Performance Analysis of Classification Models
Heart Failure Classification Problem

Amir Ragaie 21010300
Abdallah Adel 21010808
Ebrahim Alaa Eldin 21010017
Team 24

March 2025

Abstract

This report evaluates the performance of three machine learning models (Decision Tree, Bagging, and AdaBoost) using accuracy, F1-score, and confusion matrices. The results highlight model strengths.

Contents

1 Introduction and Dataset Specification

This dataset contains medical records for heart disease prediction, with 918 samples and 12 features. The features include both continuous variables like Age, RestingBP (resting blood pressure), Cholesterol, and MaxHR (maximum heart rate), as well as categorical variables like Sex, ChestPainType (type of chest pain), and ExerciseAngina (exercise-induced angina). The target variable 'HeartDisease' is binary, where 1 indicates presence of heart disease and 0 indicates absence. The dataset provides comprehensive information about patients' cardiovascular health indicators, making it valuable for developing predictive models for heart disease diagnosis.

2 Dataset and Preprocessing

The dataset used is the **Heart Disease dataset**. Key preprocessing steps include:

- Encoding categorical variables:
  - **Sex**: Converts Male (M) $\rightarrow$ 1 and Female (F) $\rightarrow$ 0.
  - **ExerciseAngina**: Converts Yes (Y) $\rightarrow$ 1 and No (N) $\rightarrow$ 0.
  - This transformation ensures these categorical variables can be used in machine learning models.
  - Identifies non-binary categorical columns (those that are not numerical).
  - Uses One-Hot Encoding (OHE) to convert these categorical features into numerical form.
  - Replaces original categorical columns with their one-hot encoded versions.
- Splitting data into different sets:
  - **Training Set (70%)**: Used for model training.
  - **Validation Set (10%)**: Used for hyperparameter tuning.
  - **Test Set (20%)**: Used for final model evaluation.

3 Models and Hyperparameter Tuning

We trained the following models:

- **Decision Tree**
  - **Parameters Being Tuned:**
    - **max_depth**: Specifies the depth of the tree, ranging from 5 to 50.
    - **min_samples_split**: Controls the minimum number of samples required to split a node, tested with values from 2 to 100.
- **Bagging Classifier**
  - **Parameter Being Tuned:**
    - **n_estimators**: The number of base learners (Decision Trees) in the bagging ensemble. The tested values range from 20 to 50, increasing in steps of 5.
- **AdaBoost Classifier**
  - **Parameter Being Tuned:**
    - **n_weak_learners**: The number of weak learners (Stumps) used in AdaBoost. The values tested are 20, 50, and 100.

Each model underwent hyperparameter tuning using validation data.

4 Results and Performance Comparison

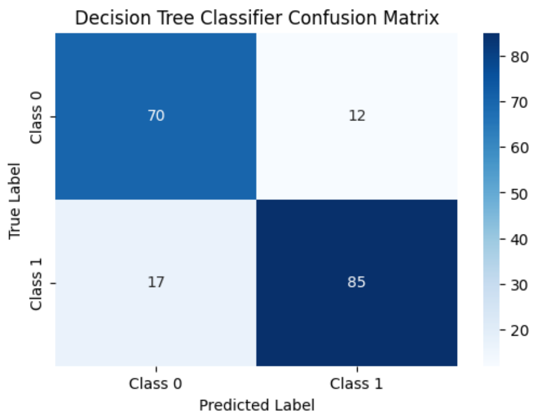
4.1 Accuracy and F1-Score

The following table summarizes the performance metrics:

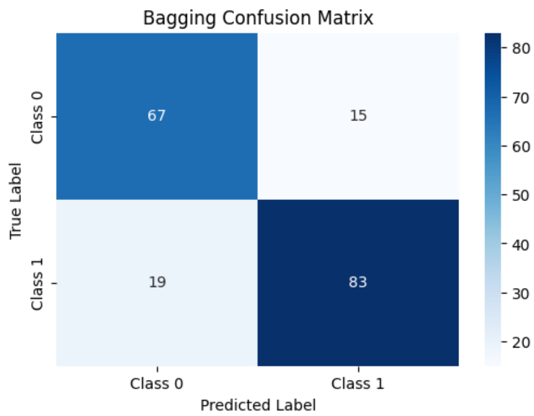
Model	Accuracy	F1-Score
Decision Tree	0.842	0.854
Bagging Classifier	0.815	0.830
AdaBoost Ensemble	0.783	0.800

Table 1: Performance Metrics of Models

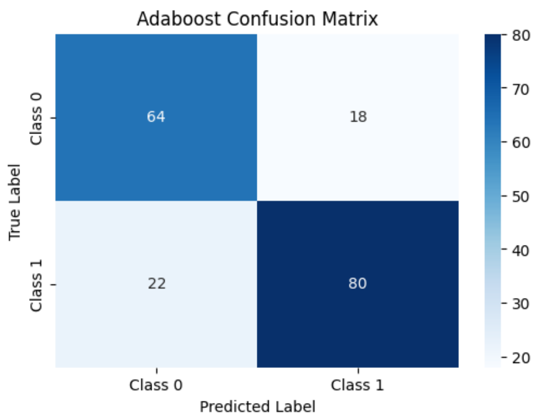
4.2 Cofusion Matrices



(a) Confusion Matrix for Decision Tree



(b) Confusion Matrix for Bagging



(c) Confusion Matrix for AdaBoost

Figure 1: Comparison of Confusion Matrices for Decision Tree, Bagging, and AdaBoost

5 Analysis and Insights

5.1 Best Performing Model: Decision Tree

- The **Decision Tree achieved the highest accuracy (0.842)**, making the most correct predictions.
- It also obtained the **highest F1-score (0.854)**, indicating a good balance between precision and recall.

5.2 Bagging vs. Decision Tree

- Surprisingly, **Bagging (0.815 accuracy) performed worse than a single Decision Tree**.
- The main reason might be **dataset characteristics**:
  - **Bagging works best when Decision Trees are unstable** (i.e., when small changes in data lead to significantly different trees).
  - If the dataset does not have high variance or noise, Bagging may not provide significant improvements.
  - If the Decision Tree is already optimal, Bagging may not help much.
- **F1-Score of Bagging (0.830)** is close to that of the Decision Tree, but still slightly lower.

5.3 AdaBoost vs. Other Models

- AdaBoost has the lowest **Accuracy (0.783)**, suggesting it struggles in this dataset.
- The **F1-Score (0.800)** indicates more misclassifications compared to other models.
- **AdaBoost is sensitive to noisy data and outliers**, which might have led to performance degradation.

6 Confusion Matrices and Insights

6.1 Decision Tree Classifier

	Predicted Class 0	**Predicted Class 1**
Actual Class 0	70	12
Actual Class 1	17	85

Table 2: Confusion Matrix for Decision Tree Classifier

Observations:

- **False Positives (FP):** 12 instances of Class 0 were misclassified as Class 1.
- **False Negatives (FN):** 17 instances of Class 1 were misclassified as Class 0.
- The model shows a slight tendency to misclassify Class 1 instances as Class 0.
- However, it achieves high performance with **85 correctly classified Class 1 instances**.

6.2 Bagging Classifier

	Predicted Class 0	**Predicted Class 1**
Actual Class 0	67	15
Actual Class 1	19	83

Table 3: Confusion Matrix for Bagging Classifier

Observations:

- **False Positives (FP):** 15 instances of Class 0 were misclassified as Class 1.
- **False Negatives (FN):** 19 instances of Class 1 were misclassified as Class 0.
- Bagging shows a slightly higher misclassification rate than the Decision Tree.
- The increased misclassification may be due to **bias introduced by aggregating multiple Decision Trees**.

6.3 AdaBoost Classifier

	Predicted Class 0	**Predicted Class 1**
Actual Class 0	64	18
Actual Class 1	22	80

Table 4: Confusion Matrix for AdaBoost Classifier

Observations:

- **False Positives (FP):** 18 instances of Class 0 were misclassified as Class 1.
- **False Negatives (FN):** 22 instances of Class 1 were misclassified as Class 0.
- AdaBoost has the highest misclassification rates among the three models.
- The lower performance may be due to **AdaBoost's sensitivity to noisy data**, leading to overfitting on certain misclassified points.

7 Key Misclassification Patterns

- **Decision Tree is the most balanced** with relatively low false positives and false negatives.
- **Bagging increases misclassification slightly**, likely due to averaging multiple Decision Trees, which can introduce bias.
- **AdaBoost has the worst misclassification rates**, showing it is less stable on this dataset, likely due to its **sensitivity to noise**.

8 Final Comparison

Model	Strengths	Weaknesses
Decision Tree	High accuracy, interpretable	Can overfit to training data
Bagging	Reduces variance, more stable	May not help if base model is already optimal
AdaBoost	Can improve weak learners	Sensitive to noisy data, lowest accuracy

Table 5: Strengths and Weaknesses of the models.

9 Conclusion

- **Decision Tree is the best performer** in this dataset, correctly classifying the most instances.
- **Bagging, while reducing variance, does not significantly improve accuracy** and slightly increases misclassification.
- **AdaBoost is the least effective model**, as it struggles with both false positives and false negatives.

10 Bonus

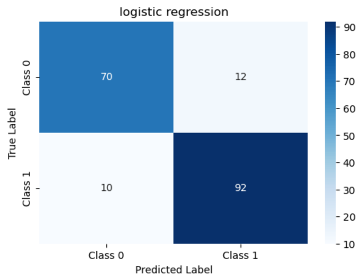
10.1 Accuracy and F1-Score

The following table summarizes the performance metrics:

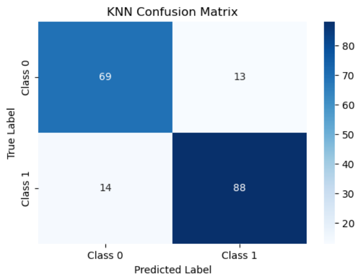
Model	Accuracy	F1-Score
KNN	0.853	0.867
Logistic Regression	0.880	0.893
Neural Network	0.869	0.884

Table 6: Performance Metrics of Models

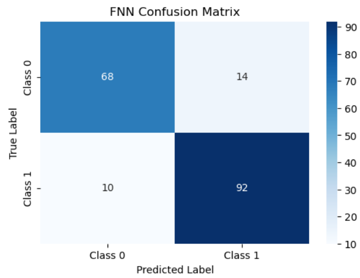
10.2 Cofusion Matrices



(a) Confusion Matrix for Logistic Regression



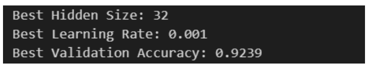
(b) Confusion Matrix for KNN



(c) Confusion Matrix for Neural Network

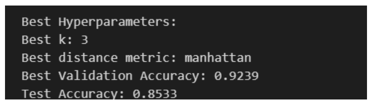
Figure 2: Comparison of Confusion Matrices for Logistic Regression, FNN, KNN

10.3 Tuning



```
Best Hidden Size: 32
Best Learning Rate: 0.001
Best Validation Accuracy: 0.9239
```

(a) NN



```
Best Hyperparameters:
Best k: 3
Best distance metric: manhattan
Best Validation Accuracy: 0.9239
Test Accuracy: 0.8533
```

(b) KNN

Figure 3: Hyperparamters Tuning