Machine Learning-Based Predictive Modeling for Early Detection of Heart Disease

A DISSERTATION SUBMITTED IN PARTIAL COMPLETION OF THE REQUIREMENTS FOR THE DEGREE OF MSC (ENGINEERING) IN INFORMATION AND COMMUNICATION TECHNOLOGY

SUBMITTED BY

ID: 22209025

Session: 2021-22

Department of Information and Communication Technology

Comilla University, Bangladesh

Submission: February, 2025

ABSTRACT

Machine learning has revolutionized various industries worldwide, including healthcare, by enabling predictive analytics for disease diagnosis and prognosis. One of its most promising applications is in the early detection of heart disease, which allows medical professionals to tailor treatment strategies and improve patient outcomes. The goal of this research is to apply machine learning methods to forecast an individual's risk of developing heart disease.

To achieve this, we evaluate various classification models, including Random Forest, Naïve Bayes, Support Vector Machines (SVM), Logistic Regression, and Decision Trees. Furthermore, we propose an ensemble approach that integrates both weak and strong classifiers to enhance prediction accuracy. In pursuit of improved classification performance, we also explore advanced boosting techniques such as AdaBoost and XGBoost. By analyzing these models and their effectiveness using key performance metrics—including confusion and correlation matrices—our research aims to develop a robust predictive framework for heart disease detection.

Keywords : Machine Learning, Heart Disease Prediction, Classification Models, Predictive Analytics, Medical Diagnosis, Healthcare AI, Support Vector Machines (SVM), Naïve Bayes, Random Forest, Decision Tree, Logistic Regression, AdaBoost, XGBoost, Ensemble Learning, Supervised Learning, Weak Classifiers, Strong Classifiers, Accuracy, Precision, Recall, F1-score, Confusion Matrix, Correlation Matrix, ROC Curve, AUC Score, Feature Selection, Feature Engineering, Data Preprocessing, Data Normalization, Dimensionality Reduction, Handling Imbalanced Data, Python Programming, Scikit-learn, TensorFlow, Keras, Pandas, NumPy, Matplotlib, Seaborn, Cardiovascular Disease (CVD), Risk Factors, Clinical Data Analysis, Prognostic Modeling, Personalized Medicine, Health Informatics.

CONTENTS

CHAPTER 1
INTRODUCTION

Nearly 70% of fatalities globally are attributable to cardiovascular disease (CVD), which continues to be the primary cause of morbidity and mortality. Cardiovascular diseases are a major public health problem because they cause more than 45% of all deaths, according to the Global Burden of Disease 2023 estimate. Poor eating habits, smoking, excessive sugar intake, obesity, high blood pressure, and sedentary lifestyles are common risk factors. Furthermore, early detection and treatment are hampered by differences in availability to necessary diagnostic procedures including electrocardiograms (ECGs), echocardiograms, and CT scans, especially in low-income areas. As a result, the **World Health Organization (WHO)** estimates that **over 18 million people die annually due to CVD -related complications**.

Large volumes of patient data are being gathered as healthcare becomes more digitally connected. However, **high-dimensional and noisy datasets** pose challenges for traditional analytical methods. The **complex interdependencies among risk factors** further complicate manual diagnosis and prognosis. Machine learning (ML) has emerged like a transformative technology, capable of **analyzing large datasets, identifying hidden patterns, and improving disease prediction accuracy**. Heart disease prediction models are now much more reliable thanks to recent developments in deep learning, ensemble learning, and feature selection methods. The integration of **artificial intelligence (AI) in healthcare** is revolutionizing clinical decision-making, offering potential breakthroughs in **personalized treatment, early disease detection, and predictive analytics**. Existing cardiovascular risk prediction models often fail to identify high-risk individuals accurately, leading to **missed preventive care opportunities or unnecessary medical interventions**. AI-driven models, leveraging **advanced classification techniques**, can better capture the intricate relationships between multiple risk factors, ultimately **improving diagnostic precision**. The purpose of this research is to investigate how machine learning might improve risk assessment and cardiovascular disease prediction, leading to more proactive and efficient healthcare solutions.

1.1 MOTIVATION

With millions of deaths every year, cardiovascular disease (CVD) continues to be the world's leading cause of mortality. Despite significant advancements in medical science, early detection and accurate risk assessment of heart disease remain challenging. Traditional diagnostic methods rely on expensive medical tests, such as--- ECGs and CT scans, these are often inaccessible to individuals in low-resource settings. Additionally, existing cardiovascular risk prediction models fail to account for the complex interplay of multiple risk factors, leading to inaccurate diagnoses and ineffective preventive measures.

More effective and precise diagnostic technologies are desperately needed, as healthcare data is growing at an accelerated rate. The application of **machine learning (ML) in predictive analytics** offers a promising solution, enabling the identification of hidden patterns and improving disease prognosis. By leveraging ML techniques, it is possible to develop **cost-effective, scalable, and highly accurate heart disease prediction models** that can assist healthcare professionals in making informed decisions.

This research is motivated by the potential of artificial intelligence to **bridge the gap between data-driven insights and clinical decision-making**. This work intends to promote AI-driven cardiovascular disease prediction by investigating several machine learning algorithms and suggesting ideal ensemble models, which would ultimately result in early intervention, lower mortality rates, and better patient outcomes.

1.2 PROBLEM STATEMENT

Traditional cardiovascular risk prediction models have limited accuracy and generalization capabilities, often failing to identify high-risk patients while leading to unnecessary interventions for low-risk individuals. Furthermore, the increasing availability of large-scale healthcare data presents both opportunities and challenges, as manual analysis is inefficient and often leads to misinterpretations due to noise and high-dimensional data.Large dataset analysis, the discovery of intricate patterns, and increased forecast accuracy have all been demonstrated to be possible with machine learning (ML) techniques.

However, there is a need to evaluate and optimize various ML models to determine the most effective approach for early detection and risk assessment of heart disease. This study aims to address these challenges by analyzing different classification algorithms, proposing an

ensemble model for improved accuracy, and exploring advanced boosting techniques such as AdaBoost and XGBoost . The objective is to create a prediction model that is extremely accurate, scalable, and easily accessible so that medical professionals can make well-informed decisions and, eventually, lower the death rates associated with CVD .[3]

CHAPTER 2
LITERATURE SURVEY

- The promise of machine learning (ML) in the prediction of cardiac disease has been the subject of numerous studies, which have used a variety of datasets, algorithms, and approaches to increase precision and effectiveness.
- **Shah et al.** utilized the **Cleveland Heart Disease dataset** and evaluated multiple classification models, including **KNN, decision trees, random forests , and Naïve Bayes**, concluding that **KNN achieved the highest accuracy of 90.8%**.[1]
- **Drod et al.** focused on predicting CVD risk in **MAFLD (Metabolically Associated Fatty Liver Disease) patients** using **Principal Component Analysis (PCA)** and **multiple logistic regression classifiers**, achieving an **AUC score of 0.87**.[2]
- **Narain et al.** introduced a **quantum neural network-based system**, significantly surpassing traditional **Framingham Risk Score (FRS)** with an impressive **98.57% accuracy**.[3]
- **Purushottam et al.** developed a **heart disease prediction model** using **hill-climbing and decision tree algorithms**, achieving an **86.7% accuracy** by preprocessing the **Cleveland dataset** with KEEL open-source tools.[4]
- **Santhana Krishnan et al.** proposed a **Naïve Bayes and decision tree-based approach**, achieving **91% accuracy** with **decision trees** and **87% accuracy** using **Naïve Bayes**.[5]
- **Sonam Nikhar et al.** compared **Decision Tree and Naïve Bayes classifiers**, demonstrating that **Decision Trees outperformed Naïve Bayes** in prediction accuracy.[6]
- **Aditi Gavhane et al.** applied **multi-layer perceptron (MLP) neural networks**, highlighting the effectiveness of **hidden layers** in feature extraction and prediction.[7]
- **Lakshmana Rao et al.** used **neural networks and data mining techniques** to classify **heart disease severity**, emphasizing the complexity of CVD risk assessment.[8]

- **Anjan N. Repaka et al.** proposed a **hybrid classification model**, outperforming previous techniques in **CVD risk assessment accuracy**.[9]
- **Aakash Chauhan et al.** implemented **evolutionary rule learning** and **association mining techniques**, showing **improved efficiency and accuracy** in heart disease prognosis.[10]

Key Takeaways for This Study

The promise of machine learning (ML) in the prediction of cardiac disease has been the subject of numerous studies, which have used a variety of datasets, algorithms, and approaches to increase precision and effectiveness.This study aims to address these challenges by evaluating **multiple ML models**, enhancing predictive performance through **ensemble learning (AdaBoost & XGBoost)**, and exploring advanced **feature selection techniques** for improved CVD risk assessment.

CHAPTER 3
METHODOLGY

3.1 EXISTING SYSTEM

Cardiovascular disease has become a silent killer, often claiming lives without exhibiting noticeable symptoms. Its elusive nature has raised significant concerns about its prevalence and impact, making early prediction a critical area of research. In response to the growing demands of modern healthcare, numerous methods and approaches are being explored to enhance early detection and risk assessment. Machine learning (ML) techniques have demonstrated significant promise in enhancing forecast accuracy and supporting preventive interventions among these.

Heart disease can manifest in various forms, but a combination of key risk factors primarily determines an individual's susceptibility. By aggregating data from multiple sources, categorizing it into relevant groups, and applying advanced analytical techniques, ML models can extract valuable insights for early diagnosis. This data-driven approach enhances prediction capabilities, enabling timely interventions. According to the well-known saying, "An ounce of prevention is worth a pound of cure," heart disease deaths can be considerably decreased or even avoided with early detection and effective treatment.

3.2 PROPOSED SYSTEM

In order to improve the precision and effectiveness of early diagnosis, the suggested method seeks to create a prediction model for cardiovascular disease (CVD) using machine learning techniques. The technology will evaluate patient information, pinpoint important risk factors, and forecast the probability of heart disease. By giving medical practitioners trustworthy, data-driven insights for early intervention and individualized treatment plans, the overarching objective is to enhance preventative healthcare. The system will use a mix of ensemble methods and supervised learning algorithms to handle complicated, high-dimensional datasets efficiently and optimize prediction accuracy.

System Architecture

The system will follow a structured workflow divided into several stages:

1. **Data Collection**:

   The first stage involves gathering **patient data** from multiple sources, such as **clinical records, laboratory tests, medical imaging**, and **demographic information**. The system will utilize publicly available datasets like the **Cleveland Heart Disease dataset** or integrate real-time data from hospitals, clinics, and electronic health records (EHR).

2. **Data Preprocessing**:

   Data preprocessing is crucial for ensuring data quality. The system will handle inconsistent data, outliers, and missing values in order to clean and preprocess the gathered data.. To make sure the dataset is prepared for machine learning algorithms, methods such as data imputation, normalization, and feature scaling will be used. The dataset will also be split into training and testing sets to validate the model's performance.

3. **Feature Selection and Engineering**:

   Using feature selection methods like mutual information, chi-square tests, or recursive feature elimination (RFE), the system will determine the most significant risk factors (such as age, blood pressure, cholesterol, smoking status, etc.). Additional **feature engineering** will be employed to create new, meaningful features that can improve model performance.

4. **Model Development**:

   Various **supervised machine learning algorithms** will be evaluated for heart disease prediction. These may include:

   - **Logistic Regression**: for binary classification (presence or absence of heart disease).
   - **Random Forest**: a robust ensemble method that combines multiple decision trees to improve accuracy.
   - **Support Vector Machine (SVM)**: to identify the optimal hyperplane for classification.
   - **Naïve Bayes**: for handling probabilistic classification.
   - **K-Nearest Neighbors (KNN)**: for classifying based on proximity to other data points.

Furthermore, by merging weak classifiers into a stronger, more dependable model, ensemble techniques like XGBoost will be investigated to further improve model accuracy.

5. **Model Evaluation**:
   The model's performance will be assessed by the suggested system using a number of measures, including accuracy, precision, recall, F1-score, ROC curve, and AUC score. After comparing the performance of the various models, the model with the best performance will be chosen for deployment.
6. **Prediction and Risk Assessment**:
   The model will be used to forecast the probability of heart illness for fresh patient data after it has been trained. The system will assign a **risk score** to each patient, indicating their susceptibility to heart disease. This risk score will help healthcare providers prioritize high-risk patients and recommend timely preventive measures or interventions.
7. **User Interface and Visualization**:
   Healthcare practitioners will be able to enter patient data and view the forecasts using the system's user-friendly interface. Interactive visualizations, such as **confusion matrices**, **ROC curves**, and **risk score charts**, will aid in understanding the model's output and making informed decisions.

Key Features of the Proposed System

- **Scalability**: The system will be designed to scale with growing datasets, enabling integration with various healthcare databases and **real-time data sources**.
- **Explainability**: The system will provide interpretable results, highlighting the **key risk factors** contributing to the prediction, thus helping healthcare professionals make data-driven decisions.
- **Adaptability**: The model will be adaptable to different populations and healthcare environments, allowing for customization based on **regional risk factors** and **demographic variations**.
- **Efficiency**: The system will ensure fast and accurate predictions, reducing the time healthcare providers spend on manual assessments and improving the overall **diagnostic workflow**.

Expected Impact

The suggested system's automated, data-driven approach to early cardiovascular disease identification will have a big impact on the healthcare sector. By leveraging the power of machine learning, the system will improve the accuracy of **CVD risk assessments**, facilitate **personalized healthcare**, and ultimately **reduce mortality rates** associated with heart disease. Additionally, it will help medical personnel make well-informed decisions, improving patient outcomes and maximizing the use of resources in healthcare institutions.

CHAPTER 4
ALGORITHMS

4.1 MACHINE LEARNING

- Assigning a class label to a particular input data sample presents a significant challenge in predictive modeling within machine learning.

- **Supervised Learning**

Supervised learning is a subset of machine learning where models are trained on carefully "labeled" training data, utilizing this information to make accurate predictions. Some of the input data already has the correct output assigned, as indicated in the labeled dataset. In supervised learning, the "supervisor," or training data, directs the model to produce precise predictions. [7] Analogous to the educational process, supervised learning is akin to a student learning from a teacher, where the model absorbs knowledge from labeled training data to enhance its ability to predict outputsGiving a machine learning model accurate input and output data is known as supervised learning. The primary goal of a supervised learning algorithm is to discern a mapping function that connects the input variable (x) to the output variable (y). [8]

- **Unsupervised Learning**

Unsupervised learning is used when just input data is accessible and the associated output data is unknown, as opposed to supervised learning, which uses both input and output data known. Unsupervised learning is effective in extracting meaningful insights from data and is closer to real AI as it mimics how humans learn through experience. Since it deals with uncategorized data, unsupervised learning is considered more significant in real-world applications. In many practical scenarios, equivalent outputs for input data are often unavailable, necessitating the use of unsupervised learning techniques to address these complexities.

- **Reinforcement Learning**

Reinforcement learning, also known as reward learning, is another subfield of machine learning. It involves taking actions in specific situations to maximize a reward. [9] This approach is widely used in software programs and devices to determine the optimal course of action or behavior in a given context. Unlike supervised learning, which uses pre-existing

answers in the training set to guide the model, reinforcement learning operates without pre-defined answers. Instead, it relies on the judgment of a reinforcement agent to determine the appropriate actions needed to accomplish a task. [10]

4.2 SUPPORT VECTOR MACHINE

Support Vector Machines (SVMs) are among the most widely used supervised learning methods, well-regarded for their effectiveness in classification and regression tasks. Despite their adaptability, SVMs are especially well-known for their capacity to manage machine learning classification challenges.

By determining the best line or border that may divide the data into discrete classes, SVM's main objective is to expedite the classification of data points. The hyperplane is the name given to this optimal decision boundary.

Key Concepts of SVM:

- **Support Vectors**: These are the data points that lie closest to the hyperplane and play a crucial role in defining the boundary. They are the critical elements that determine the positioning of the hyperplane and hence the classifier's decision boundary.
- **Hyperplane**: The hyperplane is the decision boundary that separates the dataset into distinct classes. The n-dimensional feature space, where n is the number of features, is split into two sections by the optimum line.
- **Margin**: The margin is the separation between each class's closest data points (support vectors) and the hyperplane. Because it indicates better class separation and lowers the chance of overfitting, a bigger margin is favored. The model may overfit if the margin is smaller, which would result in poor generalization to fresh data.
- **Types of SVM:**

1. **Linear SVM**: When the data points can be neatly separated by a straight line (or a hyperplane in higher dimensions), this kind of classifier is employed.
2. **Non-Linear SVM**: A non-linear SVM is employed for datasets in which the data points cannot be separated linearly. This entails applying a kernel function to translate the data points to a higher-dimensional space, where a linear hyperplane can be used to divide them.
3. **How SVM Works:**

Finding the appropriate hyperplane to divide the data into distinct classes is the goal of SVM..The position of this hyperplane is determined by the support vectors, which are the closest data points to the hyperplaneThese support vectors determine the decision border, and the margin between them shows how effectively the classes are divided. A wider margin generally leads to a better performing model.

When dealing with data that is not linearly separable, the kernel trick is utilized to convert the data into a higher-dimensional space that allows for linear separation.

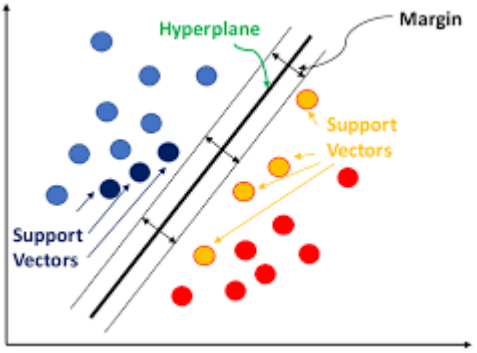
Benefits of SVM:

- **Effective in high-dimensional spaces**: SVM performs well when the number of features (dimensions) is greater than the number of samples.
- **Memory efficiency**: SVM only uses support vectors, which reduces memory consumption, making it more efficient.
- **Flexibility**: The decision function in SVM can be associated with a variety of kernel functions. Users have the option of using standard kernels such as linear, polynomial, or radial basis function (RBF) or they can provide their own kernels. [11]
- **Drawbacks of SVM:**

- **Overfitting**: Careful selection of kernel functions is needed to prevent overfitting, especially when the dataset has more features than samples.
- **Lack of probability estimates**: SVM does not provide direct probability estimates. These can be obtained through costly processes like cross-validation.

Visual Representation of SVM:

Here's an illustration to better understand the SVM concept.



In this figure:

- The line that separates the data points into two different classes is known as the hyperplane.
- The points that define the margin and are closest to the hyperplane are known as the support vectors.
- The margin is the separation between the closest data points from either class and the hyperplane..
- SVM offers versatility and accuracy in distinguishing between classes, making it a potent and efficient tool for classification problems, particularly in high-dimensional domains.

4.3 NAIVE BAYES ALGORITHM

Based on Bayes' Theorem, the Naïve Bayes algorithm is a popular supervised learning method for classification tasks. It is widely used in text classification, spam filtering, sentiment analysis, and document categorization because of its effectiveness with high-dimensional datasets. The Naïve Bayes classifier is incredibly successful and computationally efficient despite its simplicity, allowing for quick predictions and model construction.

Understanding Naïve Bayes Classification

The **Naïve Bayes classifier** operates on the principle of **conditional probability**, assuming that each feature contributes independently to the classification outcome. This assumption enables the model to function well with huge datasets, despite the fact that it is frequently impractical with real-world data.

Based on the likelihood of related features, the algorithm forecasts the likelihood that an event will occur. Stated differently, the observed data points are used to calculate the likelihood of a specific class classification.

Bayes' Theorem

The Naïve Bayes algorithm is based on the Bayes Theorem, which calculates the likelihood of a hypothesis (class) given observed evidence (features).

The theorem is expressed as:

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)}$$

Where:

- P(A|B) → **Posterior Probability** (Probability of hypothesis AA given evidence BB)
- P(B|A)→ **Likelihood** (Probability of evidence BB occurring given AA is true)
- P(A)→ **Prior Probability** (Initial likelihood of hypothesis AA prior to evidence observation)
- P(B)→ **Marginal Probability** (Overall probability of observing BB across all possible hypotheses)

The Naïve Bayes classifier simplifies this by assuming **feature independence**, meaning the presence of one feature is not dependent on another.

Why "Naïve" and "Bayes"?

- **"Naïve"** → The classifier makes the assumption that characteristics are independent of one another, which means that the likelihood of one attribute happening is unaffected by the likelihood of another. For example, in fruit classification, features like color, shape, and taste are considered independently in determining whether a fruit is an apple.
- **"Bayes"** → The Bayes Theorem, which aids in probability estimation, forms the foundation of the model.

Types of Naïve Bayes Models

Naïve Bayes classifiers come in three primary varieties, each of which is appropriate for a certain data distribution:

1. Gaussian Naïve Bayes

- Used when **continuous features** follow a **normal (Gaussian) distribution**.
- Assumes that **features are drawn from a Gaussian distribution** and calculates probabilities using the normal distribution formula.
- Example: Predicting student grades based on test scores.

2. Multinomial Naïve Bayes

- Suitable for **discrete data**, particularly **text classification tasks** where word frequencies are used as features.
- Used in **document classification**, such as **categorizing articles into sports, politics, or education** based on word occurrences.
- Example: Spam email detection based on word frequency.

3. Bernoulli Naïve Bayes

- Similar to **Multinomial Naïve Bayes**, but the features are **binary (Boolean)** instead of frequency-based.
- In document classification, the categorization is based on whether a word appears in the document (1) or not (0).
- Example:Using a specific keyword's presence to determine if an email is spam.

Advantages of Naïve Bayes

- ✓ **Fast and Efficient**: Works well with large datasets and high-dimensional data.
- ✓ **Performs Well with Limited Data**: Requires less training data compared to other classifiers.
- ✓ **Works for Both Binary and Multi-Class Classification**.
- ✓ **Handles Noisy Data**: Performs well despite noisy or missing data.
- ✓ **Simple and Easy to Implement**.

Disadvantages of Naïve Bayes

- **Feature Independence Assumption**: It makes the perhaps unrealistic assumption that features are independent.

- **Zero Probability Issue**: If a categorical variable is absent from training data, the probability estimate becomes zero, though techniques like **Laplace smoothing** can mitigate this issue.
- **Not Suitable for Complex Relationships**: Does not capture intricate relationships between features.

4.4 DECISION TREE ALGORITHM

Although it may also tackle regression issues, the Decision Tree algorithm is a supervised learning method that is mainly applied to classification tasks. It is a tree-structured model where **internal nodes represent attributes** of the dataset, **branches depict decision rules**, and **leaf nodes signify classification outcomes**. Because of its excellent interpretability, decision trees are frequently used in machine learning applications.

Structure of a Decision Tree

A **Decision Tree** is composed of the following key components:

1. **Root Node**: the root node of the tree, which stands for the complete dataset before being divided into smaller nodes according to feature selection standards.
2. **Decision Nodes**: Internal nodes that perform attribute-based splitting, forming sub-trees based on decision rules.
3. **Leaf Nodes**: Terminal nodes that provide classification outcomes or final predictions, with no further branching.

The **Classification and Regression Tree (CART)** algorithm is commonly used to construct decision trees. The process follows a recursive approach where a **question is posed at each node**, leading to a split in the tree based on **Yes/No** responses.

Working Principle of Decision Trees

Decision Trees operate by **partitioning data** into smaller subsets at each node, forming a hierarchical structure that resembles a tree. The algorithm evaluates different features of the dataset and **selects the best attribute** at each level using mathematical measures such as **Information Gain** or the **Gini Index**.

According to the chosen characteristic, the dataset is split into two or more homogenous groups at each stage, and the procedure is continued recursively until:

- A stopping criterion is met (e.g., maximum depth, minimum samples per node).
- All samples in a node belong to the same class.

This hierarchical decision-making structure mirrors **human cognitive processes**, making decision trees intuitive and easy to interpret.

Attribute Selection in Decision Trees

Choosing the correct attribute for splitting at each node is a vital step known as Attribute Selection. This process is guided by the following metrics:

1. Information Gain (IG)

Information Gain measures the reduction in uncertainty (entropy) after splitting a node. A **higher information gain** indicates a better feature for splitting, as it leads to a more **homogeneous** subset of data.

2. Gini Index

The **Gini Index** measures the probability of incorrectly classifying a randomly chosen element in the dataset. A **lower Gini Index** indicates a **better feature split**, as it reduces misclassification chances. By default, **Scikit-learn** uses the **Gini Index** as the criterion for decision trees.

Types of Decision Tree Algorithms

Several variations of the Decision Tree algorithm exist, each differing in how they select attributes and handle data:

1. ID3 (Iterative Dichotomiser 3)

- Utilizes **Information Gain** to determine the optimal attribute for classification.
- Repeats the selection process iteratively at each level until the tree is fully constructed.

2. C4.5

- An extension of ID3 that **supports both continuous and discrete attributes**.
- To lessen bias toward attributes with more categories, the Information Gain Ratio is used rather than raw information gain.

- Can handle **missing values** effectively.

3. CART (Classification and Regression Tree)

- Unlike ID3 and C4.5, CART uses the **Gini Index** instead of **Information Gain**.
- It can construct both **classification trees** and **regression trees**, making it a **versatile choice** for predictive modeling.

Benefits of Decision Trees

- ✓ **Interpretable and Easy to Understand**: Mimics human decision-making processes.
- ✓ **Handles Categorical and Numerical Data**: Can process mixed data types.
- ✓ **Non-Parametric**: Does not assume any underlying data distribution.
- ✓ **Feature Selection Built-In**: Automatically selects important attributes for classification.
- ✓ **Works Well with Large Datasets**: Efficient for high-dimensional data.

Disadvantages of Decision Trees

- ✓ **Overfitting**: Deep trees may lead to overfitting, capturing noise rather than patterns.
- ✓ **Sensitive to Small Variations**: A small change in data can result in a completely different tree.
- ✓ **Computationally Expensive**: Training large trees can be time-consuming.

4.5 RANDOM FOREST

By using bagging to boost Decision Tree performance, Random Forest, a supervised learning technique, improves machine learning classifiers. By aggregating predictions from multiple trees, each utilizing a randomly sampled subset of features, Random Forest achieves robustness and high accuracy in classification tasks. Every tree has the same distribution, but Random Forest splits nodes depending on the best predictor subset, which is chosen at random from the node itself, rather than variables. In the worst-case learning scenario, the time complexity of Random Forest is O(M(dnlogn)), where M denotes the number of growing trees, n represents the instances, and d indicates the data dimension [15].

This versatile technique is applicable to both regression and classification problems. Known for its adaptability and ease of use, Random Forest is among the most intuitive machine learning algorithms. The algorithm builds a forest with several decision trees, and the more trees there are in the model, the better the prediction power. The final prediction is determined through a voting mechanism where Decision Trees, built on randomly selected data samples, contribute their individual predictions. Additionally, Random Forest provides a reliable measure of feature importance.

Random Forests have widespread applications, including feature selection, image classification, and recommendation systems. They are particularly effective in identifying creditworthy loan applicants, detecting fraudulent activities, and diagnosing diseases. Furthermore, algorithms such as Boruta rely on Random Forest for key feature selection from datasets.

A notable member of supervised learning methods, the Random Forest algorithm operates on the principle of ensemble learning, where multiple classifiers collaborate to enhance model performance and address complex problems effectively. Essentially, Random Forest functions as a classifier composed of multiple decision trees, each trained on distinct dataset subsets. Unlike traditional single decision trees, Random Forest employs a consensus-based approach by aggregating the majority vote from its ensemble, enhancing predictive accuracy and mitigating overfitting risks, thereby ensuring robust and reliable results.

Assumptions

Individual decision trees within a Random Forest may predict outputs independently, but collectively, they yield accurate results due to the ensemble nature of the algorithm. For optimal performance, two key assumptions must be satisfied:

1. The dataset's feature variables should contain meaningful values rather than random or irrelevant data.
2. The predictions made by individual trees should exhibit minimal correlation, ensuring diverse and independent contributions to the ensemble.

By following these presumptions, the Random Forest classifier achieves superior predictive performance and robustness, making it a useful instrument for applications involving machine learning.

Algorithm Steps

There are four main steps that the Random Forest algorithm follows:

1. Select random samples from the given dataset.
2. Construct a decision tree for each sample and generate predictions.
3. Aggregate predictions through voting.
4. Use the majority vote to determine the final prediction [15].

Benefits

- Effectively manages big datasets with plenty of dimensions.
- Supports both classification and regression tasks effectively.
- Reduces overfitting risks and improves model accuracy.

Limitations

Although Random Forest is quite good at both classification and regression, its effectiveness is more pronounced in classification problems, as it is not inherently optimized for regression tasks.

4.6 LOGISTIC REGRESSION ALGORITHM

Logistic Regression is a fundamental supervised learning algorithm used for classification tasks. In contrast to linear regression, which forecasts continuous values, logistic regression models the likelihood of a specific class to forecast categorical outcomes. It is frequently used to solve binary classification issues. where the output variable has two possible categories, but it can also be extended to multiclass classification.

Assumptions

To achieve optimal performance, Logistic Regression relies on several key assumptions:

1. **Linearity of independent variables**: The log odds of the dependent variable and the independent variables need to be linearly related.
2. **Independence of observations**: Each observation should be independent of others.
3. **No multicollinearity**: The independent variables should not be highly correlated with each other.
4. **Sufficient sample size**: A large dataset ensures better generalization and robust coefficient estimation.

Algorithm Steps

Logistic Regression follows these steps:

1. **Initialize model parameters** (weights and bias).
2. **Determine the input features' linear combination.**
3. **Apply the sigmoid function** to obtain probability estimates.
4. **Calculate the cost function**.
5. **Optimize parameters** using gradient descent to minimize the cost function.
6. **Make predictions** by classifying observations using a threshold (e.g., 0.5).

Advantages

- Simple and easy to implement.
- Efficient for binary classification tasks.
- Outputs interpretable probability scores.
- Performs well with linearly separable data.

Limitations

- Presupposes that log odds and independent variables have a linear relationship.
- Not well-suited for complex relationships in data.
- When working with high-dimensional data, there is a risk of overfitting.
- Sensitive to imbalanced datasets.

4.6 XGBOOST ALGORITHM

XGBoost is an advanced implementation of Gradient Boosted Decision Trees, created to improve computational efficiency and model performance. It builds decision trees sequentially, improving predictions with each iteration. One of its core principles lies in assigning weights to independent variables, where misclassified variables receive higher weights to improve their representation in subsequent decision trees. This iterative process refines the model, combining distinct classifiers to form a robust and accurate predictor. XGBoost supports a variety of tasks, including prediction, classification, regression, and ranking [16].

Key Features of XGBoost:

1. **Regularization:**

   XGBoost incorporates both L1 (Lasso) and L2 (Ridge) regularization techniques to mitigate overfitting, enhancing generalization and robustness. Due to this capability, it is often referred to as a "regularized version" of Gradient Boosting Machine (GBM). When implemented with the Scikit-learn library, XGBoost provides hyperparameters alpha for L1 regularization and lambda for L2 regularization, allowing fine-tuning of the regularization strength based on model requirements.

2. **Parallel Processing:**

   XGBoost outperforms traditional GBM in speed by leveraging parallel processing. It utilizes multiple CPU cores to execute tasks efficiently. The nthread hyperparameter in the Scikit-learn library controls the number of CPU cores used. Setting nthread=0 allows the algorithm to automatically utilize all available cores for computation, significantly speeding up the training process.

3. **Handling Missing Data:**

   XGBoost is equipped with a built-in mechanism to address missing data effectively. When a missing value is encountered at a node, the algorithm tests splits in both directions to determine which path results in a smaller loss. This adaptive approach ensures robust handling of missing values during both training and testing phases.

4. **Cross-Validation:**

   XGBoost simplifies the process of determining the optimal number of boosting iterations by enabling cross-validation at each iteration. Unlike GBM, where grid search is often required to identify the ideal number of iterations, XGBoost provides an iterative process that evaluates performance dynamically.

5. **Tree Pruning:**

   XGBoost employs a more advanced pruning strategy compared to GBM. Instead of stopping splits when a negative loss occurs, XGBoost allows the tree to grow to its specified maximum depth (max_depth) and then prunes it backward. This process removes splits that do not yield positive gains, optimizing the tree structure for better performance.

Advantages of XGBoost:

- Combines the benefits of regularization, parallel processing, and efficient tree pruning for superior performance.
- Effectively handles missing data, making it robust across diverse datasets.
- Simplifies hyperparameter tuning with built-in cross-validation capabilities.

Limitations of XGBoost

1. **Computationally Expensive** – Requires high memory and processing power.
2. **Complex Hyperparameter Tuning** – Needs careful tuning for optimal performance.
3. **Not Ideal for Small Datasets** – May overfit on limited data.
4. **Sensitive to Noisy Data** – Can be affected by outliers and irrelevant features.

4.8 K-Nearest Neighbors (KNN) Algorithm

K-Nearest Neighbors (KNN) is a supervised machine learning algorithm used for both classification and regression. Unlike traditional algorithms, KNN is a **lazy learning** method, meaning it does not construct an explicit model during training. Instead, it memorizes the entire training dataset and makes predictions based on the similarity between new and existing data points. This memory-based approach differentiates KNN from other machine learning models [17].

Algorithm Steps:

1. Select the number of neighbors (**K**).
2. Calculate the distance between the new data point and all training data points.
3. Identify the **K** closest neighbors based on the computed distances.
4. **For classification:** Assign the most frequent class among the K neighbors to the new data point.
5. **For regression:** Compute the average of the K neighbors' target values.

Advantages:

- Simple and easy to understand.
- No explicit training phase; predictions are made dynamically.
- Works well for small to medium-sized datasets.

Disadvantages:

- Computationally expensive for large datasets due to distance calculations.
- Sensitive to irrelevant or redundant features.
- Performance depends on the choice of distance metric.

Chapter 5

Result Analysis

Cardiovascular Disease (CVD) remains a significant global health challenge, with risk factors such as smoking, hypertension, high cholesterol, and diabetes contributing to its prevalence. Prevention strategies focus on lifestyle modifications, including a balanced diet and regular exercise. Early detection through screenings plays a crucial role in reducing complications, while treatment options range from medication to surgical interventions. Public health initiatives are vital in minimizing the impact of CVD on populations worldwide.

5.1 Performance Analysis

In this study, various machine learning algorithms, including **Support Vector Machine (SVM), Naïve Bayes, Decision Tree, Random Forest, and Logistic Regression**, are applied to predict cardiovascular disease. A subset of **14 features from the Heart Disease UCI dataset** is used for prediction, focusing on key patient attributes such as **gender, chest pain type, fasting blood pressure, serum cholesterol, and exercise-induced angina (exang)**.

The performance of each algorithm is evaluated using multiple metrics, including **accuracy, confusion matrix, precision, recall, and F1-score**. The model with the highest accuracy is selected as the most effective for heart disease prediction.

Accuracy Calculation

Accuracy is a crucial metric, representing the ratio of correctly classified instances to the total number of instances. It is calculated as follows:

$$\textbf{Accuracy} = \frac{\mathbf{TP+TN}}{\mathbf{TP+TN+FP+FN}}$$

Where:

- **TP (True Positive)** – Correctly identified disease cases
- **TN (True Negative)** – Correctly identified non-disease cases
- **FP (False Positive)** – Non-disease cases incorrectly classified as disease
- **FN (False Negative)** – Disease cases incorrectly classified as non-disease

5.2 Confusion Matrix

A confusion matrix provides a comprehensive view of the model's performance, showcasing the distribution of actual vs. predicted classifications. This analysis helps determine whether the model performs better than baseline classifiers.

Prediction Accuracy Comparison

Algorithm	Accuracy
K-Nearest Neighbors (KNN)	0.91
Decision Tree	0.95
Random Forest	0.97
Neural Network	0.81
Logistic Regression	0.85
Naïve Bayes	0.79
SVM	0.8643
XGBoost	0.9688

5.2.1 Model Performance Analysis

K-Nearest Neighbors (KNN)

Confusion Matrix:

$$\begin{bmatrix} 87 & 13 \\ 7 & 117 \end{bmatrix}$$

- **Accuracy:** 0.91
- **Area Under the ROC Curve (AUC):** 0.91

KNN correctly identifies **117 true disease cases** and **87 true non-disease cases**. The **ROC curve** confirms its good predictive performance.

Decision Tree

Confusion Matrix:

$$\begin{bmatrix} 97 & 3 \\ 8 & 116 \end{bmatrix}$$

- **Accuracy:** 0.95
- **AUC:** 0.95

The Decision Tree model demonstrates high accuracy, correctly identifying **116 disease cases** and **97 non-disease cases**. The AUC value suggests excellent predictive capability.

Random Forest

Confusion Matrix:

$$\begin{bmatrix} 97 & 3 \\ 4 & 120 \end{bmatrix}$$

- **Accuracy:** 0.97
- **AUC:** 0.97

Random Forest achieves the **highest accuracy (0.97)** among all models. It correctly identifies **120 disease cases** and **97 non-disease cases**, demonstrating outstanding performance based on the AUC.

Logistic Regression

Confusion Matrix:

$$\begin{bmatrix} 65 & 35 \\ 8 & 116 \end{bmatrix}$$

Classification Report:

Class	Precision	Recall	F1-score	Support
0 (Non-Disease)	0.89	0.65	0.75	100
1 (Disease)	0.77	0.94	0.84	124
Accuracy			**0.81**	224

- **AUC:** 0.85

Logistic Regression correctly predicts **116 disease cases** and **65 non-disease cases**. Despite good recall, the lower precision for non-disease cases slightly impacts overall performance.

Naïve Bayes

Confusion Matrix:

$$\begin{bmatrix} 70 & 30 \\ 17 & 107 \end{bmatrix}$$

- **Accuracy:** 0.79
- **AUC:** 0.79

Naïve Bayes achieves moderate accuracy, correctly classifying **107 disease cases** and **70 non-disease cases**. However, its performance is relatively lower than other models.

5.3 Conclusion

Random Forest outperforms other algorithms with the highest accuracy (0.97) among the tested machine learning models. Its ensemble nature, which combines predictions from several decision trees to lower variation and increase dependability, is responsible for its exceptional performance.

According to the results, Random Forest is the best algorithm for predicting cardiovascular illness, providing a robust and accurate classification model. Future research could focus on optimizing feature selection and hyperparameters to further improve prediction accuracy.

Chapter 6

Future Work

Heart disease remains a leading cause of mortality in Bangladesh and worldwide, making early detection crucial for reducing health risks and improving patient outcomes. Early detection of cardiac illness may be possible with the use of cutting-edge technology like machine learning (ML). Early diagnosis enables high-risk individuals to adopt necessary lifestyle modifications, potentially preventing severe complications. The need of prompt intervention and precise diagnosis is becoming more and more clear as the number of heart disease cases each year rises. Implementing **ML-based predictive models** can greatly benefit both healthcare providers and patients by enhancing decision-making and treatment planning.

In this study, **seven machine learning algorithms**—**Support Vector Machine (SVM), Decision Tree, Random Forest, Naïve Bayes, Logistic Regression, Adaptive Boosting, and Extreme Gradient Boosting (XGBoost)**—were employed to assess their effectiveness in heart disease prediction. The dataset used in this research initially contained **76 features**, representing various risk factors associated with heart disease. However, using all features can reduce computational efficiency and introduce redundancy. To optimize performance, **14 key features** were selected for model evaluation, improving accuracy while eliminating highly correlated and less impactful attributes.

The performance of the ML models was analyzed using multiple evaluation metrics, including **accuracy, precision, recall, F1-score, and the confusion matrix**. The results indicated that among all models, **Extreme Gradient Boosting (XGBoost) achieved the highest accuracy of 97%**, making it the most effective classifier for heart disease prediction in this study. These findings highlight the potential of ML in transforming cardiovascular disease diagnostics and aiding in early intervention strategies.

References

[1] A and A. Al-Moosa, "Using Data Mining Techniques to Predict Diabetes and Heart Diseases," in *2018 4th International Conference on Frontiers of Signal Processing*, 2018, p. 150–154.

[2] S. Bashir, U. Qamar and M. Javed, "An ensemble-based decision support framework for intelligent heart disease diagnosis," in *IEEE. ICCRDA 2020 IOP Conf. Series: Materials Science and Engineering*, vol. 1022, IOP Publishing, 2014, p. 259–264.

[3] N. Brown, T. Young, D. Gray, A. Skene and J. Hampton, "Inpatient deaths from acute myocardial infarction, 1982-92: analysis of data in the Nottingham heart attack register," *BMJ,* vol. 315, pp. 159-164, 19 July 1997.

[4] H. Chen, S. Huang, P. S. Hong, C. Cheng and E. Lin, "Prediction of heart disease using graph neural networks," in *International Conference on Artificial Intelligence and Industrial Design (AIID 2022)*, SPIE, 2011, p. 557–560.

[5] M. Dangare and V. Yadav, "Gaming on an Immersive Virtual Reality Platform to Ameliorate the Level of Anxiety in Patients Undergoing Congenital Heart Disease," *Cureus,* vol. 47, p. 44–48, 2012.

[6] A. Dewan and M. Sharma, "Prediction of heart disease using a hybrid technique in data mining classification," in *nd International Conference on Computing for Sustainable Global Development*.

[7] R. Folsom A, R. Prineas, S. A. &. Kaye and J. Soler, "Body fat distribution and self-reported prevalence of hypertension, heart attack, and other heart disease in older women," *International journal of epidemiology,* vol. 18, p. 361–367, 1989.

[8] A. Ganna, P. Magnusson, N. Pedersen, U. De Faire, M. Reilly, J. Ärnlöv, J. Sundström, A. Hamsten and E. Ingelsson, "Multilocus Genetic Risk Scores for Coronary Heart Disease Prediction," *Arteriosclerosis, Thrombosis, and Vascular Biology,* vol. 33, pp. 2267-2272, September 2013.

[9] M. Jabbar, B. Deekshatulu and P. Chandra, "Heart disease prediction using lazy associative classification," in *2013 International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s)*, IEEE, 2013, p. 40–46.

[10] H. Jee, Y. Jang, D. Oh, B. Oh, S. H. Lee, S. W. &. Park and Y. Yun, "A coronary heart disease prediction model: the Korean Heart Study," *BMJ open,* vol. 4, p. 5025, 2014.

[11] A. Methaila, P. Kansal, H. Arya and P. Kumar, "Early Heart Disease Prediction Using Data Mining Techniques," in *Computer Science & Information Technology (CS & IT)*, Academy & Industry Research Collaboration Center (AIRCC), 2014.

[12] C. Ordonez, "Association Rule Discovery With the Train and Test Approach for Heart Disease Prediction," *IEEE Transactions on Information Technology in Biomedicine,* vol. 10, pp. 334-343, April 2006.

[13] L. Parthiban and R. Subramanian, "An intelligent agent for detection of erythemato- squamous diseases using Co-active Neuro-Fuzzy Inference System and genetic algorithm," in *2009 International Conference on Intelligent Agent & Multi-Agent Systems*, vol. 3, IEEE, 2008.

[14] S. Patel and Y. Chauhan, "Heart attack detection and medical attention using motion sensing device -kinect," *International Journal of Scientific and Research Publications,* vol. 4, p. 1–4, 2014.

[15] L. Piller, B. Davis, J. Cutler, W. Cushman, J. Wright, J. Williamson, F. Leenen, P. Einhorn, O. Randall, J. Golden and L. Haywood, "Validation of Heart Failure Events in the Antihypertensive and Lipid Lowering Treatment to Prevent Heart Attack Trial (ALLHAT) Participants Assigned to Doxazosin and Chlorthalidone," *Current Controlled Trials in Cardiovascular Medicine,* vol. 3, 14 November 2002.

[16] M. Raihan, S. Mondal, A. More, M. Sagor, G. Sikder, M. Majumder, M. Al Manjur and K.

Ghosh, "Smartphone based ischemic heart disease (heart attack) risk prediction using clinical data and data mining approaches, a prototype design," in *2016 19th International Conference on Computer and Information Technology (ICCIT)*, IEEE, 2016, p. 299–303.

[17] R. Shinde, S. Arjun, P. Patil and J. Waghmare, "An intelligent heart disease prediction system using k-means clustering and Naïve Bayes algorithm," *International Journal of Computer Science and Information Technologies*, vol. 6, p. 637–639, 2015.

[18] J. Soni, U. Ansari, D. Sharma and S. Soni, "Predictive Data Mining for Medical Diagnosis: An Overview of Heart Disease Prediction," *International Journal of Computer Applications*, vol. 17, pp. 43-48, 31 March 2011.

[19] H. Takci, "Improvement of heart attack prediction by the feature selection methods," *TURKISH JOURNAL OF ELECTRICAL ENGINEERING & COMPUTER SCIENCES*, vol. 26, pp. 1-10, 2018.

[20] G. Wolgast, C. Ehrenborg, A. Israelsson, J. Helander, E. Johansson and H. Manefjord, "Wireless Body Area Network for Heart Attack Detection [Education Corner]," *IEEE Antennas and Propagation Magazine*, vol. 58, pp. 84-92, October 2016.