ARTIFICIAL INTELLIGENCE IN HEALTHCARE OF HEART DISEASE

A PROJECT REPORT

Submitted by

Roll Number	Registration Number	Student Code	Student Name
23010201094	23012005209	BWU/MCA/23/101	Rupak Pal
23010201096	23012005212	BWU/MCA/23/104	Suman Roy
23010201112	23012005228	BWU/MCA/23/121	Sathi Raut
23010201093	23012005208	BWU/MCA/23/100	Partha Dey

in partial fulfillment for the award of the degree

of

MASTERS OF COMPUTER APPLICATION

Department of Computational Sciences

BRAINWARE UNIVERSITY
398, Ramkrishnapur Road, Barasat, North 24 Parganas, Kolkata - 700125

May , 2025

ARTIFICIAL INTELLIGENCE IN HEALTHCARE OF HEART DISEASE

Submitted by

Roll Number	Registration Number	Student Code	Student Name
23010201094	23012005209	BWU/MCA/23/101	Rupak Pal
23010201096	23012005212	BWU/MCA/23/104	Suman Roy
23010201112	23012005228	BWU/MCA/23/121	Sathi Raut
23010201093	23012005208	BWU/MCA/23/100	Partha Dey

in partial fulfillment for the award of the degree

of

MASTERS OF COMPUTER APPLICATIONS

in

Department of Computational Sciences

BRAINWARE UNIVERSITY
398, Ramkrishnapur Road, Barasat, North 24 Parganas, Kolkata - 700 125

BRAINWARE UNIVERSITY

398, Ramkrishnapur Road, Barasat, North 24 Parganas, Kolkata - 700 125

[DEPARTMENT OF COMPUTATIONAL SCIENCES]

BONAFIDE CERTIFICATE

Certified that this project report "**AI IN HEALTHCARE OF HEART DISEASE "** is the bonafide work of "**Rupak Pal, Suman Roy, Sathi Raut, Partha Dey** who carried out the project work under my supervision.

SIGNATURE
Dr. JAYANTA AICH
HEAD OF THE DEPARTMENT
Department Of Computational Sciences
BRAINWARE UNIVERSITY
398, Ramkrishnapur Road, Barasat,
North 24 Parganas, Kolkata - 700125

SIGNATURE
Ms. DEBASHRI DEBNATH
SUPERVISOR
ASSISTANT PROFESSOR
Department Of Computational Sciences

Acknowledgement

Project Title: ARTIFICIAL INTELLIGENCE IN HEALTHCARE OF HEART DISEASE

Project Group ID: MCA23B003

We, the undersigned project members, would like to express our sincere gratitude to Ms.DEBASHRI DEBNATH, Project Mentor, Department of Computational Sciences, Brainware University, for their invaluable guidance, support, and encouragement throughout the course of this project. Their expert advice and constructive feedback were crucial in the successful completion of this work.

We also extend our heartfelt thanks to Dr. Jayanta Aich, Head of the Department of Computational Sciences and Dr.S. Senthil Kumar , the course professor for providing the necessary resources and support throughout this project.

We acknowledge with gratitude the support and cooperation of all faculty members and staff of the Department of Computational Sciences. Their insights and suggestions helped us stay on course and improve the quality of our work.

We especially thank our families, friends, and peers for their encouragement and support throughout this journey and for motivating us to achieve our goals.

Finally, we express our deepest gratitude to Brainware University for providing us with the opportunity to undertake this project as part of our academic curriculum.

Project Members:

Student Code	Student Name	Signature
BWU/MCA/23/101	RUPAK PAL	
BWU/MCA/23/104	SUMAN ROY	
BWU/MCA/23/121	SATHI RAUT	
BWU/MCA/23/100	PARTHA DEY	

Date: 23/05/2025
Department of Computational Sciences
Brainware University

ABSTRACT

The study's main goal is to use machine learning (ML) techniques to identify chronic heart failure. To replicate the procedure utilized in clinical practice, the study built models based on various combinations of feature categories, including clinical traits, echocardiography, and laboratory results. The phases of the suggested ML technique include feature selection and classification. Early sickness prediction is vital because of the increase in the prevalence of heart disease. The project's major goal is to identify people more likely to develop heart disease based on a range of medical markers. Various methods, including KNN and logistic regression, were used to predict and identify persons with heart disease. This method for predicting cardiac illness improves patient care, simplifies disease diagnosis, and enables the parallel processing of enormous amounts of data. According to the findings, the Random Forest algorithm has the greatest accuracy of 90.16%.

This report represents the mini-project assigned to Third-semester students for the partial fulfillment of MCA481, given by the Department of computer application, BRAINWARE UNIVERSITY. Cardiovascular diseases are the most common cause of death worldwide over the last few decades in developed as well as underdeveloped and developing countries. Early detection of cardiac diseases and continuous supervision of clinicians can reduce the mortality rate. However, it is not possible to monitor patients every day in all cases accurately and consultation of a patient for 24 hours by a doctor is not available since it requires more sapience, time, and expertise. In this project, we have developed and researched heart disease prediction through the various heart attributes of the patient and detected impending heart disease using Machine learning techniques like backward elimination algorithm, **random forest** , logistic regression, and REFCV on the dataset available publicly on Kaggle Website, further evaluating the results using confusion matrix and cross-validation. The early prognosis of cardiovascular diseases can aid in making decisions on lifestyle changes in high-risk patients and in turn reduce complications, which can be a great milestone in the field of medicine.

TABLE OF CONTENTS

LIST OF FIGURES

List of Symbols, Abbreviations and Nomenclature

Here's a list of symbols, abbreviations, and nomenclature from the given code:

Symbols and Variables:

1.Importing Libraries

- import numpy as np
- import pandas as pd
- import matplotlib.pyplot as plt
- import seaborn as sns
- %matplotlib inline
- import os
- import warnings
    - Suppress warnings: warnings.filterwarnings('ignore')

2. Loading and Exploring the Dataset

- Load Dataset: dataset = pd.read_csv("heart.csv")
- Dataset Summary:
    - dataset.describe()
    - dataset.info()
- Feature Description:
  A list info is created with descriptions for dataset columns, iterated to print.

3. Target Variable Analysis

- Unique Values: dataset["target"].unique()
- Correlation Check: dataset.corr()["target"].abs().sort_values(ascending=False)
- Target Distribution:
    - Countplot: sns.countplot(y)
    - Value counts: target_temp = dataset.target.value_counts()
    - Percentages:

- Percentage without heart problems
- Percentage with heart problems

4. Train-Test Split

- Split predictors (X) and target (y):
    - X_train, X_test, Y_train, Y_test = train_test_split(predictors, target, test_size=0.20, random_state=0)

5. Model Training and Evaluation

- Logistic Regression:
    - Train: lr.fit(X_train, Y_train)
    - Predict: Y_pred_lr
    - Accuracy: accuracy_score
- Naive Bayes:
    - Train: nb.fit(X_train, Y_train)
    - Predict: Y_pred_nb
    - Accuracy: accuracy_score
- Support Vector Machine:
    - Train: sv.fit(X_train, Y_train)
    - Predict: Y_pred_svm
    - Accuracy: accuracy_score
- K-Nearest Neighbors:
    - Train: knn.fit(X_train, Y_train)
    - Predict: Y_pred_knn
    - Accuracy: accuracy_score
- Decision Tree:
    - Optimize with random seed:
      Loop through 200 iterations, tracking the best accuracy.
    - Train: dt.fit(X_train, Y_train)

- Predict: Y_pred_dt
- Accuracy: accuracy_score

- Random Forest:
  - Optimize with random seed:
    Loop through 2000 iterations, tracking the best accuracy.
  - Train: rf.fit(X_train, Y_train)
  - Predict: Y_pred_rf
  - Accuracy: accuracy_score
- XGBoost:
  - Train: xgb_model.fit(X_train, Y_train)
  - Predict: Y_pred_xgb
  - Accuracy: accuracy_score

6. Neural Network Training

- Model Architecture:
  - Layers:
    - Input layer: model.add(Dense(11, activation='relu', input_dim=13))
    - Output layer: model.add(Dense(1, activation='sigmoid'))
  - Compile: model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
  - Train: model.fit(X_train, Y_train, epochs=300)
- Prediction and Accuracy:
  - Predict: Y_pred_nn = model.predict(X_test)
  - Round Predictions: rounded = [round(x[0]) for x in Y_pred_nn]
  - Accuracy: accuracy_score

7. Accuracy Comparison

- Accuracy Scores:
  List of scores for all models:
  scores = [score_lr, score_nb, score_svm, score_knn, score_dt, score_rf, score_xgb, score_nn]
- Visualization:
  - Bar chart: plt.bar(algorithms, scores)

8. Final Print Statements

- Accuracy of each algorithm is printed in a loop:
  - print("The accuracy score achieved using "+algorithms[i]+" is: "+str(scores[i])+" %")

Abbreviations:

- ECG:electrocardiogram
- SVM:Support Vector Machine
- NN:neural network
- PCA:Principal Component Analysis
- MIT:Massachusetts Institute of Technology
- DWT:Discrete Wavelet Transform
- GBT:Gradient-Boosted Trees
- RF:Random Forest
- CNN: Convolutional Neural Network
- RNN: Recurrent Neural Network
- LSTM: Long Short-Term Memory
- DNN: Deep Neural Network
- ROC: Receiver Operating Characteristic
- AUC: Area Under the Curve

- CART: Classification and Regression Trees
- KNN: k-Nearest Neighbors
- Naive Bayes: Naive Bayes Classifier
- ACC: American College of Cardiology
- AHA: American Heart Association
- CAD:Coronary Artery Disease
- CHD:Coronary Heart Disease
- CVD:Cardiovascular Disease
- Framingham:Framingham Heart Study
- LDL:Low-Density Lipoprotein
- HDL:High-Density Lipoprotein
- CRP:C-reactive Protein
- BMI:Body Mass Index
- BP:Blood Pressure
- TC:Total Cholesterol
- HbA1c:Glycated Haemoglobin
- MI:Myocardial Infarction
- PCI:Percutaneous Coronary Intervention
- CABG:Coronary Artery Bypass Grafting
- LVH:Left Ventricular Hypertrophy
- EKG:Electrocardiogram

Nomenclature:

1. Libraries Used:
    - numpy: For numerical computations.
    - pandas: For data manipulation and analysis.

- matplotlib.pyplot and seaborn: For data visualization.
- os: To interact with the file system.
- warnings: To suppress unnecessary warnings.

2. Dataset Operations:
    - Dataset Reading: The dataset is read using pd.read_csv("heart.csv").
    - Basic Data Information:
        - dataset.describe() provides summary statistics.
        - dataset.info() gives an overview of data types and missing values.
    - Feature Description (info): A manual description of each column in the dataset.
3. Target Variable:
    - Distribution of target variable analyzed using sns.countplot() and value counts.
    - Calculation of percentages of patients with and without heart disease.
4. Feature-Target Splitting:
    - Features (X) and target (y) are split:
        - Predictors: dataset.drop("target", axis=1).
        - Target: dataset["target"].
5. Train-Test Splitting:
    - Data split into training and testing sets using train_test_split() with a test size of 20%.
6. Machine Learning Models:
    - Logistic Regression (LogisticRegression): Trained using lr.fit(), accuracy evaluated using accuracy_score.
    - Naive Bayes (GaussianNB): Trained using nb.fit(), accuracy evaluated.
    - Support Vector Machine (SVM): Linear kernel used with svm.SVC, accuracy calculated.

- K-Nearest Neighbors (KNN) (KNeighborsClassifier): Trained with n_neighbors=7, accuracy evaluated.
- Decision Tree (DecisionTreeClassifier):
  - Iterates over multiple random states to find the best accuracy.
- Random Forest (RandomForestClassifier):
  - Iterates over 2000 random states to find the best accuracy.
- XGBoost (XGBClassifier): Trained and tested for accuracy.

7. Neural Network Implementation:
   - Built using keras.Sequential with:
     - Input layer: 13 features.
     - Hidden layer: 11 nodes, activation: relu.
     - Output layer: 1 node, activation: sigmoid.
   - Compiled with binary_crossentropy loss and adam optimizer.
   - Trained for 300 epochs. Predictions rounded for accuracy evaluation.
8. Comparison of Model Performances:
   - Accuracy scores of all algorithms stored in the list scores.
   - Algorithms' names stored in the list algorithms.
   - Bar plot created using matplotlib to visualize accuracy comparison.

Visualizations in the Code:

- Count Plot: Visualizes the distribution of the target variable.
- Bar Plot: Compares accuracy scores of different algorithms.

Key Notes:

- Decision Tree and Random Forest use a random state loop to optimize accuracy.
- XGBoost and Neural Network are advanced models included.
- Neural Network accuracy improves with higher epochs and more nodes.

CHAPTER 1

INTRODUCTION

Computers can create and use learning methods thanks to a branch of artificial intelligence known as machine learning. We used a variety of supervised and unsupervised learning classifiers to predict and assess the dataset's accuracy. Cardiovascular illnesses are a broad term that refers to a variety of conditions that could damage the heart and circulatory system. Long considered one of the deadliest illnesses to exist, heart disease. "Cardiovascular disease" is the top cause of death worldwide, according to the WHO. (CVD). Four out of every five CVD fatalities are caused by heart attacks or strokes, with one-third of these deaths happening before the age of 70. Heart disease is one of the leading worldwide causes of death. For bettering patient outcomes, early detection and prediction of cardiac illness are essential. Forecasting the risk of cardiovascular disease has been effectively accomplished using machine learning techniques. These methods use statistical algorithms to analyze huge amounts of patient data to identify the most important risk factors for cardiac disease (Mohan et al., 2019). Blood flow to the heart or brain is blocked, which results in cardiac attacks. Basic medical facilities can monitor blood pressure, glucose, and lipid levels at home in people who are at risk for heart disease. To make accurate choices and offer the public high-quality services, healthcare professionals must possess these skills. The technique that the healthcare organization offers to professionals who lack more knowledge and expertise is crucial because current approaches cannot arrive at reliable conclusions. We predict heart disease based on health parameters using machine learning algorithms like logistic regression, Random Forest, and support vector machines. Unsupervised learning techniques, including clustering algorithms, have also been utilized to uncover hidden structures within patient data, facilitating the identification of novel risk groups and contributing to more personalized treatment approaches. Additionally, hybrid models that combine multiple ML techniques have been developed to enhance predictive performance further.The integration of ML into cardiovascular care enables continuous monitoring of vital health parameters, such as blood pressure, glucose levels, and lipid profiles, even in home settings. This proactive approach allows for timely interventions and supports healthcare professionals in making informed decisions, ultimately leading to improved patient care and resource optimization.As ML technologies continue to evolve, their application in CVD prediction and management holds great promise for advancing public health outcomes and fostering a more responsive healthcare system.

CHAPTER 2

LITERATURE REVIEW

2.1 Literature 1

Heart Disease Prediction Using Machine Learning Techniques

Heart disease, also called cardiovascular disease, is considered one of the deadliest diseases that cause high mortality worldwide. Early detection or prediction is a challenging task in the medical field. There is a massive amount of data in the healthcare industry, and processing this amount of data is a tedious task. A computer-aided system that predicts cardiac disease can save time and money. Researchers have researched several computer-assisted diagnoses for disease prediction and prognosis. In this paper, the authors provide an extensive literature survey of various classification approaches such as Machine Learning, Feature Selection, Hybrid, Ensemble, and Deep Learning used by researchers in the last decade for Heart Disease prediction. Furthermore, as the paper focuses on Machine Learning techniques, comparative analysis of the performance and accuracy of various Machine Learning techniques are summarized in tabular form. Additionally, this work critically assesses earlier methods and outlines their shortcomings. Finally, the article offers some potential future research direction in machine learning-based automated heart disease prediction. [1]

2.2 Literature 2

Heart Disease Detection Using Artificial Intelligence

As per the recent study by WHO, heart related diseases are increasing. 17.9 million people die every-year due to this. With growing population, it gets further difficult to diagnose and start treatment at early stage. But due to the recent advancement in technology, Machine Learning techniques have accelerated the health sector by multiple researches. Thus, the objective of this paper is to build a ML model for heart disease prediction based on the related parameters. We have used a benchmark dataset of UCI Heart disease prediction for this research work, which consist of 14 different parameters related to Heart Disease. Machine Learning algorithms such as Random Forest, Support Vector Machine (SVM), Naive Bayes and Decision tree have been used for the development of model. In our research we have also tried to find the correlations between the different attributes available in the dataset with the help of standard Machine Learning methods and then using them efficiently in the prediction of chances of Heart disease. Result shows that compared to other ML techniques, Random Forest gives more accuracy in less time for the prediction.

This model can be helpful to the medical practitioners at their clinic as decision support system. [2]

2.3 Literature3

Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques

Heart disease is one of the most significant causes of mortality in the world today. Prediction of cardiovascular disease is a critical challenge in the area of clinical data analysis. Machine learning (ML) has been shown to be effective in assisting in making decisions and predictions from the large quantity of data produced by the healthcare industry. We have also seen ML techniques being used in recent developments in different areas of the Internet of Things (IoT). Various studies give only a glimpse into predicting heart disease with ML techniques. In this paper, we propose a novel method that aims at finding significant features by applying machine learning techniques resulting in improving the accuracy in the prediction of cardiovascular disease. The prediction model is introduced with different combinations of features and several known classification techniques. We produce an enhanced performance level with an accuracy level of 88.7% through the prediction model for heart disease with the hybrid random forest with a linear model (HRFLM). [3]

2.4 Literature 4

Artificial intelligence in the diagnosis and detection of heart failure: the past, present, and future

Artificial Intelligence (AI) performs human intelligence-dependent tasks using tools such as Machine Learning, and its subtype Deep Learning. AI has incorporated itself in the field of cardiovascular medicine, and is increasingly employed to revolutionize diagnosis, treatment, risk prediction, clinical care, and drug discovery. Heart failure has a high prevalence, and the mortality rate following hospitalization is 10.4% at 30 days, 22% at 1 year, and 42.3% at 5 years. Early detection of heart failure is of vital importance in shaping the medical and surgical interventions specific to HF patients. This has been accomplished with the advent of the Neural Network (NN) model, the accuracy of which has proven to be 85%. AI can be of tremendous help in analyzing raw image data from cardiac imaging techniques (such as echocardiography, computed tomography, cardiac MRI amongst others) and electrocardiogram recordings through incorporation of an algorithm. The use of decision trees by Rough Sets (RS), and logistic regression (LR) methods utilized to construct a decision-making model to diagnose congestive heart failure, and the role of AI in early detection of future mortality and destabilization episodes has played a vital role in optimizing cardiovascular disease outcomes. The review highlights the major achievements

of AI in recent years that have radically changed nearly all areas of HF prevention, diagnosis, and management. [4]

2.5 Literature 5

Heart disease prediction using supervised machine learning algorithms: Performance analysis and comparison

Machine learning and data mining-based approaches to prediction and detection of heart disease would be of great clinical utility, but are highly challenging to develop. In most countries there is a lack of cardiovascular expertise and a significant rate of incorrectly diagnosed cases which could be addressed by developing accurate and efficient early-stage heart disease prediction by analytical support of clinical decision-making with digital patient records. This study aimed to identify machine learning classifiers with the highest accuracy for such diagnostic purposes. Several supervised machine-learning algorithms were applied and compared for performance and accuracy in heart disease prediction. Feature importance scores for each feature were estimated for all applied algorithms except MLP and KNN. All the features were ranked based on the importance score to find those giving high heart disease predictions. This study found that using a heart disease dataset collected from Kaggle three-classification based on k-nearest neighbor (KNN), decision tree (DT) and random forests (RF) algorithms the RF method achieved 100% accuracy along with 100% sensitivity and specificity. Thus, we found that a relatively simple supervised machine learning algorithm can be used to make heart disease predictions with very high accuracy and excellent potential utility. [5]

2.6 Literature 6

Heart Disease Prediction Using Artificial Intelligence Ensemble Network

Heart disease has climbed its way to the top of the list of the primary causes of death all over the world. In the past, individuals also referred to heart disease as cardiovascular disease when talking about it. Heart disease and stroke are the primary causes of death in India, together accounting for one death in every four that occur there. The use of machine learning to the process of forming judgements and creating predictions based on the large quantities of data created by the healthcare business is highly valuable. This is because machine learning can analyse patterns in the data to make more accurate predictions. According to the information that was provided by the WHO, cardiovascular disease is the primary cause of around 24 percent of deaths in India that are attributed to non-communicable illnesses. These deaths are mostly caused by coronary artery disease (CVD). In addition, coronary heart disease is the main cause of death in industrialised nations like the United States of America and other rich countries. Around 17 million people each year

lose their lives to cardiovascular disease, making it the leading cause of death on a global scale; the incidence of cardiovascular disease mortality is greatest in Asia. The Cleveland heart disease dataset's primary objective was to provide information that could be used to conduct an analysis of the system. The implementation of the prediction model takes use of a broad range of feature integrations, in addition to numerous techniques of categorization that are already widely known to the general public.. [6]

2.7 Literature 7

Heart Disease Prediction Using Machine Learning Techniques: A Quantitative Review Conference paper

Cardiovascular disease refers to any critical condition that impacts the heart. Because heart diseases can be life-threatening, researchers are focusing on designing smart systems to accurately diagnose them based on electronic health data, with the aid of machine learning algorithms. This work presents several machine learning approaches for predicting heart diseases, using data of major health factors from patients. The paper demonstrated four classification methods: Multilayer Perceptron (MLP), Support Vector Machine (SVM), Random Forest (RF), and Naïve Bayes (NB), to build the prediction models. Data preprocessing and feature selection steps were done before building the models. The models were evaluated based on the accuracy, precision, recall, and F1-score. The SVM model performed best with 91.67% accuracy.[7]

2.8 Literature 8

Prediction of Coronary Heart Disease using Supervised Machine Learning Algorithms

New technologies like Machine learning and Big data analytics have been proven to provide promising solutions to biomedical communities, healthcare problems, and patient care. They also help in early prediction of disease by accurate interpretation of medical data. Disease management strategies can further be improved by the detection of early signs of disease. This early prediction, moreover, can be helpful in controlling the symptoms of the disease as well as the proper treatment of disease. Machine learning approaches can be used in the prediction of chronic diseases, such as kidney and heart diseases, by developing the classification models. In this paper, we propose a preprocessing extensive approach to predict Coronary Heart Diseases (CHD). The approach involves replacing null values, resampling, standardization, normalization, classification, and prediction. This work aims to

predict the risk of CHD using machine learning algorithms like Random Forest, Decision Trees, and K-Nearest Neighbours. Also, a comparative study among these algorithms on the basis of prediction accuracy is performed. Further, K-fold Cross Validation is used to generate randomness in the data. These algorithms are experimented over "Framingham Heart Study" dataset, which is having 4240 records. In our experimental analysis, Random Forest, Decision Tree, and K-Nearest Neighbour achieved an accuracy of 96.8%, 92.7%, and 92.89% respectively. Therefore, by including our preprocessing steps, Random Forest classification gives more accurate results than other machine learning algorithms. [8]

2.9 Literature

Prediction of hospitalization due to heart diseases by supervised learning methods Background

In 2008, the United States spent $2.2 trillion for healthcare, which was 15.5% of its GDP. 31% of this expenditure is attributed to hospital care. Evidently, even modest reductions in hospital care costs matter. A 2009 study showed that nearly $30.8 billion in hospital care cost during 2006 was potentially preventable, with heart diseases being responsible for about 31% of that amount. Our goal is to accurately and efficiently predict heart-related hospitalizations based on the available patient-specific medical history. To the best of our knowledge, the approaches we introduce are novel for this problem. The prediction of hospitalization is formulated as a supervised classification problem. We use de-identified Electronic Health Record (EHR) data from a large urban hospital in Boston to identify patients with heart diseases. Patients are labeled and randomly partitioned into a training and a test set. We apply five machine learning algorithms, namely Support Vector Machines (SVM), AdaBoost using trees as the weak learner, logistic regression, a naïve Bayes event classifier, and a variation of a Likelihood Ratio Test adapted to the specific problem. Each model is trained on the training set and then tested on the test set. All five models show consistent results, which could, to some extent, indicate the limit of the achievable prediction accuracy. Our results show that with under 30% false alarm rate, the detection rate could be as high as 82%. These accuracy rates translate to a considerable amount of potential savings, if used in practice. [9]

2.10 Literature

Heart Disease Detection Using Feature Extraction and Artificial Neural Networks: A Sensor-Based Approach

study presents a novel technique for identifying individuals using feature extraction methods and signal processing approaches. It uses an artificial neural network (ANN) technique to identify scent patterns in individuals using ten metal oxide semiconductor sensors. Sensor data is scanned and extracted before using ANN patterns. Before using ANN patterns to generate patterns from sensor data, it is important to scan and extract sensory information from that data. Each participant is recognized and scanned for a totally of 1000 different characteristics during the course of the multiple investigations, which are conducted across a variety of time periods that include 5, 10, 15, and 20 people. Because of the varying time periods, signals from sensors are received in analog form, which is then transformed by Arduino into digital form. It is necessary to train an architecture on the data set that has been created. The benchmarks that are employed for the assessment of the model that is presented for the identification of human odor include sensitivity, f-measures, accuracy, and specificity, among other things. Experiments are carried out using the assessment measures, and the findings demonstrate that this model has an accuracy of greater than 85 % in most cases. The research demonstrates the potential of feature extraction methods in identifying individuals and enhancing human odor identification. [10]

CHAPTER 3

THEORY

Machine Literacy ways have been around us and have been compared and used for analysis for numerous kinds of data wisdom operations. The major provocation behind this exploration-grounded design was to explore the point selection styles, data medication, and processing behind the training models in machine literacy. With first-hand models and libraries, the challenge we face moment is data where besides their cornucopia, and our cooked models, the delicacy we see during training, testin,g and factual confirmation has advanced friction. Hence this design is carried out with the provocation to explore behind the models, and further apply Logistic Retrogression. model to train the attained data. likewise, as the whole machine literacy is motivated to develop an applicable computer- grounded system and decision support that can prop to early discovery of heart complaint, in this design we've developed a model which classifies if case will jave heart complaint in ten times or not grounded on colorful features(i.e., implicit threat factors that can beget heart complaint) using logistic retrogression. Hence, the early prognostic of cardiovascular conditions can prop in making opinions on life changes in high-threat cases and in turn reduce the complications, which can be a great corner in the field of drugs.

- **Objective**:

- To develop a machine learning model to predict the future possibility of heart disease by implementing Logistic Regression.

- To determine significant risk factors based on medical datasets which may lead to heart disease.

- To analyze feature selection methods and understand their working principle.

Methodology

The suggested method's major goal is to forecast the likelihood of developing heart disease to quickly and accurately diagnose the condition. In our strategy, we employ several data mining approaches and machine learning algorithms, including “Naive Bayes”, “k Nearest Neighbor (KNN)”, “Decision tree”, “Artificial Neural Network (ANN)”, and “Random Forest”, to predict the occurrence of heart disease based on a few health-related variables[20].

Our methodology begins with data collection, where we gather relevant health-related variables from a diverse set of individuals. These variables may include age, gender, blood pressure, cholesterol levels, smoking habits, family history, and other risk factors associated with heart disease.

Once the data is collected, we preprocess it by cleaning any inconsistencies or missing values. We also perform feature selection to identify the most significant variables that contribute to the prediction of heart disease. This step helps reduce dimensionality and improve the efficiency and accuracy of our models.

Next, we apply several data mining approaches and machine learning algorithms to build predictive models. The Naive Bayes algorithm is utilized to estimate the probability of heart disease based on the given input variables. It assumes independence between the predictors, making it a fast and simple algorithm for classification tasks.

The k Nearest Neighbor (KNN) algorithm is employed to classify individuals based on their similarity to other individuals in the dataset. It calculates the distance between data points and assigns a class label based on the nearest neighbors. KNN is effective when the data exhibits local patterns and can provide accurate predictions.

The Decision Tree algorithm is utilized to create a tree-like model that predicts heart disease based on a series of if-else conditions. Decision trees are interpretable and can handle both categorical and numerical data, making them useful for understanding the underlying rules for heart disease prediction.Lastly, the Random Forest algorithm is utilized to create an ensemble of decision trees. It combines multiple decision trees and aggregates their predictions to make a final prediction. Random Forest is known for its ability to handle high-dimensional data and reduce overfitting, providing robust predictions.

After building the predictive models, we evaluate their performance using various evaluation metrics such as accuracy, precision, recall, and F1 score. We also employ techniques like cross-validation to assess the models' generalization capabilities and ensure they can perform well on unseen data.

In conclusion, our methodology combines data mining approaches and machine learning algorithms to predict the likelihood of developing heart disease. By leveraging a diverse set

of health-related variables and utilizing multiple algorithms, we aim to create accurate and efficient models that can assist in the early diagnosis and prevention of heart disease.

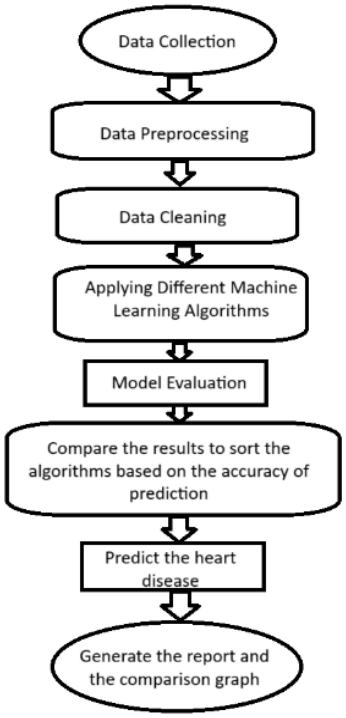


Figure 1: Flowchart of Heart Disease Prediction

MATERIALS

DATASET

The dataset is publicly available on the Kaggle Website at [4] which is from an ongoing cardiovascular study on residents of the town of Framingham, Massachusetts. It provides patient information which includes over 4000 records and 14 attributes. The attributes include: age, sex,chest pain type, resting blood pressure, serum cholesterol, fasting, sugar blood, resting electrocardiographic results, maximum heart rate, exercise-induced angina, ST depression induced by exercise, the slope of the peak exercise, number of major vessels, and target ranging from 0 to 2, where 0the is the absence of heart disease. The data set is in CSV (Comma Separated Value) format which is further prepared to data frame as supported by the panda's library in Python[6].

The datasets consists of several medical predictor variables and one target variable, Outcome. Predictor variables include age, gender, cp, ca, etc.

- Age.
- Sex.
- Chest Pain type (CP).
- Trestbps (on admission to the hospital, resting blood pressure in mm Hg.).
- Cholesterol.
- Restecg (resting electrocardiographic results: assesses the heart's activity.).
- Thalach (attained maximum heart rate).
- Exang (Angina caused by exercise is a common complaint of cardiac patients, particularly when exercising in the cold).
- Old peak (Exercise-induced ST depression compared to rest).
- Slope (the curve of the ST segment of the peak activity).
- Ca (fluoroscopy coloration of a lot of major vessels (0-3))
- Thal (normal, fixed defect, reversable defect).

```
In [2]: import numpy as np
        import pandas as pd
        from sklearn.model_selection import train_test_split
        from sklearn.linear_model  import LogisticRegression
        from sklearn.metrics import accuracy_score

In [5]: heartdataset=pd.read_csv("heart_disease_data.csv")
        heartdataset.head()
        #first 5 rows
```

Out[5]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

```
In [7]: heartdataset.tail()
        #last 5 rows
```

Out[7]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

Figure 2: Dataset

```
dataset
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
298	57	0	0	140	241	0	1	123	1	0.2	1	0	3	0
299	45	1	3	110	264	0	1	132	0	1.2	1	0	3	0
300	68	1	0	144	193	1	1	141	0	3.4	1	2	3	0
301	57	1	0	130	131	0	1	115	1	1.2	1	1	3	0
302	57	0	1	130	236	0	0	174	0	0.0	1	1	2	0

303 rows × 14 columns

Figure 2.1: Data Collection

```
dataset.describe()
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	2.313531	0.544554
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	0.612277	0.498835
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.000000	1.000000

Figure 3: Data preprocessing

METHODS

1. Backend Development: Implements 8 models as Logistic regression, Support Vector Machine, Random Forest Model, Naive Bayes, K-Nearest Neighbours, Decision Tree, XG Boost, Neural Network for generating responses, a custom tokenizer for text preprocessing, and data handling from a CSV file to prepare input-output pairs.

3.1.1 Relevent Models

LOGISTIC REGRESSION:

The logistic regression machine learning technique is used when the dependent variables can be categorized. The results of logistic regression are generated based on the characteristics that are given.

It is possible to calculate the probability of classification issues with two outcomes using supervised learning methods like logistic regression. It can also be used to forecast many classes.

We utilized this function to successfully convert any integer into a value between 0 and 1, which we then used to determine the probability of class identification. For instance, there are two categories of heart disease: those who are affected and those who are not.

“Logistic regression” is a member of the “supervised machine learning model” family in the context of “artificial intelligence”. It is likewise regarded as a “discriminative model”, which denotes that it makes an effort to discriminate between classes. It cannot, as the name implies, generate information of the class that it is trying to predict (for example, a picture of a cat), unlike a generative algorithm like “naive bayes”. We previously discussed how the beta coefficients of the model are calculated using “logistic regression”, which maximizes the log likelihood function. When seen in the perspective of “machine learning”, this modifies slightly. In “machine learning”, the loss function is the negative log likelihood, and the global maximum is found using gradient descent. The estimations stated above can also be reached in this manner.

Additionally, “logistic regression” is susceptible to overfitting, especially when the model contains a large number of predictor variables. When a model has high dimensionality, regularisation is often employed to penalize big coefficients in the parameters. To understand more about the logistic regression machine learning model, refer to “Scikit-learn”.

For problems involving categorization and prediction, logistic regression is frequently utilized. Some examples of these use cases are:

Fraud detection: Teams can uncover data anomalies that are indicative of fraud with the aid of logistic regression models. In order to better safeguard their customers, banking and other financial organisations may find that certain behaviours or attributes are more

frequently associated with fraudulent operations. In order to remove false user accounts from their datasets when conducting data analysis on company performance, SaaS-based organisations have also started to implement these practises.Disease Prediction: This analytics strategy can be applied to medicine to forecast the likelihood of disease or illness in a certain group. Healthcare organisations can set up preventative care for those who have a higher risk of developing a certain ailment.

Churn Prediction: Different organizational functions may exhibit particular behaviours that are indicative of churn. For instance, management and human resources teams may want to know whether there are high performers inside the firm who may leave; this kind of information can spark discussions to address problem areas within the company, such as culture or salary. As an alternative, the sales team would want to find out which of its customers might decide to do business elsewhere. In order to prevent income loss, this may inspire teams to develop a retention plan.

Some methods of “logistic regression”:

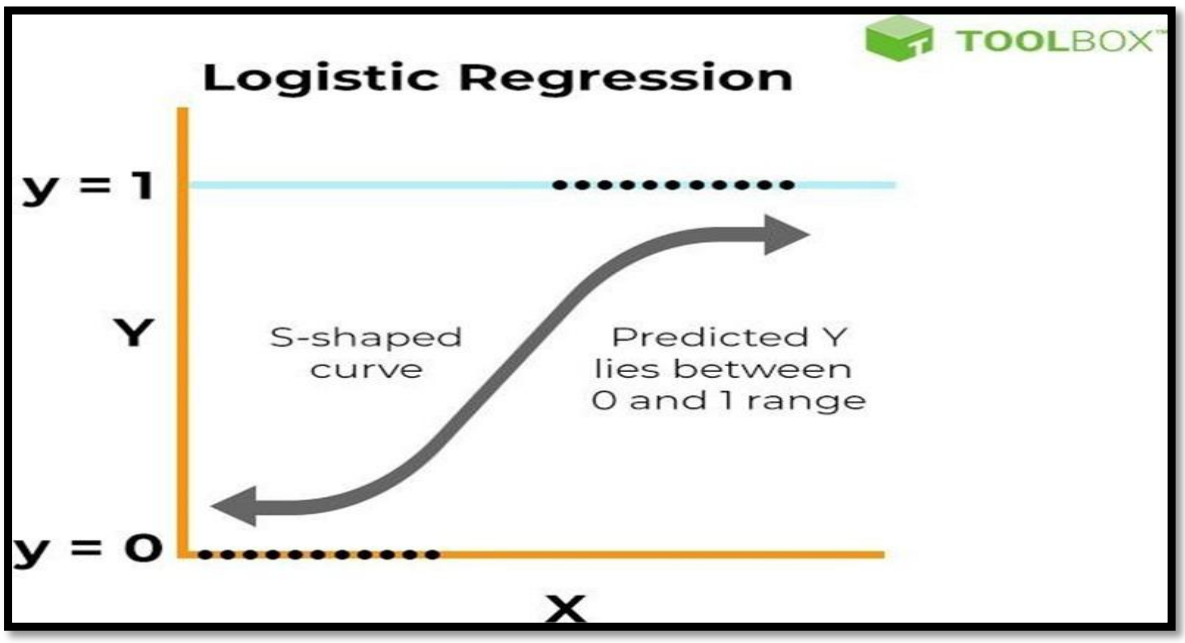


Figure 4: Logistic Regression

Logesticis

+ Code + Markdown

```
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()

logreg.fit(X_train, Y_train)

y_pred_lr = logreg.predict(X_test)
# print(y_pred_lr)SS
```

```
[0 1 1 0 0 0 0 0 0 0 1 1 0 1 1 1 0 1 0 1 1 0 0 0 1 0 0 0 1 1 1 0 1 1 1 1 0
 1 0 0 1 1 0 0 0 1 1 1 0 1 1 1 1 1 1 1 0 1 1 1 1 1]
c:\Users\USER\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\linear_model\_logistic.py:444
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```
score_lr = round(accuracy_score(y_pred_lr,Y_test)*100,2)

print("The accuracy score achieved using Logistic Regression is: "+str(score_lr))
```

```
The accuracy score achieved using Logistic Regression is: 85.25
```

Figure 4.1: Logistic Regression

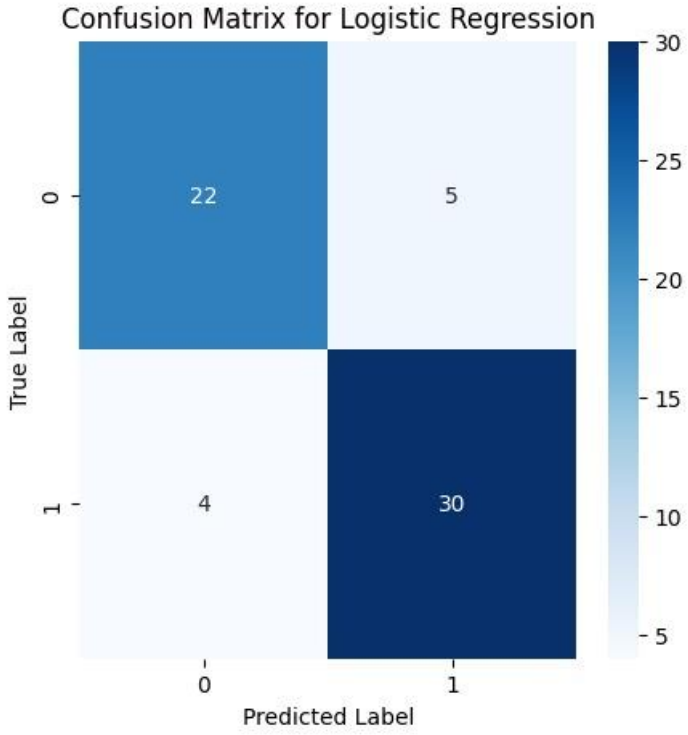


Figure 4.2: Confusion matrix for Logistic Regression

Metrics for Logistic Regression:

Precision: 0.8571

Recall: 0.8824

F1 Score: 0.8696

SVM: Support Vertor Machine

Several studies have used the Support Vector Machine (SVM) that Vapnik and Cortes suggested with success for gender categorization issues (Yadav, 2021). The separation hyperplane in an SVM classifier is chosen to reduce the anticipated classification error of the unobserved test patterns. SVM is a potent predictor that can tell one class from another.

SVM places the test image in the class that is farthest from the closest point in the training image.

Using the SVM training method, a model that predicts whether the test image corresponds to this class or another was developed. Even if we restrict ourselves to single pose (frontal) detection, SVM still needs a lot of training data to determine the emotional decision boundary, and the computational cost is high.

Support Vector Machine or SVM is one of the most popular Supervised literacy algorithms, which is used for Bracket as well as Retrogression problems. still, primarily, it's used for Bracket problems in Machine literacy. The thing of the SVM algorithm is to produce the stylish line or decision boundary that can insulate n- dimensional space into classes so that we can fluently put the new data point in the correct order in the future. This stylish decision boundary is called a hyperplane. SVM chooses the extreme points vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is nominated as Support Vector Machine. Consider the below illustration in which there are two different orders that are classified using a decision boundary or hyperplane.

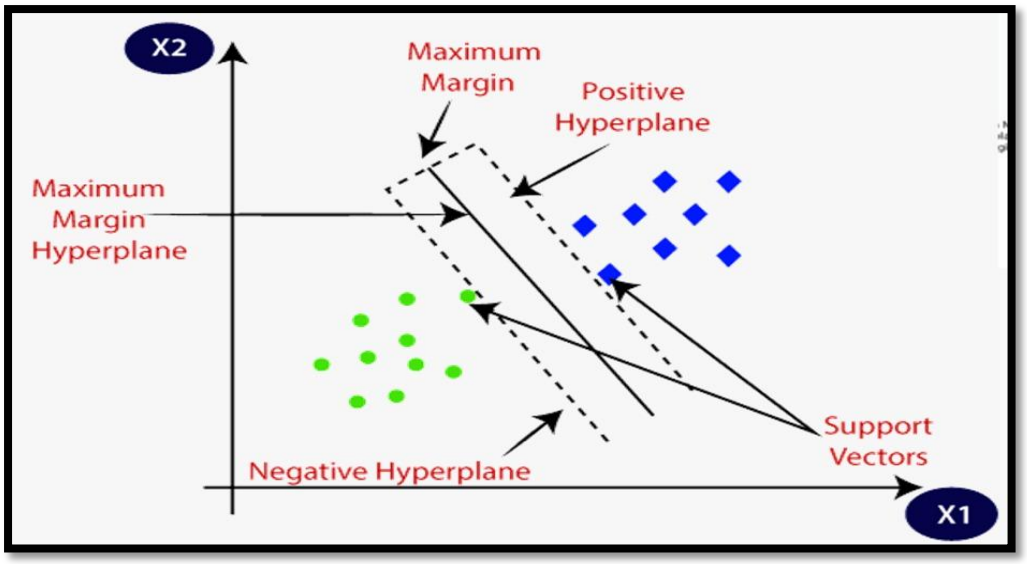


Figure 5: Support Vector Machine

SVC

```
from sklearn.svm import SVC # "Support vector classifier"
classifier = SVC(kernel='linear', random_state=0)
classifier.fit(X_train,Y_train)
```

```
SVC(kernel='linear', random_state=0)
```

```
#accuracy on training data
X_train_Prediction=classifier.predict(X_train)
training_data_accuracy=accuracy_score(X_train_Prediction,Y_train)
print('Accuracy on Training Data:',training_data_accuracy)
```

```
Accuracy on Training Data: 0.8553719008264463
```

```
#accuracy on test data
X_test_Prediction=classifier.predict(X_test)
test_data_accuracy=accuracy_score(X_test_Prediction,Y_test)
print('Accuracy on Test Data:',test_data_accuracy)
```

```
Accuracy on Test Data: 0.819672131147541
```

Figure 5.1: Support Vector Machine

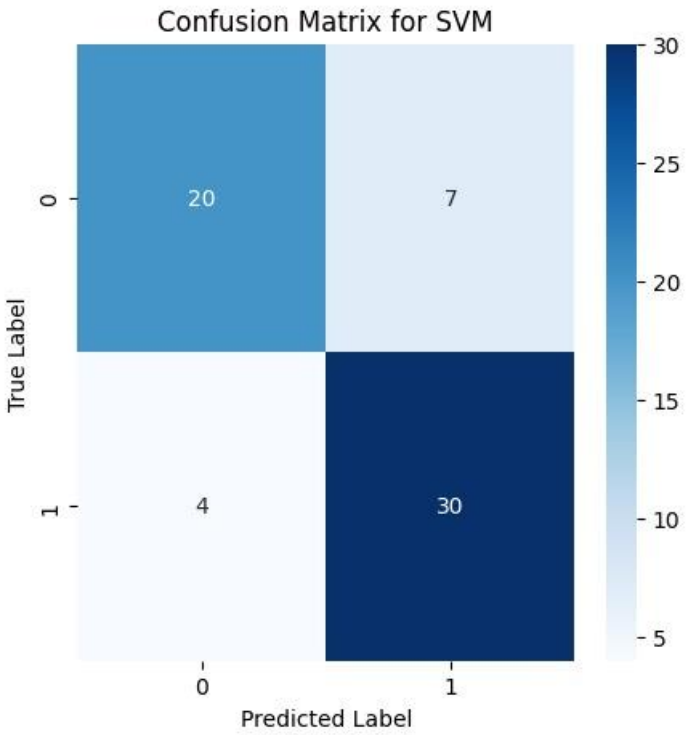


Figure 5.2: Confusion matrix for Support Vector Machine

Metrics for Support Vector Machine:

Precision: 0.8108

Recall: 0.8824

F1 Score: 0.8451

RANDOM FOREST:

Leo Breiman and Adele Cutler are the creators of the widely used machine learning algorithm known as random forest, which combines the output of various decision trees to produce a single outcome. Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems.

Since the “random forest model” consists of several “decision trees”, it is helpful to start by briefly describing the “decision tree algorithm”. A “decision tree” starts with a basic question, such as "Should I surf?" From there, you can ask a series of questions to determine the answer, such as "Is this a long-lasting swelling?" or "The wind blows at sea?". These questions form decision nodes in the tree and serve as a means of partitioning the data. Each question helps make a final decision represented by a leaf node. Observations

that meet the criteria follow the yes branch, while observations that do not follow the alternate path. “Decision trees” attempt to find the best splits on a subset of data and are typically trained using the “Classification Regression Tree (CART) algorithm”. Metrics such as “Gini contamination”, “information gain”, and “mean squared error (MSE)” can be used to assess the quality of the splits.

This decision tree is an example of a classification problem with and without class label surfing.

“Decision trees” are a popular algorithm for “supervised learning”, but they can be prone to problems such as bias and overfitting. However, when “multiple decision trees” form an ensemble, the “random forest algorithm” predicts more accurate results, especially if the individual trees are not mutually correlated.

With the help of “random forest classifier” we have been able to achieve 93% accuracy.

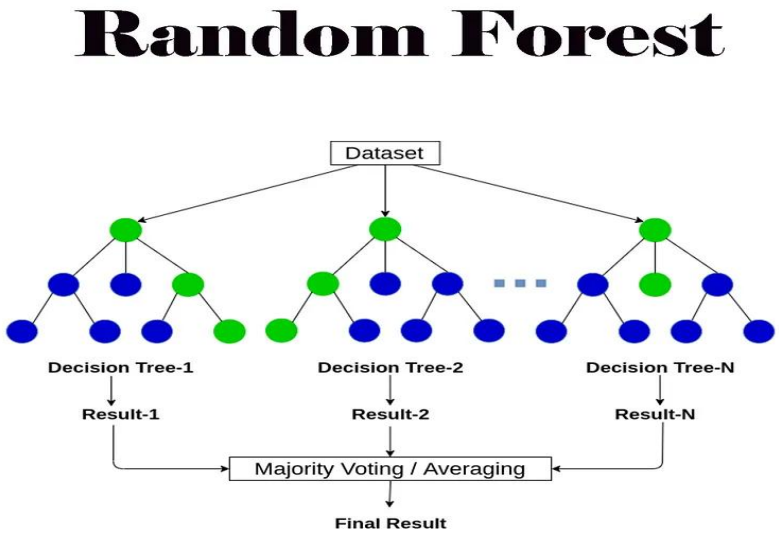


Figure 6: Random Forest Model

Gini Impurity Formula

$$G(node) = \sum_{k=1}^{c} pk(1 - pk)$$

- c: Total number of classes.
- pk: Proportion of samples belonging to class k at the node.

This form still calculates the **probability of misclassification** at a node, just written differently.

RANDOM FOREST

```
from sklearn.ensemble import RandomForestClassifier
max_accuracy = 0
for x in range(2000):
    rf = RandomForestClassifier(random_state=x)
    rf.fit(X_train,Y_train)
    Y_pred_rf = rf.predict(X_test)
    current_accuracy = round(accuracy_score(Y_pred_rf,Y_test)*100,2)
    if(current_accuracy>max_accuracy):
        max_accuracy = current_accuracy
        best_x = x
# print(max_accuracy)
# print(best_x)
rf = RandomForestClassifier(random_state=best_x)
rf.fit(X_train,Y_train)
Y_pred_rf = rf.predict(X_test)
```

✓ 0.6s

```
score_dt = round(accuracy_score(Y_pred_rf,Y_test)*100,2)
print("The accuracy score achieved using Random Forest is: "+str(score_dt)+" %")
```

✓ 0.0s

```
The accuracy score achieved using Random Forest is: 93.61 %
```

Figure 6.1: Random Forest Model

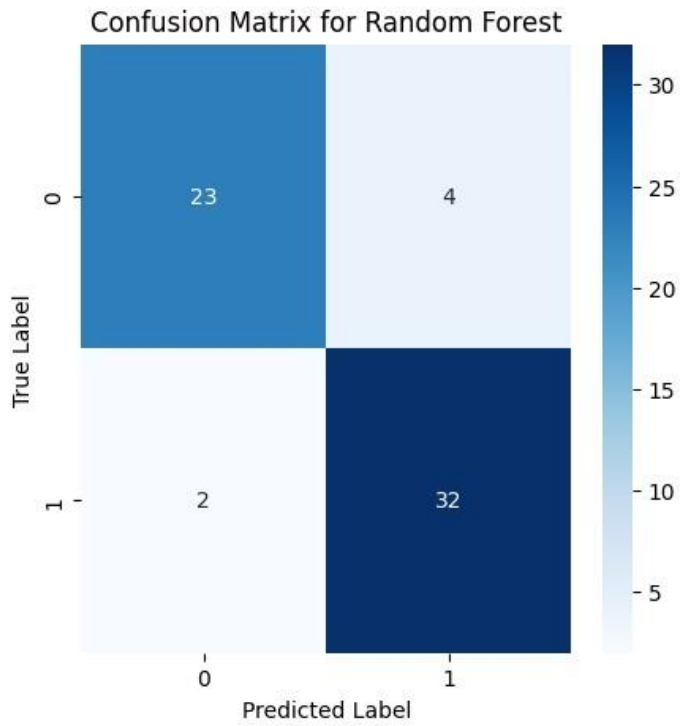


Figure 6.2: Confusion matrix for Random Forest

Metrics for Random Forest:

Precision: 0.8889

Recall: 0.9412

F1 Score: 0.9143

Naive Bayes classifiers:

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other. To start with, let us consider a dataset.

One of the most simple and effective classification algorithms, the Naïve Bayes classifier aids in the rapid development of machine learning models with rapid prediction capabilities.

Naïve Bayes algorithm is used for classification problems. It is highly used in text classification. In text classification tasks, data contains high dimension (as each word represent one feature in the data). It is used in spam filtering, sentiment detection, rating classification etc. The advantage of using naïve Bayes is its speed. It is fast and making prediction is easy with high dimension of data.

This model predicts the probability of an instance belongs to a class with a given set of feature value. It is a probabilistic classifier. It is because it assumes that one feature in the model is independent of existence of another feature. In other words, each feature contributes to the predictions with no relation between each other. In real world, this condition satisfies rarely. It uses Bayes theorem in the algorithm for training and prediction.

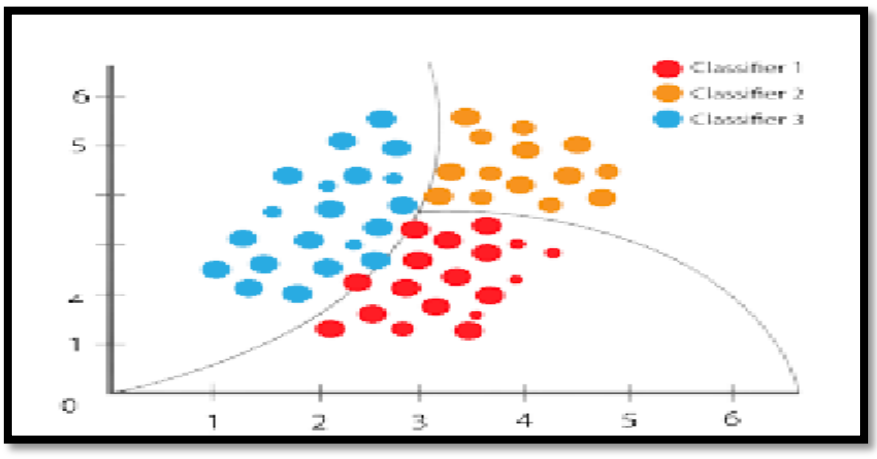


Figure 7: Naïve Bayes

Naives Bayes Classifier

```
from sklearn.metrics import classification_report
from sklearn.naive_bayes import GaussianNB

model4 = GaussianNB() # get instance of model
model4.fit(x_train, y_train) # Train/Fit model

y_pred4 = model4.predict(x_test) # get y predictions
print(classification_report(y_test, y_pred4)) # output accuracy
```

```
              precision    recall  f1-score   support

           0       0.81      0.75      0.78        28
           1       0.80      0.85      0.82        33

    accuracy                           0.80        61
   macro avg       0.80      0.80      0.80        61
weighted avg       0.80      0.80      0.80        61
```

```
#Acurency
X_train_Prediction=model4.predict(x_train)
training_data_accuracy=accuracy_score(X_train_Prediction,y_train)
t_data=r2_score(X_train_Prediction,y_train)
print(training_data_accuracy)
print(t_data)
```

```
0.8553719008264463
```

Figure 7.1: Naïve Bayes

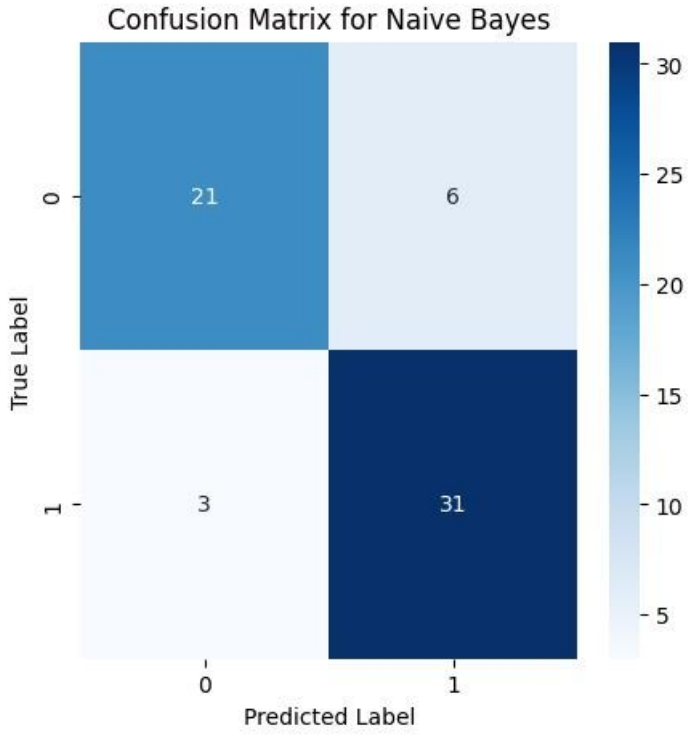


Figure 7.2: Confusion matrix for Naïve Bayes

Metrics for Naive Bayes:

Precision: 0.8378

Recall: 0.9118

F1 Score: 0.8732

KNN: K-Nearest Neighbor

k- NN is a type of bracket where the function is only approached locally and all calculation is remitted until function evaluation. Since this algorithm relies on distance for bracket, if the features represent different physical units or come in extensively different scales also homogenizing the training data can ameliorate its delicacy dramatically.(3) Both for bracket and retrogression, a useful fashion can be to assign weights to the benefactions of the neighbors, so that the nearer neighbors contribute further to the average than the more distant bones. For illustration, a common weighting scheme consists in giving each neighbor a weight of 1/ d, where d is the distance to the neighbor.(4) The neighbors are taken from a set of objects for which the class(for k- NN bracket) or the object property value(for k- NN retrogression) is known. This can be allowed as the training set for the algorithm, though no unequivocal training step is needed. A peculiarity of the k- NN algorithm is that it's sensitive to the original structu K-Nearest Neighbor of the data.

The k-nearest neighbor classifier can be viewed as assigning the k-nearest neighbors a weight and all others 0 weight. This can be generalized to weighted nearest neighbor classifiers. That is, where the ith nearest neighbor is assigned a weight, with. An analogous result on the strong consistency of weighted nearest neighbor classifiers also holds. Let denote the weighted nearest classifier with weights. Subject to regularity conditions, which in asymptotic theory are conditional variables that require assumptions to differentiate among parameters with some criteria. On the class distributions, the excess risk has the following asymptotic expansion.

$$\mathcal{R}_{\mathcal{R}}(C_n^{wnn}) - \mathcal{R}_{\mathcal{R}}(C^{\text{Bayes}}) = \left(B_1 s_n^2 + B_2 t_n^2\right)\{1 + o(1)\},$$

for constants B_1 and B_2 where $s_n^2 = \sum_{i=1}^{n} w_{ni}^2$ and $t_n = n^{-2/d} \sum_{i=1}^{n} w_{ni} \left\{ i^{1+2/d} - (i-1)^{1+2/d} \right\}$.

The optimal weighting scheme $\{w_{ni}^*\}_{i=1}^n$, that balances the two terms in the display above, is given as follows: set $k^* = \lfloor Bn^{\frac{4}{d+4}} \rfloor$,

$$w_{ni}^* = \frac{1}{k^*}\left[1 + \frac{d}{2} - \frac{d}{2k^{*2/d}}\{i^{1+2/d} - (i-1)^{1+2/d}\}\right]$$

for $i = 1, 2, \ldots, k^*$ and

$$w_{ni}^* = 0$$

for $i = k^* + 1, \ldots, n$.

Figure 8: K-Nearest Neighbor

K-Nearest Neighbors

```
from sklearn.metrics import classification_report
from sklearn.neighbors import KNeighborsClassifier

model2 = KNeighborsClassifier() # get instance of model
model2.fit(x_train, y_train) # Train/Fit model

y_pred2 = model2.predict(x_test) # get y predictions
print(classification_report(y_test, y_pred2)) # output accuracy
```

```
              precision    recall  f1-score   support

           0       0.78      0.75      0.76        28
           1       0.79      0.82      0.81        33

    accuracy                           0.79        61
   macro avg       0.79      0.78      0.78        61
weighted avg       0.79      0.79      0.79        61
```

```
#accurency score
X_train_Prediction=model2.predict(x_train)
training_data_accuracy=accuracy_score(X_train_Prediction,y_train)
t_data=r2_score(X_train_Prediction,y_train)
print(training_data_accuracy)
print(t_data)
```

```
0.8636363636363636
```

Figure 8.1 K-Nearest Neighbor

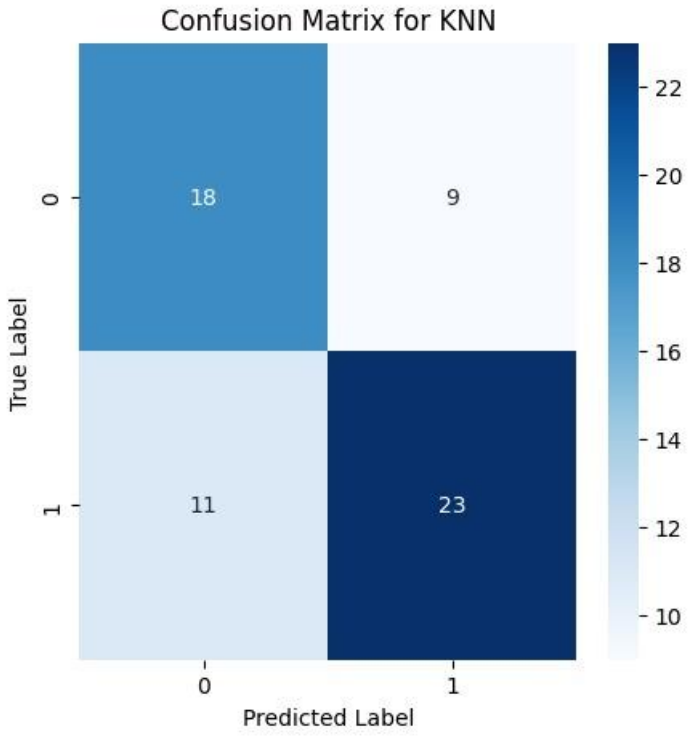


Figure 8.2: Confusion matrix for KNN: K-Nearest Neighbor

Metrics for K-Nearest Neighbors:

Precision: 0.7188

Recall: 0.6765

F1 Score: 0.6970

DECISSION TREE

A decision tree is a flowchart-like structure used to make decisions or predictions. It consists of nodes representing decisions or tests on attributes, branches representing the outcome of these decisions, and leaf nodes representing final outcomes or predictions. Each internal node corresponds to a test on an attribute, each branch corresponds to the result of the test, and each leaf node corresponds to a class label or a continuous value.

Root Node: Represents the entire dataset and the initial decision to be made.

Internal Nodes: Represent decisions or tests on attributes. Each internal node has one or more branches.

Branches: Represent the outcome of a decision or test, leading to another node.

Leaf Nodes: Represent the final decision or prediction. No further splits occur at these nodes.

The process of creating a decision tree involves:

1. Selecting the Best Attribute: Using a metric like Gini impurity, entropy, or information gain, the best attribute to split the data is selected.

2. Splitting the Dataset: The dataset is split into subsets based on the selected attribute.

3. Repeating the Process: The process is repeated recursively for each subset, creating a new internal node or leaf node until a stopping criterion is met (e.g., all instances in a node belong to the same class or a predefined depth is reached).

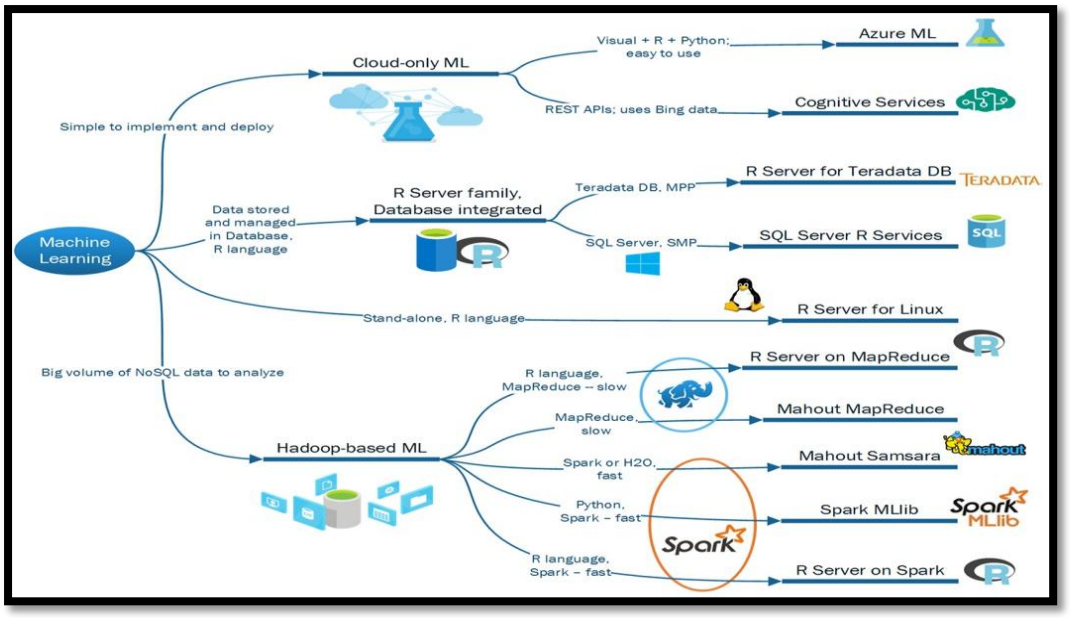


Figure 9: Decission Tree

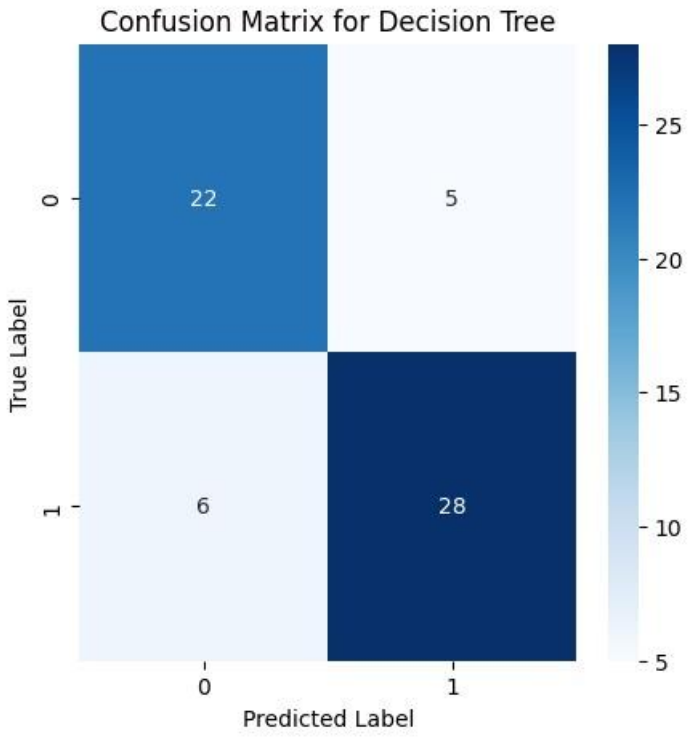


Figure 9.1: Confusion matrix for Decision Tree

Metrics for Decision Tree:

Precision: 0.8485

Recall: 0.8235

F1 Score: 0.8358

XGBoost

XGBoost (Extreme Gradient Boosting) is a powerful machine learning algorithm based on gradient boosting, where multiple weak models (typically decision trees) are combined to form a stronger model. It works by sequentially training decision trees, with each new tree focusing on correcting the errors made by previous ones. XGBoost enhances this process by using regularization (L1 and L2) to prevent overfitting, making it more robust. It also improves speed and efficiency through parallelization, allowing faster training. The algorithm handles missing data automatically and uses advanced techniques like "weighted quantile sketch" for efficient decision tree splitting. XGBoost updates the model iteratively by adding predictions from new trees, scaled by a learning rate to control the model's complexity. This approach leads to highly accurate predictions while reducing the risk of overfitting, making XGBoost one of the most popular algorithms for both classification and regression tasks.

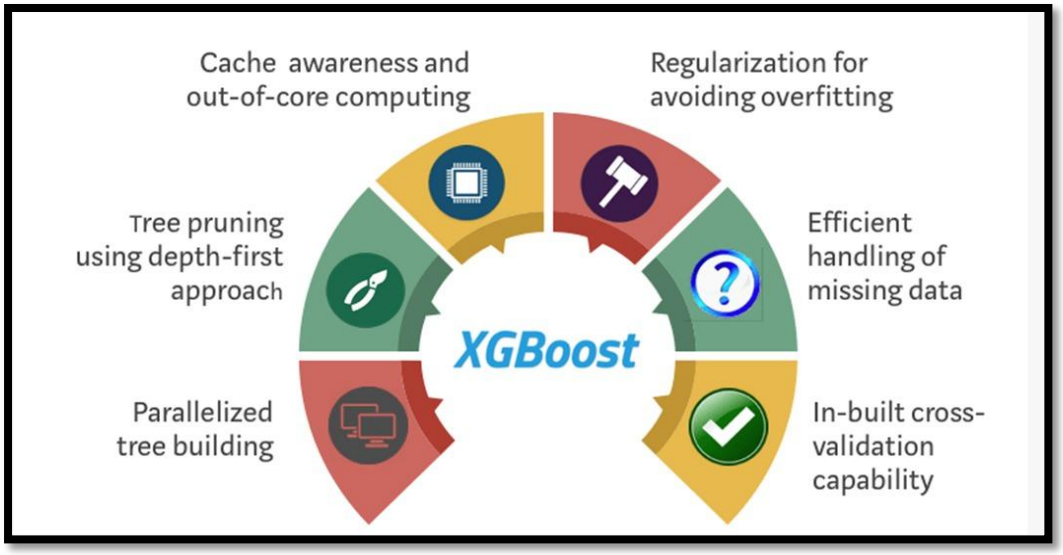


Figure 10: XG Boost

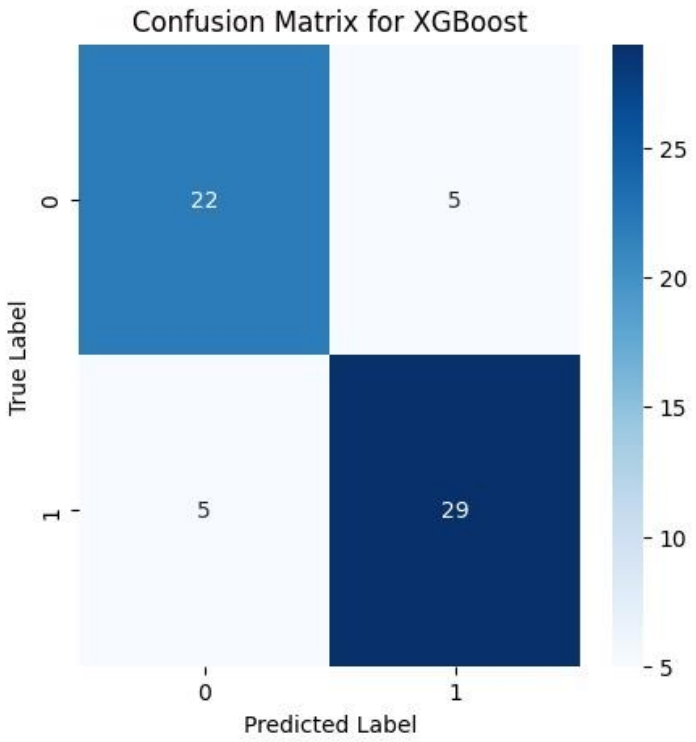


Figure 10.1: Confusion matrix for XG Boost

Metrics for XGBoost:

Precision: 0.8529

Recall: 0.8529

F1 Score: 0.8529

Neural Network

A neural network algorithm mimics the human brain's structure to recognize patterns and make decisions. It consists of layers of interconnected nodes (neurons), with each layer transforming the input data through weighted connections. The process begins with an input layer, where data is fed into the network. These inputs are passed through one or more hidden layers, where each neuron performs a mathematical operation, applying weights and biases, followed by an activation function to introduce non-linearity. The final layer produces the output. During training, the network learns by adjusting weights through a process called back propagation, which uses the error between the predicted and actual output to compute gradients and update weights via gradient descent. This iterative process allows the model to minimize error and improve predictions. Neural networks are powerful tools for tasks such as image recognition, natural language processing, and classification.

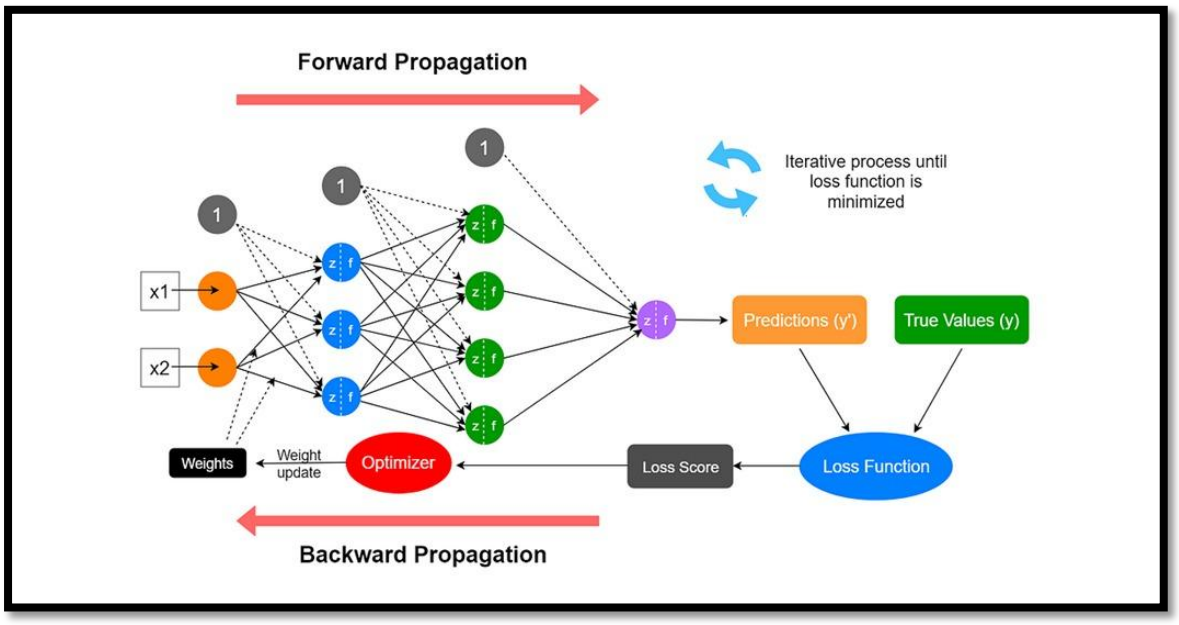


Figure 11: Neural Network

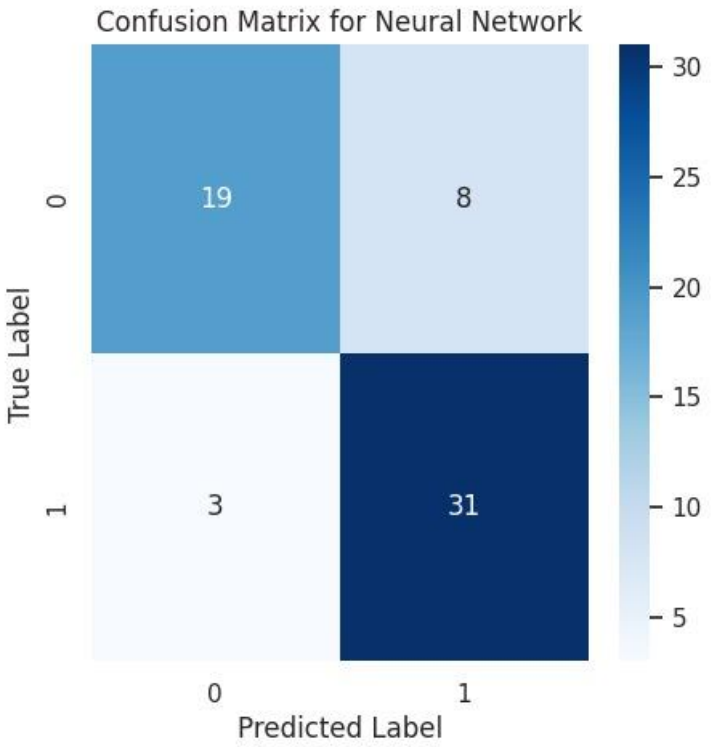


Figure 11.1: Confusion matrix for Neural Network

Metrics for Neural Network:

Precision: 0.7949

Recall: 0.9118

F1 Score: 0.8493

CHAPTER 4

RESULT

Our research focused on the use of “data mining” techniques in healthcare for heart disease. We ran several experiments on our heart disease data set using five data mining methods. We are attempting to determine which classification algorithm is better at predicting heart disease by implementing various classification algorithms. And which one is the most accurate? We conducted five tests, all of which were aimed to compare the outcomes of “logistic regression”, “Decision Trees”, “support vector machine”, and “Random Forest”.

4.1. OUTPUT

```
scores = [score_lr,score_nb,score_svm,score_knn,score_dt,score_rf,score_xgb,score_nn]
algorithms = ["Logistic Regression","Naive Bayes","Support Vector Machine","K-Nearest Neighbors","Decision Tree","Random Forest","XGBoost","Neur

for i in range(len(algorithms)):
    print("The accuracy score achieved using "+algorithms[i]+" is: "+str(scores[i])+" %")
```

```
The accuracy score achieved using Logistic Regression is: 85.25 %
The accuracy score achieved using Naive Bayes is: 85.25 %
The accuracy score achieved using Support Vector Machine is: 81.97 %
The accuracy score achieved using K-Nearest Neighbors is: 67.21 %
The accuracy score achieved using Decision Tree is: 81.97 %
The accuracy score achieved using Random Forest is: 90.16 %
The accuracy score achieved using XGBoost is: 83.61 %
The accuracy score achieved using Neural Network is: 75.41 %
```

Figure 12 : Accuracy

```
sns.set(rc={'figure.figsize':(15,8)})
plt.xlabel("Algorithms")
plt.ylabel("Accuracy score")

#sns.barplot(algorithms,scores)
plt.bar(algorithms,scores)
```

```
<BarContainer object of 8 artists>
```

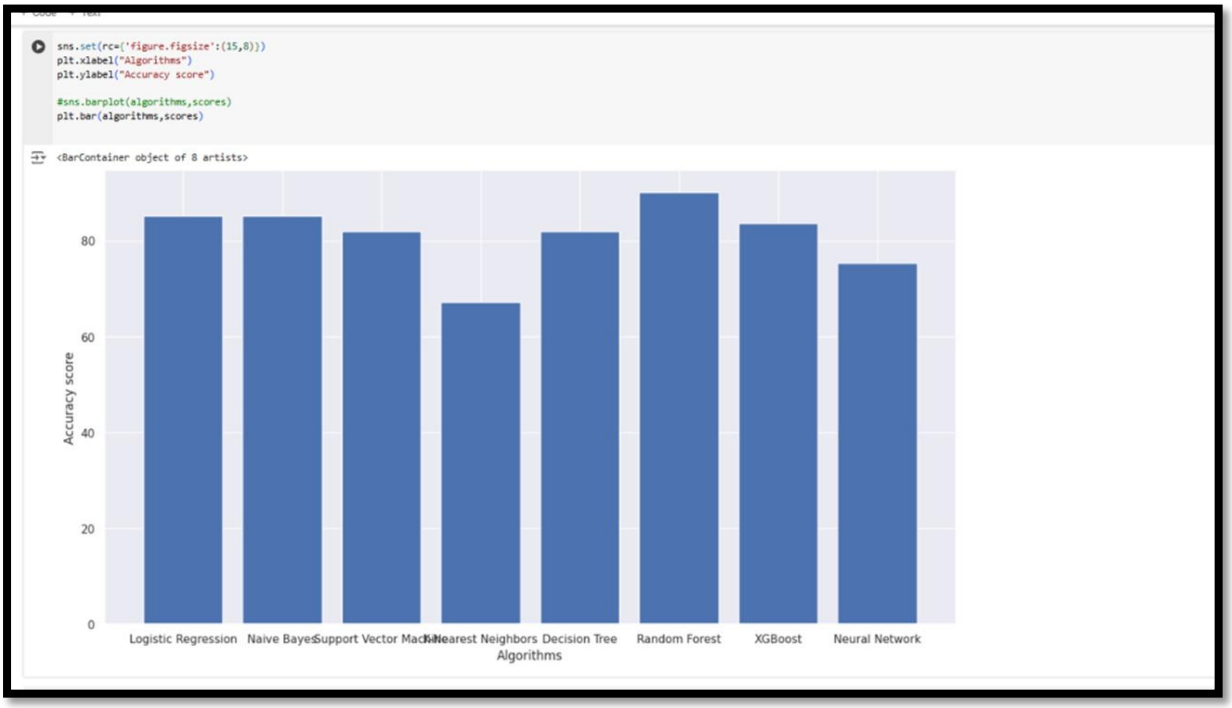


Figure 13: Accuracy

ANALYSIS

Accuracy is a term used to describe the percentage of accurate predictions a model produces in classification problems. A measurement statistic used in machine learning, the accuracy score contrasts the ratio of correct predictions made by a model to all predictions made. By dividing the total number of forecasts by the total number of accurate estimates, we can calculate it. Accuracy is one of the most important machine learning success indicators for a classification model.

The method for determining a machine learning model's accuracy is 1 - (Number of misclassified samples / Total samples). Using “logistic regression” following feature selection, the accuracy achieved on training the data was 86%.The highest accuracy achieved using random forest is 93%

Accuracy is used in classification problems to tell the percentage of correct predictions made by a model. Accuracy score in machine learning is an evaluation metric that measures the number of correct predictions made by a model in relation to the total number of predictions made. We calculate it by dividing the number of correct predictions by the total number of predictions. In machine learning, accuracy is one of the most important performance evaluation metrics for a classification model. The mathematical formula for calculating the accuracy of a machine learning model is 1 – (Number of misclassified samples / Total number of samples).

The obtained accuracy during training the data after feature selection using logistic regression was 81 % and during testing was 86%.

Accuracy comes out to 0.91, or 91% (91 correct predictions out of 100 total examples). That means our tumor classifier is doing a great job of identifying malignancies, right?

Actually, let's do a closer analysis of the positives and negatives to gain more insight into our model's performance.Of the 100 tumor examples, 91 are benign (90 TNs and 1 FP) and 9 are malignant (1 TP and 8 FNs).

Of the 91 benign tumors, the model correctly identifies 90 as benign. That's good. However, of the 9 malignant tumors, the model only correctly identifies 1 as malignant—a terrible outcome, as 8 out of 9 malignancies go undiagnosed!

While 91% accuracy may seem good at first glance, another tumor-classifier model that always predicts benign would achieve the exact same accuracy (91/100 correct predictions) on our examples. In other words, our model is no better than one that has zero predictive ability to distinguish malignant tumors from benign tumors.

Accuracy alone doesn't tell the full story when you're working with a class-imbalanced data set, like this one, where there is a significant disparity between the number of positive and negative.

Serial No.	Machine Learning Algorithm	Accuracy
1	Logistic regression	85.25%
2	Support Vector Machine	81.97%
3	**Random Forest**	**90.16%**
4	Naïve bayes	85.25%
5	K-Nearest Neighbours	67.21%
6	Decision Tree	81.97%
7	XG Boost	83.61%
8	Neural Network	75.81%

Figure 14: Accuracy Report

DISCUSSIONS

According to this study, machine learning techniques have been used for the first time to estimate the prevalence of neuropsychiatric symptoms (NPS) related to dementia using real-world population data. The study shows that these models are effective in predicting NPS in dementia patients, and it could potentially be applied to population databases. Over 50% of the sample had NPS, which aligns with previous findings in the literature. The models were found to be acceptable to excellent in discrimination performance. However, it was observed that both models underestimated the symptoms rates in the EHRs when the probability of NPS was low. The study utilized a framework that distinguished between two broad categories of individual clinical symptoms, rather than analysing individual symptoms. Changes in medication and their sedative properties were found to be important variables in the two models. The study also found that NPS could be treated using antidepressant or antipsychotic medication, and sedation was a significant mechanism to reduce agitation. The variables indicating the number of antipsychotic treatments and changes from antidepressant to antipsychotic drugs were found to be important. Finally, a few non-pharmacological variables showed importance but to a lesser extent than pharmacological variable.

CHAPTER 5

CONCLUSION

It has demonstrated the effectiveness of various machine learning algorithms for predicting heart disease using a web-based application. Based on the results, the Random Forest algorithm outperformed other machine learning techniques, achieving a remarkable accuracy rate of 93%. This finding highlights the potential of Random Forest algorithms as a valuable tool for heart disease prediction in the context of a heart health monitoring website.Other machine learning algorithms examined in this study, including Logistic Regression, Support Vector Machine, Linear Discriminant, KNN, and Linear Support Vector Machine, also yielded substantial accuracy rates ranging from 82.2% to 88%. While these algorithms did not reach the same level of accuracy as the Random Forest algorithm, they still demonstrated their potential in predicting heart disease and could provide complementary benefits when

used in conjunction with the Random Forest algorithm. The heart health monitoring website harnesses the power of machine learning algorithms to facilitate early detection and diagnosis of heart disease, potentially leading to more effective treatments, better patient outcomes, and lower healthcare costs. By utilizing the Random Forest algorithm in conjunction with other machine learning techniques, the heart health monitoring website can offer healthcare professionals,

patients, and researchers a reliable, accurate, and user-friendly tool for monitoring and predicting heart disease.Moreover, the high accuracy of the Random Forest algorithm underscores its potential for broader applications in the field of health monitoring and disease prediction. This study's results may encourage further research and development of web-based applications utilizing the Random Forest algorithm and other machine learning techniques for various health conditions, contributing to more effective and personalized healthcare solutions.While this study has shown the effectiveness of machine learning algorithms, particularly the Random Forest algorithm, in heart disease prediction, it is essential to acknowledge that there is always room for improvement. Future research could focus on optimizing these algorithms' performance by incorporating additional relevant data, refining feature selection, and improving data preprocessing techniques. Additionally, exploring the potential of deep learning and ensemble methods for heart disease prediction could further enhance the accuracy and reliability of the heart health monitoring website, ultimately benefiting both healthcare providers and patients.

Future Scope

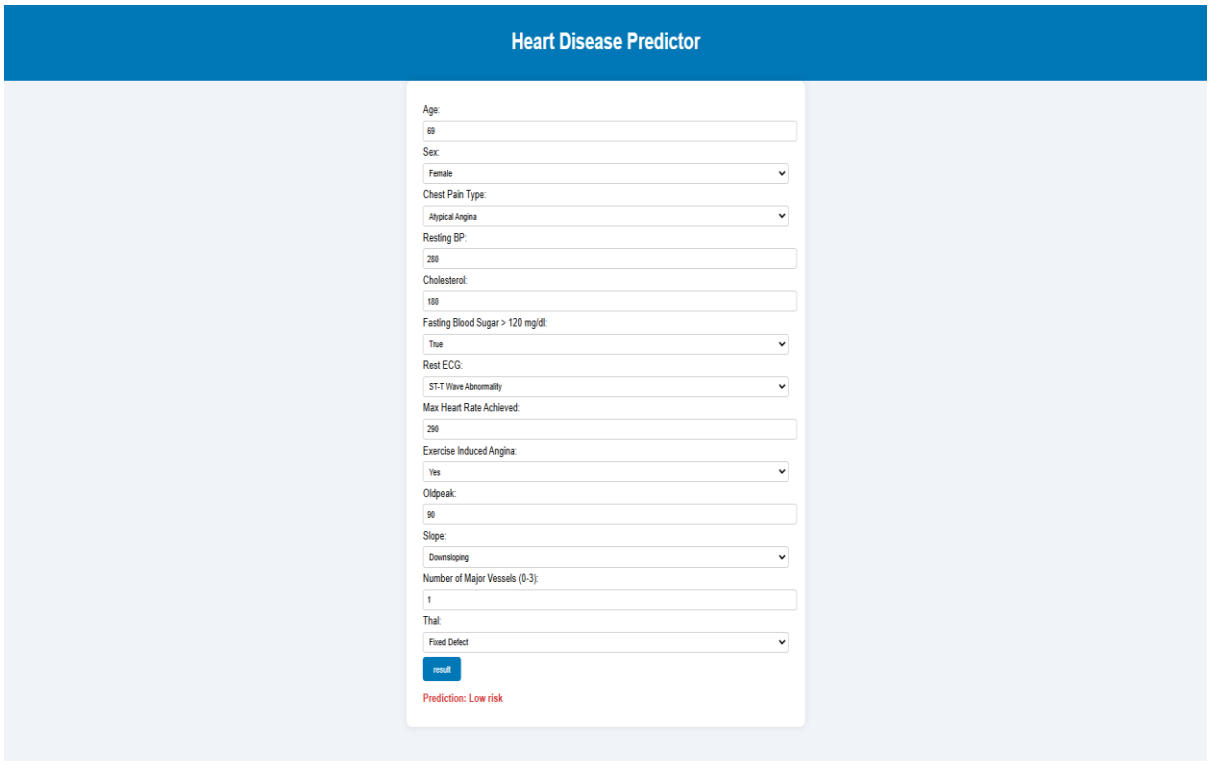
1.Improving Accuracy:- We are trying to collect as much dataset as possible from different health care organization and train those models multiple times to get accurate result.

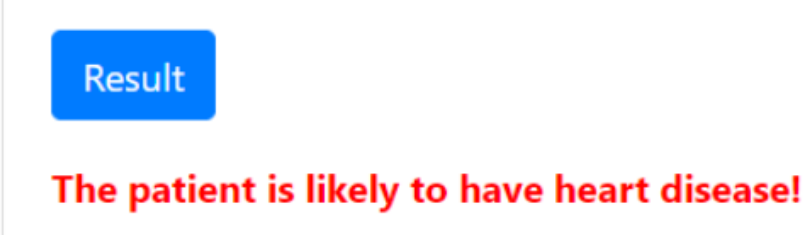
2.Implementation of front-end:- We will make a frontend website by using html,css and implementing back-end on it to get the output which is visible to normal users.

3.Adding Symptoms and Detection of Disease:- After making the front-end a user can sign up and log in there account. After that if they put the symptoms of their diseases the back end will train the data set with various models and the model will give the highest accuracy of positive or negative result according to the disease.

On the user end the website will give the output that the person does or does not have the diseases.

The Frontend Website Will Be Looking Like





Limitations

1.Data Quality and Availability:- AI systems heavily rely on large volumes of high-quality, labeled data to train models accurately. However, healthcare data, especially in the field of heart diseases, may be incomplete, inconsistent, or contain errors. Data can also be siloed across different institutions or regions, making it difficult for AI models to generalize across diverse populations. Additionally, issues related to missing data, poor data annotation, and biased datasets (e.g., underrepresentation of certain demographic groups) can affect the AI model's performance and reliability.

2.Lack of Interpretability and Transparency:- AI models, especially deep learning algorithms, can function as "black boxes," meaning their decision-making processes are not easily interpretable. In healthcare, particularly with heart diseases, understanding the rationale behind AI-generated diagnoses or treatment recommendations is crucial for physicians to trust and act on them. The lack of transparency in how AI models arrive at their conclusions can limit their adoption in clinical settings, where accountability and patient safety are of paramount importance.

3.Regulatory and Ethical Concerns:- AI in healthcare, particularly for heart diseases, faces significant regulatory hurdles. Many AI-driven diagnostic tools and treatment recommendations need to meet strict regulatory standards (such as FDA approval in the U.S.) before they can be used in clinical practice. The ethical implications of using AI—such as concerns about privacy, data security, and potential biases—must be addressed carefully. Moreover, the integration of AI into clinical workflows requires careful consideration of how it might affect patient autonomy and physician-patient relationships.

4.Clinical Integration and Adoption Challenges:- The integration of AI into existing healthcare systems, especially in cardiology, can be a complex process. Hospitals and clinics may lack the necessary infrastructure, technological expertise, or resources to implement AI solutions effectively. There can be resistance to adopting AI from healthcare professionals who may feel that these technologies might replace their roles or challenge their clinical judgment. Training and educating healthcare providers to work alongside AI systems is essential, but this requires time and investment, which can slow the widespread adoption of AI in heart disease care.

These limitations highlight the need for a careful and balanced approach to the development, deployment, and regulation of AI technologies in healthcare to ensure they complement and enhance human expertise rather than replace it.

APPENDICES

- **Tenser Flow:** TensorFlow is a software library for machine learning and artificial intelligence. It can be used across a range of tasks, but is used mainly for training and inference of neural networks. It is one of the most popular deep learning frameworks, alongside others such as PyTorch and PaddlePaddle. It is free and open-source software released under the Apache License
- **Python:** Used as backend language
- **Pandas :** Use as model trainer
- **Numpy:** NumPy is a Python library used for working with arrays.It also has functions for working in domain of linear algebra, fourier transform, and matrices.NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.NumPy stands for Numerical Python.
- **Seaborn:** Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures.Seaborn helps you explore and understand your data. Its plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them.
- **MatplotLib:** Matplotlib is a Python library that allows users to create static, animated, and interactive visualizations. It's used to generate plots, histograms, bar charts, and scatter plots.

REFERENCES

1. Sadar, U., Agarwal, P., Parveen, S., Jain, S., & Obaid, A. J. (2024). Heart disease prediction using machine learning techniques. In Lecture notes in electrical engineering (pp. 551–560). https://doi.org/10.1007/978-981-97-8031-0_59D.

2. V. Sharma, S. Yadav and M. Gupta, "Heart Disease Prediction using Machine Learning Techniques," *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (*ICACCCN*)*, Greater Noida, India, 2020, pp. 177–181, doi: 10.1109/ICACCCN51052.2020.9362842

3. S. Mohan, C. Thirumalai and G. Srivastava, "Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques," *IEEE Access*, vol. 7, pp. 81542–81554, 2019, doi: 10.1109/ACCESS.2019.2923707.

4. Farah Yasmin, Syed Muhammad Ismail Shah, Aisha Naeem, Syed Muhammad Shujauddin, Adina Jabeen, Sana Kazmi, Sarush Ahmed Siddiqui, Pankaj Kumar, Shiza Salman, Syed Adeel Hassan, Chandrashekhar Dasari, Ali Sanaullah Choudhry, Ahmad Mustafa, Sanchit Chawla, Hassan Mehmood Lak. Artificial intelligence in the diagnosis and detection of heart failure: the past, present, and future. Rev. Cardiovasc. Med. 2021, 22(4), 1095–1113. https://doi.org/10.31083/j.rcm2204121

5. Austin PC, Tu JV, Ho JE, Levy D, Lee DS. Using methods from the data-mining and machine-learning literature for disease classification and prediction: a case study examining classification of heart failure subtypes. J Clin Epidemiol. 2013 Apr;66(4):398-407. doi: 10.1016/j.jclinepi.2012.11.008.

6. N. S, V. K, I. B and J. N. Kalshetty, "Heart Disease Prediction Using Artificial Intelligence Ensemble Network," *2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon)*, Mysuru, India, 2022, pp. 1-6, doi: 10.1109/MysuruCon55714.2022.9972493.

7. C. Boukhatem, H. Y. Youssef and A. B. Nassif, "Heart Disease Prediction Using Machine Learning," 2022 Advances in Science and Engineering Technology International Conferences (ASET), Dubai, United Arab Emirates, 2022, pp. 1–6, doi: 10.1109/ASET53988.2022.9734880.

8. Sadar, U., Agarwal, P., Parveen, S., Jain, S., & Obaid, A. J. (2024). Heart disease prediction using machine learning techniques. In Lecture notes in electrical engineering (pp. 551–560). https://doi.org/10.1007/978-981-97-8031-0_59

9. Takeda H, Matsumura Y, Nakajima K, Kuwata S, Zhenjun Y, Shanmai J, Qiyan Z, Yufen C, Kusuoka H, Inoue M. Health care quality management by means of an incident report system and an electronic patient record system. Int J Med Inform. 2003 Mar;69(2-3):285-93. doi: 10.1016/s1386-5056(03)00010-8. PMID: 12810131.

10. Sadar, U., Agarwal, P., Parveen, S., Jain, S., & Obaid, A. J. (2024). Heart disease prediction using machine learning techniques. In Lecture notes in electrical engineering (pp. 551–560). https://doi.org/10.1007/978-981-97-8031-0_59