# Ensemble Learning Strategies for Enhancing Predictive Models in Cardiology

Ensemble Medi-Tech
Ayantanu Laha
Rajarshi Saha
ayantanulaha@gmail.com
saharajarshi7126@gmail.com

April 30, 2024

**Abstract**

The study aims to create a predictive model for cardiovascular disease (CVD) risk using machine learning techniques. An anonymized dataset including various factors such as age, gender, chest pain types, blood pressure, cholesterol, and more were analyzed. After comparing with the individual model performances we achieved a better accuracy using our proposed Stacking model. The final model achieves an accuracy of 93.68 %, 99.02 %, 98 %, and 73.66 % for Dataset 1, Dataset 2, Dataset 3, and Dataset 4 accordingly. It identifies key risk factors and assists doctors in appropriate actions. This underscores the potential of machine learning in CVD risk assessment, emphasizing its role in preventive care and the need for further research for wider clinical use.

## 1 Introduction

### 1.1 Objective

Our project focuses on creating a predictive model for cardiovascular disease (CVD). This model aims to analyze various factors associated with CVD risk, such as age, gender, medical history, and lifestyle factors, to predict the likelihood of an individual developing CVD.

### 1.2 Motivation

The motivation behind our project stems from the alarming increase in CVD-related deaths worldwide. As this health issue continues to pose a significant threat to public health, there is an urgent need for effective preventive measures. By developing a predictive model, we aim to empower healthcare professionals with a tool that can accurately assess CVD risk in individuals, enabling timely intervention and preventive strategies.

### 1.3 Method

We employ advanced machine learning techniques such as Decision Trees, k-nearest Neighbors, Bayesian decision, Random Forest, Support Vector Machines, and Logistic Regression to analyze complex datasets containing various CVD risk factors. Additionally, we use ensemble modeling, combining multiple models' predictions to enhance accuracy by leveraging their strengths and minimizing weaknesses. Through iterative refinement and validation, our goal is to develop a comprehensive and accurate predictive tool. This tool aims to assist healthcare professionals in identifying individuals at risk of CVD, facilitating timely intervention and personalized preventive strategies.

## 2 Literature Survey

Daza et al. [5] proposed a method using stacking models. They have shown that among four stacking models, stacking 1 combined with oversampling and AdaBoost-SVM with hyperparameter tuning exhibited superior performance, achieving a high accuracy of 88.24%. Alqahtani et al. [3] proposed six classification methods to predict cardiovascular disease. They proposed an ensemble learning method and voting-based decision-making to detect CV disease. They have used the random forest feature extraction process to identify the most informative features. They have also used Neural networks DNN and KDNN. The maximum accuracy they achieved is using an RF classifier, 88.65%. Mohd Syafiq Asyraf Suhaimi et al. [4] have used an ensemble approach, especially using stacking. They have used Logistic regression as the meta-level classifier, whereas RF and KNN are used as base-level classifiers. In the case of accuracy, the proposed model they used shows the same performance as RF individually, that is 82.42%, but the AUC score and F1 score are the highest. B. Rupa Devi et al. [7] proposed an ensemble method that uses a combination of two DNNs and as a meta-model XGBoost is used. Comparing the performances between models like RF, MLP, DT, and XGB, they have found that RF gives the best accuracy of 90%, whereas their proposed model Deep Neural Embedding (DEM) gives 93%, and that is the highest accuracy they have shown. G. Manikandan et al. [10] compared the performance of Logistic regression, Decision tree, and Support Vector Machine (SVM) with and without Boruta Feature Selection. They have shown that the Boruta feature selection enhanced the performance of the predictive models and achieved the highest accuracy of 88.52% using Logistic Regression. Ahmed Al Ahdal et al. [1] used feature selection methods, temporal analysis, and clustering techniques. They have shown that the RF and XGBoost achieve higher accuracy than other models, that is 96.72% and 95.08%, respectively. Subhash Mondal et al. [11] have used two different datasets. They have taken ML classifiers like XGB, RF, DT, SVM, and LR and applied hyperparameters Randomized CV, and Grid Search CV on each model and applied for both datasets. The best accuracy they achieved among all the classifiers irrespective of the dataset, is 96.88%.

# 3 Proposed Methodology

In this section, we detail the methodology utilized for building the models, comprising the following stages: Data Set, Cleaning and Pre-processing, Training and Testing Data, Modeling, and Evaluation, which was developed in the data science environment *Anaconda*, in the open-source application *VS Code*, and the Python programming language.

## 3.1 Data Set

- **Dataset 1:** `https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction`
  - This dataset consists of 1190 observations and 12 attributes (11 predictor variables and 1 predicted variable).
- **Dataset 2:** `https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset`
  - This dataset consists of 1025 observations and 14 attributes (13 predictor variables and 1 predicted variable).
- **Dataset 3:** `https://data.mendeley.com/datasets/dzz48mvjht/1`
  - This dataset consists of 1000 observations and 14 attributes (13 predictor variables and 1 predicted variable).
- **Dataset 4:** `https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset`
  - This dataset consists of 70000 observations and 12 attributes (11 predictor variables and 1 predicted variable).

## 3.2 Cleaning and Pre-processing

- The data was reviewed using Python commands, and it was observed that there are no missing values present. Consequently, there was no need to perform any data transformations, as it was determined that the values within the features were suitable for our proposed algorithm.

### 3.2.1 Dataset 1

Regarding the correlation of the variables, a strong relationship between the predictor variables and the predicted variables is shown in Figure 1.

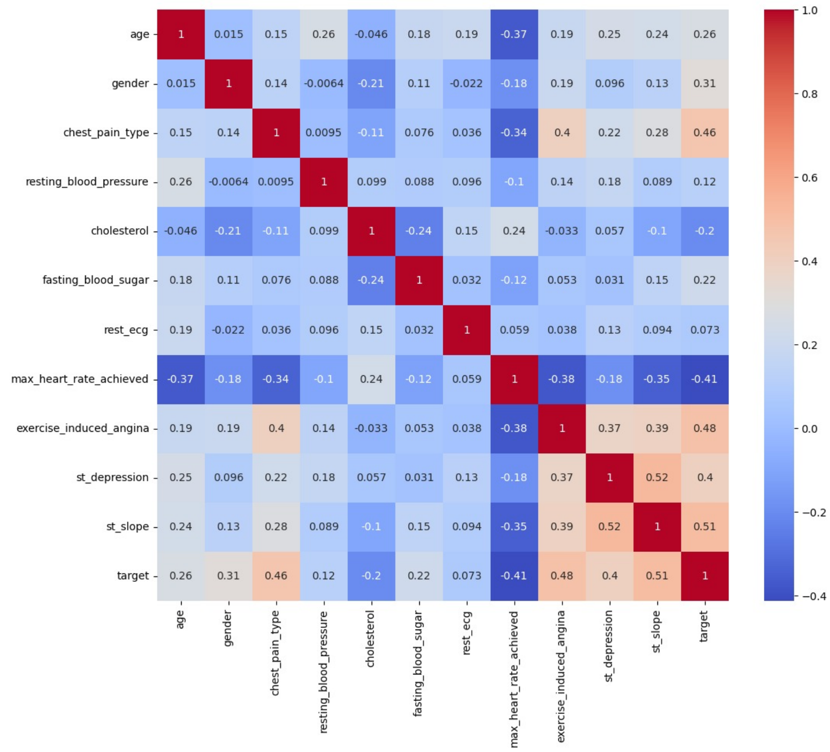



Figure 1: Correlation plot for Dataset 1.

The male-female ratio in this dataset is shown in Figure 2.

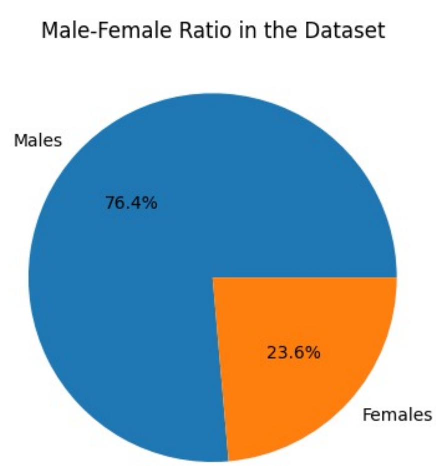



Figure 2: Male-Female ratio for Dataset 1.

The percentage of heart disease patients is shown in Figure 3.

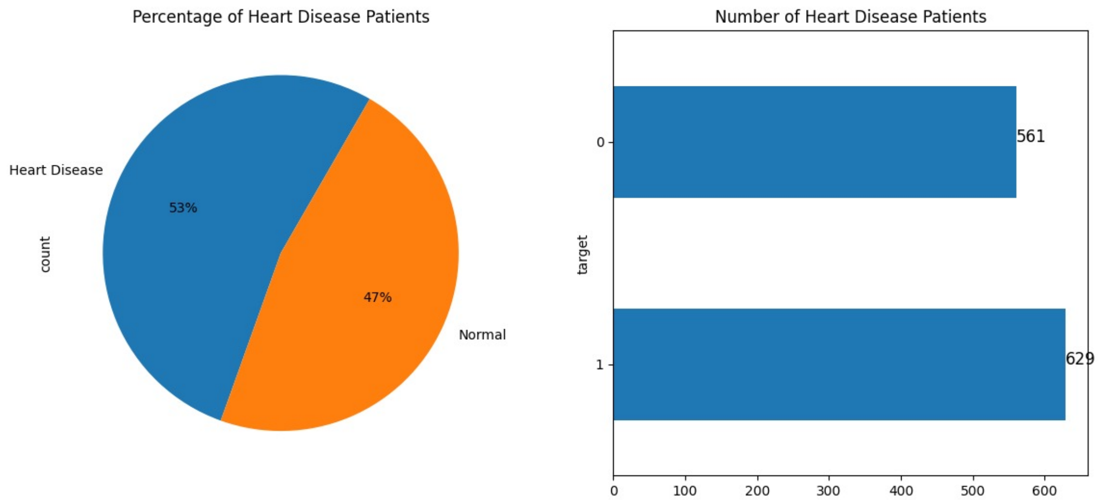



Figure 3: Percentage of Heart Disease Patients for Dataset 1.

### 3.2.2 Dataset 2

Regarding the correlation of the variables, a strong relationship between the predictor variables and the predicted variables is shown in Figure 4.

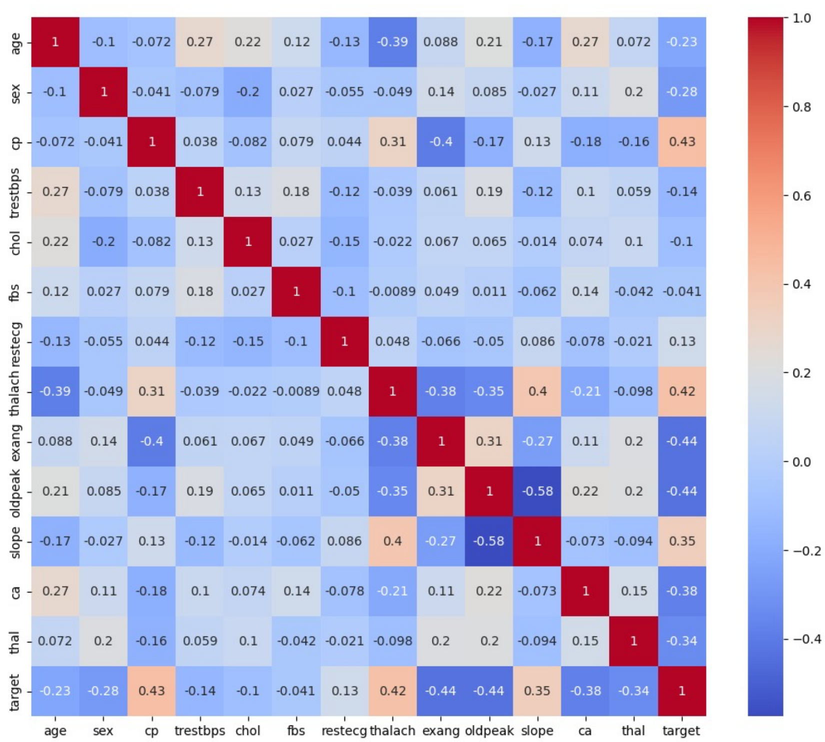



Figure 4: Correlation plot for Dataset 2.

The male-female ratio in this dataset is shown in Figure 5.

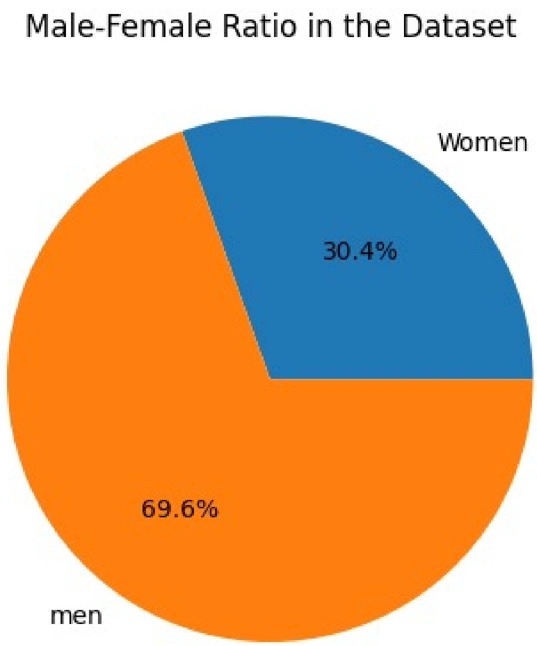



Figure 5: Male-Female ratio for Dataset 2.

The percentage of heart disease patients is shown in Figure 6.

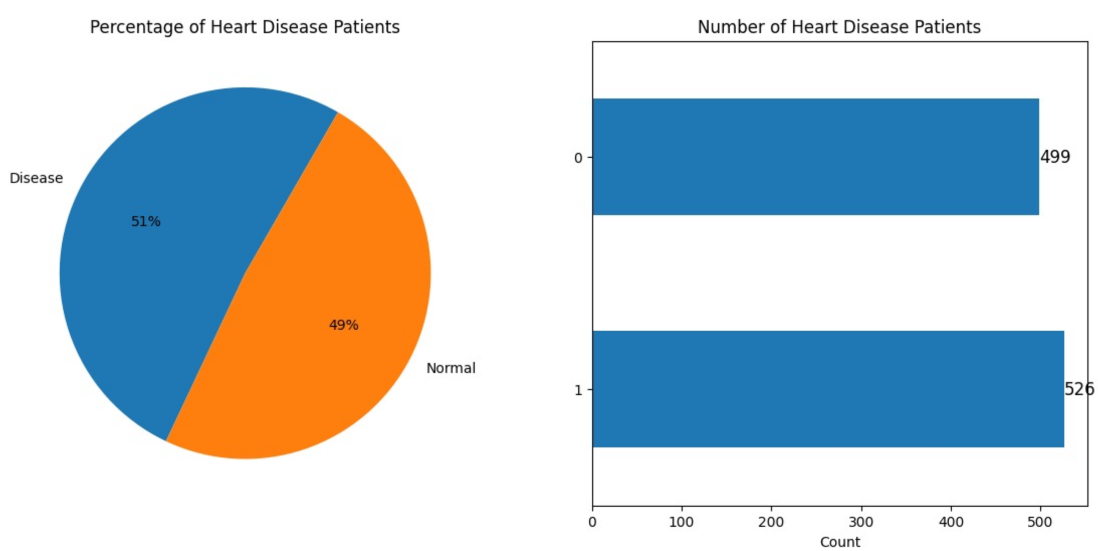



Figure 6: Percentage of Heart Disease Patients for Dataset 2.

### 3.2.3 Dataset 3

Regarding the correlation of the variables, a strong relationship between the predictor variables and the predicted variables is shown in Figure 7.

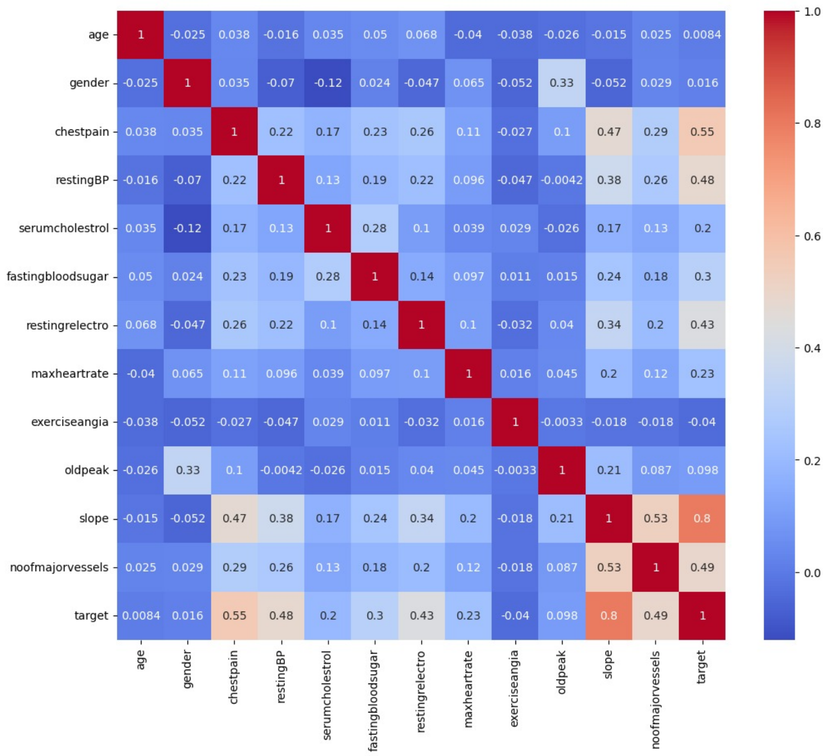



Figure 7: Correlation plot for Dataset 3.

The male-female ratio in this dataset is shown in Figure 8.

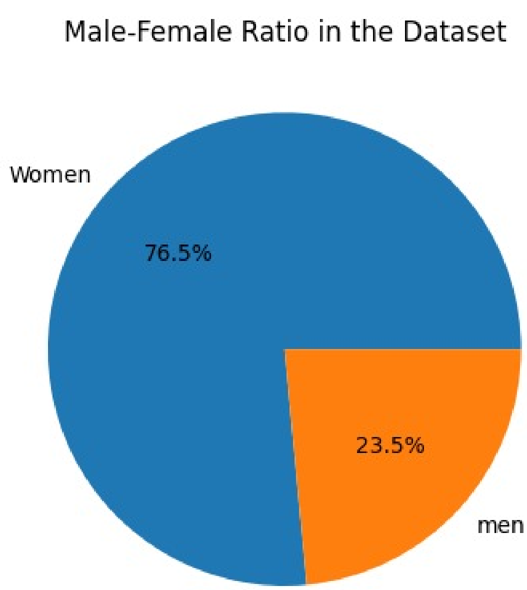



Figure 8: Male-Female ratio for Dataset 3.

The percentage of heart disease patients is shown in Figure 9.

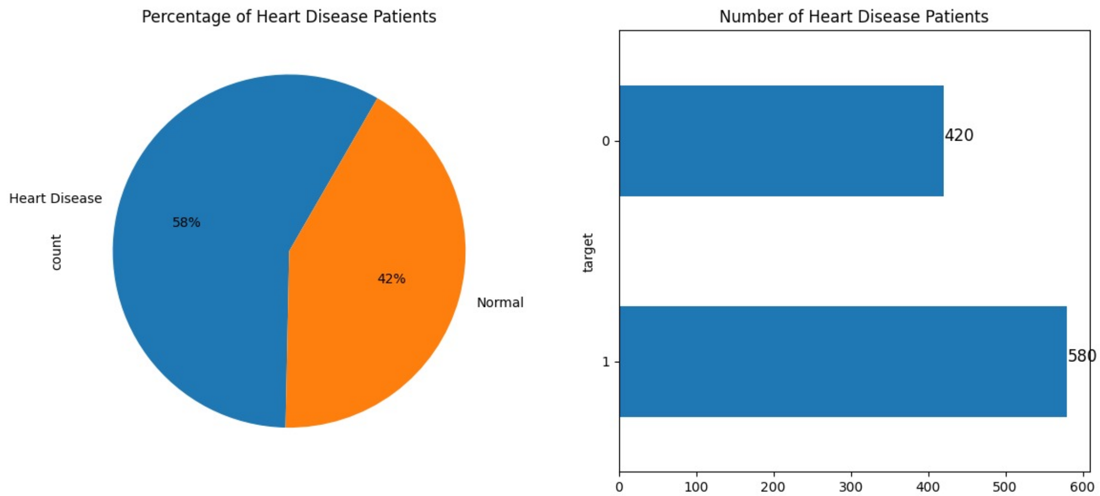



Figure 9: Percentage of Heart Disease Patients for Dataset 3.

### 3.2.4 Dataset 4

Regarding the correlation of the variables, a strong relationship between the predictor variables and the predicted variables is shown in Figure 10.

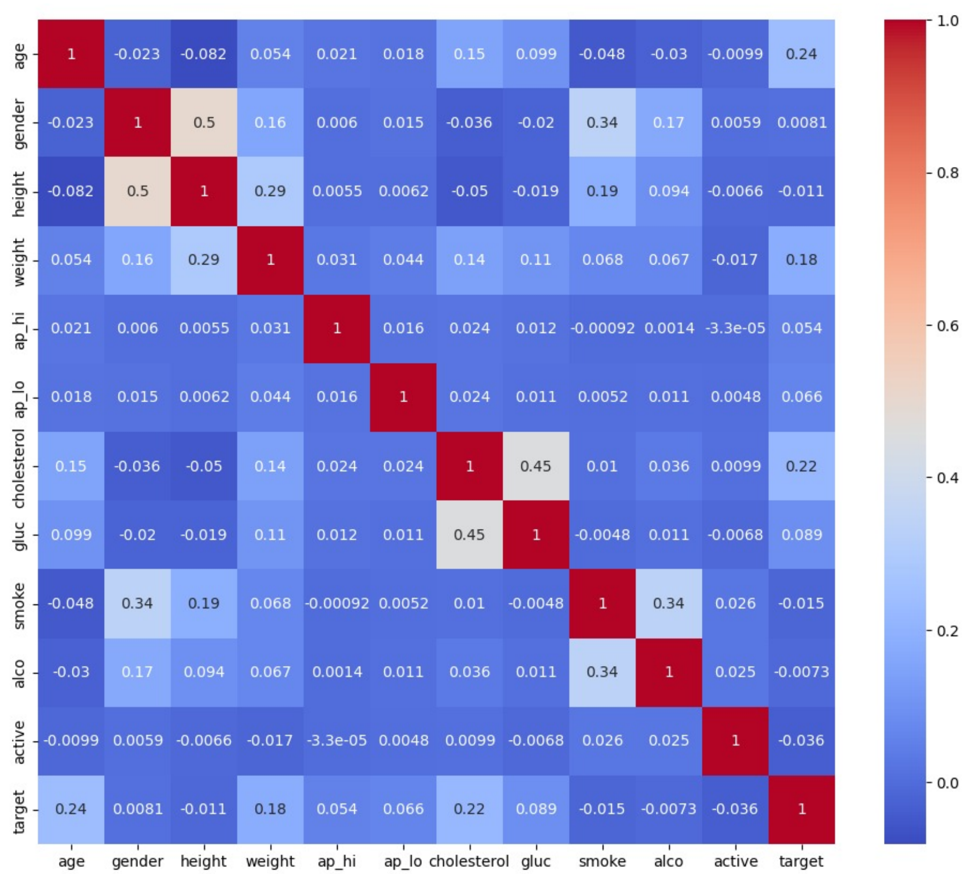



Figure 10: Correlation plot for Dataset 4.

The male-female ratio in this dataset is shown in Figure 11.

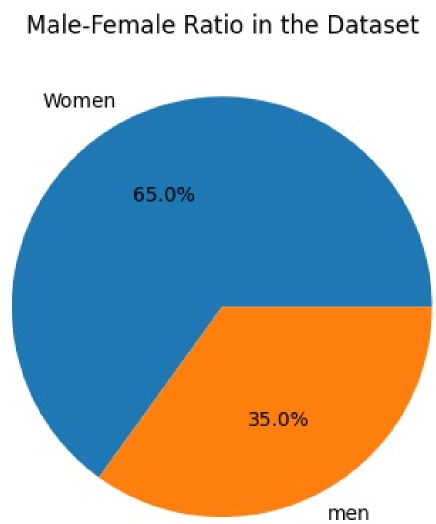



Figure 11: Male-Female ratio for Dataset 4.

The percentage of heart disease patients is shown in Figure 12.

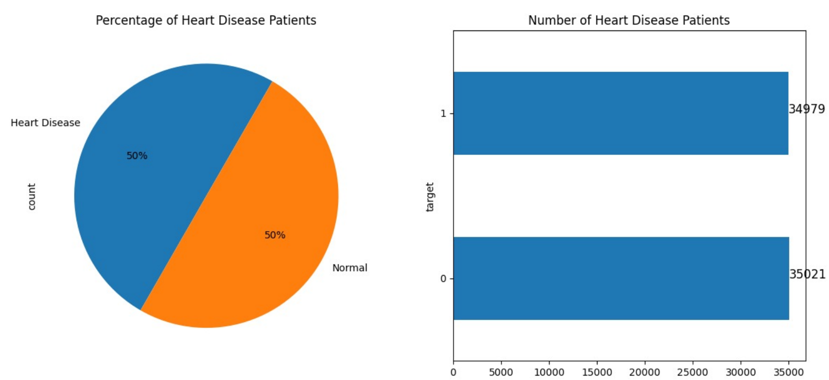



Figure 12: Percentage of Heart Disease Patients for Dataset 4.

## 3.3 Train-Test-Validation Spliting

In this step, we split the dataset into training, validation, and testing sets using the `train_test_split` function from the `sklearn.model_selection` module. Therefore, the splits are as follows:

- Training set: 64% of the original dataset
- Validation set: 16% of the original dataset
- Testing set: 20% of the original dataset

## 3.4 Modeling and evaluation

For the modeling stage, we employed a diverse set of machine learning algorithms to explore different modeling approaches and capture the complexity of the data. The models utilized include Decision Tree, Gradient Boost, K-nearest neighbors (KNN), Random Forest, Logistic regression, Support Vector Machines (SVM), and Bayesian networks. In addition to the individual models, four combined models were proposed using a technique known as stacking.[17] Each stacking model consists of two levels:

#### 3.4.1 Stacking 1

- **Level 0**: Decision Tree, Random Forest, Bayesian Networks, and Gradient Boosting are utilized as base models to generate predictions on the dataset.
- **Level 1**: Majority Voting is employed to combine the predictions from the base models. It learns to predict the target variable based on the outputs of the base models.

#### 3.4.2 Stacking 2

- **Level 0**: Similar to Stacking 1, Decision Tree, SVM, KNN, and Gradient Boosting serve as base models.
- **Level 1**: Majority Voting is employed to combine the predictions from the base models. It learns to predict the target variable based on the outputs of the base models.

#### 3.4.3 Stacking 3

- **Level 0**: Again, Decision Tree, Random Forest, KNN, and Gradient Boosting are employed as base models.
- **Level 1**: Majority Voting is employed to combine the predictions from the base models. It learns to predict the target variable based on the outputs of the base models.

#### 3.4.4 Stacking 4

- **Level 0**: Utilizing the same base models—Decision Tree, Logistic Regression, SVM, and Gradient Boosting.
- **Level 1**: Majority Voting is employed to combine the predictions from the base models. It learns to predict the target variable based on the outputs of the base models.

By stacking multiple models in this manner, the combined models aim to exploit the complementary strengths of individual algorithms and improve overall predictive performance.

Next, we combine the four stacking models using a technique called majority voting. This ensemble method aggregates the predictions of the individual stacking models and selects the final prediction based on the majority vote.

The process involves:

- Obtaining predictions from each of the four stacking models.
- For each data point, determine the most frequently predicted class among the stacking models.
- Assigning the class with the highest frequency as the final prediction for the data point.

By leveraging the collective decision-making of multiple models through majority voting, the ensemble aims to enhance predictive accuracy and robustness, effectively capturing diverse perspectives from the constituent models which is shown in Figure 13.

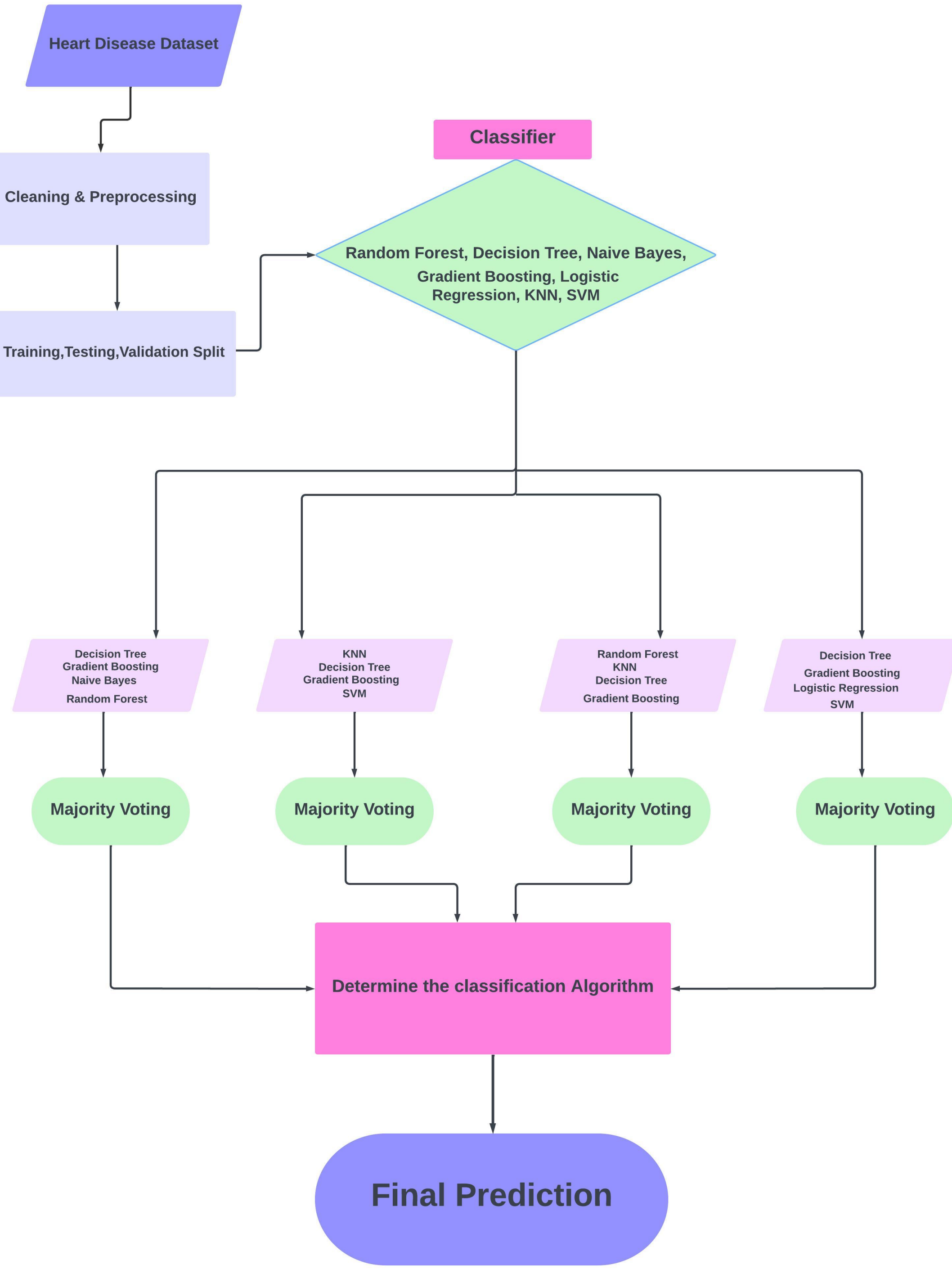



Figure 13: Proposed Model

On the other hand, various metrics were used to evaluate the effectiveness of the models, such as Accuracy, Precision, F1-Score, and Recall presented in the experimental result Section.

# 4 Experimental Result

## 4.1 Parameters used

In stacking models, parameters are configured at two levels: base learners and the final meta-learner. Base learner parameters, like decision tree depth or k-nearest neighbors, are algorithm-specific, while the meta-learner's parameters govern how it combines base model predictions for the final output. The parameters we used are mentioned in fig **??**.

```
rf=RandomForestClassifier(criterion='entropy', max_features='sqrt')
dt= DecisionTreeClassifier (criterion='entropy', max_depth=5, min_samples_leaf=5,
splitter='random')
gb=GradientBoostingClassifier(max_depth=4, min_samples_leaf=20)
gnb-GaussianNB(var_smoothing=0.004328761281083057)
svm=SVC(C=100, gamma 'auto', probability=True)
knn=KNeighbors Classifier (metric='manhattan', n_neighbors=9, weights='distance')
lr=LogisticRegression (C=0.615848211066026, max_iter=500, penalty='11', solver='liblinear')
```

Figure 14: Stacking Models

## 4.2 Performance Metric

### 4.2.1 Accuracy

Accuracy measures the proportion of correctly classified instances among all instances, calculated by dividing the number of correct classifications by the total instances.

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

### 4.2.2 Precision

Precision quantifies the number of correctly predicted positive cases out of all instances predicted as positive. It is calculated by dividing the number of true positives by the sum of true positives and false positives.

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

### 4.2.3 Recall

Recall measures the proportion of correctly predicted positive cases out of all actual positive cases. It is calculated by dividing the number of true positives by the sum of true positives and false negatives.

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

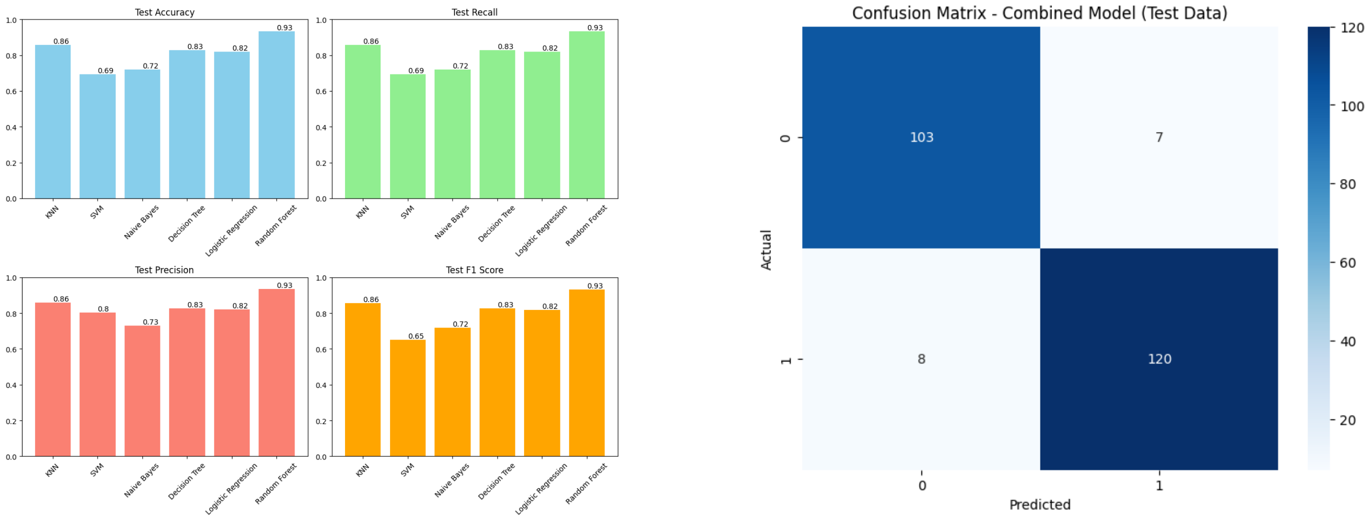
### 4.2.4 F1 Score

The F1 score combines both precision and recall into a single metric by taking their harmonic mean. It provides a balance between precision and recall.

$$\text{F1 Score} = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)}$$

## 4.3 Model Performance

### 4.3.1 Experimented on dataset 1




(a) Individual Performance (b) Confusion Matrix

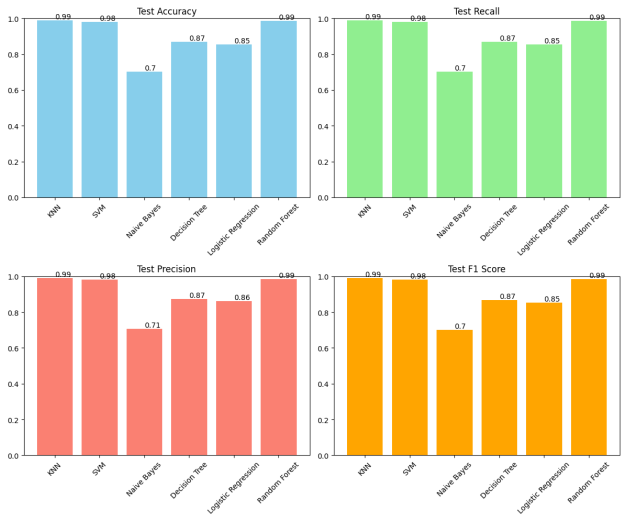
Figure 15: For Dataset with 1190 instances

Performance Comparison between different Algorithms

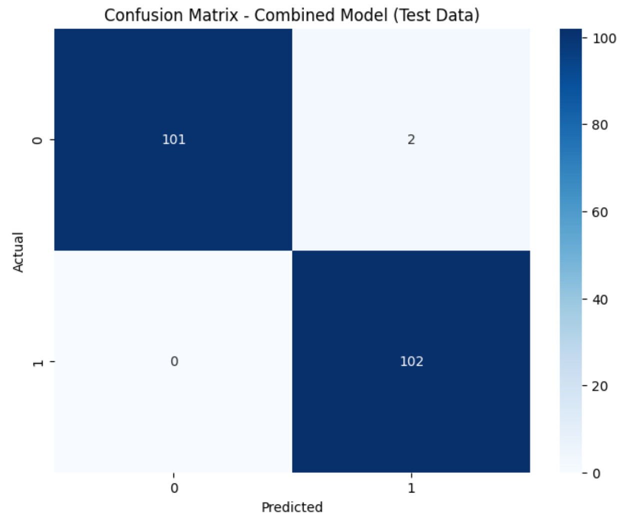
| Model Type | Algorithm | Accuracy | | Precision | | Recall | | F1 Score | |
|---|---|---|---|---|---|---|---|---|---|
| | | Validation | Test | Validation | Test | Validation | Test | Validation | Test |
| Individual Model | KNN | 80.63% | 85.71% | 80.81% | 85.75% | 80.63% | 85.71% | 80.54% | 85.68% |
| | SVM | 68.06% | 69.33% | 80.16% | 80.47% | 68.06% | 69.33% | 63.74% | 65.11% |
| | NB | 73.40% | 71.85% | 73.62% | 72.91% | 73.29% | 71.85% | 73.29% | 71.82% |
| | DT | 81.15% | 82.77% | 81.17% | 82.76% | 81.15% | 82.77% | 81.12% | 82.75% |
| | LR | 81.16% | 81.93% | 81.30% | 81.94% | 81.15% | 81.93% | 81.08% | 81.94% |
| | RF | 89% | 93.28% | 89.02% | 93.41% | 89% | 93.28% | 88.99% | 93.29% |
| Stack | Stacking 1 | 89.53% | 93.70% | 89.66% | 93.70% | 89.53% | 93.70% | 89.50% | 93.70% |
| | Stacking 2 | 88.48% | 91.60% | 88.48% | 91.73% | 88.48% | 91.60% | 88.48% | 91.61% |
| | Stacking 3 | 89.53% | 91.18% | 89.54% | 91.28% | 89.53% | 91.18% | 89.52% | 91.19% |
| | Stacking 4 | 90.05% | 92.02% | 90.08% | 92.02% | 90.05% | 92.02% | 90.04% | 92.02% |
| Majority Voting | Combined | 90.58% | 93.68% | 90.63% | 93.70% | 90.57% | 93.70% | 90.56% | 93.70% |

Table 1: Dataset 1 (1190)

### 4.3.2 Experimented on dataset 2




(a) Individual Performance




(b) Confusion Matrix

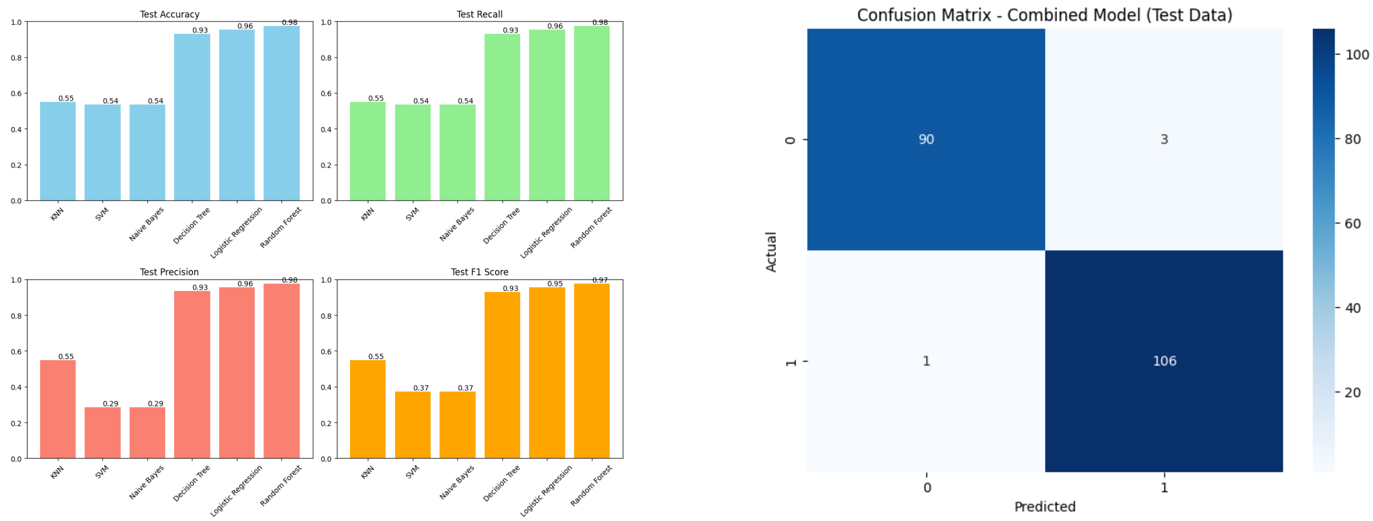
Figure 16: For Dataset with 1025 instances

Performance Comparison between different Algorithms

| **Model Type** | **Algorithm** | **Accuracy** | | **Precision** | | **Recall** | | **F1 Score** | |
|---|---|---|---|---|---|---|---|---|---|
| | | Validation | Test | Validation | Test | Validation | Test | Validation | Test |
| Individual | KNN | 96.95% | 99.02% | 97.12% | 99.04% | 96.95% | 99.02% | 96.95% | 99.02% |
| | SVM | 96.34% | 98.04% | 96.59% | 98.12% | 96.34% | 98.05% | 96.34% | 98.05% |
| | NB | 70.73% | 70.24% | 70.75% | 70.59% | 70.73% | 70.24% | 70.73% | 70.13% |
| | DT | 88.41% | 86.83% | 88.42% | 87.26% | 88.42% | 86.83% | 88.41% | 86.79% |
| | LR | 81.09% | 85.37% | 81.22% | 86.26% | 81.09% | 85.36% | 81.08% | 85.28% |
| | RF | 97% | 98.54% | 96.96% | 98.54% | 97% | 98.54% | 96.95% | 98.54% |
| Stack | Stacking 1 | 98.78% | 99.02% | 98.81% | 99.04% | 98.78% | 99.02% | 98.78% | 99.02% |
| | Stacking 2 | 98.78% | 99.02% | 98.81% | 99.04% | 98.78% | 99.02% | 98.78% | 99.02% |
| | Stacking 3 | 96.95% | 98.54% | 96.96% | 98.54% | 96.95% | 98.54% | 96.95% | 98.54% |
| | Stacking 4 | 98.78% | 99.02% | 98.81% | 99.04% | 98.78% | 99.02% | 98.78% | 99.02% |
| Majority Voting | Combined | 98.78% | 99.02% | 98.80% | 99.04% | 98.78% | 99.02% | 98.78% | 99.02% |

Table 2: Dataset 2 (1025)

### 4.3.3 Experimented on dataset 3




(a) Individual Performance

(b) Confusion Matrix

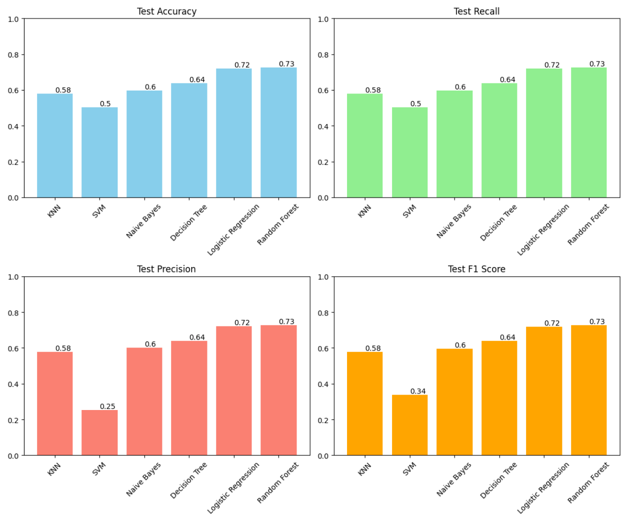
Figure 17: For Dataset with 1000 instances

Performance Comparison between different Algorithms

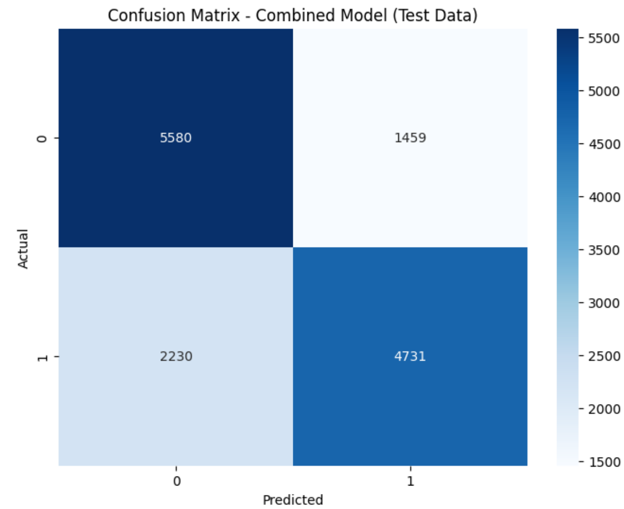
| **Model Type** | **Algorithm** | **Accuracy** | | **Precision** | | **Recall** | | **F1 Score** | |
|---|---|---|---|---|---|---|---|---|---|
| | | Validation | Test | Validation | Test | Validation | Test | Validation | Test |
| Individual Model | KNN | 49.37% | 55.00% | 47.44% | 54.88% | 49.38% | 55.00% | 48.19% | 54.91% |
| | SVM | 61.87% | 53.50% | 38.28% | 28.62% | 61.87% | 53.50% | 47.30% | 37.29% |
| | NB | 61.87% | 53.50% | 38.28% | 28.62% | 61.87% | 53.50% | 47.30% | 37.29% |
| | DT | 93.75% | 93.00% | 93.94% | 93.47% | 93.75% | 93.00% | 93.75% | 93.00% |
| | LR | 95.62% | 95.50% | 95.65% | 95.50% | 95.62% | 95.50% | 95.63% | 95.49% |
| | RF | 97% | 97.50% | 96.87% | 97.53% | 97% | 97.50% | 96.87% | 95.49% |
| Stack | Stacking 1 | 96.25% | 98.00% | 96.25% | 98.02% | 96.25% | 98.00% | 96.25% | 98.00% |
| | Stacking 2 | 98.12% | 97.00% | 98.18% | 97.01% | 98.12% | 97.00% | 98.12% | 97.00% |
| | Stacking 3 | 98.12% | 97.00% | 98.18% | 97.07% | 98.12% | 97.00% | 98.12% | 96.99% |
| | Stacking 4 | 97.50% | 98.50% | 97.52% | 98.50% | 97.50% | 98.50% | 97.49% | 98.50% |
| Majority Voting | Combined | 97.50% | 98.00% | 97.52% | 98.01% | 97.50% | 98.00% | 97.49% | 97.99% |

Table 3: Dataset 3 (1000)

### 4.3.4 Experimented on dataset 4




(a) Individual Performance




(b) Confusion Matrix

Figure 18: For Dataset with 70000 instances

Performance Comparison between different Algorithms

| **Model Type** | **Algorithm** | **Accuracy** | | **Precision** | | **Recall** | | **F1 Score** | |
|---|---|---|---|---|---|---|---|---|---|
| | | Validation | Test | Validation | Test | Validation | Test | Validation | Test |
| Individual Model | KNN | 57.96% | 57.91% | 58.03% | 57.91% | 57.96% | 57.91% | 57.94% | 57.89% |
| | SVM | 49.59% | 50.26% | 75.00% | 25.28% | 49.58% | 50.26% | 32.88% | 33.63% |
| | NB | 60.19% | 59.80% | 60.29% | 59.98% | 60.19% | 59.80% | 60.03% | 59.65% |
| | DT | 63.84% | 63.90% | 63.85% | 63.94% | 63.84% | 63.90% | 63.82% | 63.89% |
| | LR | 72.13% | 71.91% | 73.33% | 72.01% | 72.13% | 71.91% | 72.09% | 71.87% |
| | RF | 72% | 72.68% | 72.37% | 72.23% | 72% | 71.27% | 72.25% | 72.65% |
| Stack | Stacking 1 | 73.62% | 73.66% | 73.97% | 73.89% | 73.62% | 73.66% | 73.54% | 73.59% |
| | Stacking 2 | 73.40% | 73.68% | 73.85% | 73.98% | 73.40% | 73.68% | 73.30% | 73.58% |
| | Stacking 3 | 73.13% | 73.29% | 73.84% | 73.77% | 73.13% | 73.29% | 72.96% | 73.13% |
| | Stacking 4 | 73.63% | 73.69% | 74.19% | 74.06% | 73.63% | 73.69% | 73.51% | 73.58% |
| Majority Voting | Combined | 73.64% | 73.66% | 74.09% | 73.95% | 73.64% | 73.65% | 73.54% | 73.56% |

Table 4: Dataset 4 (70000)

## 4.4 Comparison with different Literature Survey

### 4.4.1 Dataset 1

| Paper | Algorithm | Accuracy (%) | Precision (%) | Recall (%) | F1 Score (%) |
|---|---|---|---|---|---|
| Doppala et al. [8] | DT | 82.56 | 83 | 79 | 81 |
| | RF | 90.75 | 88 | 93 | 90 |
| | NB | 84.24 | 82 | 85 | 84 |
| | LR | 84.03 | 81 | 87 | 84 |
| | SVM | 81.52 | 82 | 84 | 83 |
| | Ensemble Model | 93.39 | 99 | 88 | 90 |
| Daza et al. [5] | RF | 88.5 | 88.95 | 88.8 | 88.5 |
| | Ensemble Model | 91.5 | 91.5 | 91.6 | 91.49 |
| Ozcan et al. [13] | Cart Algorithm | 87.25 | 88.24 | 84.51 | - |
| Tiwari et al. [14] | RF | 90.21 | 87.31 | - | 91.05 |
| | KNN | 80.85 | 78.67 | - | 82.62 |
| | SVM | 82.55 | 80.14 | - | 84.16 |
| | Ensemble Model | 92.34 | 92 | - | 92.74 |
| Our approach | KNN | 85.71 | 85.75 | 85.71 | 85.68 |
| | SVM | 69.33 | 80.47 | 69.33 | 65.11 |
| | NB | 71.85 | 72.91 | 71.85 | 71.82 |
| | DT | 82.77 | 82.76 | 82.77 | 82.75 |
| | LR | 81.93 | 81.94 | 81.93 | 81.94 |
| | RF | 93.28 | 93.41 | 93.28 | 93.29 |
| | Proposed Method | 93.68 | 93.7 | 93.7 | 93.7 |

Table 5: Performance Comparison for **Dataset 1**

### 4.4.2 Dataset 2

| Paper | Algorithm | Accuracy (%) | Precision (%) | Recall (%) | F1 Score (%) |
|---|---|---|---|---|---|
| Asyraf et al. [4] | LR | 81.32 | - | 91.49 | 83.5 |
| | KNN | 74.73 | - | 89.36 | 78.5 |
| | RF | 82.42 | - | 85.11 | 83.3 |
| | Ensemble Model | 82.42 | - | 91.49 | 84.3 |
| Nawaz et al. [12] | SVM | 75 | - | 75 | - |
| | KNN | 66.73 | - | 68 | - |
| | NB | 83 | - | 78 | - |
| | RF | 92.48 | - | 91.17 | - |
| | Grad. Desc. Opti. | 98.54 | - | 99.43 | - |
| Our Approach | KNN | 99.02 | 99.04 | 99.02 | 99.02 |
| | SVM | 98.04 | 98.12 | 98.05 | 98.05 |
| | NB | 70.24 | 70.59 | 70.24 | 70.13 |
| | DT | 86.83 | 87.26 | 86.83 | 86.79 |
| | LR | 85.37 | 86.26 | 85.36 | 85.28 |
| | RF | 98.54 | 98.54 | 98.54 | 98.54 |
| | Proposed Method | 99.02 | 99.04 | 99.02 | 99.02 |

Table 6: Performance Comparison for **Dataset 2**

#### 4.4.3 Dataset 3

| Paper | Algorithm | Accuracy (%) | Precision (%) | Recall (%) | F1 Score (%) |
|---|---|---|---|---|---|
| Doppala et al. [8] | DT | 95 | 96 | 95 | 95 |
| | RF | 95.12 | 97 | 94 | 96 |
| | NB | 94.25 | 94 | 95 | 94 |
| | LR | 95.25 | 97 | 97 | 97 |
| | SVM | 93.15 | 93 | 95 | 93 |
| | Ensemble Model | 96.75 | 98 | 96 | 97 |
| Mondal et al. [11] | RF | 96 | 96 | 96 | 96 |
| | DT | 93 | 92 | 92 | 92 |
| | LR | 87 | 87 | 86 | 86 |
| | SVM | 78 | 77 | 78 | 77 |
| | Ensemble Model | 96.88 | 96 | 97 | 97 |
| Our Approach | KNN | 55 | 54.88 | 55 | 54.91 |
| | SVM | 53.5 | 28.62 | 53.5 | 37.29 |
| | NB | 53.5 | 28.62 | 53.5 | 37.29 |
| | DT | 93 | 93.47 | 93 | 93 |
| | LR | 95.5 | 95.5 | 95.5 | 95.49 |
| | RF | 97.5 | 97.53 | 97.5 | 95.49 |
| | Proposed Method | 98 | 98.01 | 98 | 97.99 |

Table 7: Performance Comparison for **Dataset 3**

#### 4.4.4 Dataset 4

| Paper | Algorithm | Accuracy (%) | Precision (%) | Recall (%) | F1 Score (%) |
|---|---|---|---|---|---|
| Delima et al. [6] | LR | 72.35 | - | - | - |
| | DT | 61.72 | - | - | - |
| | RF | 68.94 | - | - | - |
| | SVM | 72.16 | - | - | - |
| | KNN | 68.34 | - | - | - |
| | Genetic Algorithm ANN | 73.43 | - | - | - |
| Alfaidi et al. [2] | LR | 85.54 | 90.01 | 84.34 | 87.09 |
| | RF | 86.03 | 92.82 | 82.19 | 87.19 |
| | DT | 85.93 | 93.57 | 81.24 | 86.97 |
| | NB | 83.38 | 93.65 | 76.44 | 84.18 |
| | KNN | 84.56 | 88.97 | 83.66 | 86.24 |
| | SVM | 86.63 | 96.06 | 80.16 | 87.39 |
| | MLP | 87.23 | 95.23 | 82.01 | 88.13 |
| Alqahtani et al. [3] | RF | 88.65 | 90.03 | 88.03 | 88.02 |
| | KNN | 86.45 | 87.53 | 86.21 | 86.25 |
| | DT | 86.35 | 86.23 | 86.22 | 86.22 |
| | Stacking | 86.49 | 87.32 | 86.02 | 86.01 |
| | Ensemble | 88.7 | 88.02 | 88.02 | 88.01 |
| Our Approach | KNN | 57.91 | 57.91 | 57.91 | 57.89 |
| | SVM | 50.26 | 25.28 | 50.26 | 33.63 |
| | NB | 59.8 | 59.98 | 59.8 | 59.65 |
| | DT | 63.9 | 63.94 | 63.9 | 63.89 |
| | LR | 71.91 | 72.01 | 71.91 | 71.87 |
| | RF | 72.68 | 72.23 | 71.27 | 72.65 |
| | Proposed Method | 73.66 | 73.95 | 73.65 | 73.56 |

Table 8: Performance Comparison for **Dataset 4**

### 4.5 Complexity Analysis

In this subsection, we present the computational complexity of individual and ensemble algorithms. In machine learning, model complexity often depends on the number of extracted features and samples in the training set. In ensemble learning, the execution time of meta-learners is typically negligible, and they have minimal impact on the running time of base classifiers. Stacking applies several individual learners to training data and then combines the output of them using a meta-learner.
The overall complexity of stacking is $O(f_1 + f_2 + f_3, \ldots, f_n)$, where $n = 1, \ldots, N$, and $f_n$ denotes the time complexity of each individual learner.[16] To write computation complexity we assume :

- $n$: number of training examples
- $d$: number of dimensions of the data
- $k$: number of neighbors

For more details, click here [9].

| **Algorithm** | **Training Time Complexity** |
|---|---|
| K Nearest Neighbors | $O(knd)$ |
| Logistic Regression | $O(nd)$ |
| SVM | $O(n^2)$ |
| Decision Tree | $O(n \log(n) d)$ |
| Random Forest | $O(n \log(n) dk)$<br>(k=number of Decision Trees) |
| Naive Bayes | $O(nd)$ |

Table 9: Complexity Analysis of Machine Learning Algorithms

# 5 Summary

We have studied four different datasets and applied six different classification algorithms to each. Upon comparing our results with those of existing research projects, we have observed significant improvements using our approach of ensemble modeling on three of the datasets. Our proposed method performs better than previous approaches on these three datasets. However, for dataset 4, we have not achieved satisfactory accuracy. Future work will focus on improving the model for this dataset.

# References


[1] Ahmed Al Ahdal, Manik Rakhra, Rahul R. Rajendran, Farrukh Arslan, Moaiad Ahmad Khder, Binit Patel, Balaji Ramkumar Rajagopal, and Rituraj Jain. [retracted] monitoring cardiovascular problems in heart patients using machine learning. *Journal of Healthcare Engineering*, 2023:9738123, Feb 2023.

[2] Aseel Alfaidi, Reem Aljuhani, Bushra Alshehri, Hajer Alwadei, and Sahar Sabbeh. Machine learning: Assisted cardiovascular diseases diagnosis. *International Journal of Advanced Computer Science and Applications*, 13, 01 2022.

[3] Abdullah Alqahtani, Shtwai Alsubai, Mohemmed Sha, Lucia Vilcekova, and Talha Javed. Cardiovascular disease detection using ensemble learning. *Computational Intelligence and Neuroscience*, 2022:5267498, Aug 2022.

[4] Syafiq Asyraf, Nor Azuana Ramli, and Noryanti Muhammad. Heart disease prediction using ensemble of k-nearest neighbour, random forest and logistic regression method. page 040009, 03 2024.

[5] Alfredo Daza, Juana Bobadilla, Juan Carlos Herrera, Angelica Medina, Nemias Saboya, Karoline Zavaleta, and Segundo Siguenas. Stacking ensemble based hyperparameters to diagnosing of heart disease: Future works. *Results in Engineering*, 21:101894, 2024.

[6] Allemar Jhone Delima and Jan Carlo Arroyo. An optimized neural network using genetic algorithm for cardiovascular disease prediction. *Journal of Advances in Information Technology*, 13:95–99, 01 2022.

[7] B. Rupa Devi, U. Sivaji, Thammisetty Swetha, J. Avanija, A. Suresh, and K. Reddy Madhavi. Advanced cardiovascular disease prediction: A comparative analysis of ensemble stacking and deep neural networks. *International Journal of Intelligent Systems and Applications in Engineering*, 12(6s):46–55, Nov. 2023.

[8] Bhanu Prakash Doppala, Debnath Bhattacharyya, Midhunchakkaravarthy Janarthanan, and Namkyun Baik. A reliable machine intelligence model for accurate identification of cardiovascular diseases using ensemble techniques. *Journal of Healthcare Engineering*, 2022:2585235, Mar 2022.

[9] Paritosh Kumar. Time Complexity of ML Models — medium.com. `https://medium.com/analytics-vidhya/time-complexity-of-ml-models-4ec39fad2770#:~:text=Time%20complexity%20can%20be%20seen,http://bigocheatsheet.com/`. [Accessed 30-04-2024].

[10] G. Manikandan, B. Pragadeesh, V. Manojkumar, A.L. Karthikeyan, R. MANIKANDAN, and Amir Gandomi. Classification models combined with boruta feature selection for heart disease prediction. *Informatics in Medicine Unlocked*, 44, 01 2024.

[11] Subhash Mondal, Ranjan Maity, Yachang Omo, Soumadip Ghosh, and Amitava Nag. An efficient computational risk prediction model of heart diseases based on dual-stage stacked machine learning approaches. *IEEE Access*, 12:7255–7270, 2024.

[12] Muhammad Saqib Nawaz, Bilal Shoaib, and Muhammad Adeel Ashraf. Intelligent cardiovascular disease prediction empowered with gradient descent optimization. *Heliyon*, 7(5):e06948, 2021.

[13] Mert Ozcan and Serhat Peker. A classification and regression tree algorithm for heart disease modeling and prediction. *Healthcare Analytics*, 3:100130, 2023.

[14] Achyut Tiwari, Aryan Chugh, and Aman Sharma. Ensemble framework for cardiovascular disease prediction. *Computers in Biology and Medicine*, 146:105624, 2022.

[15] Ilias Tougui, Abdelilah Jilbab, and Jamal El Mhamdi. Heart disease classification using data mining tools and machine learning techniques. *Health and Technology*, 10(5):1137–1144, Sep 2020.

[16] Zhongliang Zhao, Mostafa Karimzadeh, Florian Gerber, and Torsten Braun. Mobile crowd location prediction with hybrid features using ensemble learning. *Future Generation Computer Systems*, 110:556–571, 2020.

[17] Z.H. Zhou and S. Liu. *Machine Learning*. Springer Nature Singapore, 2021.