HEART DISEASE PREDICTION USING MACHINE LEARNING ALGORITHMS

A Project report submitted
In partial fulfilment for the degree

B. Tech
in
Computer Science and Engineering
by

Prince Mehta - 21033440025
Mohammad Aamir Faraz - 21033440019
Ankit Kumar – 21033220008
Sanjana Kumari – 21033440034
Md. Anas – 21033440016
pursuing in
Department of Computer Science and Engineering

RAMGARH ENGINEERING COLLEGE
(Estd. By Govt. of Jharkhand & run by Techno India under PPP)

June, 2025

CERTIFICATE

This is to certify that the project report entitled **HEART DISEASE PREDICTION USING MACHINE LEARNING ALGORITHMS** submitted by **Prince Mehta(21033440025), Mohammad Aamir Faraz(21033440019), Ankit Kumar(21033440008), Sanjana Kumari(21033440034), Md. Anas(21033440016)** to the Ramgarh Engineering College, Ramgarh in partial fulfillment for the award of the degree of **B. Tech in (Computer Science and Engineering)** is a bonafide record of project work carried out by him/her under my/our supervision. The contents of this report, in full or in parts, have not been submitted to any other Institution or University for the award of any degree or diploma.

Signature of Supervisor

Ms. Jyoti Kumari
(Assistant Professor)
Department of CSE

Counter signature of HOD with seal

Mr. Ashim Kumar Mahato
(Assistant Professor)
Head of the Department
Department of CSE

Signature of External Examiner

Ramgarh, Jharkhand

June, 2025

DECLARATION

We declare that this project report titled submitted in "**Heart Disease Prediction using Machine Learning Algorithms",** submitted in partial fulfillment of the degree of **B. Tech in Computer Science and Engineering,** is a record of original work carried out by us under the supervision of **Ms. Jyoti Kumari** has not formed the basis for the award of any other degree, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

Prince Mehta – 21033440025

Mohammad Aamir Faraz – 210033440019

Ankit Kumar – 21033440008

Sanjana Kumari – 21033440034

Md. Anas - 21033440016

Ramgarh, Jharkhand

June, 2025

ACKNOWLEDGEMENT

We pay our veneration and heartfelt gratitude to Ms. Jyoti Kumari of the Computer Science and Engineering Department, in the absence of whose erudite guidance, we could not have presented this thesis. We shall remain indebted to them throughout my life for their magnanimous help.

We are also grateful to Ms. Ankita Roy, Project Coordinator of the Department and Mr. Ashim Kr. Mahato, HOD of Computer Science and Engineering Department, whose immense help, inspiration and advice helped me to present the project report.

We shall be failing in my duties, if we do not express our sense of gratitude and respect to all the faculty members of Computer Science and Engineering Department, who ushered us to the successful completion of the project report. We would like to thank all the staff members of Computer Science and Engineering Department for their kind cooperation extended towards us.

Finally, we pay our respect and love to our beloved parents and friends without whose unimpeded support and encouragement, this thesis would not have seen the light of the day.

Prince Mehta – 21033440025
Mohammad Aamir Faraz – 210033440019
Ankit Kumar – 21033440008
Sanjana Kumari – 21033440034
Md. Anas - 21033440016

Ramgarh, Jharkhand
June, 2025

ABSTRACT

Machine learning is widely used in various industries worldwide, including healthcare. It has become increasingly popular due to its ability to predict the presence or absence of various health conditions such as locomotor disorders, heart diseases, and more. By accurately predicting such conditions well in advance, machine learning can provide crucial insights to doctors, allowing them to tailor their diagnosis and treatment plans for each patient. The utilization of machine learning algorithms for predicting heart disease is a growing area of research in healthcare. This study aimed to investigate the potential of using such algorithms and propose enhancements to the reference project. The research involved using an open-source dataset of medical records and implementing several machine learning models, such as K-Nearest Neighbors, Support Vector Classifier, Logistic Regression, Decision Trees, and Random Forest. The proposed enhancement included integrating a logistic regression to enhance the predictive performance. The research process included data preprocessing, model implementation, and evaluation. The results demonstrated that the proposed enhancement significantly improved the predictive performance, achieving an accuracy of 90. The study concludes that machine learning can be a valuable tool for predicting heart disease and proposes further research directions to enhance the model's interpretability and scalability.

Keywords: Heart Disease Prediction, Logistic Regression, Diseases diagnosis, Confusion Matrix, Supervised Learning.

Contents

List of Figures

List of Tables

Chapter 1

1. Introduction

Heart disease is a term that encompasses a range of conditions affecting the heart and blood vessels, such as coronary artery disease, heart failure, arrhythmias, and valve disorders. It is a leading cause of mortality worldwide, responsible for around 31% of all global deaths. Understanding the different types of heart disease, their causes, and risk factors is crucial for predicting and preventing heart disease.

Coronary Artery Disease (CAD) is the most common type of heart disease, and it occurs when plaque builds up in the arteries that supply blood to the heart. This can lead to a reduction in blood flow, resulting in chest pain (angina) or a heart attack.

Heart Failure happens when the heart is unable to pump enough blood to meet the body's requirements. It can be caused by various factors, including high blood pressure, coronary artery disease, and diabetes.

1.1 Overview

Heart disease remains one of the leading causes of death globally, accounting for nearly one-third of all deaths worldwide. Early detection and accurate prediction are vital in reducing mortality and improving patient outcomes. Heart disease can manifest in various forms such as coronary artery disease, heart failure, arrhythmias, and valve disorders, with risk factors including hypertension, high cholesterol, diabetes, smoking, obesity, and family history.

Predictive models using machine learning techniques have emerged as powerful tools in identifying individuals at risk of developing heart disease. By analyzing patterns in medical data, these models help clinicians make timely and personalized healthcare decisions. In this study, a heart disease prediction model was developed and deployed using **Streamlit**, making it accessible via a web-based interface. This enables real-time, user-friendly predictions and bridges the gap between technical models and practical use.

1.2 Problem Statement

Despite progress in machine learning for heart disease prediction, most models remain inaccessible to doctors and end users due to their academic or offline nature. Traditional methods lack precision, and there's a critical need for a reliable, real-time, and user-friendly tool to support both clinical decision-making and personal risk assessment**.**

1.3 Motivation

The motivation behind this project stems from the increasing global burden of heart disease and the need for efficient tools that aid in early detection. While many researchers have implemented machine learning models for heart disease prediction, most have not deployed them for practical interface use by clinicians or the general public.

This project stands out by deploying the trained machine learning model on **Streamlit**, offering a user-friendly interface that makes predictions accessible in real-time. This reduces the gap between data science and clinical practice and encourages the adoption of predictive analytics in healthcare.

1.4 Objective

- To use open-source medical data for heart disease prediction.
- To apply and compare machine learning algorithms like Logistic Regression, SVM, KNN, Decision Trees, and Random Forest.
- To evaluate models using performance metrics.
- To deploy the final model via a web-based interface using Streamlit.

1.5 Aim

The aim of this study is to build and deploy a reliable machine learning-based heart disease prediction system that leverages patient data to provide accurate risk assessments and can be used effectively through a web-based interface.

1.6 Benefits

This study offers several significant benefits, starting with early detection, which helps identify high-risk individuals for timely medical intervention and preventive care. The model's deployment through a **web-based platform** using **Streamlit** enhances accessibility, making it easy to use for both healthcare professionals and individuals.

The system improves efficiency by automating the prediction process, saving time and reducing manual effort. It is also scalable, allowing future integration with additional features and real-world healthcare data. Moreover, the user-friendly interface ensures that even non-technical users can interact with the tool effectively, supporting wider adoption in practical healthcare settings.

Chapter 2

2. Literature Review

Saba Bashir et al. (2014) [1] implemented an ensemble-based decision support framework combining Naïve Bayes, Decision Tree (using Gini Index), and Support Vector Machine classifiers with a majority voting scheme to predict heart disease. The model achieved 81.82% accuracy, 73.68% sensitivity, and 82.86% specificity on the UCI heart disease dataset. As heart disease remains a leading global cause of death, the study emphasizes the need for intelligent data mining tools to improve diagnostic accuracy. Their system incorporates data preprocessing, feature selection, and integrates heterogeneous classifiers to outperform individual models in reliability and predictive performance.

Shinde et al. (2015) [2] developed an intelligent heart disease prediction system that integrates K-means clustering and the Naïve Bayes algorithm to enhance diagnostic accuracy. The system first applies K-means clustering to group patient data based on similarities, which helps in reducing noise and improving classification performance. Then, Naïve Bayes is used to classify individuals as either having heart disease or not, based on selected attributes. Experimental results on standard datasets demonstrated that the hybrid approach improves prediction accuracy compared to using Naïve Bayes alone, making it a viable tool for early and efficient heart disease diagnosis in clinical settings.

Dewan and Sharma (2015) [3] proposed a classification technique for heart disease prediction by combining Decision Tree and K-Nearest Neighbor (KNN) algorithms. Their approach aims to improve the diagnostic accuracy by leveraging the strength of Decision Tree in feature selection and the precision of KNN in classification. Using the UCI heart disease dataset, the model achieved better accuracy compared to individual classifiers. The study highlights that integrating multiple data mining methods can enhance performance and reliability, making The approach a practical solution for effective clinical decision support in heart disease diagnosis.

Dangare, C. S., and Apte, S. S. (2012) [4] proposed an enhanced heart disease prediction system by applying multiple data mining classification techniques such as Decision Tree, Naïve Bayes, and Neural Networks. Their model emphasizes the inclusion of 13 medical parameters (including obesity, smoking, and physical inactivity), improving the prediction process beyond

the traditional 6–8 attributes. The system achieved an accuracy of up to 86.53%, showcasing that increasing the number of attributes and refining the preprocessing steps significantly enhances diagnostic outcomes. Their findings reinforce the value of comprehensive datasets and hybrid analytical methods in clinical decision-making.

M. Raihan et al. (2016) [5] developed a smartphone-based heart attack (Ischemic Heart Disease) risk prediction system using clinical data and data mining approaches, incorporating statistical analysis with a user-friendly Android application. Based on data from 835 patients, the app classifies users into high, medium, or low-risk categories using various input factors. The system initially achieved 76.05% accuracy and 89.25% sensitivity, which improved to 86% accuracy and 87.6% sensitivity using the C4.5 decision tree algorithm. The model enables early screening and risk assessment without requiring mandatory lipid profile values, making it suitable for preventive care, especially in resource-limited settings.

Ordonez (2006) [6] introduced an algorithm that applies association rule mining with search constraints and a train-test validation approach to enhance heart disease prediction. By using constraints such as maximum rule size, item filtering, attribute grouping, and rule filtering on antecedents/consequents, the approach significantly reduces medically irrelevant rules. The use of an independent test set ensures better generalization of the discovered patterns. Experiments conducted on a real medical dataset containing 655 records and 25 attributes produced a concise set of association rules with high predictive accuracy, offering valuable insights into the relationship between heart perfusion data, risk factors, and disease presence in specific arteries.

Masethe et al. (2014) [7] implemented several classification algorithms such as J48, Naïve Bayes, REPTree, CART, and Bayes Net to accurately predict heart disease, achieving a maximum accuracy of 83%. Recognizing heart disease as a major global health concern, their study aimed to support medical practitioners by reducing complexity in diagnosis through intelligent data analysis. Using the WEKA tool and heart disease datasets, the researchers conducted various experiments to evaluate the effectiveness of each algorithm. Their results demonstrated that data mining techniques are valuable in medical decision-making, offering a non-invasive, efficient, and accurate approach to identifying high-risk patients.

Choi et al. (2017) [8] developed a recurrent neural network (RNN) model to detect early onset of heart failure using longitudinal electronic health records (EHR). Their model captures temporal patterns in patient data and outperforms traditional machine learning methods like

logistic regression and multilayer perceptrons, achieving higher accuracy and AUC scores. The study demonstrates that RNNs are effective for modeling complex, time-dependent health data to support early diagnosis and preventive care.

F. S. Alotaibi (2019) [9] implemented a machine learning model to predict heart failure disease using clinical datasets and evaluated multiple classification algorithms including Decision Tree, Support Vector Machine, and Naïve Bayes. The study emphasized the significance of data preprocessing and feature selection in improving model performance. Among the models tested, the Decision Tree achieved the highest accuracy of 85.7%, demonstrating its effectiveness for heart failure prediction. The research contributes to developing intelligent decision support systems aimed at improving early diagnosis and treatment outcomes.

Sangeetha et al. (2014) [10] developed a comprehensive heart disease prediction model using machine learning algorithms such as Gradient Boosting Classifier, Support Vector Machine, and Logistic Regression on the UCI dataset. After preprocessing and structuring data in CSV format, they evaluated performance using confusion matrix metrics. Their results highlighted Logistic Regression as the most effective model, achieving 85% accuracy, alongside high precision and recall. The study underscores the utility of ML in enhancing the speed and accuracy of clinical decision-making for cardiac patient assessment.

Kumar et al. (2015) [11] conducted a structured review on the use of machine learning (ML) in heart disease prediction, categorizing the literature into five thematic clusters: diagnostics, ML models, feature engineering, emerging technologies, and multi-disease applications. The study highlights that deep learning models—especially CNNs and hybrid architectures like CNN-LSTM—consistently outperform traditional models in sensitivity, specificity, and AUC. The authors emphasize the role of wearable devices, federated learning (FL), and explainable AI (XAI) in transforming healthcare diagnostics while addressing critical challenges like data privacy, class imbalance, and model interpretability. They also discuss regulatory and ethical concerns, such as compliance with GDPR and HIPAA. This review offers valuable insights into the evolution of ML-based cardiovascular diagnostics and presents a forward-looking perspective on integrating AI with clinical workflows.

Khan et al. (2024) [12] conducted a comprehensive analysis of heart disease detection using machine learning, ensemble, and deep learning algorithms. Their study evaluated a wide range of models—including Decision Tree, Random Forest, SVM, K-Nearest Neighbors, Naïve Bayes, Neural Networks, and advanced ensemble methods—on clinical datasets. They also explored the impact of feature selection, hyperparameter tuning, and deep learning frameworks. Results showed that ensemble techniques and deep learning models consistently outperformed traditional methods, delivering higher accuracy, specificity, and sensitivity. The paper highlights current trends and challenges in the field and outlines future directions for enhancing clinical decision support systems with improved interpretability and ethical compliance.

Bhatt et al. (2013) [13] proposed an efficient heart disease prediction model using multiple machine learning techniques, including Random Forest, Decision Tree, XGBoost, and Multilayer Perceptron (MLP). The authors addressed key limitations of previous studies such as small datasets and high overfitting risk by using a large, real-world dataset comprising 70,000 patient records from Kaggle. They introduced a preprocessing phase involving outlier removal, feature binning, and categorical conversion, and applied k-modes clustering to improve classification. The study achieved its highest accuracy of 87.28% using the MLP model with cross-validation. The authors concluded that MLP outperformed other models in precision, recall, and AUC score (0.95). This research emphasizes the role of machine learning and clustering in predictive healthcare modeling and supports its integration into early diagnostic systems.

Jindal et al. (2021) [14] proposed a heart disease prediction system that leverages machine learning algorithms such as K-Nearest Neighbors (KNN), Logistic Regression, and Random Forest to improve diagnostic accuracy. Using a dataset from the UCI repository containing 304 patient records with 14 medical attributes, the study evaluated these models for their effectiveness in classifying individuals at risk of cardiovascular disease. Among the algorithms tested, KNN achieved the highest prediction accuracy of 85.52%, outperforming the others in both speed and cost-efficiency. The paper emphasizes the significance of combining multiple classification methods to enhance the reliability of heart disease prediction models and suggests that such systems can aid healthcare professionals in early diagnosis and treatment planning.

Li et al. (2020) [15] proposed a machine learning-based intelligent system for heart disease diagnosis utilizing both standard and novel feature selection techniques. The study employed classification algorithms such as Logistic Regression, Support Vector Machine (SVM), Naïve Bayes, K-Nearest Neighbor, Decision Tree, and Artificial Neural Network. To optimize the

model's performance, the authors incorporated four state-of-the-art feature selection methods (Relief, MRMR, LASSO, LLBFS) and introduced a new technique—Fast Conditional Mutual Information (FCMIM). Using the Cleveland dataset, they implemented Leave-One-Subject-Out (LOSO) cross-validation and achieved the highest accuracy of 82.37% with the FCMIM-SVM combination. The study demonstrates that effective feature selection significantly improves classification accuracy and execution time, thereby supporting the use of ML systems in clinical heart disease diagnosis and e-healthcare decision-making.

Comparative Table

Study	Techniques Used	Accuracy	Strengths	Limitations
Saba Bashir et al. [1]	Naïve Bayes, Decision Tree (Gini), SVM with majority voting	81.82%	Improved reliability with ensemble; High specificity (82.86%)	Moderate sensitivity; No user deployment
Shinde et al. [2]	K-means clustering + Naïve Bayes	Not specified	Improved classification by reducing noise via clustering	Accuracy not reported; No deployment
Dewan & Sharma [3]	Decision Tree + KNN	Higher than individual models	Combines feature selection and classification strength	Limited dataset; No deployment
Dangare & Apte [4]	Decision Tree, Naïve Bayes, Neural Network	86.53%	Uses 13 attributes; richer dataset	Not publicly available; Not deployed
Ordonez [5]	Association Rule Mining with constraints	Not specified	High generalizability; Removes irrelevant rules	Rule-based, may lack flexibility; No app deployment
Masethe et al. [6]	J48, Naïve Bayes, REPTree, CART, Bayes Net	83%	Multiple algorithm comparison	No deployment; Limited integration
Choi et al. [7]	Recurrent Neural Network (RNN)	Higher AUC than MLP/logistic regression	Captures time-series from EHR	High complexity; No user-level application
Alotaibi [8]	DT, SVM, Naïve Bayes	85.7%	Highlights preprocessing importance	No external deployment; Model only
Sangeetha et al. [9]	GBC, SVM, Logistic Regression	85%	Strong evaluation with confusion matrix	No practical deployment
Kumar et al. [10]	CNN, CNN-LSTM, FL, XAI	Varies; DL models better	Emphasizes explainability, privacy, tech trends	Theoretical focus; No tool built
Khan et al. [11]	Ensemble & Deep Learning models	Higher accuracy, sensitivity	Highlights importance of tuning and DL	No real-time implementation
Bhatt et al. [12]	RF, DT, XGBoost, MLP	87.28% (MLP)	Large real-world dataset; AUC 0.95	Not deployed for clinical users
Jindal et al. [13]	KNN, Logistic Regression, RF	85.52% (KNN)	Speed and cost-efficient	Dataset limited (304); No mobile/web interface

Li et al. [14]	SVM, DT, ANN + Feature Selection (FCMIM, LASSO, etc.)	82.37% (FCMIM-SVM)	Advanced FS improves speed and accuracy	No deployment; Only model optimization
Our Work • Deploy model via web & Android app for public access • They can train models with their own dataset and predict based upon that. • Available for patient/doctor use directly				

Table. 2.1. Comparative Table

2.1 Research Gap

While numerous studies have explored heart disease prediction using machine learning models—ranging from classical classifiers to deep learning architectures—**most remain confined to academic research or offline environments**. They typically evaluate model performance on benchmark datasets like UCI but **do not translate into accessible, user-ready tools**. Existing systems often:

- Lack **real-time accessibility** for doctors and patients
- Do not allow **custom dataset training** by end users
- Are not integrated into **cross-platform applications** (e.g., web or Android)
- Focus more on **accuracy benchmarks** rather than **practical deployment and usability**
- Miss interactive interfaces for **parameter tuning or visual performance comparisons**

Moreover, **none of the reviewed studies** offered a fully **deployable solution** where end users (health professionals or individuals) could interactively:

- Train multiple ML models,
- Adjust parameters,
- Upload their own data,
- View performance metrics in real-time,
- And receive predictions on an accessible platform (web + mobile).

Chapter 3

3. Problem Statement

Despite significant advancements in healthcare technologies and diagnostic methodologies, early and accurate prediction of heart disease remains a persistent challenge due to the complex interplay of risk factors and individual patient variability. Traditional clinical methods often depend on manual scoring systems that may lack precision and consistency across diverse populations. While numerous machine learning-based models have demonstrated high accuracy in research settings, a critical gap lies in their limited accessibility and deployment. These models are often confined to academic environments, lacking integration into user-friendly, real-time systems for practical use. As a result, both healthcare professionals and end users face difficulties in leveraging these predictive tools during clinical decision-making or personal health assessments. There is a pressing need for a reliable, accessible, and real-time heart disease prediction system that bridges this gap — empowering doctors with clinical support tools and enabling individuals to assess their heart health risk with ease and confidence.

3.1 Project Planning

Developing an accessible and customizable heart disease prediction system using machine learning requires careful and user-centric project planning. The goal is to address the gap between advanced ML models and their practical usability by both healthcare professionals and the general public. The following structured steps were followed to ensure effectiveness, accessibility, and real-world utility:

i. Defining the Problem and Objective

The core objective was to develop a real-time, accessible, and accurate heart disease prediction tool that could be used by both doctors and patients. The problem addressed was the **inaccessibility and lack of deployment of ML models in existing research**, which remain academic prototypes. Our system was designed to:

- Predict heart disease using clinical data
- Allow users to upload their own datasets
- Enable model selection and parameter customization
- Offer seamless access via web and Android platforms

ii. Data Collection and Preprocessing

We utilized the widely recognized **UCI Heart Disease Dataset** as the base dataset for model training and validation. Users were also given the option to **upload custom datasets** for personalized predictions. The following preprocessing steps were applied:

- Handling missing values and data inconsistencies
- Normalizing features to standardize scale
- Encoding categorical variables
- Splitting the dataset into training and testing sets

iii. Feature Selection and Engineering

Key features from the dataset—such as age, cholesterol levels, chest pain type, and heart rate—were selected based on domain relevance. Feature engineering was applied to:

- Transform raw values into model-friendly formats
- Create meaningful interactions (e.g., combining ST depression and exercise data)
- Improve model accuracy through dimensionality reduction and analysis

iv. Model Selection and Evaluation

Seven different machine learning algorithms were implemented and made selectable via the UI:

- Logistic Regression
- Decision Tree
- Random Forest
- K-Nearest Neighbors (KNN)
- Support Vector Machine (SVM)
- Gaussian Naive Bayes
- Artificial Neural Network (ANN)

Each model was evaluated using:

- Accuracy
- Precision
- Recall
- F1 Score
- Confusion Matrix

This allowed users to **compare performance in real-time** and choose the best-suited model for their specific dataset.

v. Deployment and Implementation

To maximize accessibility and usability:

- The predictive system was deployed as a **web application using Streamlit**
- A companion **Android app** was developed for mobile access
- Users can upload datasets, tune model parameters, and get predictions with ease
- Confusion matrix visualizations and classification reports are dynamically generated
- Trained models can be **downloaded** for external use

This deployment ensures that the system is not only accurate but also **usable in real-world clinical and personal health monitoring scenarios**.

3.2 Project Analysis

To ensure the effectiveness, usability, and impact of a machine learning–based heart disease prediction system, it is important to analyze various technical and practical factors throughout the project lifecycle. This analysis ensures that the model not only performs well but is also usable and accessible to both healthcare professionals and the general public. Key factors considered in our project include:

a) Data Quality and Accessibility

The system was built using the **UCI Heart Disease Dataset**, a widely accepted benchmark in clinical ML research. Additionally, users can **upload their own datasets**, which introduces variability in data formats and quality. Therefore, robust preprocessing pipelines were implemented to handle:

- Missing or inconsistent values
- Outlier detection
- Input validation and feedback to the user

This ensures both **data quality** and **user accessibility**, making the system reliable for diverse input scenarios.

b) Feature Engineering and Flexibility

Essential medical features such as age, cholesterol, chest pain type, and maximum heart rate were selected based on domain knowledge. Feature engineering involved:

- Normalization and encoding for ML compatibility
- Derived features (e.g., combining stress test results with ECG patterns)
- Real-time feature handling for user-uploaded data

This step enhanced model performance while maintaining **clinical relevance and flexibility** for non-technical users.

c) Model Selection and Evaluation Framework

Seven machine learning models were included in the system—ranging from simple classifiers (Logistic Regression) to more complex ones (Neural Networks). A built-in evaluation module allowed users to:

- **Compare models** using accuracy, precision, recall, F1 score, and confusion matrix
- **Tune hyperparameters** via the UI
- Select the best model for their dataset without coding knowledge

This user-driven evaluation mechanism ensures **transparency, adaptability, and real-time performance analysis**.

d) Interpretability and User Trust

For a prediction tool to be useful in real-world health settings, it must be **interpretable and trustworthy**. Our platform includes:

- Visual representations of model performance (confusion matrices, classification reports)
- Explanations of algorithm behavior and decision boundaries (for decision trees, logistic regression, etc.)
- Feature importance insights (where applicable)

These tools support doctors and patients in **understanding and trusting** the model's predictions, improving adoption and decision-making.

Chapter 4

4. Methodology

In this chapter, shall detail and list out the machine learning and data mining method that were used in this project. We also explain the dataset and how it was collected, and then show the preprocessing of the dataset step-by-step. We shall first review the working of each machine learning model used in the project and then move on to the dataset. This chapter is important because it lets the reader get used to the project environment which consists of the machine learning models and the dataset. Understanding the dataset is of critical importance because some of its characteristics are the reason why some models perform the way they do, and we discuss more about this in Chapter 5.

4.1 Machine Learning Algorithms:

I. K-nearest neighbors (KNN)

K-nearest neighbors (KNN) is a machine learning algorithm that can be used for classification problems. It works by looking at the k closest data points in the training set to a new observation and assigns a label based on the majority class among these k neighbors. Think of it like asking your friends for advice on a decision and going with what most of them suggest.

KNN is a flexible algorithm that can work well on datasets with non-linear decision boundaries, and it doesn't require any training on the data before making predictions. One of the benefits of KNN is its simplicity and ease of implementation.

However, KNN can be computationally expensive, especially on large datasets, because it needs to calculate distances between each observation and every other observation in the training set. Also, it's important to choose an appropriate value for k to avoid overfitting or underfitting the data.

Steps in KNN Classification:

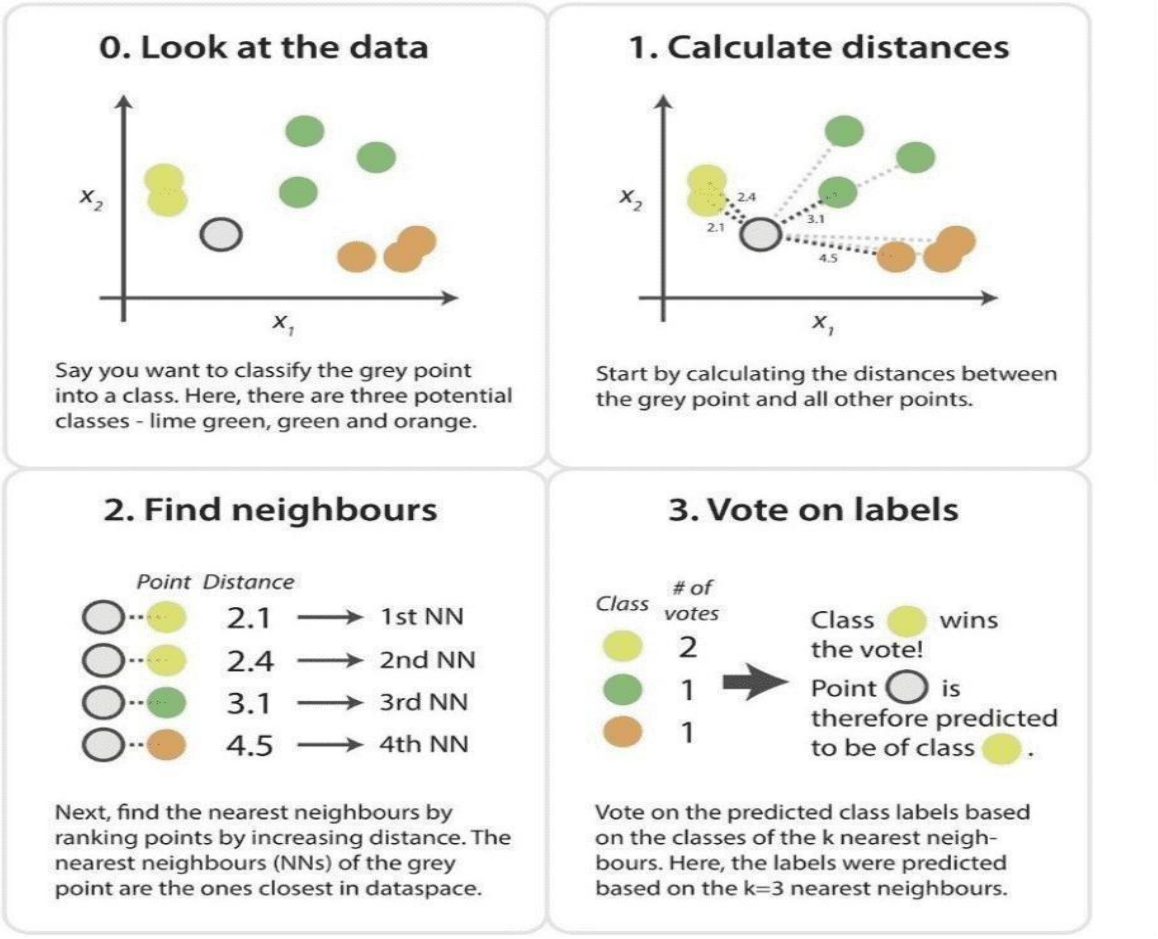

Fig. 4.1. Steps in KNN Classification

II. *Support vector machine (SVM)*

Support vector machine (SVM) is a popular machine learning algorithm that is used for classification and regression analysis. It was first introduced in 1992 by Vladimir Vapnik and his colleagues, and has become a valuable tool in the field of machine learning.

The basic idea behind SVM is to find a straight line, or hyperplane, that separates two groups of data points. The goal is to find the hyperplane that maximizes the distance between the two groups of points, which is called the margin. This approach works well for linearly separable data, but for non-linearly separable data, SVM uses a kernel function to map the data to a higher-dimensional space where it can be more easily separated. In Fig. 4.2. the working of Support Vector Machine in separating two classes is explained.

SVM can be used for both linear and non-linear classification problems. For linear classification, SVM finds a hyperplane that separates the data. In non-linear classification, SVM maps the data to a higher-dimensional space using a kernel function and then finds a hyperplane that separates the data in that space.

One type of SVM, called support vector classifier (SVC), is used for binary classification problems. SVC finds the hyperplane that separates two classes of data in such a way that the margin between the two classes is maximized. SVC can be used for both linear and non-linear classification problems.

In linear SVC, the goal is to find the weight vector w and bias term b that define the hyperplane that separates the data with the maximum margin. This can be expressed as an optimization problem, which can be solved using various optimization algorithms such as gradient descent or quadratic programming.

SVC has several advantages over other classification algorithms such as decision trees and logistic regression. For example, it can handle high-dimensional feature spaces and is less prone to overfitting. It can also handle non-linearly separable data by using kernel functions to map the data into a higher-dimensional feature space. Additionally, SVM has been shown to be effective in a wide range of applications such as image classification, text classification, and bioinformatics.

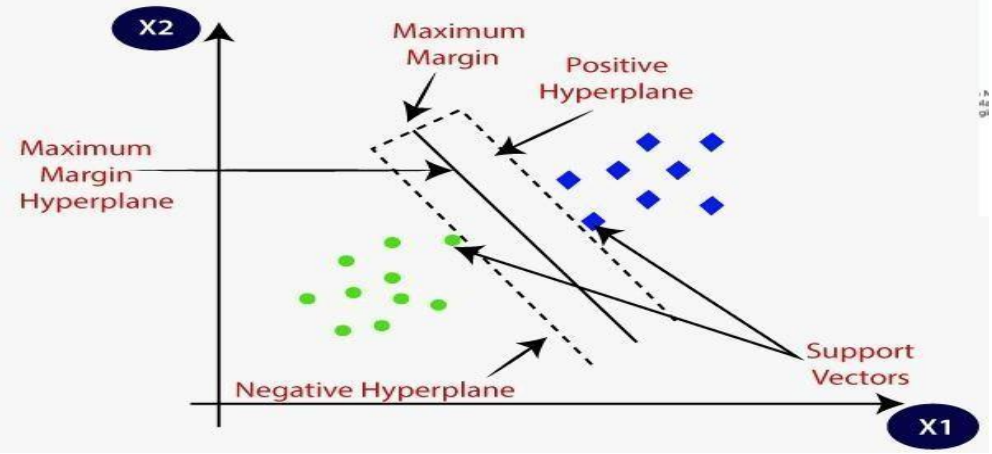

Fig. 4.2. Demonstration of working of Support Vector Machine in separating two classes.

III. Decision Tree Classifier

Decision tree is a popular algorithm in machine learning that helps to make decisions based on various outcomes. It's like a flowchart where each internal node represents a decision based on a feature, and each leaf node represents a final outcome or a class label.

Decision trees can be used for both classification and regression tasks. For example, in classification tasks, decision trees are used to classify a set of observations into one of several classes, while in regression tasks, decision trees are used to predict a continuous variable.

The algorithm starts by selecting the most informative feature based on a measure of impurity or information gain and splitting the data based on this feature. This process is recursively repeated until each leaf node represents a pure class or a stopping criterion is met. In Fig. 4.3. working of the Decision Tree is explained.

The great thing about decision trees is that they are intuitive and easy to interpret. Additionally, they can handle both numerical and categorical data and can work well even with noisy or missing data. However, decision trees are prone to overfitting if not properly tuned or if the tree becomes too complex.

In this project, we make use of the CART model. As its name suggests, it can be used for both, classification and regressions tasks. To evaluate splits in the decision tree, it uses the Gini impurity as a cost function and tries to minimize the cost while trying to split nodes. Sometimes, entropy is also used as a measure to evaluate splits. Entropy is given by eq. (3.1) as,

$$H(P) = -\sum_{i=1}^{n} p_i \log(p_i) \qquad (3.1)$$

The Gini index or Gini impurity refers to how homogeneous or heterogeneous the elements are and it is scored at a scale of 0 to 1. Hence, if the Gini impurity is 0, then all the elements are similar, and if it is 1, then all the elements or samples are maximally unequal. It can be calculated by eq. (3.2) as,

$$\text{Gini index} = 1 - \sum_{i=1}^{n} p_i^2 \qquad (3.2)$$

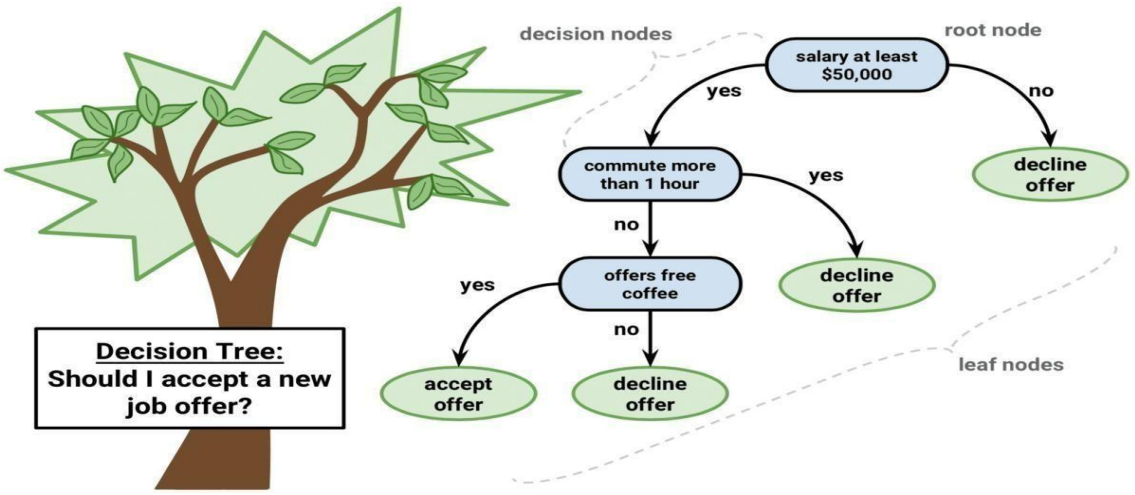

Fig. 4.3. Demonstration of working of Decision Tree.

IV. Random Forest Classifier

Random forest is a popular algorithm in machine learning that helps us make predictions. It can be used for tasks like classification and regression, and was first introduced by a man named Leo Breiman in 2001. Since then, it has become a favorite tool of many data scientists.

The way random forest works is by creating a group of decision trees that work together to make more accurate predictions. These decision trees are built using random subsets of the training data and features, which helps to reduce overfitting and improve the accuracy of the model. Each decision tree in the random forest produces a classification result, and the final output of the random forest is determined by combining the results of all the decision trees. In Fig. 4.4. A Random Forest consisting of K Decision Trees.

One of the great things about random forest is that it can handle complex and high- dimensional data. It's also less prone to overfitting than other algorithms like decision trees, logistic regression, and support vector machines. Another advantage is that it's easy to use and can be applied to many different problems, such as image classification, text classification, and bioinformatics.

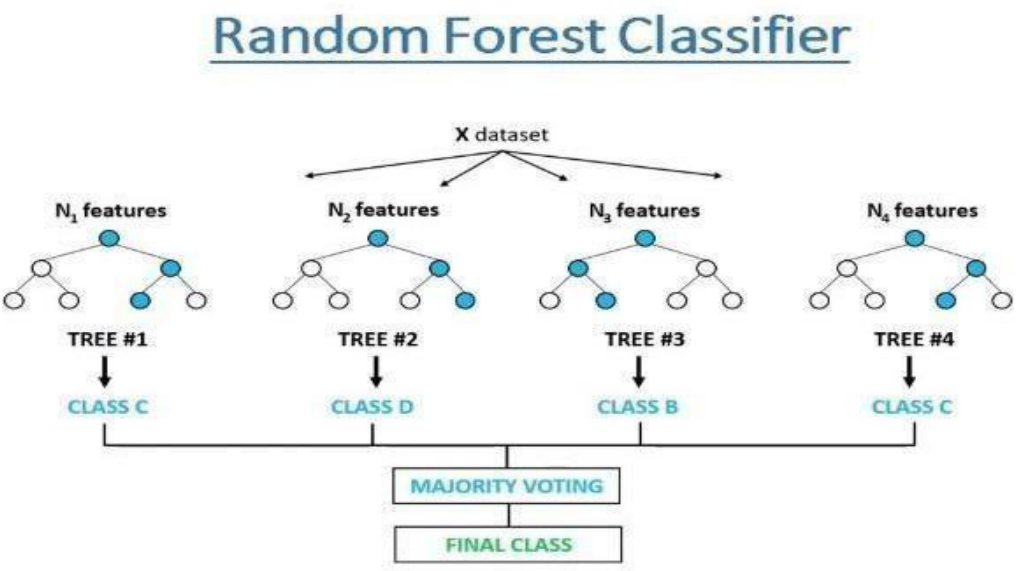

Fig. 4.4. A Random Forest consisting of K Decision Trees

V. Gaussian Naïve Bayes

Gaussian Naive Bayes is a machine learning algorithm that's often used for classification tasks. It's based on probability theory and works by calculating the probability of each class given the input features.

To do this, the algorithm makes two assumptions: that the features are independent and that they follow a Gaussian (or normal) distribution. Based on these assumptions, it calculates the mean and variance of each feature for each class. In Fig. 4.5. there is demonstration of Gaussian Naive Bayes.

Once these statistics are calculated, the algorithm can use them to compute the likelihood of each input feature given each class. It then combines this likelihood with the prior probability of each class (which is just the proportion of training examples that belong to that class) to calculate the posterior probability of each class given the input features.

Finally, the algorithm selects the class with the highest posterior probability as the output. Overall, Gaussian Naive Bayes is a pretty straightforward and effective algorithm that can be used in a variety of classification tasks.

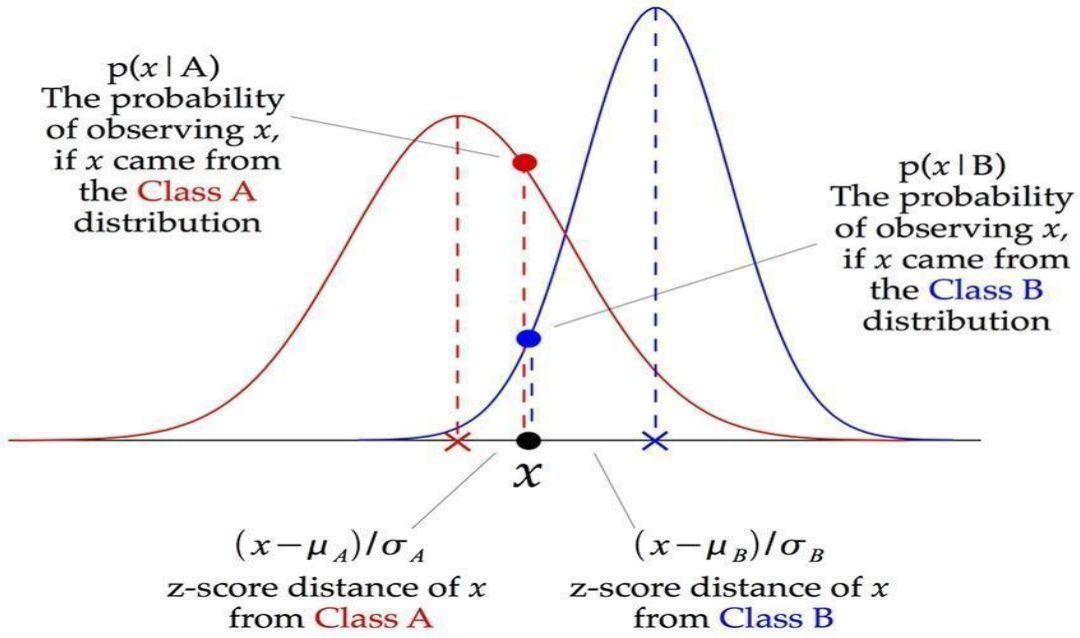

Fig. 4.5. Demonstration of Gaussian Naive Bayes.

VI. Logistic Regression

Logistic regression is a machine learning algorithm that is commonly used to solve binary classification problems. This means that it predicts whether the outcome will be a "yes" or "no" based on the input features. For example, it might predict whether an email is spam or not based on the words in the email.

The algorithm works by using a special mathematical function called the sigmoid function, also known as the logistic function. This function takes in a number as input and outputs a value between 0 and 1, which can be interpreted as the probability of the binary outcome.

To make a prediction, logistic regression calculates a linear combination of the input features and their corresponding weights, which gives a numerical score for each example. This score is then plugged into the sigmoid function, which outputs the probability of the positive class.

To find the best weights that will give the most accurate predictions, the algorithm tries to maximize the likelihood of the training data. This involves calculating the probability of the observed outcomes given the input features, and trying to adjust the weights to make this probability as high as possible. Fig. 4.6. shows demonstration of Logistic Regression.

In short, logistic regression is a simple yet powerful algorithm that can be used for binary classification problems. It works by calculating the probability of the positive class using a special function, and then adjusting the weights to maximize the likelihood of the observed outcomes.

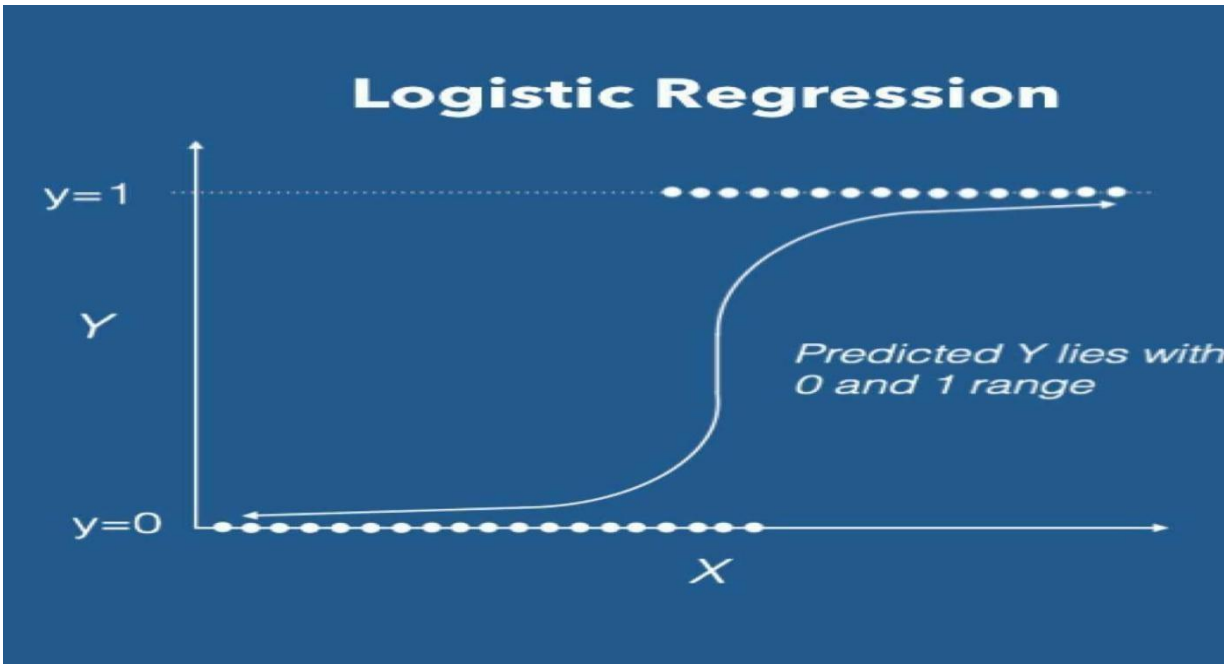

Fig. 4.6. Demonstration of Logistic Regression.

VII. Artificial Neural Network

Artificial Neural Networks (ANNs) are computer algorithms inspired by the structure and function of biological neural networks in the brain. ANNs consist of interconnected nodes, or artificial neurons, that are organized into layers. Each node receives input from other nodes or from external data sources, and performs a calculation using an activation function to compute a weighted sum of the inputs. The output of the node is then passed on to other nodes in the next layer.

In the field of heart disease research and prediction, ANNs have been widely used to predict the presence and severity of coronary artery disease (CAD) using clinical and demographic data, as well as imaging data such as echocardiography and cardiac magnetic resonance imaging. ANNs have also been used to predict the risk of sudden cardiac death using electrocardiography (ECG) data.

One of the advantages of ANNs is their ability to handle non-linear relationships between input variables and outcomes that may not be captured by traditional statistical methods. ANNs can also handle high-dimensional data, such as medical images, and can be trained using a variety of optimization algorithms, such as backpropagation, to minimize the difference between the predicted and true outcomes. Fig 4.7 demonstrates ANN.

However, there are some challenges when using ANNs. They can be computationally expensive to train, especially for large datasets or complex network architectures. ANNs are also prone to overfitting, where the network learns to model the noise in the training data rather than the underlying relationships, leading to poor performance on new data.

Despite these challenges, ANNs have shown promise in improving heart disease prediction and diagnosis. With continued research and development, ANNs have the potential to significantly advance our understanding and treatment of heart disease.

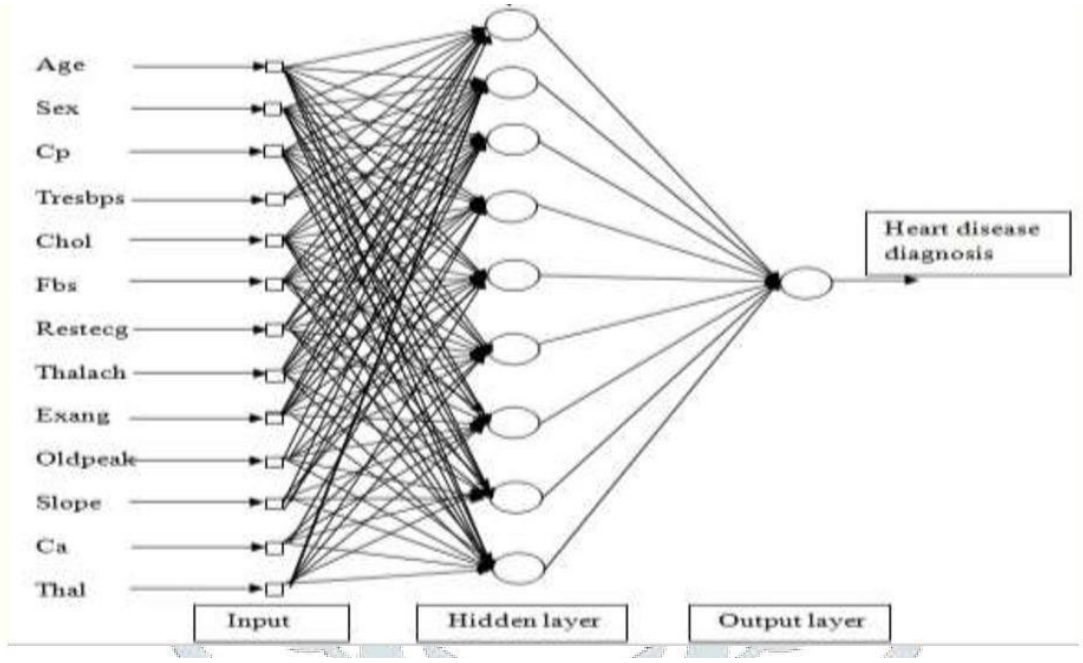

Fig. 4.7. Demonstration of ANN.

4.2 Block Diagram

To illustrate the workflow of our heart disease prediction system, the following block diagram (Fig. 4.8) outlines the step-by-step architecture implemented in the project. This end-to-end pipeline ensures that both technical and practical goals—like accuracy, usability, and accessibility—are achieved.

Workflow Steps:

1. **UCI Dataset Collection**
    - The system uses the **UCI Heart Disease Dataset** as the default dataset for training and testing.
    - Users can also **upload custom datasets** for personalized predictions.
2. **Data Preparation**
    - Patient data is loaded using Python libraries such as **NumPy** and **Pandas**.
    - The system reads both built-in and user-uploaded CSV files.
3. **Data Preprocessing**
    - Key preprocessing steps include:
        - Data cleaning
        - Filtering and duplication removal
        - Handling missing values
        - Encoding categorical features and normalization
4. **Apply Machine Learning Algorithms**
    - Seven supervised learning models are implemented:
        - Logistic Regression
        - Decision Tree
        - Random Forest
        - Support Vector Machine (SVM)
        - K-Nearest Neighbors (KNN)
        - Gaussian Naïve Bayes
        - Artificial Neural Network (ANN)
    - Users can select any model, tune hyperparameters, and train the model.
5. **Performance Analysis and Evaluation**
    - Each model is evaluated using:
        - Accuracy
        - Confusion Matrix
        - Classification Report (Precision, Recall, F1-Score)

- Correlation Matrix (for data exploration and insight)

- This comparison helps identify the best-performing model for a given dataset.

6. **Outcome**
   - A fully functional **web application** with a responsive UI allows:
     - Doctors and patients to interact with models
     - Real-time predictions using input data
     - Model training and evaluation without coding knowledge
   - The system is also deployable on Android for mobile access.

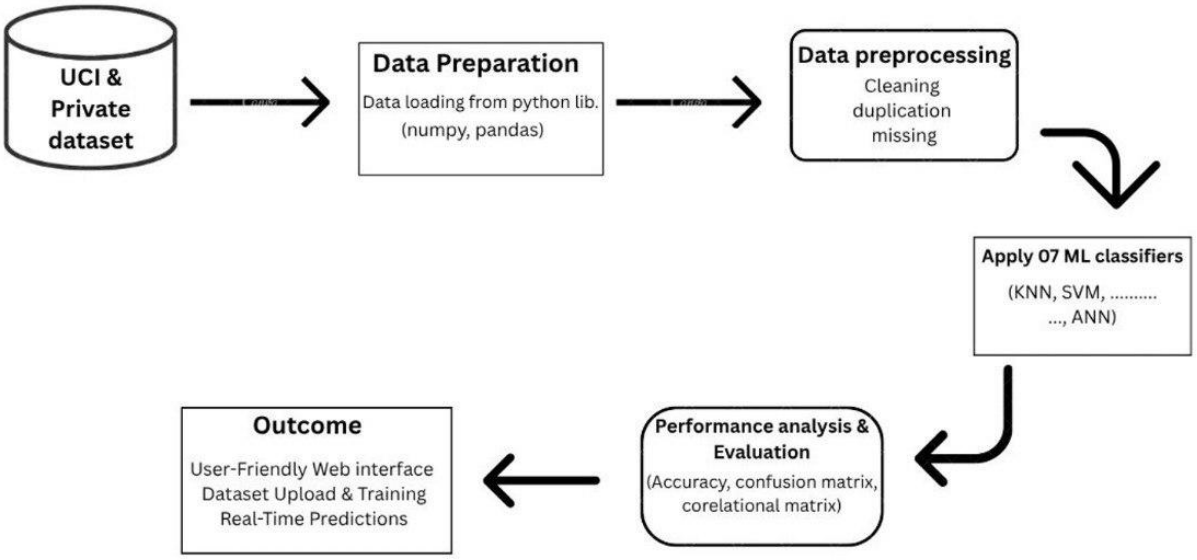

Fig. 4.8. Block Diagram for our model

4.3 Algorithm and Pseudocode:

Start

Step 1: Dataset Collection

IF user uploads a custom CSV file:

Load user-provided dataset

ELSE:

Load default UCI Heart Disease Dataset

Step 2: Data Preparation

Import libraries: NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn, Streamlit

Read the dataset using Pandas

Display basic information (shape, columns, data types)

Step 3: Data Preprocessing

Remove duplicate rows (if any)

Handle missing values (e.g., fill or drop)

Encode categorical variables (e.g., One-hot or Label Encoding)

Normalize or scale numerical features (e.g., MinMaxScaler or StandardScaler)

Split the dataset into features (X) and target (Y)

Split X and Y into training and test sets (e.g., 67% training, 33% testing)

Step 4: Machine Learning Model Selection and Training

Display model options in sidebar (via Web UI)

User selects a machine learning algorithm:

- Logistic Regression
- Decision Tree
- Random Forest
- Support Vector Machine (SVM)
- K-Nearest Neighbors (KNN)
- Gaussian Naïve Bayes
- Artificial Neural Network (ANN)

FOR selected model:

Display model-specific parameter sliders (e.g., K value, max_depth, kernel)

Train the model using the training dataset

Predict outcomes using the test dataset

Step 5: Model Evaluation and Performance Analysis

Calculate Accuracy

Generate Confusion Matrix

Generate Classification Report (Precision, Recall, F1-score)

Compute Correlation Matrix (optional for data insight)

Display all metrics and charts via Web UI

Step 6: Output and Deployment

Display final prediction results to user (Real-time predictions)

Allow users to:

- Enter new patient data and get instant prediction
- Upload new datasets and re-train models
- Download trained models (optional)

Deploy app:

- Web application via Streamlit
- Android deployment using WebView or compatible interface

End

4.4 About Dataset

To conduct our study on cardiovascular disease, we used a publicly available dataset from the Kaggle website. The dataset was originally obtained from the Machine Learning Repository of the University of California Irvine. Our objective was to develop a classification model that can predict whether a patient has a risk of developing heart disease within the next 10 years. The dataset contains anonymized patient data with 400 records and various features. We split the data into two sets, with 80% used for training our machine learning models and 20% used for testing. The dataset includes 13 columns that we used as features, as shown in Table 5.1.

No	*Feature*	*Description*
1	Age	# years
2	Sex	Value 0: female, Value 1: male
3	CP	Chest pain type Value 1/2/3/4: Typical-angina/atypical-angina/non-angina pain/asymptomatic
4	TRestbps	Resting blood sugar in Hg
5	Chol	Serum cholesterol in mg/dl
6	FBS	Fasting Blood Sugar (FBS) Value 1: if FBS > 120 mg/dl Value 0: if FBS <120 mg/dl
7	RestECG	Resting ECG result
8	ThalAch	Maximum heart rate achieved
9	ExAng	Angina following exercise
10	OldPeak	Depression following exercise relative to rest
11	Slope	Peak exercise segment Value1: up sloping Value2: flat Value 3: down sloping
12	Ca	Number of major vessels
13	Thal	Thalium
14	Target	Predicted output Value 0: Healthy patient Value 1: Ill patient

Table. 4.1. Description of the Dataset

4.5 Software and Hardware:

The project was implemented using Jupyter Notebook software running on Python 3.5. Major libraries were Scikit-Learn, NumPy, Pandas, matplotlib, seaborn, and warnings. The code was executed on a Windows 10 Home workstation with an Intel® Core™ i5-8250U CPU @ 1.60 GHz – 1.80 GHz with 8.00 GB of RAM. The specified hardware was sufficient to conduct all the computations and there were no time delays in the implementation. Now we move to Chapter 4 where we show our results of the implementation.

4.6 Histogram of Attributes:

Histograms are a type of graph that display the distribution of values in a dataset. By looking at these histograms, we can get an idea of the range of values for each attribute and how often each value occurs. Figure 4.7 shows the range of the various attributes present in our heart disease dataset through histograms.

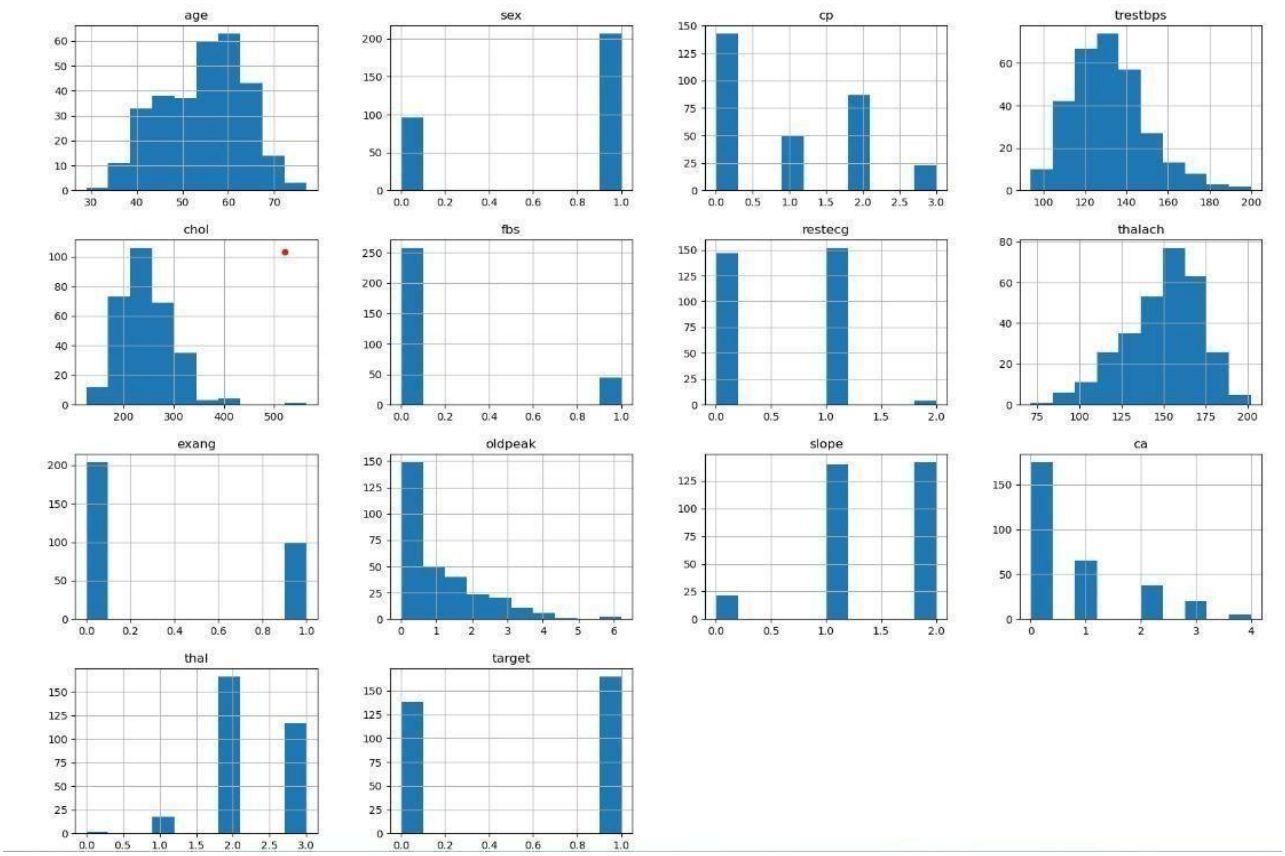

Fig. 4.9. Histogram of Attributes

4.7 Comparison of algorithms:

Sr.no.	Algorithm	Approach	Advantage	Disadvantage
1.	K-nearest neighbors (KNN)	Uses proximity to make classifications or predictions about the grouping of an individual data point.	Simple and intuitive	Requires careful normalization, sensitive to irrelevant features
2.	Support vector machine (SVM)	Creates a hyperplane in an n-dimensional space separating data into classes.	Effective in high-dimensional spaces.	Requires careful selection of kernel function, sensitive to outliers.
3.	Decision Tree Classifier	In an inverted tree diagram, the root is called the root node, and the branches represent the outcome of a decision, which are called the leaf nodes.	Easy to interpret, handles categorical and numerical data	Prone to overfitting, sensitive to small changes in data.
4	Random Forest	Uses a number of decision trees and takes their average to reach one result.	Reduces overfitting, dimensional data.	Computationally expensive, interpret

5	Gaussian Naïve Bayes	Assumes that each parameter (also called features or predictors) has an independent capacity of predicting the output variable.	Fast and effective	Assumes independence, sensitive to rare events
6	Logistic Regression	Estimates the probability of a categorical variable and maps it between 0 & 1.	Simple to implement, interpretable	Assumes linearity, requires large data.
7	Artificial Neural Network	Algorithms based on brain function and are used to model complicated patterns and forecast issues.	A neural network can implement tasks that a linear program cannot	The neural network required training to operate.

Table 4.2. Comparison of algorithms.

Chapter 5

5. Results

In this chapter, we see the results of the different classifiers. We analyze their performance in various ways which we shall explain (through different performance and evaluation metrics, accuracy, etc.

5.1 Test accuracy:

After the models were built and tested a performance analysis was performed by comparing their accuracy with each other. Table 5.1. shows accuracy of each model.

Model	**Accuracy**
K-Nearest Neighbours	0.87
Support Vector Machine	0.83
Decision Tree	0.79
Random Forest	0.84
Gaussian Naïve Bayes	0.86
Logistic Regression	0.88
Artificial Neural Network	0.56

Table 5.1. Accuracy of each model.

5.2 Correlation Matrix:

A correlation matrix is a statistical tool that helps to understand the relationship between different variables. It's a square matrix that shows how each variable is related to the others in the set.

The diagonal elements of the matrix always have a value of 1 because a variable is always perfectly correlated with itself. The off-diagonal elements show the correlation coefficient between two variables. This coefficient can range from -1 to 1, where -1 means that the variables are perfectly negatively correlated (as one variable goes up, the other goes down), 1 means that the variables are perfectly positively correlated (as one goes up, the other goes up too), and 0 means there is no correlation between the variables.

A correlation matrix can be used to identify strong and weak correlations between variables, which can help in developing predictive models, finding areas for improvement in business processes, and identifying which variables are the most important in explaining a particular phenomenon. It can also be used to detect multicollinearity, which is when two or more independent variables in a multiple regression model are highly correlated with each other. By identifying multicollinearity, we can better understand the relationship between variables and avoid inaccurate results in our models. In Fig Correlation Matrix of the given dataset is Demonstration. Fig 5.1 shows correlation matrix of the dataset.

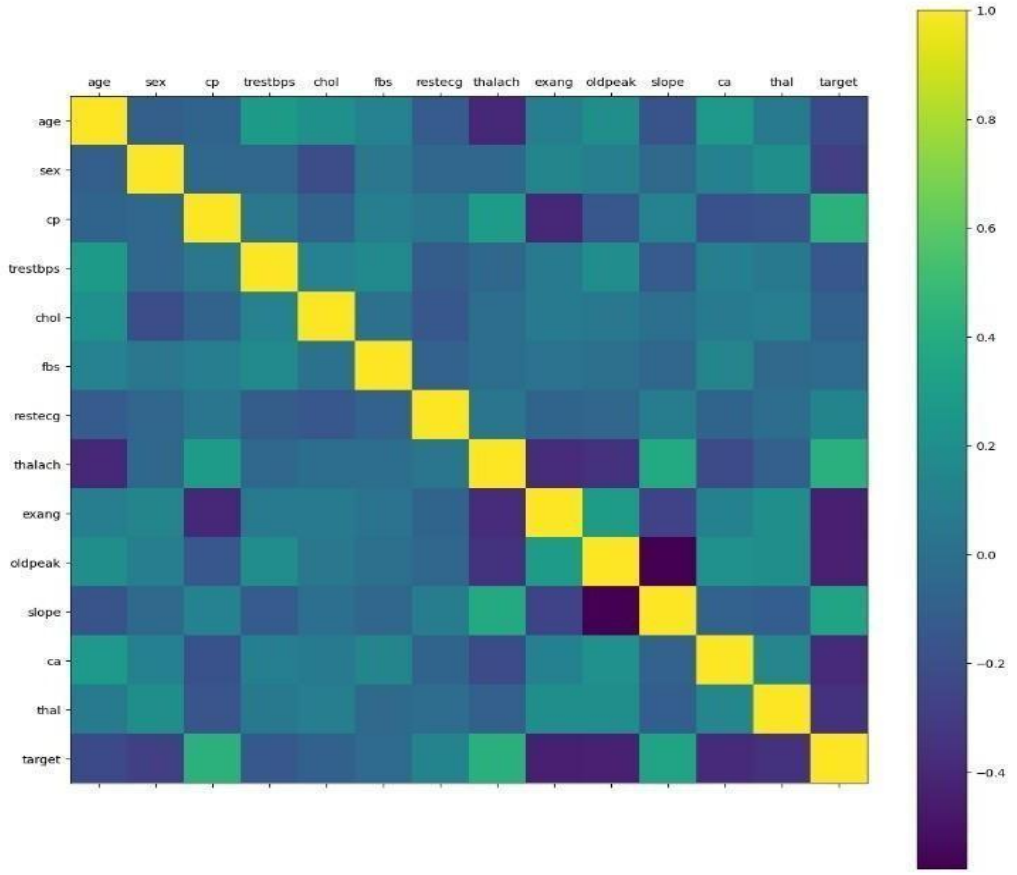

Fig. 5.1. Correlation Matrix of the given dataset

5.3. Confusion Matrix:

A confusion matrix is a tool that helps us evaluate how well a classification model is performing. It summarizes the number of correct and incorrect predictions made by the model on a set of test data.

The confusion matrix is usually presented as a table with four cells, organized into two rows and two columns. The rows represent the actual class labels of the data, while the columns represent the predicted class labels of the model.

There are four categories in the confusion matrix: true positives, false positives, true negatives, and false negatives. True positives are the data points that were correctly classified as positive by the model, while false positives are those that were incorrectly classified as positive. True negatives are the data points that were correctly classified as negative, and false negatives are those that were incorrectly classified as negative.

By analyzing the confusion matrix, we can calculate various performance metrics of the model, such as accuracy, precision, recall, and F1 score. These metrics help us assess how well the model is performing and identify areas for improvement. Fig 5.2 shows the demonstration of Confusion Matrix.

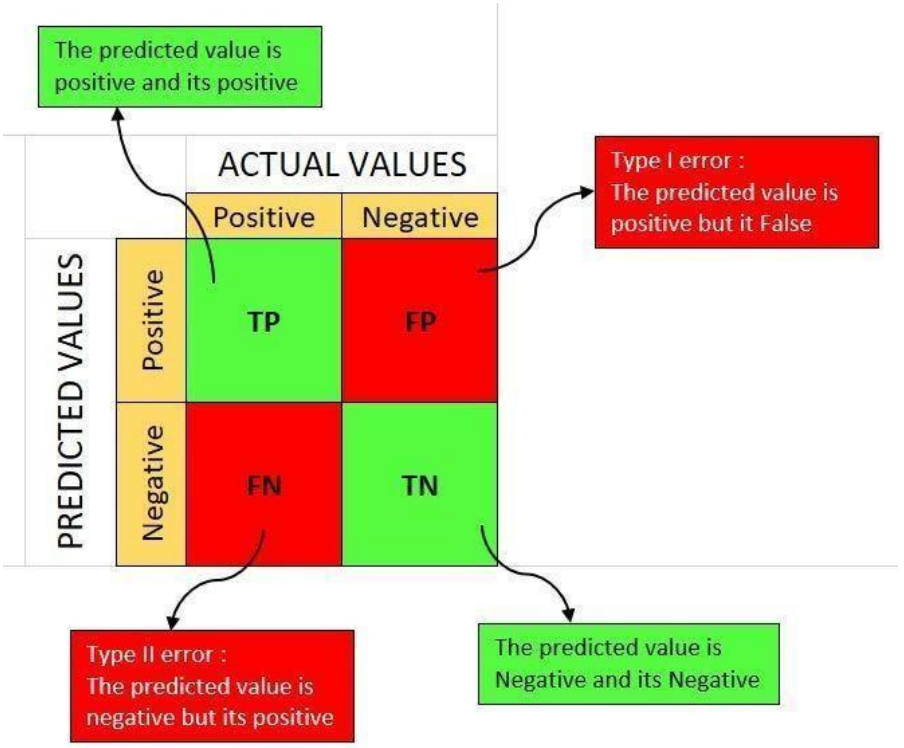

Fig 5.2. Demonstration of Confusion Matrix

From Fig 5.3. to 5.16. shows confusion matrix and output of the algorithms used in this project:

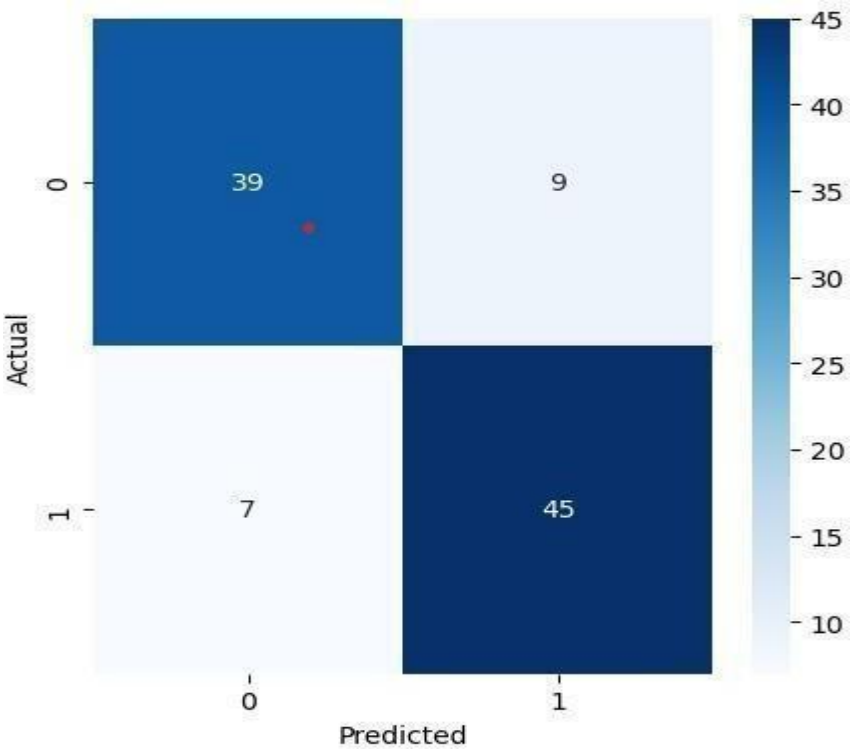

Fig. 5.3. Confusion Matrix of KNN.

↑	precision	recall	f1-score	support
0	74.29%	78.79%	76.47%	33
1	73.08%	67.86%	70.37%	28
accuracy	73.77%	73.77%	73.77%	0.7377
macro avg	73.68%	73.32%	73.42%	61
weighted avg	73.73%	73.77%	73.67%	61

Table. 5.2. Classification Report of KNN.

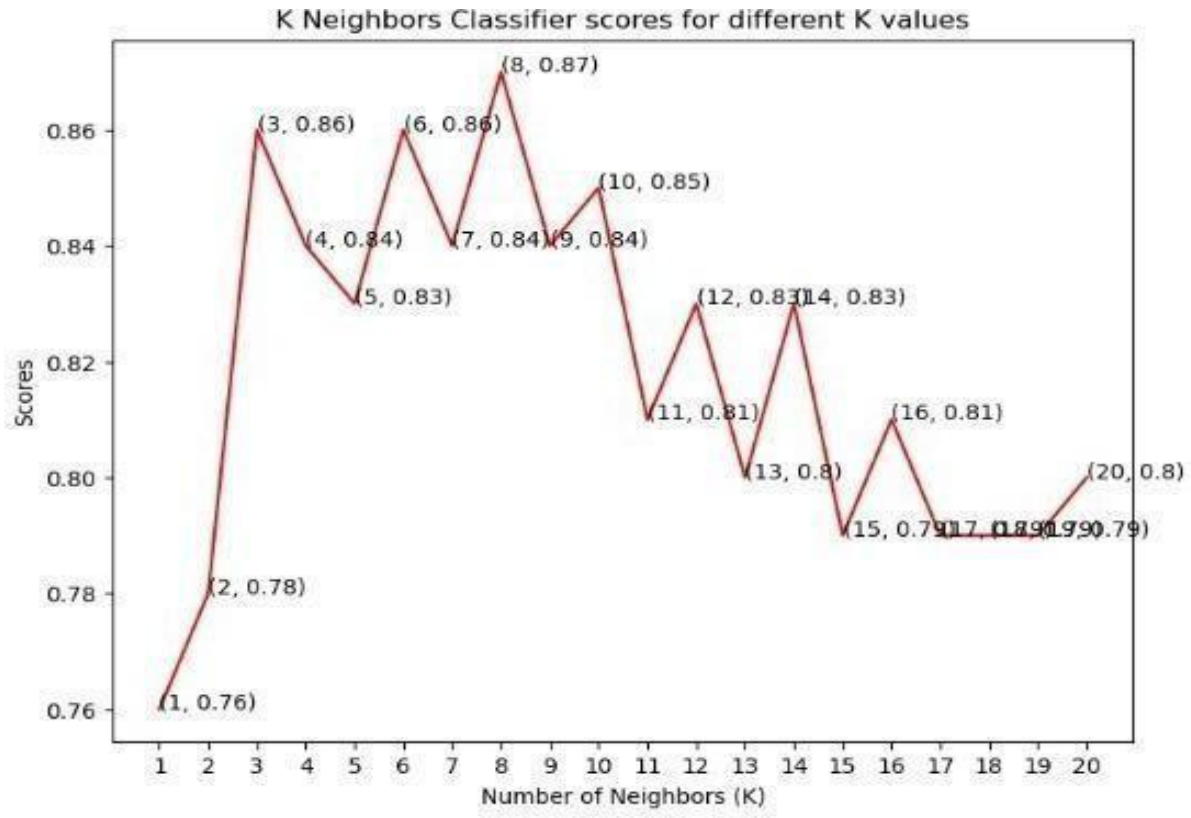

Fig5.4. Output of KNN

The figure displays the accuracy scores of the K-Nearest Neighbors (KNN) classifier for different values of K, ranging from 1 to 20. The accuracy improves significantly from 0.76 at K=1 to a peak of 0.87 at K=8, indicating optimal model performance at this point. Beyond K=8, the accuracy gradually declines, with the lowest scores observed around K=15 to K=19 (around 0.79). This suggests that K=8 offers the best balance between bias and variance for heart disease prediction in this model.

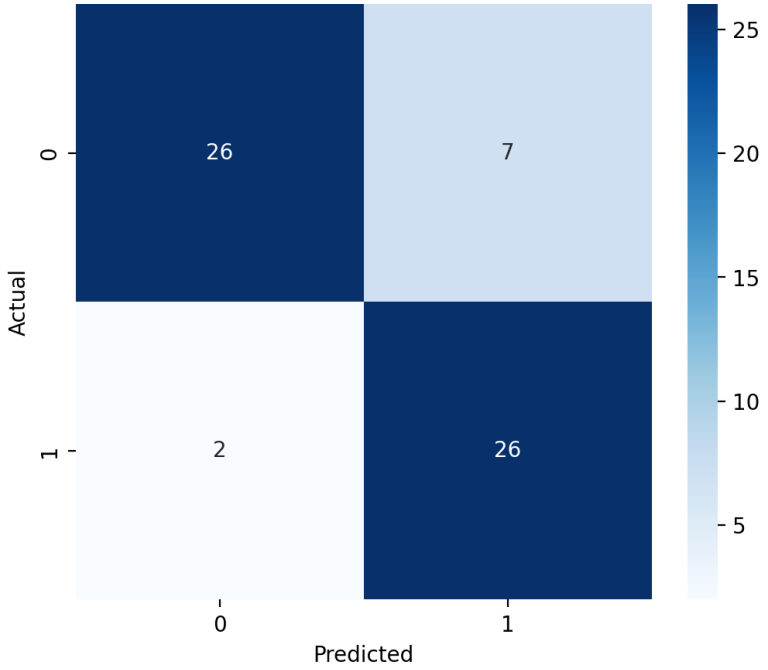

Fig. 5.5. Confusion Matrix of SVM

↑	precision	recall	f1-score	support
0	92.86%	78.79%	85.25%	33
1	78.79%	92.86%	85.25%	28
accuracy	85.25%	85.25%	85.25%	0.8525
macro avg	85.82%	85.82%	85.25%	61
weighted avg	86.40%	85.25%	85.25%	61

Table. 5.3. Classification Report of SVM

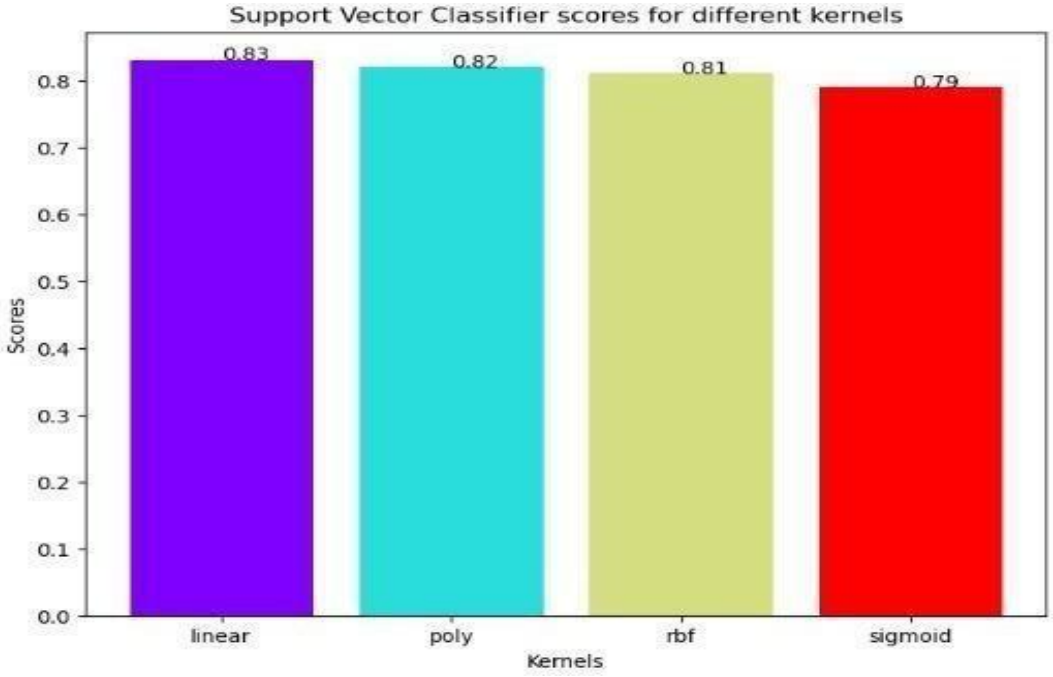

Fig5.6. Output of SVM

The bar chart illustrates the accuracy scores of the Support Vector Classifier using four different kernel functions: linear, polynomial (poly), radial basis function (rbf), and sigmoid. Among these, the linear kernel achieved the highest accuracy of 0.83, followed closely by poly (0.82) and rbf (0.81). The sigmoid kernel performed the worst with an accuracy of 0.79. This indicates that the linear kernel is most suitable for this heart disease dataset, providing the best classification performance among the tested kernels.

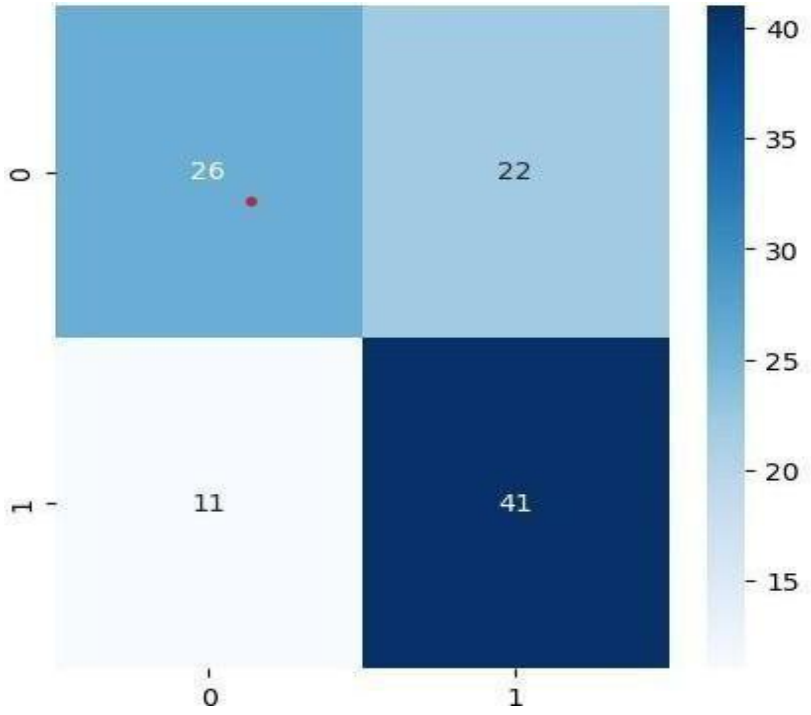

Fig. 5.7. Confusion Matrix of Decision Tree

↑	precision	recall	f1-score	support
0	75.00%	63.64%	68.85%	33
1	63.64%	75.00%	68.85%	28
accuracy	68.85%	68.85%	68.85%	0.6885
macro avg	69.32%	69.32%	68.85%	61
weighted avg	69.78%	68.85%	68.85%	61

Table. 5.4. Classification Report of Decision Tree

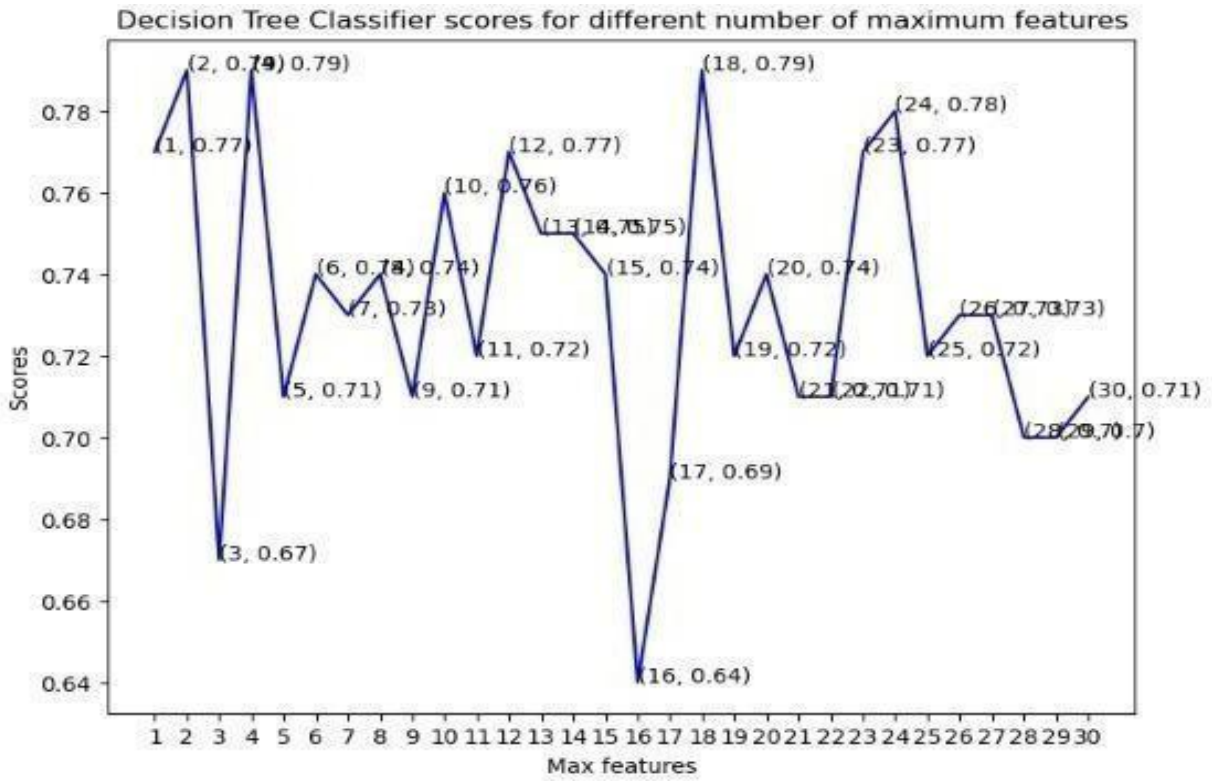

Fig5.8 Output of Decision Tree

The graph shows the accuracy of a Decision Tree Classifier for different values of maximum features (1–30). The best accuracy of 0.79 is achieved at max features = 2 and 18, while the lowest score of 0.64 occurs at 16. The results fluctuate, highlighting that performance is sensitive to the choice of max_features, and optimal values like 2 or 18 can significantly improve prediction accuracy.

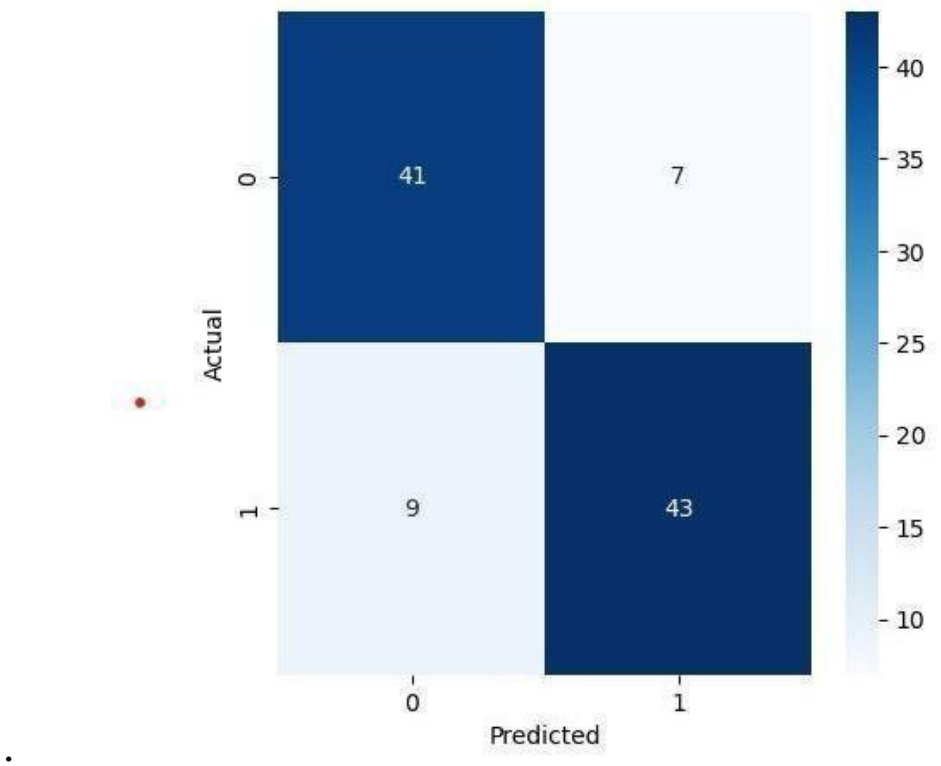

Fig. 5.9. Confusion Matrix of Random Forest

↑	precision	recall	f1-score	support
0	93.33%	84.85%	88.89%	33
1	83.87%	92.86%	88.14%	28
accuracy	88.52%	88.52%	88.52%	0.8852
macro avg	88.60%	88.85%	88.51%	61
weighted avg	88.99%	88.52%	88.54%	61

Table. 5.5. Classification Report of Random Forest

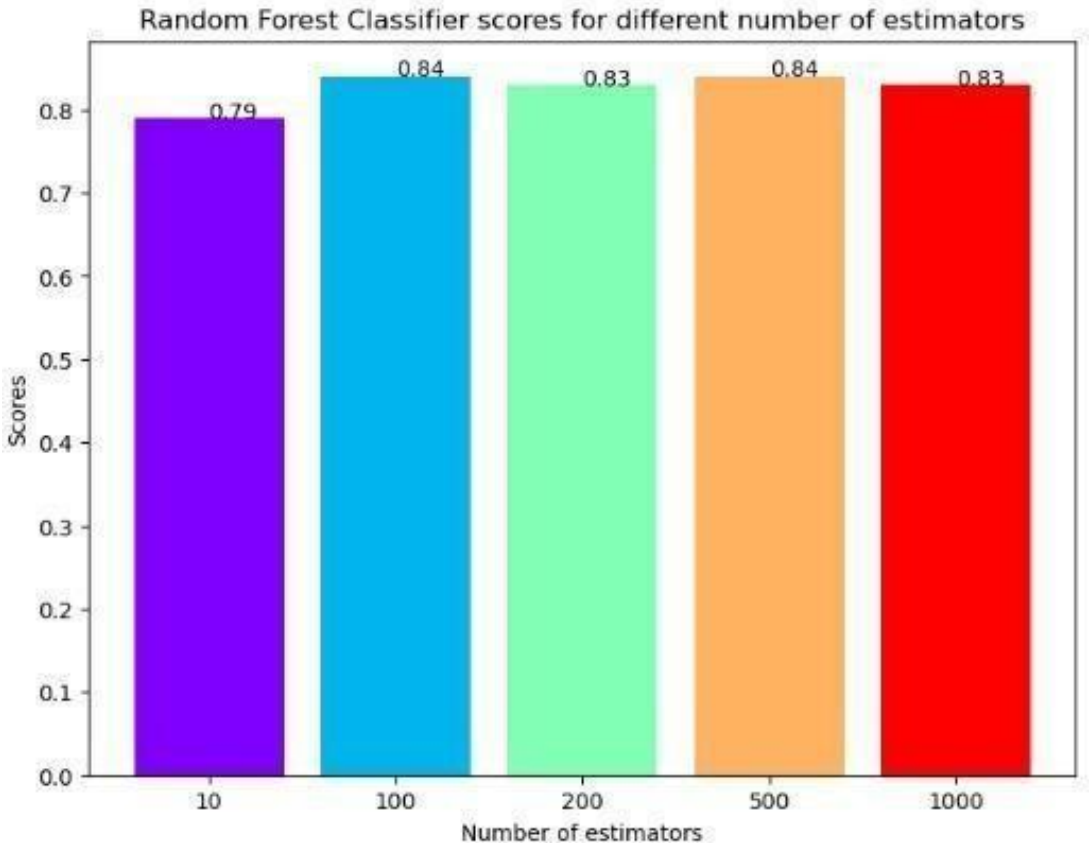

Fig5.10. Output of Random Forest

The bar chart shows the accuracy of the Random Forest Classifier with different numbers of estimators. Accuracy improves from 0.79 with 10 estimators to 0.84 with 100 and 500 estimators, and slightly drops to 0.83 for 200 and 1000 estimators. This indicates that using 100 or 500 estimators gives the best performance, while increasing beyond that offers no significant improvement.

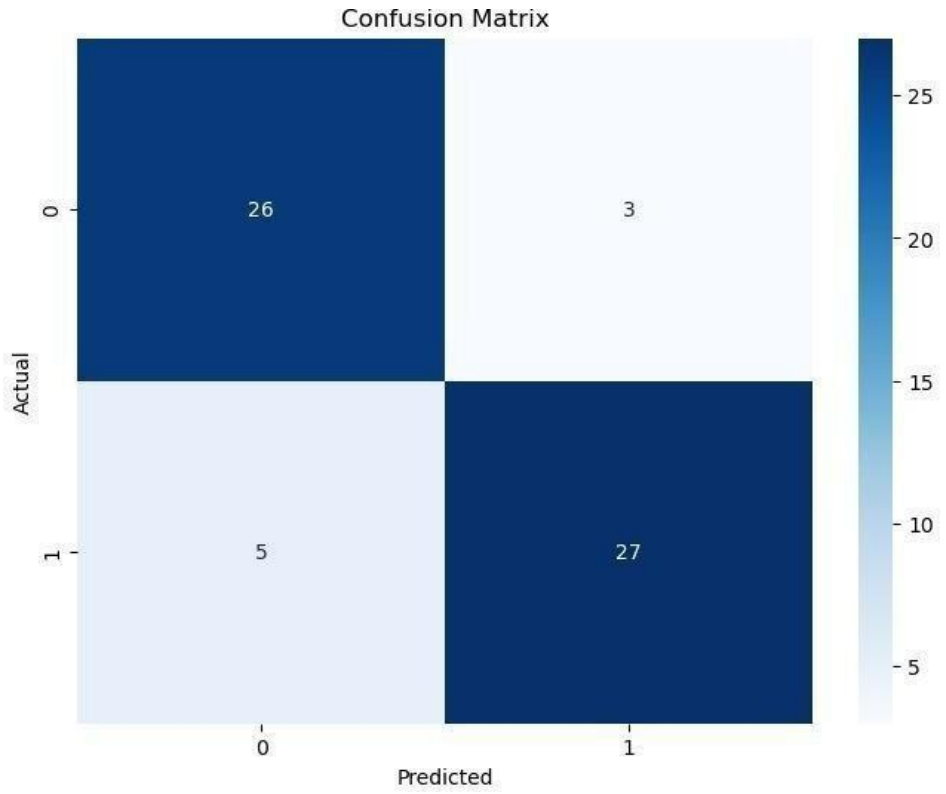

Fig. 5.11. Confusion Matrix of Gaussian Naïve Bayes

↑	precision	recall	f1-score	support
0	87.50%	63.64%	73.68%	33
1	67.57%	89.29%	76.92%	28
accuracy	75.41%	75.41%	75.41%	0.7541
macro avg	77.53%	76.46%	75.30%	61
weighted avg	78.35%	75.41%	75.17%	61

Table. 5.6. Classification Report of Gaussian Naïve Bayes

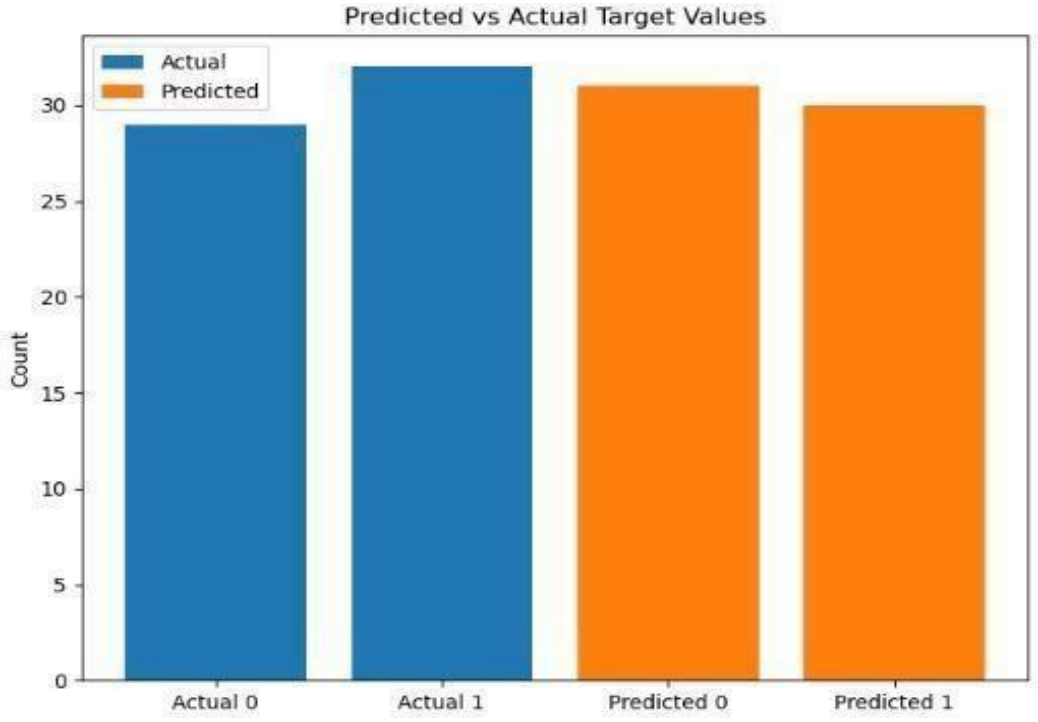

Fig5.12. Output of Gaussian Naive Bayes

The bar chart compares the actual vs. predicted target values for heart disease classification. The counts of actual class 0 and class 1 are slightly higher than the predicted ones, showing that the model has good alignment with real data but a few misclassifications. Overall, the model performs reliably, with nearly balanced prediction for both classes.

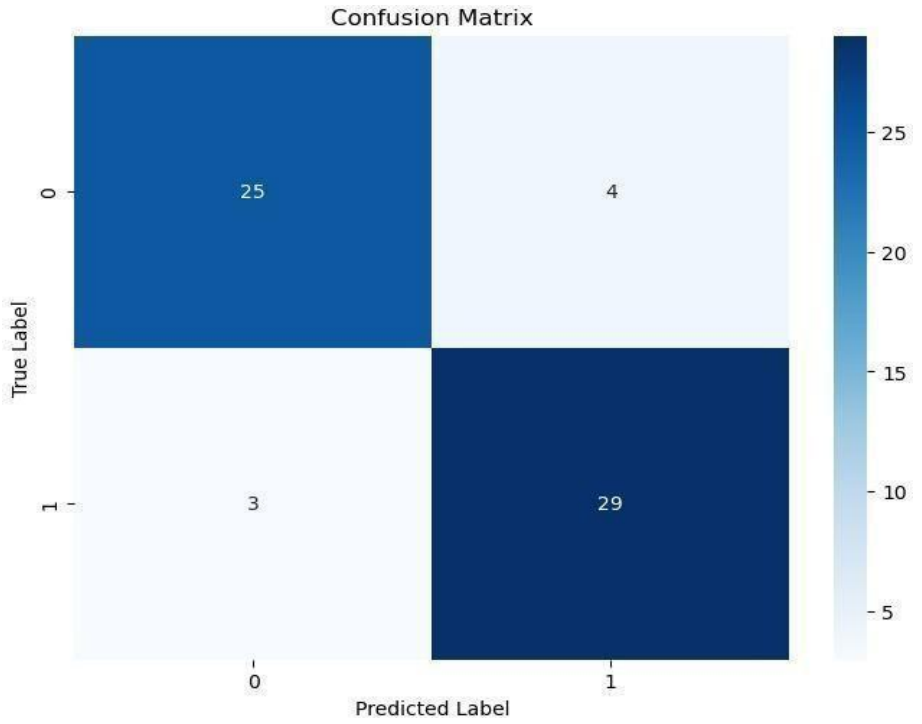

Fig. 5.13. Confusion Matrix of Logistic Regression

↑	precision	recall	f1-score	support
0	93.33%	84.85%	88.89%	33
1	83.87%	92.86%	88.14%	28
accuracy	88.52%	88.52%	88.52%	0.8852
macro avg	88.60%	88.85%	88.51%	61
weighted avg	88.99%	88.52%	88.54%	61

Table. 5.7. Classification Report of Logistic Regression

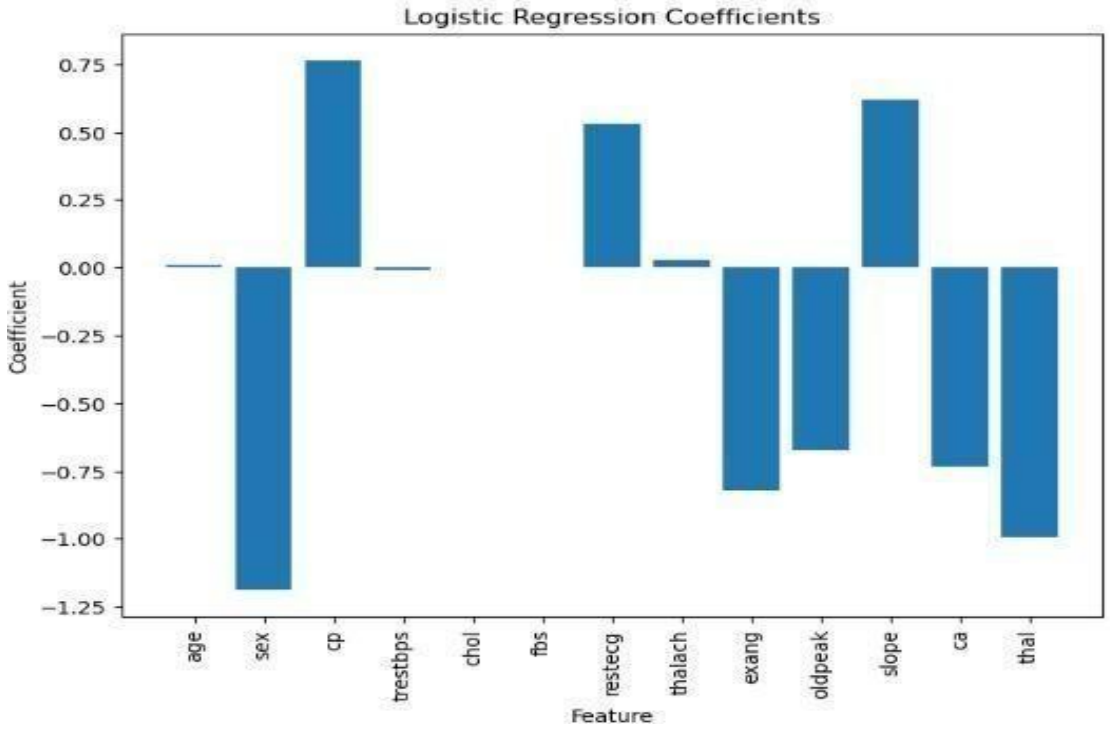

Fig5.14. Output of logistic regression

This bar chart displays the coefficients of features in the logistic regression model used for heart disease prediction. Positive coefficients (e.g., cp, restecg, slope) indicate a higher likelihood of heart disease when those feature values increase, while negative coefficients (e.g., sex, exang, thal) suggest a lower likelihood. The magnitude of the coefficient shows the feature's influence—thal, sex, and cp have the most significant impact on the model's prediction.

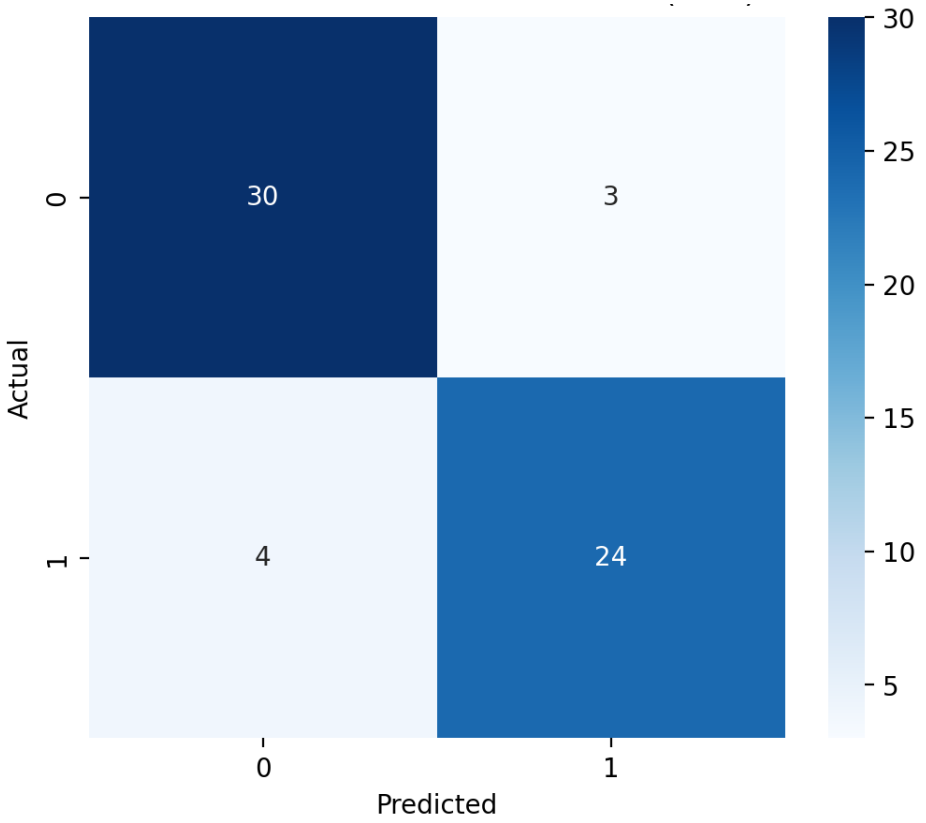

Fig. 5.15. Confusion Matrix of ANN

	precision	recall	f1-score	support
0	88.24%	90.91%	89.55%	33
1	88.89%	85.71%	87.27%	28
accuracy	88.52%	88.52%	88.52%	0.8852
macro avg	88.56%	88.31%	88.41%	61
weighted avg	88.54%	88.52%	88.51%	61

Table. 5.8. Classification Report of ANN

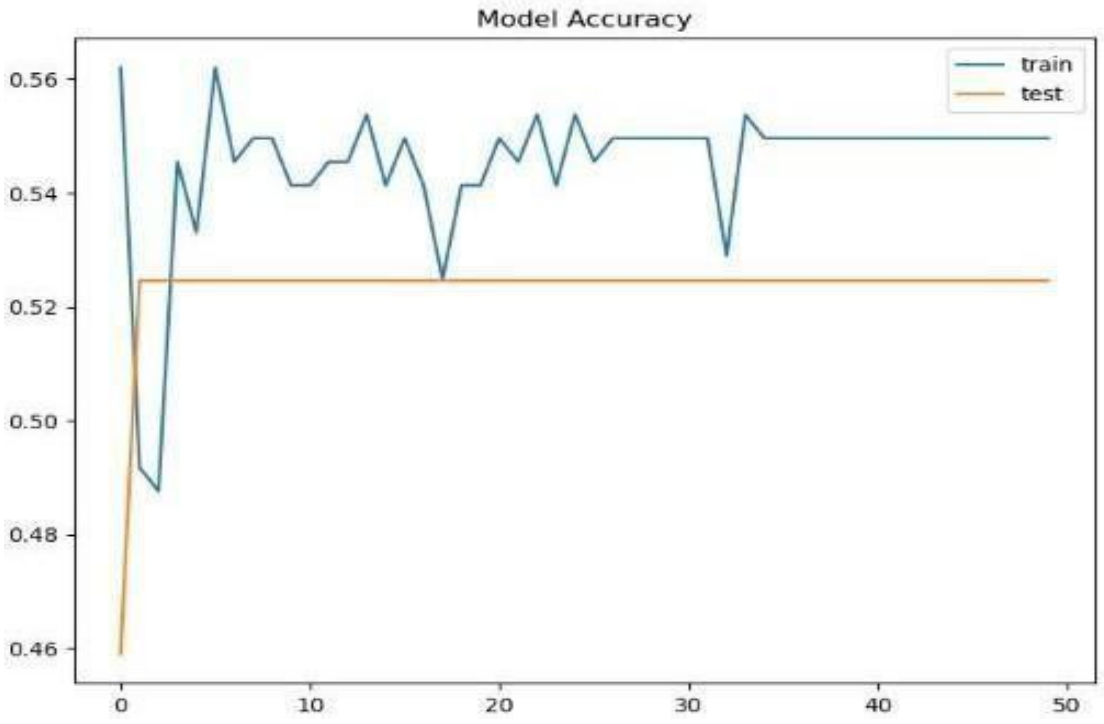

Fig 5.16. Output of ANN

This line graph shows model accuracy over training epochs for both training and testing datasets. The training accuracy fluctuates but generally stabilizes around 0.55, while the test accuracy quickly reaches and remains steady at about 0.53. The minimal gap suggests limited overfitting, but the overall low accuracy indicates underperformance, possibly due to suboptimal model architecture or features.

5.4. Result Analysis:

After building and training the machine learning models, a comprehensive performance analysis was conducted by comparing the accuracy and evaluation metrics of each algorithm. The models implemented in this system include:

- K-Nearest Neighbors (KNN)
- Support Vector Machine (SVM)
- Decision Tree
- Random Forest
- Gaussian Naive Bayes
- Logistic Regression
- Artificial Neural Network (ANN)

Each model was trained on the UCI Heart Disease dataset, and the following accuracy scores were obtained:

- K-Nearest Neighbors (k=8): 0.87
- Support Vector Machine (Linear kernel): 0.83
- Decision Tree: 0.79
- Random Forest (n_estimators=100): 0.84
- Gaussian Naive Bayes: 0.86
- Logistic Regression: 0.88
- Artificial Neural Network (ANN): 0.56

Among all models, **Logistic Regression** achieved the highest accuracy of **0.88**, making it the most effective model for this dataset. This outcome is expected since the UCI dataset is relatively simple and exhibits linearly separable patterns, which aligns well with the strengths of Logistic Regression. In contrast, the **Artificial Neural Network** (ANN) model performed the poorest with an accuracy of **0.56**, likely due to the model being under-tuned or the dataset not requiring deep non-linear representation learning.

Integration with Real-World Utility

Unlike many previous research studies that remain confined to offline evaluation or academic experimentation, our system addresses a crucial gap by making these machine learning models **accessible and usable by the public**:

- A **web-based interface** enables doctors and non-technical users to interactively choose,train, and evaluate models.
- An **Android-compatible version** makes the system accessible on mobile devices, expanding its utility in clinical and remote environments.
- Users can **upload their own patient datasets** and train models accordingly, allowing for personalized predictions and exploration.
- The platform visualizes model performance through accuracy scores, confusion matrices, and classification reports—making interpretation easier for healthcare professionals.

By offering a **deployable, real-time prediction system** with multiple models, our solution bridges the gap between academic ML model development and practical, scalable healthcare applications.

Chapter 6

6. Conclusion and Future Scope

6.1 Conclusion

This project presents a practical heart disease prediction system using machine learning, addressing the gap between research models and real-world accessibility. By analyzing key clinical features, the system enables early risk assessment through seven different ML algorithms: Logistic Regression, Decision Tree, Random Forest, SVM, KNN, Naive Bayes, and ANN.

Unlike prior studies limited to offline evaluations, our solution is deployed via a **web and Android-based platform**, allowing doctors and patients to:

- Upload custom datasets
- Select and train models
- Perform real-time risk prediction
- View detailed performance analysis

Logistic Regression achieved the highest accuracy of 88%, while ANN showed limited performance due to dataset simplicity. The system's interactive, customizable design supports wider public use and better preventive care.

In summary, this accessible, deployable tool holds strong potential to aid clinical decision-making and improve early detection of heart disease. Further clinical validation is recommended to support integration in healthcare settings.

6.1 Future Scope

The future scope for heart disease prediction systems like ours is highly promising, with several opportunities for enhancement and real-world integration. A key direction is to incorporate more diverse and personalized variables—such as genetic markers, lifestyle habits, and environmental factors—to improve prediction accuracy and individual risk assessment.

Expanding the system to integrate with **electronic health records (EHRs)** would allow clinicians to seamlessly use predictive insights during consultations. Additionally, enhancing the **Android application** to include health tracking features could empower users to monitor key health parameters and receive real-time risk updates.

The use of **big data analytics** and **advanced AI models**, including deep learning on longitudinal patient records or streaming data from **wearable devices**, can further improve prediction accuracy. Future iterations can also explore **explainable AI (XAI)** to increase model transparency for medical professionals.

Validating the system across **diverse populations**—varying in age, region, and ethnicity—will be essential to ensure fairness, generalizability, and clinical reliability. Ultimately, integrating such models into public health and clinical frameworks could significantly improve early diagnosis, prevention, and outcomes in cardiovascular care.

References

[1] Bashir, S., Qamar, U., & Javed, M. Y. (2014, November). *An ensemble-based decision support framework for intelligent heart disease diagnosis*. In Proceedings of the International Conference on Information Society (i-Society 2014) (pp. 259–264). IEEE.

[2] Shinde, R., Arjun, S., Patil, P., & Waghmare, J. (2015). An intelligent heart disease prediction system using k-means clustering and Naïve Bayes algorithm. *International Journal of Computer Science and Information Technologies*, *6*(1), 637-639.

[3] A. Dewan and M. Sharma, (2015) "Prediction of heart disease using a hybrid technique in data mining classification," *2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, 2015, pp. 704-706.

[4] Dangare, C. S., & Apte, S. S. (2012). Improved study of heart disease prediction system using data mining classification techniques. *International Journal of Computer Applications*, *47*(10), 44-48.

[5] Raihan, M., Mondal, S., More, A., Sagor, M. O. F., Sikder, G., Majumder, M. A., ... & Ghosh, K. (2016, December). 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)* (pp. 299-303). IEEE.

[6] Ordonez, C. (2006). Association rule discovery with the train and test approach for heart disease prediction. *IEEE transactions on information technology in biomedicine*, *10*(2), 334-343.

[7] Masethe, H. D., & Masethe, M. A. (2014, October). Prediction of heart disease using classification algorithms. In *Proceedings of the world Congress on Engineering and computer Science* (Vol. 2, No. 1, pp. 25-29).

[8] Choi, E., Schuetz, A., Stewart, W. F., & Sun, J. (2017). Using recurrent neural network models for early detection of heart failure onset. *Journal of the American Medical Informatics Association*, *24*(2), 361-370.

[9] Alotaibi, F. S. (2019). Implementation of machine learning model to predict heart failure disease. *International Journal of Advanced Computer Science and Applications*, *10*(6).

[10] M Sangeetha; S.Arun Kumar; K. Pazhani Bharathi; P .Kumara Guru; P.Bhuvan Prakash Reddy. “Heart Disease Prediction Using ML.” Volume. 9 Issue.3, March - 2024 International Journal of Innovative Science and Research Technology (IJISRT), www.ijisrt.com. ISSN - 2456-2165, PP :-2630-2633.

[11] Kumar, R., Garg, S., Kaur, R., Johar, M. G. M., Singh, S., Menon, S. V., ... & Lozanović, J. (2025). A comprehensive review of machine learning for heart disease prediction: challenges, trends, ethical considerations, and future directions. *Frontiers in Artificial Intelligence*, *8*, 1583459. https://doi.org/10.3389/frai.2025.1583459

[12] Khan, H., Bilal, A., Aslam, M. A., & Mustafa, H. (2024). Heart Disease Detection: A Comprehensive Analysis of Machine Learning, Ensemble Learning, and Deep Learning Algorithms. *Nano Biomedicine & Engineering*, *16*(4). https://doi.org/10.26599/NBE.2024.9290087

[13] Bhatt, C. M., Patel, P., Ghetia, T., & Mazzeo, P. L. (2023). Effective heart disease prediction using machine learning techniques. Algorithms, 16(2), 88.

[14] Jindal, H., Agrawal, S., Khera, R., Jain, R., & Nagrath, P. (2021). Heart disease prediction using machine learning algorithms. In IOP conference series: materials science and engineering (Vol. 1022, No. 1, p. 012072). IOP Publishing.

[15] Li, J. P., Haq, A. U., Din, S. U., Khan, J., Khan, A., & Saboor, A. (2020). Heart disease identification method using machine learning classification in e-healthcare. IEEE access, 8, 107562-107582.