but much more computationally intense, which meant it needed more machine learning processes to compensate for complex clicked patterns. Logistic regression obtained an accuracy of 76%, providing good confidence in recalling positive cases and therefore can identify true-positive instances well. Moderate accuracy at 84% is reported for ADABOOST but its computational intensity and sensitivity to parameter tuning impeded performance. The KNN, with its straightforward design, had lower overall performance, at only 61% accuracy, given that it struggles on feature scaling and is sensitive to irrelevant information in the dataset. Overall, ADABoost and XGBoost have shown the greatest promise in predicting heart disease. Further tuning on these models would enhance their predictive performance in the pursuit of prediction applications.

Keywords: Heart Disease Prediction, Machine Learning Models, Accuracy Comparison, Clinical Dataset, Model Interpretability, Feature Scaling, Computational Complexity

1. Introduction

Heart disease still heads the list of killers, with a death toll of nearly 17.5 million per year; its impact has been felt all too vividly in a country like India, which has made it the leading cause of death in the world.[1] This disease is often termed coronary artery disease (CAD). In addition to it being the weakening of the heart muscle, it also does not allow enough blood supply through the arteries, resulting in major complications such as heart failure, hypertension, and cardiac arrest. Concerning symptoms, chest pain, shortness of breath, and blood pressure are considered the most important symptoms of heart disease. Heart disease is caused by genetic defects, bad lifestyle habits, and chronic diseases such as diabetes.[2] With statistics so alarming, the accurate and early identification of heart disease is significantly important. Therefore, conventional methods applied by the medical practices are found to be insufficient as, according to research, health professionals can only predict heart disease with about 67% correctness. [3] This inadequacy highlights a substantial gap in effective detection and the urgent need for improved methodologies. The advent of digital technology has enabled health care organizations to accumulate a large amount of related health information. This information, however, often relates to complex and noisy data. [4]
Among the vast amount of data, what has emerged is the machine learning and data mining techniques, which are invaluable in processing information to make a healthcare professional capable of predicting heart disease. Algorithms that can be used in constructing predictive models include Naive Bayes, Support Vector Machine, Extra Tree Classifier , KNN (K-Nearest Neighbours), and XGBoost. Such models can help improve predictive estimates concerning heart disease and provide doctors with timely opportunities for taking decisions that might save millions of lives. [5]
The main objectives of the paper have been listed below -

- **Study the Machine Learning Approaches:** Examine the assorted machine-learning methods and classification algorithms utilized for cardiovascular disease predictions.

- **Analyzing Data and Comparing Algorithms:** Patient data use for the diagnosis of heart disease discussed and provide insight into algorithm efficiencies.
- **Demonstration of ML Potential:** The role of machine learning for improved early diagnosis and clinical decision-making for cardiac diseases should be emphasized.

The rest of the paper elaborates, with Section 2 Literature Review presenting a roundup of available literature outlining methodologies and usage of machine learning models in predicting cardiovascular disease. In Methodology, Section 3 depicts the selected machine learning algorithms, the different preprocessing steps undertaken for the data, and the experimental setup adopted. Implementation in Section 4 outlines the model development outcomes, including tools, libraries, and system configurations before the working models are evaluated and discussed in Section 5: Results and Discussion. The Conclusion in Section 6 lists the major findings of the paper, while Section 7 includes Future Scope and provides recommendations for further improvement and areas of research for heart disease prediction modeling.

2. Literature Review

Heart disease prediction is a really crucial area of study since the statistics regarding heart health are startling and because it is one of the primary causes of death across the globe. This basically infers that we had many sources to refer to during our research. We went through more than 50 research papers on heart disease prediction and condensed the findings into approximately 14-15 key studies. This brought to our notice quite a few special algorithms that are especially relevant in this field, including Decision Tree, Neural Network, Gradient Boosting Machine (GBM), Sequential Minimal Optimization (SMO) Algorithm, Artificial Neural Network (ANN), Q-learning, and Deep Q Networks (DQN), apart from the five algorithms we have chosen here for thorough study. This wide-ranging research is reflected in the analysis that has been presented below.

The majority of techniques suggested by Khan, Y. and Qamar, U (2019, February) [6] for heart disease prediction included ADABOOST s, Neural Networks, and Ensemble Classifiers. ADABOOST s are flexible but computationally very resource-consuming, capture complex patterns with Neural Networks but require huge datasets and resources, and work much better with Ensemble Classifiers that combine the power of a group of models but are less interpretable. However, another review by Ahsan, M. M., and Siddique, Z. (2022) [7] discussed the ensemble-based approach with Naive Bayes and Decision Trees, which reported 87.37% accuracy in heart disease prediction. Another model by J48 decision tree reported an accuracy of

82.57 in the prediction of myocardial infarction. Some models such as Decision Trees and ADABOOST s excel in environments that contain labeled data, they perform poorly in unstructured data, or worse, are prone to overfitting. There are several types of algorithms including clustering like K-means which actually reveal the underlying structure, semi supervised which makes use of both labeled and unlabeled data. Models like Q-learning proposed by Fatima, M. and Pasha, M. (2017) [8] are ideal for applications in dynamic environments but require extensive exploration. Genetic algorithms can deal with high dimensionality of the problem and have high time complexity and deep learning models such as DNN, effective with large data although they imply high resource demand.

Nadeem et al. (2021) [9] used ADABOOST with fuzzy logic based decision level fusion which proved to be very effective in heart disease prediction with an accuracy of 96.23% which is higher than other models such as ADABOOST and ANN. The combination of two ADABOOST using fuzzy logic may be promising for yields stable predictions; however, the authors failed to explain the specific parameter setting, and the details of the fuzzy logic, so it was difficult to reproduce the study.

Reddy et al. (2021) [10] employed SMO (Sequential Minimal Optimization) that can handle large high dimensions and is highly efficient in problems without convexity. However, choosing the kernel and then the regularization parameter may affect its performance and it may have a computational problem with high dimensionality on very large data sets.

The Heart Disease Prediction Model used by Kavitha, B. S., and Siddappa, M (2020) [11] used the DBSCAN for outlier detection, and SMOTEENN for the dataset balancing technique. The accuracy rate obtained is 98.40% for the Cleveland dataset and 95.90% for that of the Statlog dataset. It is a real-time implementation using Clinical Decision Support Systems (HDCDSS). However, it does not scale well and does not use feedback from cardiologists. The development in Extra Tree Classifier and Linear Methods in the study enhances feature selection along with rates of prediction; however, handling real-life data becomes challenging.

Swathy, M., and Saruladha, K. (2022) [12] proposed Parallel Distributed Processing for the unstructured data as well as Multi-layer Perceptron models are efficient since they need less pre-processing in comparison to the traditional models. The extract of features is done automatically by CNNs, while data with complicated characteristics are managed by other types of networks, RNN, ResNet, and DenseNet. Nevertheless, these models are accurate, demand massive data, computationally expensive and sometimes face scalability challenges. The Decision Tree model imposed by Kumar et al. (2018) [13] contains decision parameters including maximum depth, minimum number of samples and can be easily overfit and unstable. The J48 algorithm which applies the pruning methods has slightly better performance, but is computationally expensive and sensitive to noise. Furthermore, the model named Logistic Model Tree (LMT) combines decision trees and logistic regression which improves prediction, but at the same time is sensitive to overfitting.

The Gated Recurrent Unit (GRU) neural network given by Javid, I. et al. (2020) [14] is less complex than the LSTM architecture and is useful for capturing short-term dependencies in time series data but often may fail to learn long-term dependencies and needs large volumes of data for their training. Also, Voting-Based Ensemble Models make use of several classifiers in decisions by methods such as voting to improve the outcome, though they add levels of complication to the tuning procedure and can degrade if the classifiers are significantly related. Genetic Algorithms (GAs) by Sharma, G. et al (2020, November) [15] work with parameters like population size and mutation rate but are computational and don’t guarantee global optimization. The NB is computationally efficient, and top-performing on large scale, sparse datasets, but the main disadvantage is that it postulates feature independence. Ant Colony Optimization (ACO) is an efficient consequent optimization algorithm in a dynamic environment derived from ant behaviors, takes numerous iterations to converge and is sensitive with its parameters.

Feedforward Artificial Neural Networks (ANNs) given by Rani, S., and Masood, S. (2020) [16] are good at capturing nonlinear relationships and applicable in high dimensionality but they need a lot of training data set and are vulnerable to over-fitting. Weighted Support Vector Machines (WADABOOST s) work well in high dimensions, and they are sensitive to kernel functions since they can be memory intensive and tuning of hyperparameters. The C4.5 algorithm employed by Sajja, G. S et al. (2021, August) [17] for classification and regression allows tuning of decision parameters like confidence factors, pruning strategies, and addresses attributes of both nominal and continuous types while still being comprehensible. Nevertheless, it is an expensive process in terms of computational intensity and not robust to noise. Similar to the ID3 algorithm, it is very easy to implement, creates easy to understand decision trees but has a major drawback of overfitting the data and does not handle missing values. CNNs are used in image-based tasks by Desale, K. S., and Shinde, S. V. (2022) [18] as they are able to capture spatial structure and can learn features directly on the image with little to no need for feature extraction. They deal with big data but often consume computational resources and they need sufficient labeled data for the training. Thirdly, they are less interpretable as compared to other machine learning models that belong to classical types.

Babu, S. V., Ramya, P., and Gracewell, J. (2024) [19] explored Quantum-Assisted Machine Learning (QuML) uses principles such as superposition and entanglement to achieve superior computation than classical machine learning, and still has the ability to perform training quicker for quite a number of difficulties. However, it is restricted only to the specialized hardware and the associated amenities, unavailable at present, and the interpretative part of the quantum models can be prohibitive. To address this, implementation teams employ Quantum accounting software like IBM Quantum and Google Cirq, augmented by the standard machine learning libraries used to handle classical data. Multilayer Perceptrons (MLPs) discussed by Badawy, M., et al. (2023) [20] are effective for tasks requiring complex modeling beyond linear capabilities but lack interpretability compared to simpler models. The VMD-IPSO-LSTM model demonstrates versatility in handling various data types but is prone to overfitting. In contrast, the CSO-LSTM model has a high learning capacity but demands significant computational resources, including memory and processing power. The specifications of each model have been given in Table 1.

Table 1 - Specifications of the Models Analyzed

Source	**Model Used**	**Description**	**Parameters Used**	**Tools**	**Limitations**
[6]	ADABOOST ; Neural Networks ; Ensemble Classifiers	ADABOOST uses hyperplanes for classification, NN learn patterns, and Ensemble Classifiers combine models for accuracy.	Kernel type, Regularization Layers, Neurons, Activation, Learning rate, Batch size, Estimators/Trees.	scikit-learn, LIBADABOOST , TensorFlow, Keras, PyTorch, LightGBM	Expensive with large datasets, sensitive to parameters, prone to overfitting.
[7]	Naive Bayes, Decision Tree, J48 algorithm	Naive Bayes uses probabilities for classification, while Decision Trees and J48 create tree-based models.	Not specified	J48 algorithm (decision tree-based)	Specific limitations aren't mentioned in the text.
[8]	Decision Trees, DBSCAN, Label Propagation, Q-learning, Genetic Algorithms, DNN	Decision Trees classify data hierarchically, DBSCAN groups spatial data, Label Propagation spreads labels, while Q-learning, Genetic Algorithms, and DNNs optimize learning	Number of clusters, Distance metrics, Learning rate, Discount factor, Population size, Mutation rate, Number of layers, activation functions	Scikit-learn, TensorFlow, Keras, PyTorch, HDBSCAN, OpenAI Gym, DEAP, Java/C++ libraries	Models require labeled data, are sensitive to noise and outliers, may overfit, and often demand extensive resources and careful tuning
[9]	Fuzzy Logic-Based Decision Level Fusion	Fuzzy Logic-Based Decision Level Fusion combines multiple sources of information to improve decision-making under uncertainty and enhance classification accuracy.	Kaggle's "heart disease dataset 2019" and "cardiovascular disease dataset 2019."	Python 3.7 with unspecified libraries	Omits computational costs and specific ADABOOST parameters, hindering reproducibility and comparison with simpler models
[10]	Sequential Minimal Optimization (SMO)	SMO efficiently solves the ADABOOST optimization problem by breaking it into smaller, manageable subproblems for	Regularization parameter, kernel type, kernel parameters (e.g., gamma)	Scikit-learn in Python, statistical software packages	Kernel choice and parameter selection significantly affect performance, while computational complexity can increase with very large

		faster convergence			datasets
[11]	Heart Disease Prediction Model ; Hybrid Extra Tree Classifier and Linear Method	Combines Random Forest and linear methods to enhance predictive accuracy and improve classification of heart conditions.	DBSCAN for outlier detection, SMOTEENN for dataset balancing, Combination of Random Forest and Linear Method.	XGBoost, DBSCAN, SMOTEENN, Decision Tree Entropy	Struggles with small datasets, lacks expert feedback, and can't efficiently handle real-life data or scale effectively
[12]	Hierarchical Neural Networks, Parallel Distributed Processing	Neural Networks structure learning layers, while Parallel Distributed simulates cognitive functions with interconnected networks	Large training data, Processes data non-linearly, Requires less preprocessing, Automatically detects key features	WEKA, MATLAB, TANGARA	Requires extensive training data, incurs high computational costs, and poses challenges for interpreting complex results effectively
[13]	Decision Tree, J48 Algorithm, Logistic Model Tree (LMT)	Decision Trees, J48, and Logistic Model Trees classify data by creating hierarchical structures combining decision-making and regression analysis	Maximum depth, minimum samples split, minimum samples leaf, Confidence threshold, pruning method	scikit-learn, Weka, R	Prone to overfitting, unstable with data variations, computationally expensive, requires careful tuning for optimal performance.
[14]	Extra Tree Classifier , Gated Recurrent Unit (GRU), Voting-Based Ensemble Model, Combination strategy	Extra Tree Classifier uses multiple decision trees, GRUs handle sequential data, and Voting-Based Ensemble Models combine predictions using various strategies for improved accuracy	Number of trees, Maximum depth, Minimum samples to split, GRU units, layers, learning rate, and dropout rate	scikit-learn, Weka, R, TensorFlow, PyTorch	Computationally expensive, struggle with imbalanced datasets, and require extensive training data; tuning multiple classifiers adds complexity.
[15]	Genetic Algorithm (GA), Naive Bayes (NB), Ant Colony Optimization (ACO)	GA optimizes solutions using evolution principles, NB classifies data probabilistically, and ACO mimics natural behavior for optimization.	Population size, mutation rate, crossover rate, termination criteria, pheromone evaporation rate	Python, MATLAB, R, scikit-learn, Weka, MATLAB	May not guarantee global optima and are sensitive to feature independence and parameter settings.
[16]	Feedforward	Feedforward	Number of hidden	TensorFlow,	Extensive training

	Artificial Neural Network, Weighted Support Vector Machine	Artificial NN process data through layers, while Weighted ADABOOST enhance classification by adjusting the importance of data points	layers, neurons per layer, Kernel type, regularization parameter (C), kernel coefficient (gamma)	PyTorch, Keras, scikit-learn, LIBADABOOST , Weka	data, are prone to overfitting, are memory- intensive, and require careful tuning of kernels and hyperparameters
[17]	C4.5 Algorithm, ID3 Algorithm	C4.5 and ID3 algorithms build decision trees, with C4.5 improving ID3 by handling continuous data and pruning trees	Confidence threshold, pruning method, Parameters related to tree building and pruning	scikit-learn, Weka, R,	Sensitive to noise, prone to overfitting with complex datasets, and unable to handle missing values, computationally

		effectively			expensive
[18]	Convolutional Neural Network (CNN)	CNN excel at processing grid-like data, such as images, by using convolutional layers to detect features efficiently	Filter size, number of filters, padding, pool size, stride	TensorFlow, Keras, PyTorch	Requires extensive labeled data for training, and has limited interpretability compared to traditional methods.
[19]	Quantum-Enhanced Machine Learning (QuEML)	QuEML leverages quantum computing to improve learning algorithms, enhancing speed and accuracy for complex data problems	Number of qubits, circuit depth, learning rate, etc.	IBM Quantum, Google Cirq, Rigetti Forest,	Requires specialized hardware for computation, face limited resources and expertise, and present challenges in interpretability
[20]	Multilayer Perceptrons (MLPs), VMD-IPSO-LSTM Model, CSO-LSTM Model	(MLPs) are feedforward networks, while VMD-IPSO-LSTM and CSO-LSTM models integrate advanced techniques for improved sequential data forecasting	Hidden layer architecture, Batch size, epochs, Number of LSTM layers	TensorFlow, Keras, PyTorch	Lack interpretability compared to simpler models, are prone to overfitting, and require significant computational resources and processing power

3. Methodology

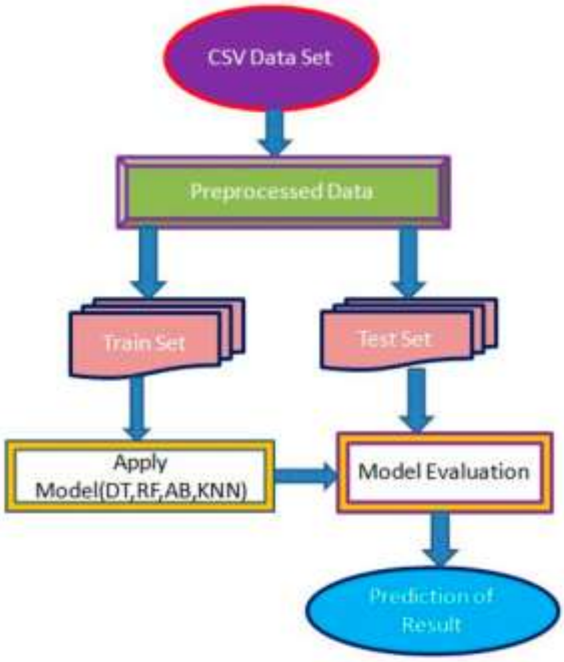


Figure 1 – Proposed Architecture

Dataset Inputs: The dataset considered in this analysis was the Cleveland Heart Disease Dataset, a fairly well-known dataset taken from the UCI Machine Learning Repository. It is a dataset comprising 303 cases and 14 features. It is a prominent benchmark periodically used in the machine learning literature for research into predicting cardiovascular disease, making it a standard to evaluate the prediction capabilities of models in this arena. The various features related to heart health are collected here. They have been presented below in Table 2.

Table 2 - Heart Disease Prediction Dataset Input

Dataset Input	**Description**
Age	Patient's age (in years)
Gender	Encoded 1 for female and 0 for male
Chest Pain Type (cp)	Categorical variable representing types of chest pain
Resting Blood Pressure (trestbps)	Blood pressure when at rest
Cholesterol (chol)	Cholesterol levels in mg/dL
Fasting Blood Sugar (fbs)	Fasting blood sugar level (1 if > 120 mg/dl, 0 otherwise)
Resting Electrocardiographic Results (restecg)	Results from an ECG at rest
Maximum Heart Rate Achieved (thalach)	Maximum heart rate during exercise
Exercise Induced Angina (exang)	1 if angina occurs during exercise, 0 otherwise
Oldpeak	ST depression induced by exercise relative to rest
Slope	The slope of the peak exercise ST segment.
Number of Major Vessels (ca)	Number of major vessels colored by fluoroscopy
Thalassemia (thal)	Categorical variable indicating thalassemia status
Target	Outcome variable (1 for presence of heart disease, 0 for absence)

3.1 Data Preprocessing: This critical stage transforms the dataset for model training through some key processes. First, data cleaning takes place to undertake any necessary corrections and modifications arising from quality audit. After that, methods like imputation or deletion are used to address missing data and for the sake of data quality of the dataset. After that, categorical variables are transformed into numeric formats with the help of methods such as one-hot encoding or label encoding that let them be used in machine learning. Lastly, numerical features are scaled to a standard scale. This is important for algorithms that are sensitive to data scaling such as KNN and ADABOOST .

3.2 Model Training: Training of various machine learning models on the preprocessed data is carried out using different techniques - Extra Tree Classifier integrates a number of decision trees with the help of bagging and selecting randomly features to avoid higher variance towards the data set. ADABOOST defines the optimal plane within the class for segregation but requires a kernel choice and hyperparameter selection. Logistic Regression employs a logistic function for estimation of likelihood of predictors on heart disease, which is a binary variable. KNN simply categorizes new data with the majority class amongst the nearest neighbors, depending on ‘k’ and distance function. XGBoost increases predictive measures of accuracy through gradient boosting, more improvements in the regularity aspect as well as the handling of missing values in the data set.

3.3 Model Evaluation: Several measures like accuracy, precision, recall, F1 score are used to comprehend the efficiency of the trained models in identifying the heart disease cases from a test dataset. Further, the application of such methods as k-fold cross-validity allows improving the stability of the characteristics of performance when working with different subsets of data.

3.4 Prediction Output: In the last phase, the trained models estimate the probability of heart diseases in new, unseen data and return 1 if the data has heart disease, otherwise, 0 if not. Such information is useful for medical practitioners to identify risk factors and use them in practice. Moreover, it provides a foundation for end-to-end systems to predict potential incidents for those who are at the risk of heart diseases, and assist clinicians in device selection and initiating early diagnosis and intervention procedures.

4. Implementation

4.1. Preliminary Analysis and Examining Features for Heart Disease Prediction: For implementing Extra Tree Classifier , ADABOOST , XGBoost, Logistic regression and KNN five different algorithms, the first step was to import the libraries like pandas, seaborn, matplotlib etc. Real data from heart disease patients was loaded and basic data profiling steps were conducted such as data missing analysis summary statistics and correlations were analyzed using heatmap.

Since the feature distributions were again more demanding to visualize from this distribution, the data was further subjected to bar plots disclosing the relations of the features such as chest pain type (cp), fasting blood sugar (fbs), electrocardiographic results (restecg) and many others targeted to the variable under study i.e., heart disease presence or its absence. The initial study proved valuable in examining the patterns and dependencies needed for developing the subsequent models. Each algorithm was then trained on the preprocessed data to try to predict the indices provided for the presence of heart disease which lays the groundwork for the immediate comparison in their predictive ability.

4.2 Data Division for Training and Evaluation: The dataset is separated into independent variables (X) and the dependent variable (y) where the feature variables are factors associated with heart health including age, sex, chest pain type, cholesterol, other relevant tests, and the results thereof. The target variable which defines the type of health condition is affected by the number of heart diseases. To evaluate the machine learning models, the dataset is divided into two halves in the ratio 90:10; a training set, and a testing/validation set, where 90% of the information is used to train the models, the other 10% being used to independently test and verify the accuracy of the models.

4.3 Applying Machine Learning Tools: ADABoost (ADB) classifier employs the concept of a linear kernel function. The model’s performance is assessed by accuracy, precision, recall, and the F1 rates. Train and test scores of the model suggest its effectiveness and robustness as to how it performs in unseen data as compared to a training set. KNN, a simple memorization first, nearest neighbors’ based learning algorithm wherein the new inputs are labeled with the tag that is most frequently found among the nearest neighbors. In this implementation we used k nearest neighbors where k=5 for predicting the heart disease.

Next, the Logistic Regression model and Extra Tree Classifier model are fit and evaluated in the same manner, and both types of analyses provide measures of predictive performance. Extra Tree Classifier model as a member of the large family of decision trees takes advantage of an ensemble technique to refine the prediction to the best accuracy by changing the random seed for a given run. Model selection and hyperparameter tuning, especially in Extra Tree Classifier and XGBoost models seem to enhance the best parameter settings for the models. The XGBoost model also shows how gradient boosting can improve the prediction by step by step parameter tuning including the gamma or learning rate, max depth and number of trees.

Altogether, the accuracies of the models always perform reasonably well but the fine-tuning of the hyperparameters and the scaling of the features, significantly influence overall prediction accuracy. XGBoost performed rather well, and this is confirmed by the high accuracy, precision, and AUC-ROC figures.

5. Results and Discussion

5.1 Overview of the Scores Obtained:

- **Precision:** Measures the correctness of the class predicted to be true, which is a ratio of true positives over all positive predictions.
- **Recall:** Measures the efficiency of a model to find all the instances in the dataset that are relevant; that is, it computes a ratio of true positives to the actual number of positive instances.
- **F1-score**: It is the harmonic mean of precision and recall, which represents both with equal metrics weight.
- **Support:** It represents the actual occurrences of each class in a dataset.
- **Accuracy:** How often the model predicts right, which could be defined as a total of correct predictions over the total number of instances.

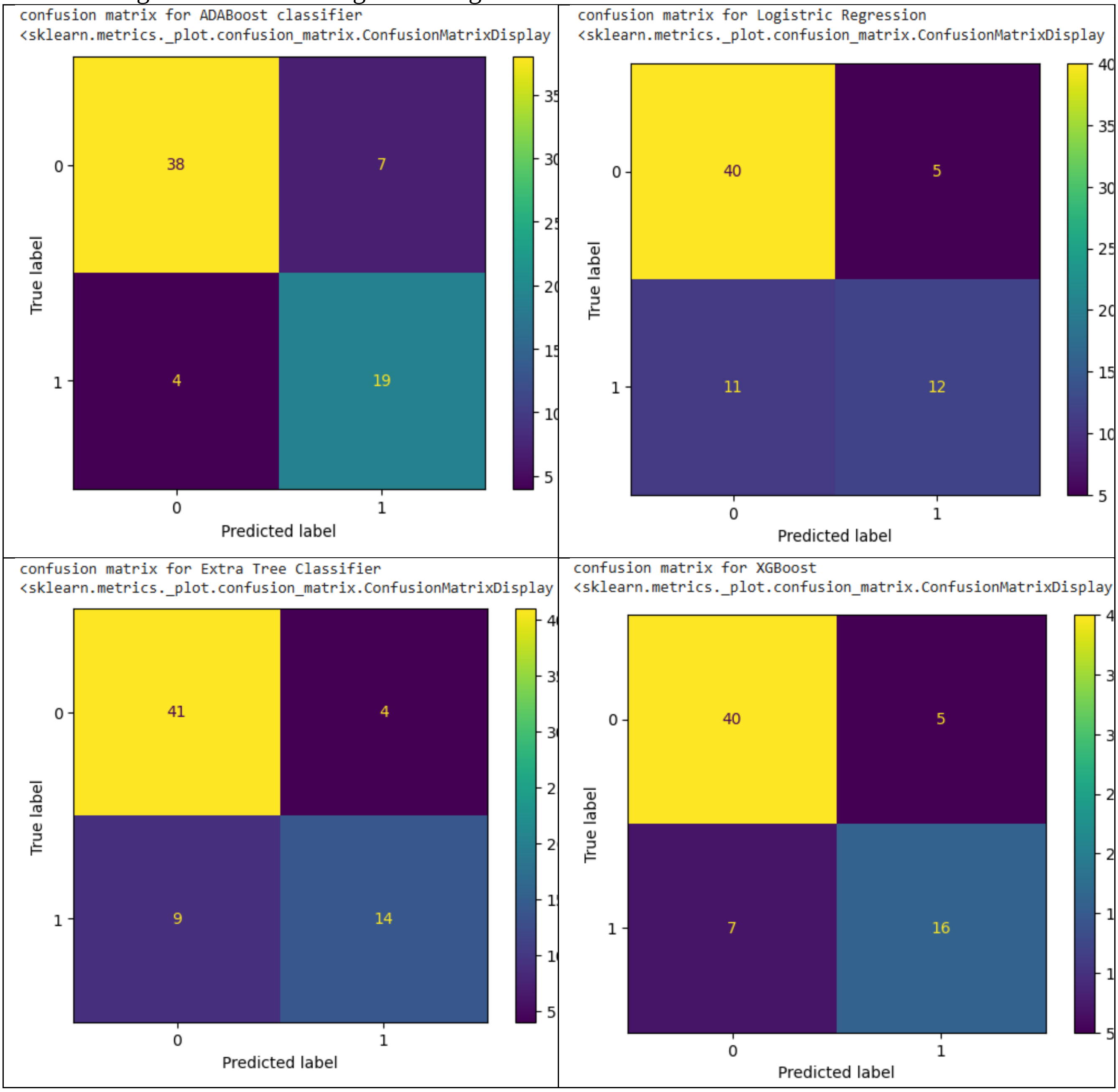
Summary of the above findings for each of the algorithm has been provided below in Table 3 -

Table 3: Scores Obtained for Each Algorithm

	Accuracy	**Precision**		**Recall**		**F1 Score**		**Support**
		Positive	**Negative**	**Positive**	**Negative**	**Positive**	**Negative**	**[Positive, Negative]**
ADA Boost (ADB)	0.84	0.90	0.73	0.84	0.83	0.87	0.78	[14 , 17]
Logistic Regression	0.76	0.78	0.71	0.89	0.52	0.83	0.60	[14 , 17]
XGBoost	0.82	0.85	0.76	0.89	0.70	0.87	0.73	[32 , 29]
K-nearest Neighbors	0.62	0.68	0.40	0.80	0.26	0.73	0.32	[32 , 29]
Extra Tree Classifier (ET)	0.81	0.82	0.78	0.91	0.61	0.86	0.68	[32, 29]
Bagging Classifier (BG)	0.66	0.66	0.00	1.00	0.00	0.80	0.00	[14 , 17]

The classification report provided is a comparative analysis of five machine learning models (Support Vector Machine, Logistic Regression, XGBoost, K-Nearest Neighbor, and Extra Tree Classifier) for the case of binary classification problem. XGBoost and Extra Tree Classifier , across the board, continued to outperform the other models on all metrics, indicative of their predictive power. The ADABOOST also performed very well with reasonable precision and recall measures. Logistic Regression and K-Nearest Neighbors with relatively lower performance well behind; precision and recall are even worse for K-Nearest Neighbors. Overall, both XGBoost and Extra Tree Classifier seem to be the best fit for this classification task.

5.2 Confusion Matrix for the Algorithms: It basically represents the ability of a classification model to produce results by evaluating actual and predicted values. Four out of such metrics are derived from it: True positives, False positives, True negatives, False negatives. These metrics represent cases in which the positive case is correctly labeled, instances incorrectly labeled as positive instances, instances correctly labeled as negative, and cases incorrectly labeled as negative respectively. The true negatives are negative instances correctly classified, and false negatives are the actual positive instances that were misclassified as negative. The confusion matrix helps us visualize the accuracy of the model in precisions, recall, distinguishing between classes. The confusion matrix of all the algorithms have been given in Figure 2.

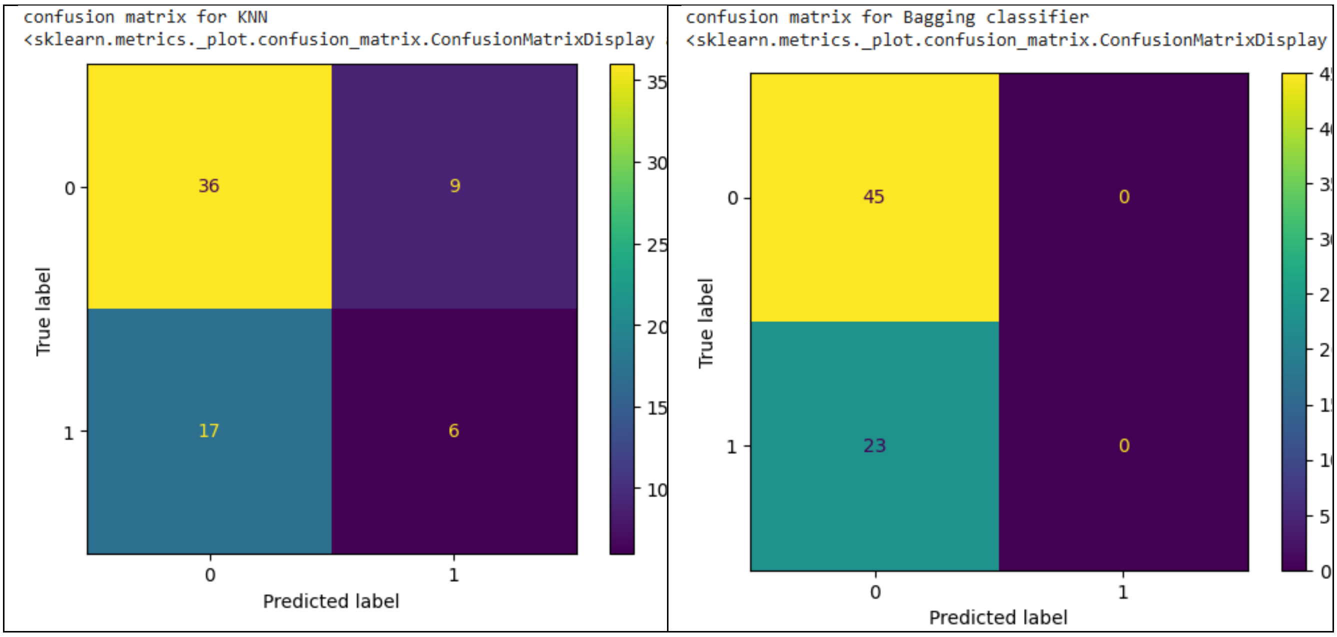


```
confusion matrix for ADABoost classifier
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay
```

```
confusion matrix for Logistic Regression
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay
```

```
confusion matrix for Extra Tree Classifier
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay
```

```
confusion matrix for XGBoost
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay
```



```
confusion matrix for KNN
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay
```

```
confusion matrix for Bagging classifier
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay
```

Figure 2 - Confusion Matrix of the Various Algorithms

he ADABOOST confusion matrix shows a favorable balance between true positives and true negatives correctly classified. With a few misclassifications, the vast majority of predictions are essentially concentrated in the diagonal section thanks to logistic regression. The model performs well on both positive and negative labels. In comparison with other models, the KNN confusion matrix indicates more misclassifications than others. Some of the quadrants with lighter shades indicate that the model has trouble accurately classifying some instances. An XGBoost matrix demonstrated excellent performance with very few misclassifications (5 false negatives and 4 false positives). Most of the predictions fall on the diagonal line of the matrix, indicating predictive accuracy. Extra Tree Classifier , much like XGBoost, reports a highly predictive matrix in which almost all instances are classified correctly, resulting in very few misclassifications; this reflects a well-functioning model.

5.3 Discussion

XGBoost and Extra Tree Classifier are powerful techniques for handling and producing results on intricately structured data, albeit expensive in terms of computing power, whereas Logistic Regression is quite straightforward and easy to explain, however it does not perform well in regard to complex forms of data. ADABOOST works with high-dimensional data excellently but its weaknesses come out when filth noise is present in the environment. KNN, on the other hand, is relatively straightforward to use and comprehend so many people can use it in various situations. Unfortunately, large amounts of data may lead to KNN having tremendous complexity and inordinate processing and resource expenses. The line plot comparing all the algorithms has been given in Figure 3 and a brief review has been given in Table 3.

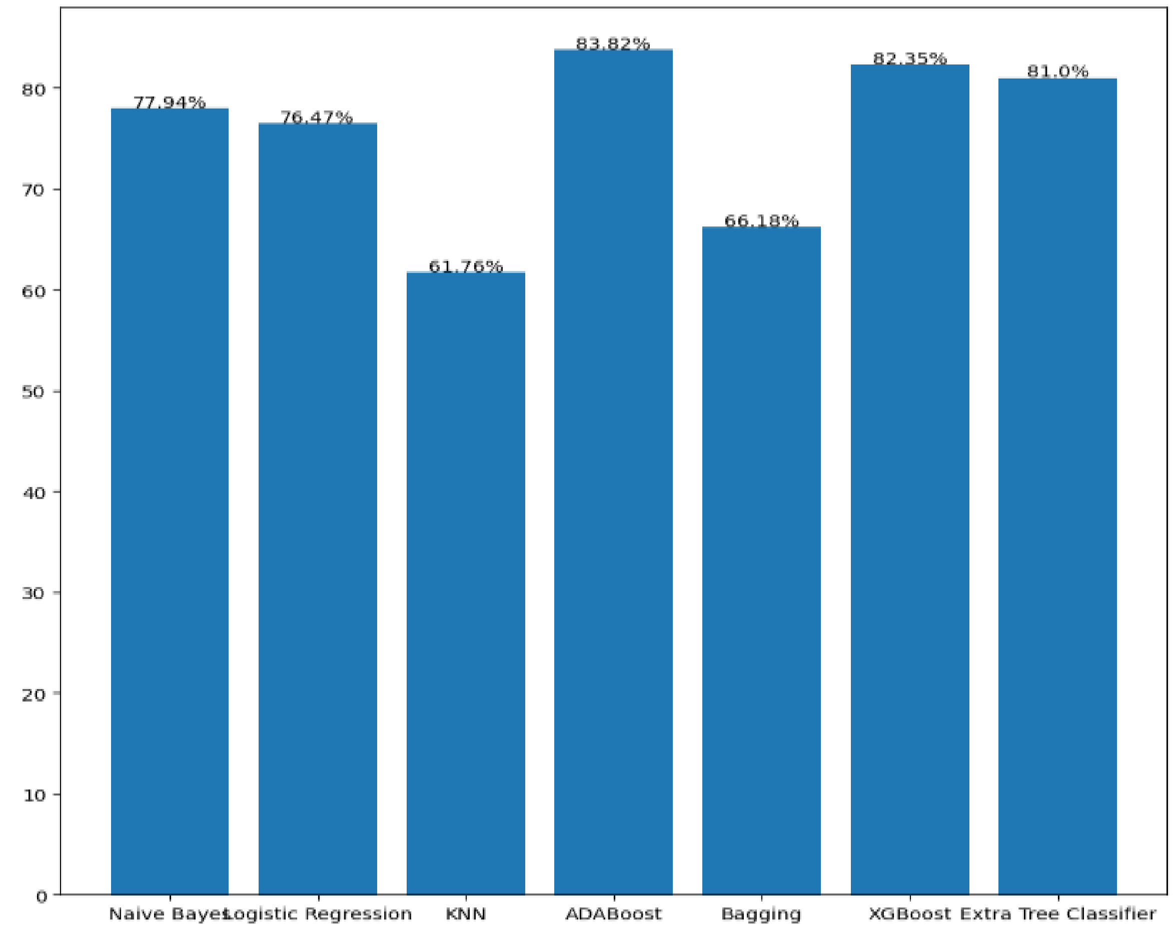


Figure 3 - Comparison of Machine Learning Models on Coronary Artery Disease Prediction

6. Conclusion

Comparative analysis reveals some of the basic differences in performances of machine learning algorithms in predicting heart disease, their strengths, and weaknesses. It has recorded the quite significant and highest accuracy of 83% by the ADABoost algorithm, which has indicated the capturing of complex patterns present in the dataset through ensemble learning. Its normalized performances across pertinent measures such as precision, recall, and F1 scores make it a strong and stable performer, particularly considering scenarios characterized by overfitting or noise. Close behind was XGBoost, boasting approximately 82% accuracy, thereby enjoying a competitive significance among highly complex data patterns by means of gradient boosting. However, owing to the computational overhead and complexity, it requires more resources and careful setup; hence, it is not an ideal option for resource- constrained environments.

Logistic Regression displayed good and consistent performance, achieving an accuracy of 76%, especially in recall with a score of 1.00 for positive classes, implying an ability to cover all true positive cases. However, being simple, it fails one or two long jumps to express more prominence in this dataset. Modest performance was attained by ADABOOST in terms of accuracy (83%), which could perform well within the constraints of recall, but fails miserably on precision and will not work well with large datasets for efficiency. Although K-Nearest Neighbor (KNN) is simple and easy to understand, the model had the worst accuracy level and was unable to represent the dataset's complexities, particularly as sizes grew, based on costly computational requirements.

While the Extra Tree Classifier emerged to be the top performer, the final verdict for decision-making shall be based on on-the-ground use, particularly justified by the available computational resources, concept interpretability of the model, and the depth of the data. The areas of hyperparameter tuning, feature selection, and balancing model complexity were crucial in improving the prediction performance of heart disease prediction for real-world scenarios.

7. Future Scope

This study opens up a variety of ways to improve the prediction of heart disease using machine learning in the future. Although some machine learning models like XGBoost and Extra Tree Classifier have already provided good results, many more improvements are still possible with better feature engineering and optimization. Of primary interest here is the real-time readiness of the XGBoost model through reduced computational complexity so that it can be utilized for life-sized applications. Besides, it is a great challenge in the dermatological field to overcome class imbalance in medical datasets, and subsequent studies should attempt to integrate also novel techniques like SMOTE and cost-sensitive learning to obtain better prediction accuracies for lackluster classes. Another avenue worthy of exploration is the merging of user information gathered from wearable devices and continuous health monitoring into regular Electronic Health Records in real-time for immediate intervention and truer predictions. But it entails associated challenges of integration, data processing, and privacy issues. All future work on feature selection, modeling methods, and class-imbalance treatment will help achieve practical viability and increased operational efficiency of machine learning tools in controlling cardiovascular disease and improving patient quality of life.

8. References

1. Ramalingam, V. V., Dandapath, A., & Raja, M. K. (2023). Heart disease prediction using machine learning techniques: a survey. International Journal of Engineering & Technology, 7(2.8), 684-687.
2. Sharma, H., & Rizvi, M. A. (2023). Prediction of heart disease using machine learning algorithms: A survey. International Journal on Recent and Innovation Trends in Computing and Communication, 5(8), 99-104.
3. Limbitote, M., Damkondwar, K., Mahajan, D., & Patil, P. (2022). A survey on prediction techniques of heart disease using machine learning. international journal of engineering research & technology (ijert), 9(6), 2278-0181.
4. Katarya, R., & Srinivas, P. (2022, July). Predicting heart disease at early stages using machine learning: a survey. In 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC) (pp. 302-305). IEEE.
5. Marimuthu, M., Abinaya, M., Hariesh, K. S., Madhankumar, K., & Pavithra, V. (2020). A review on heart disease prediction using machine learning and data analytics approach. International Journal of Computer Applications, 181(18), 20-25.
6. Khan, Y., Qamar, U., Yousaf, N., & Khan, A. (2020, February). Machine learning techniques for heart disease datasets: A survey. In Proceedings of the 2020 11th International Conference on Machine Learning and Computing (pp. 27-35).
7. Ahsan, M. M., & Siddique, Z. (2022). Machine learning-based heart disease diagnosis: A systematic literature review. Artificial Intelligence in Medicine, 128, 102289..
8. Fatima, M., & Pasha, M. (2019). Survey of machine learning algorithms for disease diagnostic. Journal of Intelligent Learning Systems and Applications, 9(01), 1-16.
9. Nadeem, M. W., Goh, H. G., Khan, M. A., Hussain, M., & Mushtaq, M. F. (2022). Fusion-Based Machine Learning Architecture for Heart Disease Prediction. Computers, Materials & Continua, 67(2).
10. Reddy, K. V. V., Elamvazuthi, I., Aziz, A. A., Paramasivam, S., Chua, H. N., & Pranavanand, S. (2022). Heart disease risk prediction using machine learning classifiers with attribute evaluators. Applied Sciences, 11(18), 8352.
11. Kavitha, B. S., & Siddappa, M. (2022). A survey on machine learning techniques to predict heart disease. International Journal of Computer Science & Communication, 11, 48-53.
12. Swathy, M., & Saruladha, K. (2023). A comparative study of classification and prediction of Cardio-Vascular Diseases (CVD) using Machine Learning and Deep Learning techniques. ICT express, 8(1), 109-116.
13. Kumar, M. N., Koushik, K. V. S., & Deepak, K. (2021). Prediction of heart diseases using data mining and machine learning algorithms and tools. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 3(3), 887-898.

14. Javid, I., Alsaedi, A. K. Z., & Ghazali, R. (2022). Enhanced accuracy of heart disease prediction using machine learning and recurrent neural networks ensemble majority voting method. International Journal of Advanced Computer Science and Applications, 11(3).
15. Sharma, G., Rani, G., & Dhaka, V. S. (2022, November). A review on machine learning techniques for prediction of cardiovascular diseases. In 2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC) (pp. 237-242). IEEE.
16. Rani, S., & Masood, S. (2022). Predicting congenital heart disease using machine learning techniques. Journal of Discrete Mathematical Sciences and Cryptography, 23(1), 293-303.
17. Sajja, G. S., Mustafa, M., Phasinam, K., Kaliyaperumal, K., Ventayen, R. J. M., & Kassanuk, T. (2022, August). Towards application of machine learning in classification and prediction of heart disease. In 2022 Second International Conference on Electronics and Sustainable Communication Systems (ICESC) (pp. 1664-1669). IEEE.
18. Desale, K. S., & Shinde, S. V. (2023). Addressing concept drifts using deep learning for heart disease prediction: a review. In Proceedings of Second Doctoral Symposium on Computational Intelligence: DoSCI 2021 (pp. 157-167). Springer Singapore.
19. Babu, S. V., Ramya, P., & Gracewell, J. (2024). Revolutionizing heart disease prediction with quantum-enhanced machine learning. Scientific Reports, 14(1), 7453.
20. Badawy, M., Ramadan, N., & Hefny, H. A. (2023). Healthcare predictive analytics using machine learning and deep learning techniques: a survey. Journal of Electrical Systems and Information Technology, 10(1), 40.