**Jana alzobidi**

**2211632**

**CCAI-312: Pattern Recognition**

**Heart Failure Prediction Dataset**

Logo, company name

Description automatically generated

**hanan awadh**

**2216930**

**Amal Faleh Alharbi**

**2210781**

# INTRODUCTION

A white rectangular box with black text

Description automatically generated

With notable variations in prevalence and prognosis among various populations, heart disease continues to be a major cause of death globally. Advanced models for predicting the risk of heart disease do exist, but their wider applicability is limited by their reliance on intricate datasets or methodology. In order to forecast the risk of heart disease in a variety of populations, this study presents a straightforward utilizing the ensemble learning technique known as bagging. We train and verify a Bagging model specifically designed for heart disease prediction, with a focus on performance, interpretability, and adaptability, using publicly available health data. We assess our Bagging model's efficacy by contrasting its results with those of two other well-known models: Random Forest and K-Nearest Neighbors (KNN). This study demonstrates how Bagging can provide scalable, effective, and easily available prediction tools for early detection and intervention. Additionally, it offers relatively accurate predictions, helping to minimize unnecessary costs for individuals who are incorrectly diagnosed as having the disease. The ultimate goal is to lower the burden of heart disease and advance health equity among various populations.

**Figure 2. Class proportions**

.

Figure 1. Class distribution

.

# PROBLEM DESCRIPTION

According to the World Health Organization, heart disease is the leading cause of death worldwide, accounting for an estimated 17.9 million deaths every year. The early detection and correct prediction of heart disease are very crucial in preventing serious outcomes. However, heart disease is a complex disease that depends on many factors such as age, lifestyle habits, and medical history, making prediction difficult.  
  
Misdiagnosis of heart disease can be very serious. On the other hand, it was stated in the studies that a lot of heart disease cases can remain undiagnosed or be misdiagnosed in younger people and women, respectively, due to atypical symptoms. Delays in treatment are usually caused by such kinds of diagnoses, resulting in increasing risks of complications such as heart attacks, strokes, or even death. On the other hand, false positives lead to unnecessary treatments, emotional trauma, and wasting of money.

Traditional models like single decision trees, though simple and interpretable, often struggle with overfitting and poor generalization, limiting their effectiveness in real-world applications. Addressing these limitations, this project employs **Bagging (Bootstrap Aggregating)** with decision trees to improve prediction performance. Bagging creates an ensemble of models trained on different subsets of data, reducing variance and enhancing accuracy.

# .DATA DESCRIPTION

## Dataset Overview

The dataset used for this project, titled Indicators of Heart Disease, contains 918 records and 12 features. The target variable, heart disease, is a binary categorical feature indicating whether an individual has ever been diagnosed with coronary heart disease or myocardial infarction, with a significant class imbalance in its distribution as shown in Figure 1 and Figure 2.

## A blue and red circle with black text Description automatically generated

## Feature Description

The dataset includes a mix of numerical, categorical, and binary features, described as follows:

The dataset includes a mix of numerical, categorical, and binary features, described as follows:

* **Numerical Features**:
  + **Age:** Age of the individual.
  + **RestingBP:** Resting blood pressure (in mm Hg).
  + **Cholesterol:** Serum cholesterol levels (in mg/dL).
  + **MaxHR**: Maximum heart rate achieved.
  + **Oldpeak**: ST depression induced by exercise relative to rest.
* **Categorical Features**:
  + **Sex**: Gender of the individual (M for Male, F for Female).
  + **ChestPainType**: Type of chest pain experienced (ATA, NAP, ASY, TA).
  + **RestingECG**: Resting electrocardiogram results (Normal, ST, LVH).
  + **ST\_Slope**: Slope of the peak exercise ST segment (Up, Flat, Down).
* **Binary Features**:
  + **FastingBS**: Fasting blood sugar > 120 mg/dL (0: No, 1: Yes).
  + **ExerciseAngina**: Exercise-induced angina (N: No, Y: Yes).
  + **HeartDisease**: Presence of heart disease (0: No, 1: Yes).

## Preprocessing:

We first checked the data's cleanness to ensure the dataset's integrity by searching for duplicates to avoid redundant information that could skew the model’s performance. We discovered no duplicates. Then, we checked for any missing values, of which we found none. Lastly, we turned categorical features into numerical ones using a manual mapping technique so the model could deal with the data.  
  
Following this, we performed Data Splitting by dividing the dataset into training and testing subsets using the train\_test\_split function from the sklearn library. We allocated 20% of the data for testing and 80% for training, with a fixed random\_state of 42 to ensure reproducibility of results.

## Statistical Analysis

Figure 3. Summary statistics of numerical features

.

# BAGGING METHODOLGY

## Bagging Definition:

bagging based on the principle of building several models (of the same kind) and training each model on a distinct subset of the data. Predictions that are more accurate are then produced by combining the output of these models. This project employed **a decision tree approach.**

## 4.2 How Bagging Works In Our Code:

**1. Data Splitting:** The dataset was divided into two sets: a training set and a testing set, with 80% allocated for training and 20% for testing.

**3. Creating the Bagging Model:** A Bagging model was created using a Decision Tree as the "base estimator," with parameters like the number of models (n\_estimators).

**4. Evaluation Using Cross-Validation:** StratifiedKFold was used to evaluate the model by splitting the data and training the model on different subsets.

**5. Predicting Outcomes**: Predictions were obtained by applying the trained model to the testing data.

**6. Calculating Performance Metrics**: Performance metrics such as Accuracy, Precision, Recall, and F1-Score were calculated to assess the model’s effectiveness.

**7. Improving the Bagging Model Using Grid Search**: The performance of the Bagging model was optimized using Grid Search , a technique that helps identify the best hyperparameters by evaluating different combinations of parameters, such as the number of decision trees (n\_estimators), the maximum number of features (max\_features), and the maximum depth (max\_depth).

**8. Comparing Model Performance:** The performance of the Bagging model was compared with K-Nearest Neighbors (KNN) and Random Forest models to assess which model performs best, using metrics like Accuracy, Precision, Recall, and F1-Score

# RESULT AND DISCUSSION

## Previous Trail:

The previous dataset was split into 80% for training and 20% for testing. The data showed an imbalanced distribution between the two classes, with 233,389 non-infected cases compared to 12,809 infected cases. This significant imbalance led to poor model performance, even after using Oversampling technique, which increases the number of records for the minority class (infected) to match the majority class (non-infected) in the training data only, without addressing the issue in the test data. The test results show weak model performance in classifying the minority class (infected), with Precision at 0.27 and Recall at 0.23. Additionally, the F1-Score was only 0.25, indicating poor balance between precision and recall. This is further highlighted by the confusion matrix, where 55,000 non-infected cases were correctly classified as "non-infected" (True Negatives), while 3,367 non-infected cases were incorrectly classified as "infected" (False Positives). On the other hand, 4,332 infected cases were misclassified as "non-infected" (False Negatives), and only 1,260 infected cases were correctly classified as "infected" (True Positives). This indicates that the model struggles with the minority class (infected), as 4,332 infected cases were misclassified as non-infected.

Hyperparameter tuning in tuning phase, Despite the promising cross-validation results, the testing metrics were less encouraging. On the test set, the model achieved an F1-Score of 0.16, Precision of 0.37, Recall of 0.10, This means that the model's performance after tuning is worse.

## Result of New Dataset:

The model was tested on a smaller dataset (918 samples), where the proportion of infected individuals was 55%, while the proportion of uninfected individuals was 44%, indicating a better balance compared to the previous dataset. The model's performance improved significantly, with the results before tuning illustrated in Figure 4 and after tuning in graph 5

A diagram of a confusion matrix

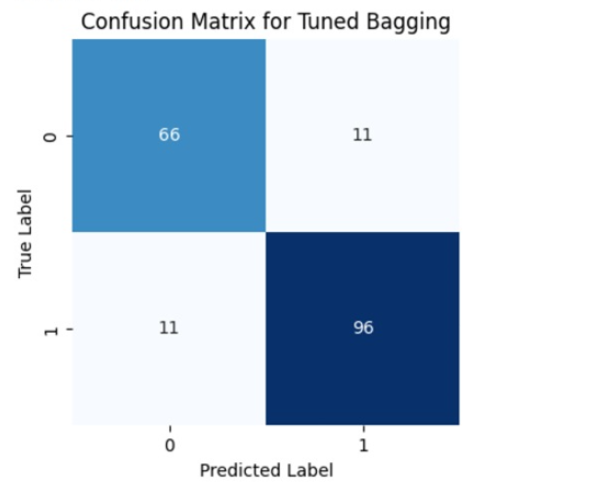
Description automatically generated

Figure 5. after tuning

Figure 4. Hyperparameter and their values

.

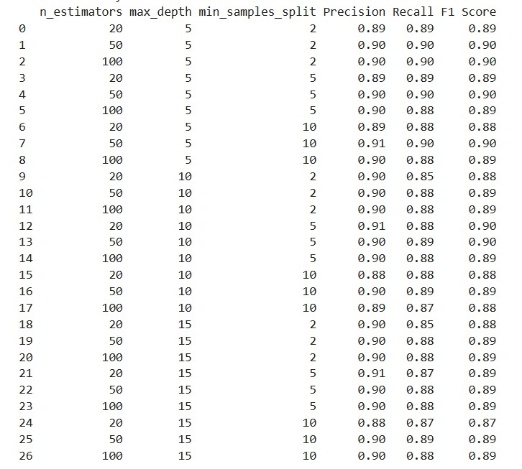
Figure 4. before tuning

.

The F1-score of the model before tuning was 0.81, with a precision of 0.82 and recall of 0.81. After tuning, these values increased to 0.90 for all three metrics. This improvement demonstrates that the model performed better on the more balanced dataset.

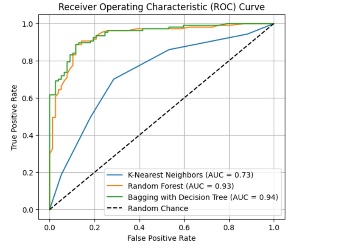
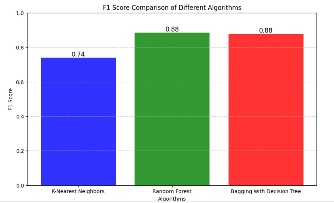
## Tuning phase:

Hyperparameter tuning is a fundamental process in machine learning, aiming to optimize model performance by systematically identifying the best combination of parameters. In this study, the BaggingClassifier was chosen as the primary model due to its ensemble nature, which combines multiple weak learners to reduce variance and improve stability. Additionally, Bagging’s ability to leverage decision trees as base estimators allows it to capture intricate patterns in the data effectively. These qualities made it a strong candidate for optimizing classification performance in this project. The Grid Search method was employed to fine-tune its parameters. The hyperparameter grid explored values for n\_estimators (ranging from 20 to 100), max\_depth (from 5 to 15), and min\_samples\_split (from 2 to 10). This approach ensured a comprehensive search across possible configurations to achieve optimal performance



The results of the Grid Search revealed that the best configuration was max\_depth: 5, min\_samples\_split: 2, and n\_estimators: 50. This configuration produced exceptional results on the test set, with an F1 Score, Precision, and Recall all reaching 0.90. Such balanced metrics highlight the model's ability to classify both majority and minority classes with high accuracy. The tuning phase demonstrated a significant improvement in classification performance, achieving Precision, Recall, and F1 Score . The confusion matrix provided further insights into the model's classification ability, showing True Negatives: 66 (correctly classified non-infected cases), False Positives: 11 True Positives: 96 and False Negatives: 11 These results demonstrate the model's proficiency in maintaining a balance between Precision and Recall, making it suitable for applications that require both metrics to be optimized.

## Compare different models:



Through the attached images, we can compare the performance of the three models (K-Nearest Neighbors, Random Forest, and Bagging with Decision Tree) based on F1-Score metrics and ROC curves (acceptance rate vs. false positive

F1-Score, both Random Forest and Bagging with Decision Tree outperform KNN with identical performance In terms of AUC, Bagging with Decision Tree appears to be the best model with a slight edge over Random Forest, while KNN lags significantly Overall, tree-based methods (Random Forest and Bagging) demonstrate significantly better performance compared to KNN, making them preferable choices when higher accuracy and balance are required

While KNN is computationally efficient for smaller datasets, its reliance on distance-based calculations often limits its effectiveness on imbalanced data. Random Forest, on the other hand, demonstrating strong overall performance. Random Forest’s ability to aggregate predictions from multiple decision trees provides a degree of stability, though it occasionally struggles with overfitting in highly imbalanced datasets.

Compared to KNN and Random Forest, Bagging demonstrated superior robustness against noise and imbalanced distributions. Its minimized false positives and false negatives underscore its suitability for applications demanding high reliability in Precision and Recall

# REFERENCES

1. Simplilearn (2024) *What is bagging in machine learning and how to perform bagging*, *Simplilearn.com*. Available at: https://www.simplilearn.com/tutorials/machine-learning-tutorial/bagging-in-machine-learning (Accessed: 16 October 2024).
2. Azevedo, N. (2024) *Data preprocessing techniques in machine learning [6 steps]*, *Scalable Path*. Available at: https://www.scalablepath.com/data-science/data-preprocessing-phase (Accessed: 21 October 2024).
3. Ibm (2024) *What is bagging?*, *IBM*. Available at: https://www.ibm.com/topics/bagging (Accessed: 16 October 2024).
4. *Bagging machine learning - javatpoint* (no date) *www.javatpoint.com*. Available at: https://www.javatpoint.com/bagging-machine-learning (Accessed: 17 October 2024).
5. Sannella, M. J. 1994. *Constraint Satisfaction and Debugging for Interactive User Interfaces*. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398., University of Washington.
6. Brownlee, J. (2021) *Bagging and random forest for imbalanced classification*, *MachineLearningMastery.com*. Available at: https://machinelearningmastery.com/bagging-and-random-forest-for-imbalanced-classification/ (Accessed: 18 October 2024).