Heart Disease Prediction using Machine Learning

Major Project Report submitted in partial fulfilment of the

requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted By

Soumyaranjan Muduli – 2101320089

Sourav Kumar Patra - 2101320090

Prakash Satapathy – 2101320064

Aryan Biswal – 2101320042

Sahil Malla - 2101320078

Sanjit Kumar Danta -2101320079

Under the supervision of

Asst. Prof. Md Shahil Khan

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

For the Academic year 2021-2025

ARYAN INSTITUTE OF ENGINEERING AND TECHNOLOGY

ARYABIHAR, BHUBANESWAR, ODISHA, PIN-752050

Affiliated to

BIJU PATNAIK UNIVERSITY OF TECHNOLOGY ROURKELA

ODISHA

APRIL-2025

ARYAN INSTITUTE OF ENGINEERING & TECHNOLOGY

CERTIFICATE

This to certify that the work which is being presented in the project **title Heart Disease Prediction using Machine Learning** in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science And Engineering and submitted in the Department of Computer Science And Engineering, Aryan Institute of Engineering and Technology, Bhubaneswar is an authentic record of work carried out by **Soumyaranjan Muduli – 2101320089, Sourav Kumar Patra – 2101320090 Prakash Satapathy – 2101320064, Aryan Biswal – 2101320042, Sahil Malla – 2101320078 and Sanjit Kumar Danta -2101320079** under the supervision of **Asst. Prof. Md Shahil Khan**, Computer Science And Engineering Department.

The above statement made is true to the best of our knowledge and belief

Soumyaranjan Muduli – 2101320089

Sourav Kumar Patra - 2101320090

Prakash Satapathy – 2101320064

Aryan Biswal – 2101320042

Sahil Malla – 2101320078

Sanjit Kumar Danta -2101320079

Place:

Date:

Project Guide **Head of the Department**

CSE

Principal

AIET, BBSR **External Examiner**

DECLARATION

We hereby declare that the project entitled **Heart Disease Prediction using Machine Learning** carried out by us under the guidance of **Asst. Prof. Md Shahil Khan** is submitted to Biju Patnaik University of Technology, Rourkela, Odisha, in the partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering.

The results embodied in this report have not been copied from any source. The results embodied this report have not been submitted to any other University or Institute for the award of any other degree.

Soumyaranjan Muduli – 2101320089

Sourav Kumar Patra - 2101320090

Prakash Satapathy – 2101320064

Aryan Biswal – 2101320042

Sahil Malla – 2101320078

Sanjit Kumar Danta -2101320079

Place:

Date:

ACKNOWLEDGEMENT

I would like to express my gratitude to all the people behind the screen who helped me to transform an idea into a real application.

I profoundly thank **Ass. Prof. Vidya Mohanty** Head of the Department of Computer Science And Engineering who has been excellent guide and also a great source of inspiration to my work. I would like to thank my internal guide **Asst. Prof. Md Shahil Khan** for his technical guidance, constant encouragement and support in carrying out my project at college. The satisfaction and euphoria that accompany the successful completion of the task would be great but incomplete without the mention of the people who made it possible with their constant guidance and encouragement crowns all the efforts with success.

In this context, I would like to thank all the other staff members, both teaching and non-teaching, who have extended their timely help and eased my task

.

Soumyaranjan Muduli – 2101320089

Sourav Kumar Patra – 2101320090

Prakash Satapathy – 2101320064

Aryan Biswal – 2101320042

Sahil Malla – 2101320078

Sanjit Kumar Danta -2101320079

Place:

Date:

ARYAN INSTITUTE OF ENGINEERING & TECHNOLOGY

Vision of the Institution

To become a leading engineering institution of the state by impacting quality technical education at affordable costs to create skilled and motivated engineers to serve the technological requirements of society in different ways.

Mission of the Institution

M1:	To impart contemporary technical education and skills to students of different socio-economic background.
M2:	To equip students with analytical learning and real-life problem solving.
M3:	To make learning a continuous endeavor compatible with market needs.
M4:	To promote the spirit of leadership, entrepreneurship, innovation and ethics

Vision of the Department

To create a distinctive culture, that could enable students and faculty members collaboratively approach to advance their knowledge about recent advancements in the domains of Computer Science and Engineering, such as Machine Learning, Data Science, IoT etc., and develop effective, implementable and environment friendly solutions towards solving the problems of present and future society.

Mission of the Department

M1:	To develop a distinctive environment where student, teacher can learn and acquire necessary knowledge and skills through effective collaboration and holistic interaction.
M2:	To create a culture of research and innovation through necessary communication with the premier institutions and contribute to the research community through high-quality publications in reputed journals.
M3:	To create a conducive ambience where students and faculty member can engage themselves for developing effective solution in the area of Machine Learning, Artificial Intelligence, Data Science, Deep Learning.
M4:	To generate a pool of professionals and entrepreneurs with the ability to address the industry and social problems through effective communication skills and should be able to provide required weightage towards societal and sustainable issue.

ARYAN INSTITUTE OF ENGINEERING & TECHNOLOGY

Program Educational Objectives (PEOs)

PEO1: Establish successful careers in Electrical Engineering and related fields by offering creative and practical solutions

PEO2: Engage in continuous learning through cutting-edge technologies for solving societal problems using logical and innovative approaches in decision-making

PEO3: Become an entrepreneur, work for a company that conducts research and development.

Program Specific Outcomes (PSOs):

Engineering Graduates will be able to:

PSO1 Apply the fundamentals of science and technology to identify, formulate, design and investigate complex engineering problems of electric circuits, control system, power electronics, electric drives and power system.

PSO2 Ability to model, simulate and assess electrical systems and components using software and hardware tools

PSO3 Empowering to socially acceptable technical solutions and relevant methodologies for sustainable development to current electrical engineering difficulties.

ARYAN INSTITUTE OF ENGINEERING & TECHNOLOGY

Program Outcomes

PO1	**Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	**Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	**Design / development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	**Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	**Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO6	**The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	**Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	**Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	**Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	**Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	**Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	**Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

MSME : Micro Small And Medium Enterprise

SVM : Support Vector Machine

SDLC : Software Development Life Cycle

ANN : Artificial Neural Network

Gaussian NB : Gaussian Naïve Bayes

IDE: Integrated Development Environment

REFCV: Recursive Feature Elimination using Cross-Validation

CV: Cross Validation

RFE: Recursive Feature Elimination

ABSTRACT

Heart disease is one of the leading causes of mortality worldwide, necessitating early and accurate prediction to prevent severe outcomes. This project explores the application of machine learning algorithms, including Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine (SVM), for predicting heart disease based on medical and demographic data.

The study uses a publicly available dataset comprising patient information such as age, cholesterol levels, blood pressure, and other clinical attributes. Data pre-processing steps like handling missing values, feature scaling, and encoding categorical variables were implemented to ensure data quality. Each algorithm was trained and evaluated using metrics like accuracy, precision, recall, F1 score, and the area under the receiver operating characteristic curve (AUC-ROC).

The results demonstrate that ensemble methods, particularly Random Forest, achieved the highest predictive accuracy, whereas Support Vector Machine showed competitive performance in certain scenarios. Logistic Regression and Decision Tree provided interpretable models, with moderate performance. The findings underline the importance of selecting appropriate algorithms based on specific requirements such as accuracy, interpretability, and computational complexity.

This project contributes to the growing field of machine learning in healthcare by providing a comparative analysis of widely-used algorithms for heart disease prediction. Future work can focus on integrating deep learning models and larger datasets to enhance prediction accuracy further.

Keywords:

- Heart Disease Detection
- Deep Learning
- Logistic Regression
- Decision Tree
- Random Forest
- KNN

CHAPTER 1

INTRODUCTION

1.1 Background

Heart disease remains a significant public health concern, accounting for a substantial portion of global mortality rates. According to the World Health Organization (WHO), cardiovascular diseases (CVDs) are the leading cause of death worldwide, claiming approximately 17.9 million lives annually. These diseases often arise due to factors such as unhealthy diets, physical inactivity, tobacco use, and harmful use of alcohol. Early diagnosis and timely intervention can significantly reduce mortality rates and improve patient outcomes.

The advent of machine learning has introduced new possibilities for medical diagnostics. By analyzing vast amounts of data, machine learning algorithms can uncover hidden patterns and relationships that traditional statistical methods might overlook. This capability is particularly valuable in predicting heart disease, where a combination of demographic, clinical, and lifestyle factors influences risk.

1.2 Importance of Heart Disease Prediction

Accurate prediction of heart disease can lead to early intervention, reducing the risk of complications and mortality. Machine learning models can assist healthcare professionals by providing a decision-support system, enabling them to identify high-risk patients efficiently. Additionally, predictive models can aid in resource allocation, optimizing medical resources by focusing on individuals most likely to benefit from intervention.

The integration of machine learning in healthcare also empowers patients with insights into their health risks, encouraging lifestyle modifications and adherence to preventive measures. The importance of such predictive systems has grown as the healthcare industry increasingly emphasizes preventive care and personalized medicine.

1.3 Objectives of the Project

The primary objectives of this project are as follows:

1. To develop machine learning models using Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine (SVM) for predicting heart disease.
2. To preprocess and analyze a heart disease dataset to ensure data quality and suitability for machine learning algorithms.

3. To evaluate and compare the performance of the selected algorithms based on metrics such as accuracy, precision, recall, F1 score, and AUC-ROC.
4. To provide insights into the strengths and weaknesses of each algorithm in the context of heart disease prediction.
5. To suggest future directions for improving predictive accuracy and practical implementation in healthcare.

1.4 Scope of the Study

This project focuses on predicting heart disease using machine learning techniques applied to a publicly available dataset. The study encompasses:

- A comparative analysis of four machine learning algorithms: Logistic Regression, Decision Tree, Random Forest, and SVM.
- The implementation of data preprocessing techniques to address challenges such as missing values, imbalanced datasets, and feature scaling.
- The evaluation of model performance using standard metrics and visualization techniques.

The project is limited to the dataset used, which may not represent the full diversity of real-world populations. While the study emphasizes accuracy and reliability, the clinical deployment of these models would require further validation and ethical considerations.

Future extensions of this work could include the integration of advanced machine learning methods, such as neural networks, and the incorporation of larger, more diverse datasets for enhanced generalizability.

CHAPTER 2

LITERATURE REVIEW

2.1 LITERATURE REVIEW

SL No	Authors	Publication Title	Date	Problem Statement	Proposed Methodology	Results
1	**Raniya R. Sarra et al**	Enhanced Heart Disease Prediction Based on Machine Learning and χ^2 Statistical Optimal Feature Selection Model	Sep 29, 2022	Need for accurate heart disease prognosis for effective treatment.	Proposed a heart disease classification model using Support Vector Machine (SVM) with χ^2 statistical feature selection to improve prediction accuracy.	Increased accuracy from 85.29% to 89.7% and reduced componential load by half.
2	**María Teresa García-Ordás et al.**	Heart disease risk prediction using deep learning techniques with feature augmentation	Feb 8, 2024	Difficulty for experts to evaluate numerous variables influencing heart disease risk.	Utilized deep learning methods combined with feature augmentation techniques to assess patient risk for cardiovascular disease.	Achieved a precision of 90%, outperforming other state-of-the-art methods by 4.4%.
3	**Shishir Rao et al.**	An explainable Transformer-based deep learning model for the prediction of incident heart failure	Jan 27, 2021	Challenges in predicting complex chronic conditions like heart failure using unexplainable deep learning models.	Developed a novel Transformer deep-learning model utilizing longitudinal electronic health records for accurate and explainable prediction.	Achieved an AUC of 0.93 and outperformed existing deep learning models.

4	**Achyut Tiwari et al.**	Ensemble Framework for Cardiovascular Disease Prediction	Jun 16, 2023	Necessity for early and accurate diagnosis to prevent heart disease progression.	Developed a stacked ensemble classifier using machine learning algorithms including ExtraTrees, Random Forest, and XGBoost.	Attained an accuracy of 92.34%, higher than existing literature.
5	**Md. Shaheenur Islam Sumon et al.**	CardioTabNet: A Novel Hybrid Transformer Model for Heart Disease Prediction using Tabular Medical Data	Mar 22, 2025	Need for early detection and prediction of cardiovascular diseases to reduce morbidity and mortality.	Proposed CardioTabNet, a hybrid Transformer model utilizing tabular medical data with feature ranking and machine learning classifiers.	Achieved 94.1% accuracy and 95.0% AUC using a hyper-tuned ExtraTree classifier.
6	**Azka Mir et al.**	A novel approach for the effective prediction of cardiovascular disease using applied artificial intelligence techniques	Jul 11, 2024	Need for effective prediction frameworks for cardiovascular disease using machine learning.	Utilized machine learning algorithms on multiple datasets with preprocessing, feature selection, and classification methods.	Achieved highest accuracy of 99.48% on Heart Statlog dataset and varying accuracies on other datasets.
7	**A. Mounika Rajeswari et al.**	Prediction of Heart Disease by Machine Learning	Apr 27, 2023	Difficulty in forecasting heart disease due to its complexity, requiring experience and advanced knowledge.	Developed an automated system for diagnosing heart disease using machine learning techniques to improve medical effectiveness and reduce costs.	Specific accuracy metrics not provided in the summary.

2.2 LITERATURE SURVEY

1. **Raniya R. Sarra et al** (2022) making an accurate and timely diagnosis of cardiac disease is critical for preventing and treating heart failure. The accuracy of results produced by traditional machine learning (ML) algorithms is satisfactory. On the other hand, deep learning algorithms result in higher prediction accuracy. In this study, we used an artificial neural network (ANN) model to construct a deep learning diagnosis system for heart disease prediction. The developed ANN prediction model achieved 93.44% accuracy, which is 7.5% higher than a traditional ML model support vector machine (SVM). Additionally, using a simpler neural network reduced the time taken for training and classification to less than a minute

2. **María Teresa García Ordás et al** (2024) María Teresa García-Ordás and her colleagues have conducted research on heart disease risk prediction using deep learning techniques combined with feature augmentation. Their study highlights how cardiovascular diseases are among the leading causes of death worldwide and emphasizes the importance of early detection. By leveraging deep learning methods, their approach improves prediction accuracy, outperforming other state-of-the-art techniques by 4.4%, achieving a precision of 90%.

3. **Shishir Rao et al. (2021).** An Explainable Transformer-Based Deep Learning Model for the Prediction of Incident Heart Failure**.** Published January 27, 2021.
   Problem Addressed:
   Challenges in predicting complex chronic conditions like heart failure due to the use of opaque, unexplainable deep learning models.
   Proposed Solution:
   Developed a novel Transformer-based deep learning model that utilizes longitudinal electronic health records (EHRs) for accurate and explainable prediction of heart failure.
   Key Results:
   Achieved an Area Under the Curve (AUC) of 0.93, indicating excellent prediction performance.
   The model outperformed existing deep learning models in heart failure prediction.
   Incorporated explainability by identifying key features influencing the prediction, improving clinical trust and interpretability.

4. **Achyut Tiwari** (2023) and his collaborators have made significant contributions to the field of cardiovascular disease (CVD) prediction through machine learning. Their notable work, "Ensemble Framework for Cardiovascular Disease Prediction," presents a robust approach to early and accurate diagnosis of heart diseases

   Objective: To develop a predictive model for early detection of cardiovascular diseases using ensemble machine learning techniques.

   Result Achieved an accuracy of **92.34%**, surpassing existing models in the literature. Demonstrated improved performance over individual classifiers like Random Forest, Support Vector Machines (SVM), Naive Bayes, Logistic Regression, and K-Nearest Neighbors (KNN) in terms of precision-recall AUC

5. **Md. Shaheenur Islam** Sumon is a researcher affiliated with Qatar University, specializing in machine learning, artificial intelligence, and biomedical data analysis. His work primarily focuses on applying advanced computational techniques to healthcare challenges, particularly in cardiovascular disease (CVD) prediction.

   CardioTabNet: A Novel Hybrid Transformer Model for Heart Disease Prediction Using Tabular Medical Data

   Overview: Introduces CardioTabNet, a hybrid transformer model leveraging tabular medical data for heart disease prediction. The model utilizes a tab transformer for feature extraction and ranking, followed by machine learning classifiers.

   Results: Achieved an average accuracy of 94.1% and an AUC of 95.0%.

   Significance: Demonstrates the potential of transformer-based models in processing structured medical data for accurate disease prediction.

   An Integrated Machine Learning Framework for Effective Prediction of Cardiovascular Diseases (MaLCaDD)

   Overview: Proposes the MaLCaDD framework, addressing data preprocessing challenges like missing values and class imbalance using techniques like mean replacement and SMOTE. Employs feature selection and an ensemble of Logistic Regression and K-Nearest Neighbors for prediction.

   Validation: Tested on three benchmark datasets, showcasing improved prediction accuracy. Significance: Highlights the importance of comprehensive data preprocessing and ensemble methods in enhancing CVD prediction models.

Deep Learning Technique for Congenital Heart Disease Detection Using Stacking-Based CNN-LSTM Models from Fetal Echocardiogram: A Pilot Study

Overview: Develops a deep learning model combining Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) networks to detect congenital heart diseases from fetal echocardiograms.

Results: Demonstrated promising accuracy in early detection, aiding in timely medical interventions.

6. **Azka Mir et al.** Azka Mir is a researcher whose work spans applied artificial intelligence (AI) in healthcare, epidemiology, and socioeconomic studies. Her contributions include developing machine learning models for disease prediction and analyzing public health data

   A Novel Approach for the Effective Prediction of Cardiovascular Disease Using Applied Artificial Intelligence Techniques

   Authors: Azka Mir, Attique Ur Rehman, Tahir Muhammad Ali, Sabeen Javaid, Maram Fahaad Almufareh, Mamoona Humayun, Momina Shaheen

   Published in: ESC Heart Failure, 2024

   Overview: This study developed a machine learning framework using five datasets (Heart UCI, Stroke, Heart Statlog, Framingham, and Coronary Heart) to predict cardiovascular diseases. The approach involved data preprocessing, feature selection, and classification using algorithms like Random Forest with AdBoost, achieving high accuracy rates.

   An Intelligent Technique for the Effective Prediction of Monkeypox Outbreak

   Authors: Azka Mir, Attique Ur Rehman, Sabeen Javaid, Tahir Muhammad Ali

   Published in: 2023 3rd International Conference on Artificial Intelligence (ICAI)

   Overview: The paper presents a machine learning-based approach to predict monkeypox outbreaks by classifying cases as confirmed, discarded, or suspected, aiding in early intervention strategies.

   The Impact of Prenatal Exposure to Ramadan on Child Anthropometric Outcomes in Pakistan

   Authors: Azka Sarosh Mir, Theresa Thompson Chaudhry

   Published in: Maternal and Child Health Journal, 2021

   Overview: This study analyzes the effects of prenatal exposure to Ramadan fasting on child growth metrics like stunting and underweight, using data from the Multiple Indicator Cluster Survey (MICS) in Pakistan.

7. **A. Mounika Rajeswari** is an emerging researcher in the field of machine learning applications for healthcare, particularly in cardiovascular disease (CVD) prediction. Her work focuses on leveraging data mining and ensemble learning techniques to enhance diagnostic accuracy and early detection of heart diseases.

   Prediction of Heart Disease by Machine Learning

   Authors: A. Mounika Rajeswari, D. V. Sathwik Reddy

   Published in: International Journal of Engineering Research & Technology (IJERT), Volume 12, Issue 04, April 2023

   Overview: This study employs machine learning techniques to predict heart disease using the Cleveland heart disease dataset from the UCI Machine Learning Repository. The focus is on identifying patterns and relationships within the data to assess the risk level of patients.

   Prediction of Heart Disease using Ensemble Learning

   Authors: A. Mounika Rajeswari, D. V. Sathwik Reddy

   Published in: Indian Journal of Science and Technology

   Overview: This paper proposes a Bagging ensemble method to predict heart disease at early stages. The study applies various machine learning algorithms, including Decision Tree, Naïve Bayes, Random Forest, and Support Vector Machine (SVM), to the Cleveland dataset. The ensemble approach aims to increase prediction accuracy by aggregating the results of individual models

CHAPTER 3

METHODOLOGY

3.1 DATE FLOW DIAGRAM

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.

2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.

4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

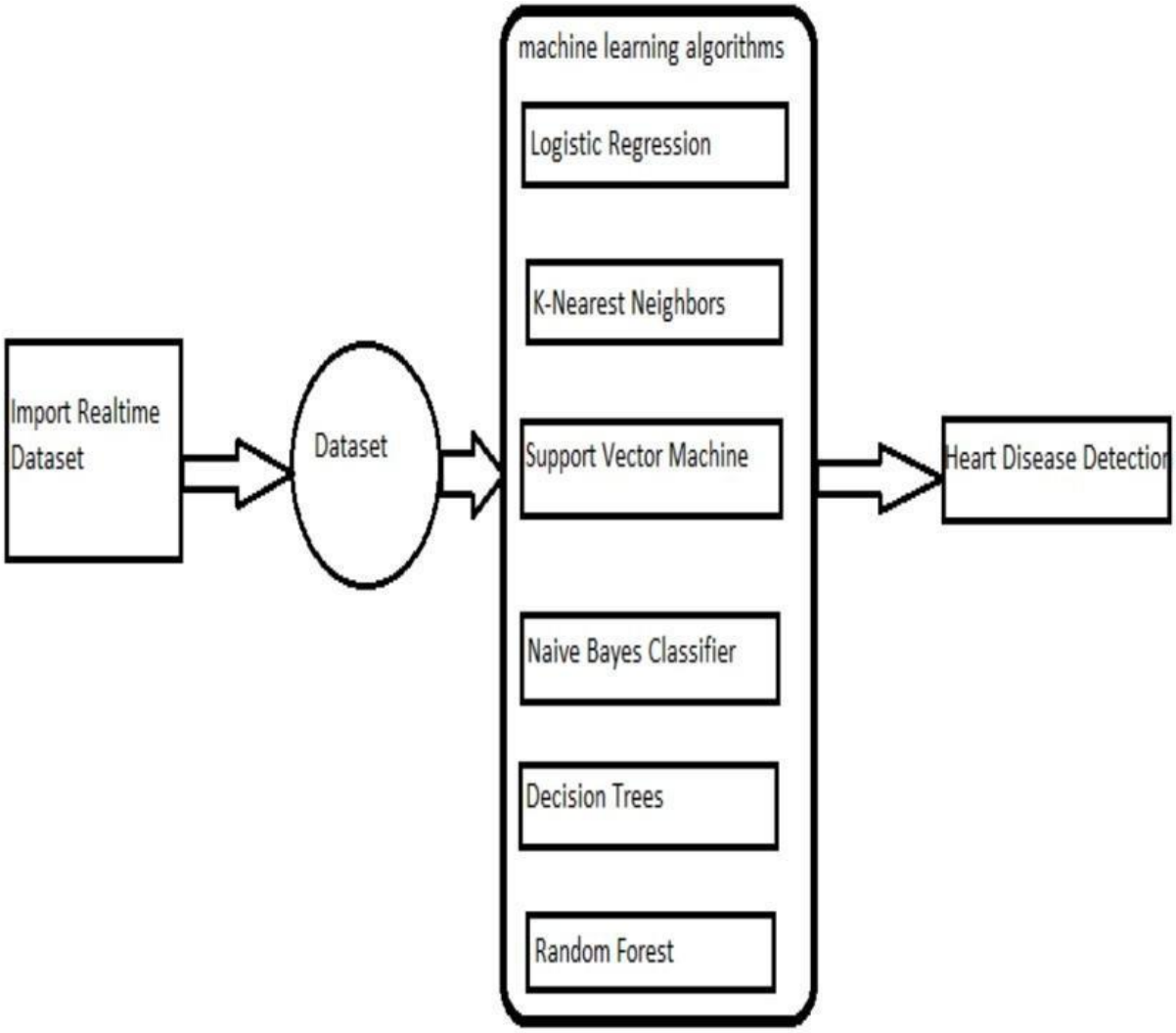


Fig 1: Date Flow Diagram

3.2 USE CASE DIAGRAM

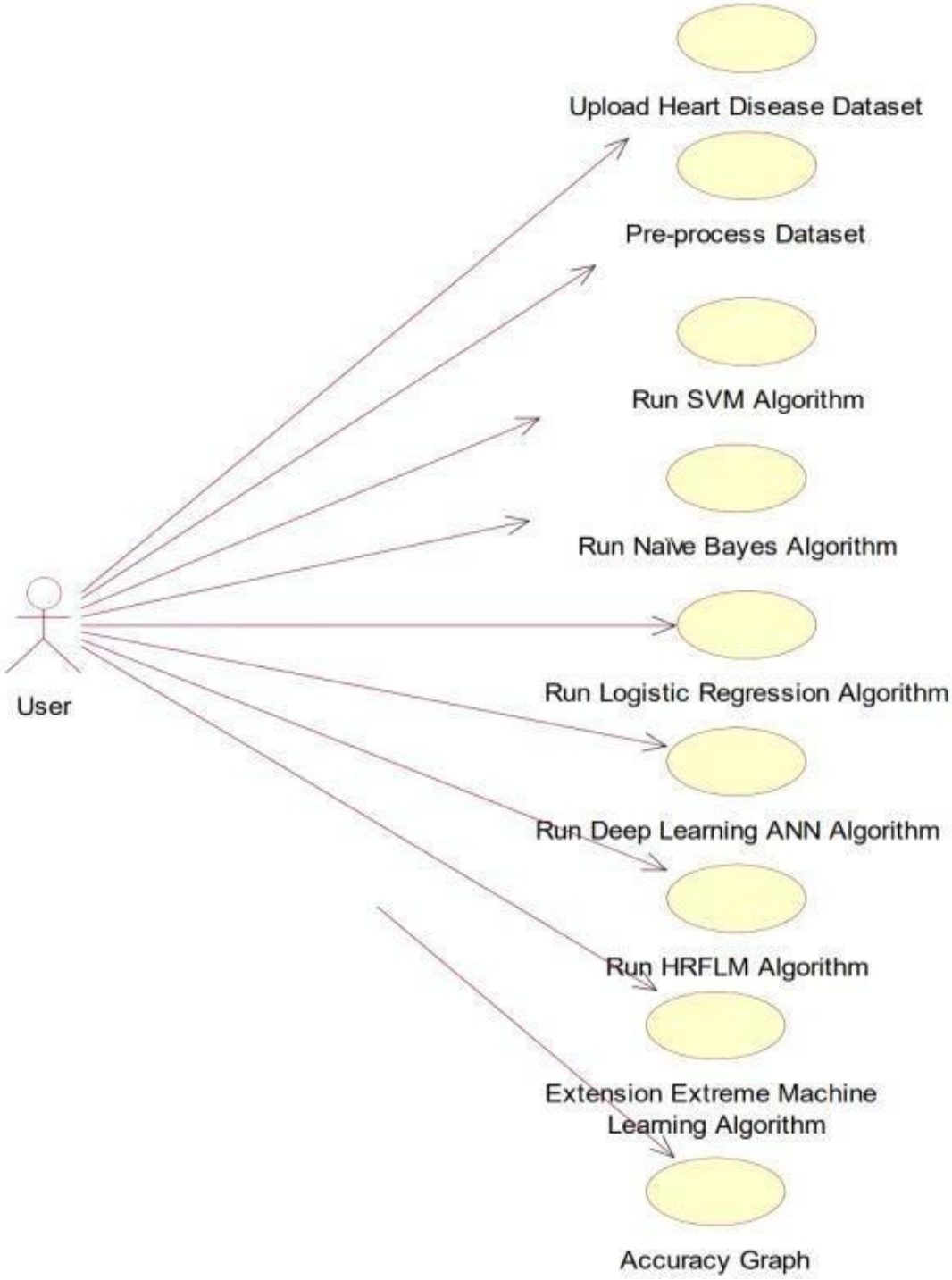


Fig 2: Use Case Diagram

3.3 CLASS DIAGRAM

Class diagrams are widely used to describe the types of objects in a system and their relationships. Class diagrams model class structure and contents using design elements such as classes, packages and objects. Class diagrams describe three different perspectives when designing a system, conceptual, specification, and implementation. These perspectives become evident as the diagram is created and help solidify the design. Class diagrams are arguably the most used UML diagram type. It is the main building block of any object oriented solution. It shows the classes in a system, attributes and operations of each class and the relationship between each class. In most modeling tools a class has three parts, name at the top, attributes in the middle and operations or methods at the bottom. In large systems with many classes related classes are grouped together to create class diagrams. Different relationships between diagrams are show by different types of Arrows. Below is a image of a class diagram. Follow the link for more class diagram examples.

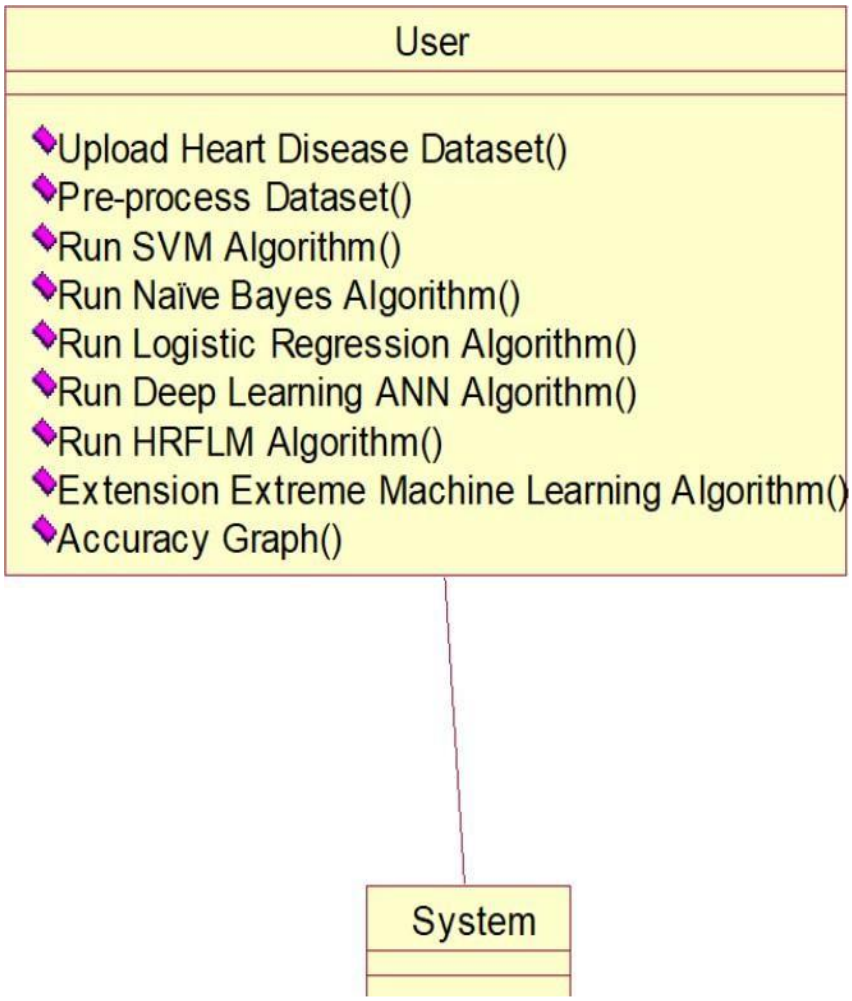


Fig 3: Class Diagram

3.4 DATA COLLECTION

Dataset Description

The dataset used in this project is the **UCI Heart Disease dataset**, a well-known resource for machine learning research in healthcare. This dataset contains records of patients, including demographic, clinical, and lifestyle attributes that influence heart disease risk.

- **Source**: UCI Machine Learning Repository
- **Number of Instances**: 303
- **Number of Features**: 14 (13 independent variables and 1 target variable)

```
#create a dataframe
df=pd.read_csv("heart.csv")
df.head(10)
```

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
5	58	0	0	100	248	0	0	122	0	1.0	1	0	2	1
6	58	1	0	114	318	0	2	140	0	4.4	0	3	1	0
7	55	1	0	160	289	0	0	145	1	0.8	1	1	3	0
8	46	1	0	120	249	0	0	144	0	0.8	2	0	3	0
9	54	1	0	122	286	0	0	116	1	3.2	1	2	2	0

Tab 01: Dataset Table

Features and Labels

The dataset includes the following features:

1. **Age**: Age of the patient (numerical).
2. **Sex**: Gender of the patient (1 = male, 0 = female).
3. **Chest Pain Type (cp)**: Types of chest pain experienced (categorical: 0 to 3).
4. **Resting Blood Pressure (trestbps)**: Resting blood pressure in mm Hg (numerical).
5. **Serum Cholesterol (chol)**: Serum cholesterol level in mg/dl (numerical).
6. **Fasting Blood Sugar (fbs)**: Fasting blood sugar > 120 mg/dl (1 = true, 0 = false).

7. **Resting ECG Results (restecg)**: Results of resting electrocardiographic tests (categorical: 0 to 2).
8. **Maximum Heart Rate Achieved (thalach)**: Maximum heart rate achieved (numerical).
9. **Exercise-Induced Angina (exang)**: Exercise-induced angina (1 = yes, 0 = no).
10. **ST Depression (oldpeak)**: ST depression induced by exercise relative to rest (numerical).
11. **Slope of ST Segment (slope)**: Slope of the peak exercise ST segment (categorical: 0 to 2).
12. **Number of Major Vessels (ca)**: Number of major vessels colored by fluoroscopy (numerical).
13. **Thalassemia (thal)**: Thalassemia status (categorical: 0 to 3).

3.5 CHECKING THE SKEWNESS OF THE DATA

For checking the attribute values and determining the skewness of the data (the asymmetry of a distribution), many distribution plots are plotted so that some interpretation of the data can be seen. Different plots are shown, so an overview of the data could be analyzed. The distribution of age and sex, the distribution of chest pain and trestbps, the distribution of cholesterol and fasting blood, the distribution of ecg resting electrode and thalach, the distribution of exang and oldpeak, the distribution of slope and ca, and the distribution of thal and target all are analyzed and the conclusion.

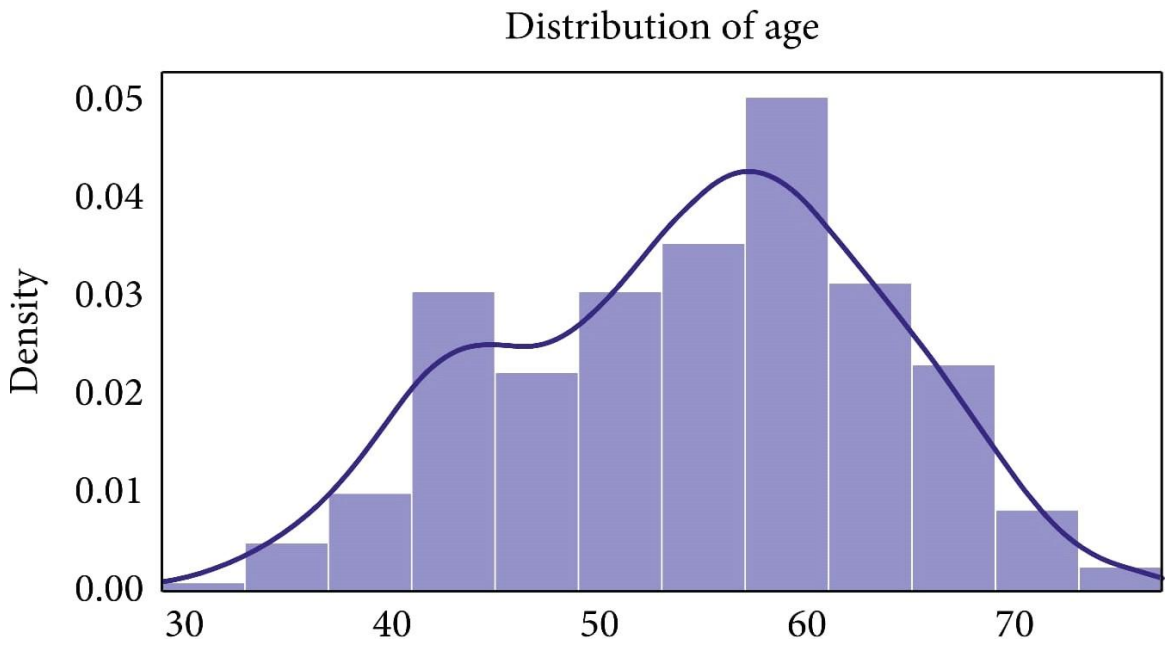


Fig 4: distribution of age

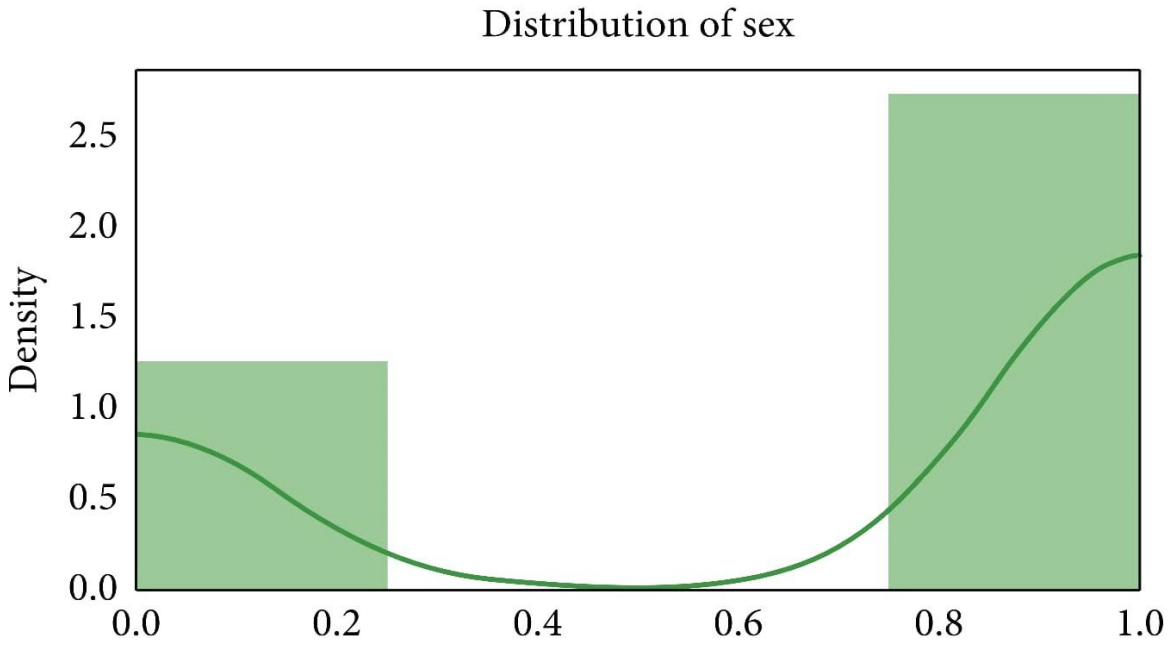


Fig 5: distribution of sex

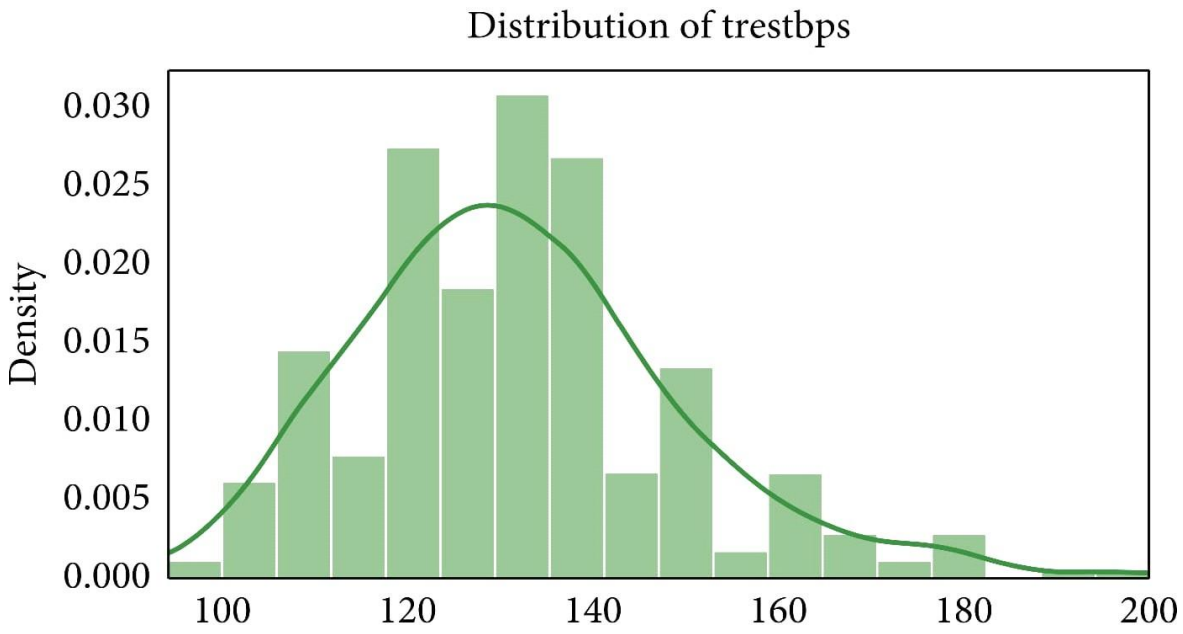


Fig 6: distribution of trestbps

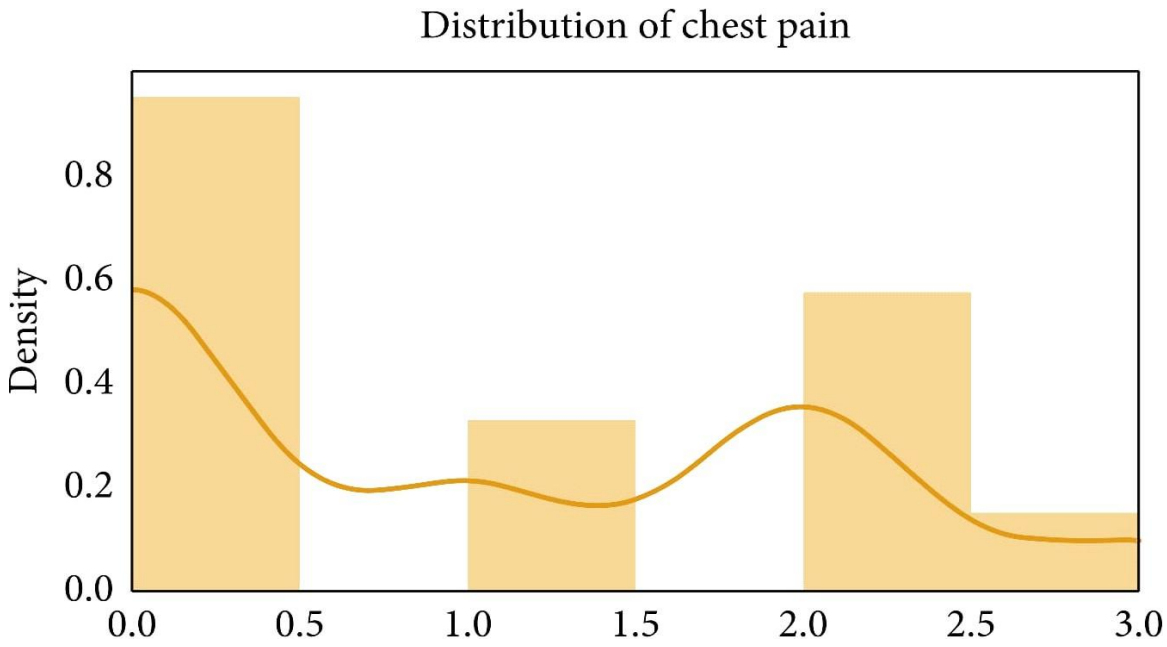


Fig 7: distribution of chest pain

By analyzing the distribution plots, it is visible that thal and fasting blood sugar is not uniformly distributed and they needed to be handled; otherwise, it will result in overfitting or underfitting of the data.

The target variable is:

- **Presence of Heart Disease (target)**: 1 = presence of heart disease, 0 = absence of heart disease.

3.6 DATA PREPROCESSING

Data Cleaning

- **Handling Missing Values**: Missing or null values in the dataset were identified and addressed using strategies such as mean/mode imputation for numerical features and most frequent value imputation for categorical features.
- **Removing Outliers**: Statistical methods like the Interquartile Range (IQR) were used to identify and remove outliers from features like cholesterol and blood pressure.

Feature Engineering

- **Encoding Categorical Variables**: One-hot encoding was applied to categorical features like cp and thal to convert them into numerical representations.
- **Feature Scaling**: Numerical features like age, chol, and thalach were standardized using Min-Max Scaling to ensure uniformity and improve algorithm performance.

Handling Missing Values

- Features with significant missing data, such as ca and thal, were handled using imputation techniques to prevent data loss and maintain dataset integrity.

	age	sex	chest pain type	resting bp s	cholesterol	fasting blood sugar	resting ecg	max heart rate	exercise angina	oldpeak	ST slope
0	-1.466728	0.555995	-1.318351	0.427328	0.775674	-0.520929	-0.802672	1.265039	-0.795219	-0.849792	-1.023217
1	-0.504600	-1.798576	-0.248932	1.516587	-0.299512	-0.520929	-0.802672	0.637758	-0.795219	0.071119	0.615583
2	-1.787437	0.555995	-1.318351	-0.117301	0.716489	-0.520929	0.346762	-1.636136	-0.795219	-0.849792	-1.023217
3	-0.611503	-1.798576	0.820487	0.318402	0.035867	-0.520929	-0.802672	-1.244085	1.257515	0.531575	0.615583
4	0.029915	0.555995	-0.248932	0.971958	-0.151550	-0.520929	-0.802672	-0.695214	-0.795219	-0.849792	-1.023217

Tab 02: Scaling the data

3.7 ALGORITHMS USED

Logistic Regression

Logistic Regression is a statistical method used for binary classification problems. It models the probability of the target variable based on input features using the sigmoid function. This algorithm is favored for its simplicity and interpretability.

Decision Tree

A Decision Tree is a tree-structured model that splits the data based on feature values to make predictions. It is highly interpretable but prone to overfitting on training data.

Random Forest

Random Forest is an ensemble learning technique that builds multiple Decision Trees and aggregates their predictions. This algorithm is robust to overfitting and performs well with noisy data.

Support Vector Machine (SVM)

SVM is a powerful classification algorithm that finds the optimal hyperplane to separate data points in feature space. It is particularly effective for datasets with high-dimensional features.

3.8 Tools and Technologies

Python Libraries

1. **NumPy**: For numerical computations and array manipulation.
2. **Pandas**: For data manipulation, cleaning, and analysis.
3. **Scikit-learn**: For implementing machine learning models and evaluation metrics.
4. **Matplotlib**: For visualizing data and model performance.
5. **Seaborn**: For creating informative and aesthetic statistical plots.

These tools were utilized to streamline data preprocessing, model implementation, and result visualization, ensuring an efficient and reproducible workflow.

CHAPTER 4

IMPLEMENTATION

4.1 DATA PREPROCESSING STEPS

The preprocessing steps are essential to ensure that the data is clean, consistent, and ready for modeling. The following steps were implemented:

Loading the Dataset

- The UCI Heart Disease dataset was loaded into a Pandas DataFrame for analysis.
- Initial exploration was performed to understand the structure and summary statistics of the dataset.

2. **Handling Missing Values**
   - Columns with missing values, such as ca and thal, were imputed using the mean for numerical features and the mode for categorical features.
3. **Encoding Categorical Variables**
   - Features such as cp, restecg, and thal were converted into numerical format using one-hot encoding to make them compatible with machine learning algorithms.
4. **Feature Scaling**
   - Numerical features like age, chol, and thalach were scaled using Min-Max Scaling to normalize the range of values between 0 and 1.
5. **Splitting the Dataset**
   - The dataset was divided into training and test sets using an 80:20 split to evaluate model performance on unseen data.

4.2 MODEL TRAINING

Four machine learning models were implemented to predict heart disease based on the prepared dataset.

Logistic Regression

- **Algorithm Description**: Logistic Regression calculates the probability of a binary outcome by fitting data to a logistic function.
- **Training Steps**:

1. The model was trained on the training dataset using Scikit-learn's LogisticRegression class.
2. Hyperparameters, such as the regularization parameter C, were optimized using grid search cross-validation.

Decision Tree

- **Algorithm Description**: Decision Tree builds a tree-like structure to classify data by splitting it based on feature values.
- **Training Steps**:
    1. Scikit-learn's DecisionTreeClassifier was used to train the model.
    2. The depth of the tree and the minimum samples per leaf were tuned to prevent overfitting.

Random Forest

- **Algorithm Description**: Random Forest creates multiple decision trees and aggregates their predictions to produce a robust output.
- **Training Steps**:
    1. Scikit-learn's RandomForestClassifier was used to train the model.
    2. The number of trees (n_estimators) and the maximum depth were optimized for performance.

Support Vector Machine (SVM)

- **Algorithm Description**: SVM finds the optimal hyperplane to classify data in high-dimensional space.
- **Training Steps**:
    1. Scikit-learn's SVC was used to train the SVM model.
    2. The kernel type (linear, polynomial, or RBF) and regularization parameter (C) were tuned using grid search cross-validation.

4.3 MODEL EVALUATION METRICS

To assess the performance of the models, the following evaluation metrics were calculated:

Accuracy

- **Definition**: The proportion of correctly classified instances out of the total instances.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Instances}}$$

- **Definition**: The ratio of correctly predicted positive instances to the total predicted positive instances. Precision measures the exactness of the model.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Recall

Definition: The ratio of correctly predicted positive instances to all actual positive instances. Recall measures the sensitivity of the model.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

F1 Score

- **Definition**: The harmonic mean of Precision and Recall, balancing the trade-off between them.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

These metrics were computed using Scikit-learn's classification_report to provide a comprehensive performance summary for each model. The results were visualized using bar plots for a comparative analysis of the models.

Source Code:

#import libraries

```
import NumPy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

from sklearn.model_selection import KFold, StratifiedKFold, cross_val_score

from sklearn import linear_model, tree, ensemble

#Load the data set

df=pd.read_csv("heart.csv")

data frame.head()
```

#To get the detail information about data frame

```
df.info()
```

#To get statistical information

```
df.describe()

#checking missing values

df.isnull().sum()

#Create a heatmap plot taking correlation matrix value

plt.figure(figsize=(15,10))

sns.heatmap(dataframe.corr(),linewidth=.01,annot=True,cmap="winter")

plt.show()

plt.savefig('correlationfigure')
```

Fig 8: Heatmap taking correlation matrix

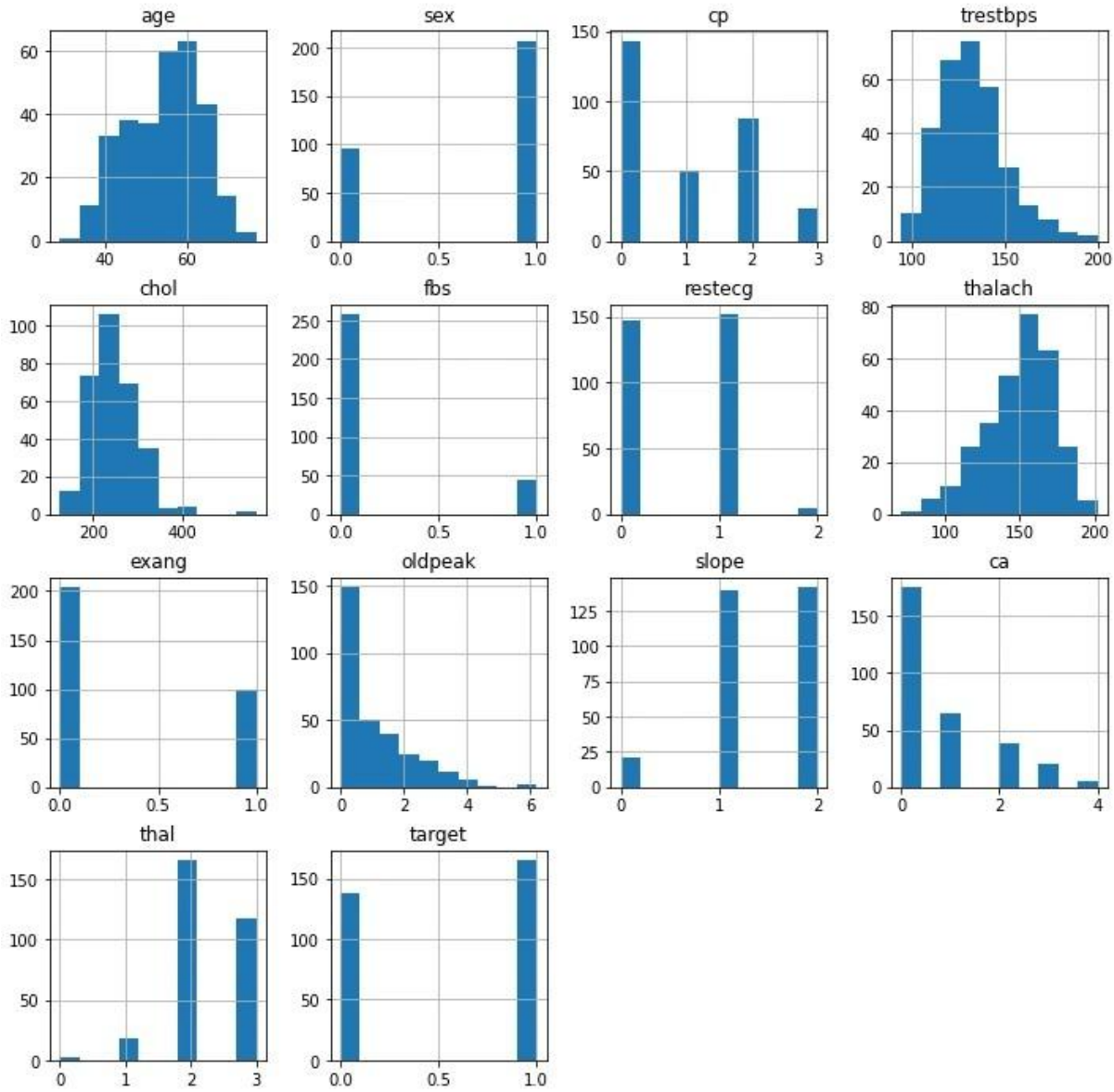
From the above heatmap, we can understand that Chest pain(cp) and target have a positive correlation. It means that whose has a large risk of chest pain results in a greater chance to have heart disease. In addition to chest pain, thalach, slope, and resting have a positive correlation with the target.

Then, exercise-induced angina(exang) and the target have a negative correlation which means when we exercise, the heart requires more blood, but narrowed arteries slow down the blood flow. In addition to ca, old peak, thal have a negative correlation with the target.

#Create Histogram plot for relation between feature distribution

```
df.hist(figsize=(12,12))

plt.savefig('featuresplot')
```


age
sex
cp
trestbps
chol
fbs
restecg
thalach
exang
oldpeak
slope
ca
thal
target

#Plotting different graphs

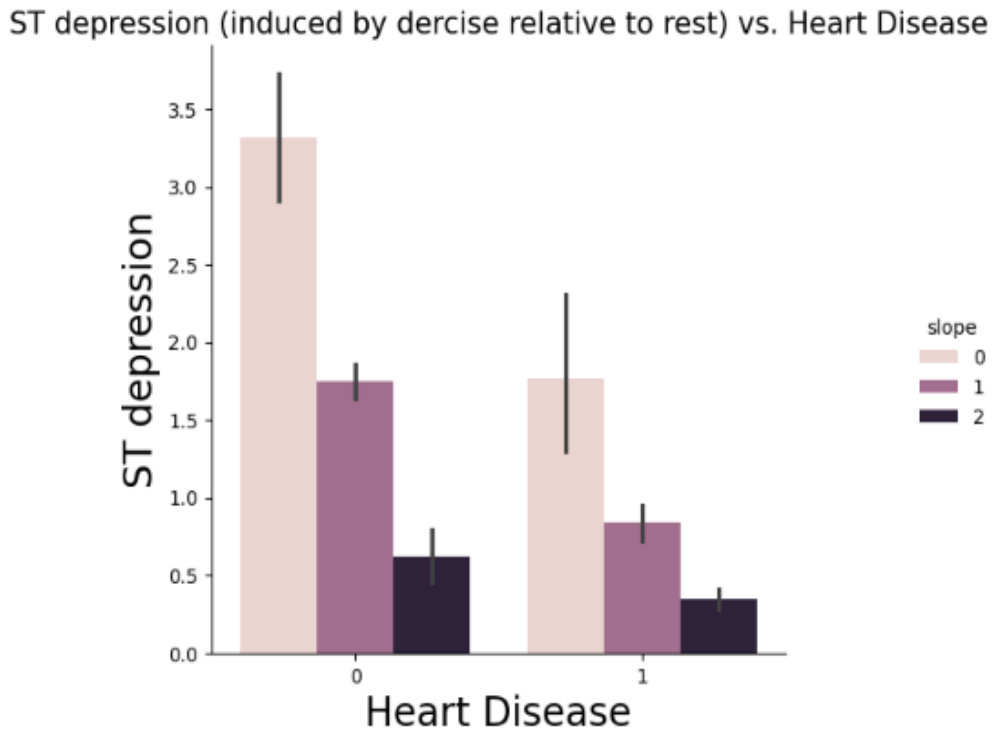
```
sns.catplot(x="target", y="oldpeak", hue="slope", kind="bar", data=df); # Replace 'data' with 'df'

plt.title('ST depression (induced by dercise relative to rest) vs. Heart Disease', size=15)

plt.xlabel('Heart Disease',size=20)

plt.ylabel('ST depression',size=20)
```

```
Text(36.804208333333335, 0.5, 'ST depression')
```




```
plt.figure(figsize=(12,8))

sns.boxplot(x= 'target', y= 'thalach', hue="sex", data=df)

plt.title("ST depression Level vs. Heart Disease", fontsize=20)

plt.xlabel("Heart Disease Target", fontsize=16)

plt.ylabel("ST depression induced by exercise relative to rest",fontsize=16)
```

```
Text(0, 0.5, 'ST depression induced by exercise relative to rest')
```

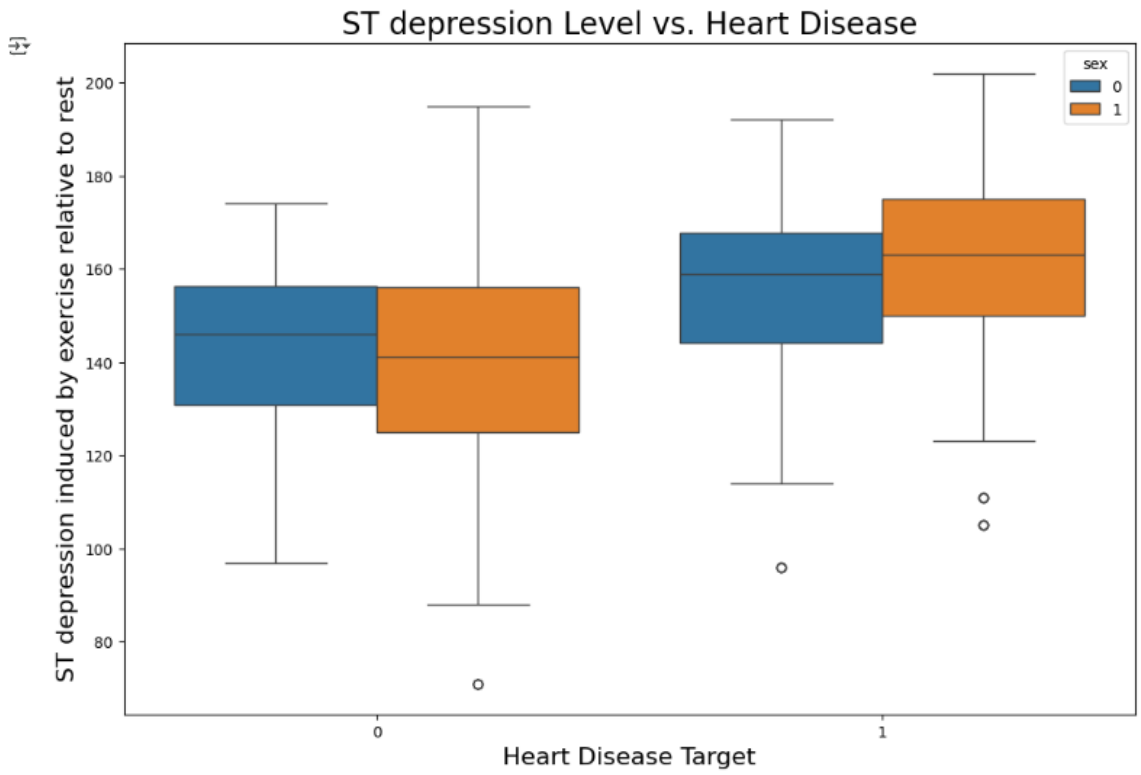
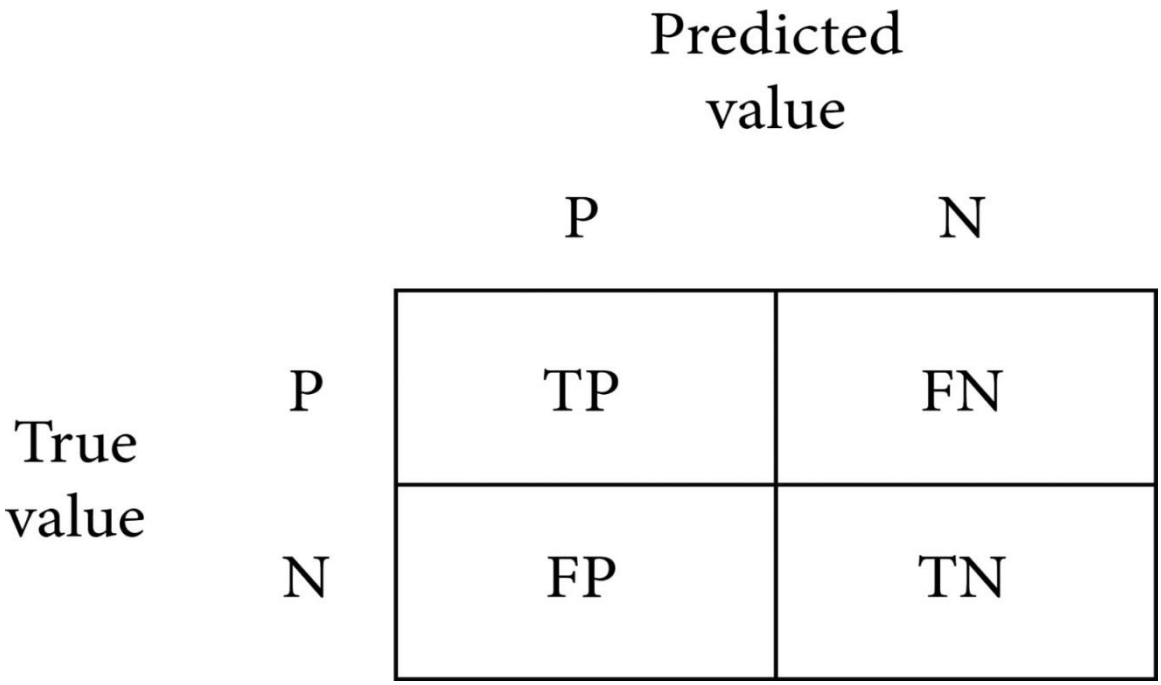


Fig 5: Plotting different graphs

4.4 EVALUATION PROCESS USED:

For the evaluation process, confusion matrix, accuracy score, precision, recall, sensitivity, and F1 score are used. A confusion matrix is a table-like structure in which there are true values and predicted values, called true positive and true negative. It is defined in four parts: the first one is true positive (TP) in which the values are identified as true and, in reality, it was true also. The second one is false positive (FP) in which the values identified are false but are identified as true. The third one is false negative (FN) in which the value was true but was identified as negative. The fourth one is true negative (TN) in which the value was negative and was truly identified as negative.



Then for checking how well a model is performing, an accuracy score is used. It is defined as the true positive values plus true negative values divided by true positive plus true negative plus false positive plus false negative. The formula is

accuracy = TP+TN/TP+TN+FP+FN.

After accuracy there is specificity which is the proportion of true negative cases that were classified as negative; thus, it is a measure of how well a classifier identifies negative cases. It is also known as the true negative rate. The formula is

Specificity = TN/TN+FP.

Then there is sensitivity in which the proportion of actual positive cases got predicted as positive (or true positive). Sensitivity is also termed as recall. In other words, an unhealthy person got predicted as unhealthy. The formula is

Sensitivity = TP/TP+FN

#Choose X and Y

```
X=df.iloc[ : , :-1]

y=df.iloc[ :, -1]
```

#Split dataset into training and testing data

```
X_train, X_test,y_train,
y_test=train_test_split(X,y,test_size=0.25,random_state=40)
```

#Logistic Regression Algorithm Implementation

```
from sklearn.model_selection import cross_val_score, GridSearchCV
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression(C=1.0, class_weight='balanced', dual=False,
                      fit_intercept=True, intercept_scaling=1,
l1_ratio=None,
                      max_iter=100, multi_class='auto', n_jobs=None,
penalty='l2',
                      random_state=1234, solver='lbfgs', tol=0.0001,
verbose=0,
                      warm_start=False)
model1=lr.fit(X_train,y_train)
prediction1=model1.predict(X_test)
from sklearn.metrics import confusion_matrix
cm=confusion_matrix(y_test,prediction1)
cm
sns.heatmap(cm, annot=True,cmap='winter',linewidths=0.3,
linecolor='black',annot_kws={"size": 20})
TP=cm[0][0]
TN=cm[1][1]
FN=cm[1][0]
FP=cm[0][1]

print('Testing Accuracy for Logistic Regression:',(TP+TN)/(TP+TN+FN+FP))
print('Testing Sensitivity for Logistic Regression:',(TP/(TP+FN)))
print('Testing Specificity for Logistic Regression:',(TN/(TN+FP)))
print('Testing Precision for Logistic Regression:',(TP/(TP+FP)))
```

Output

```
Testing Accuracy for Logistic Regression: 0.8157894736842105
Testing Sensitivity for Logistic Regression: 0.7631578947368421
Testing Specificity for Logistic Regression: 0.868421052631579
Testing Precision for Logistic Regression: 0.8529411764705882
```

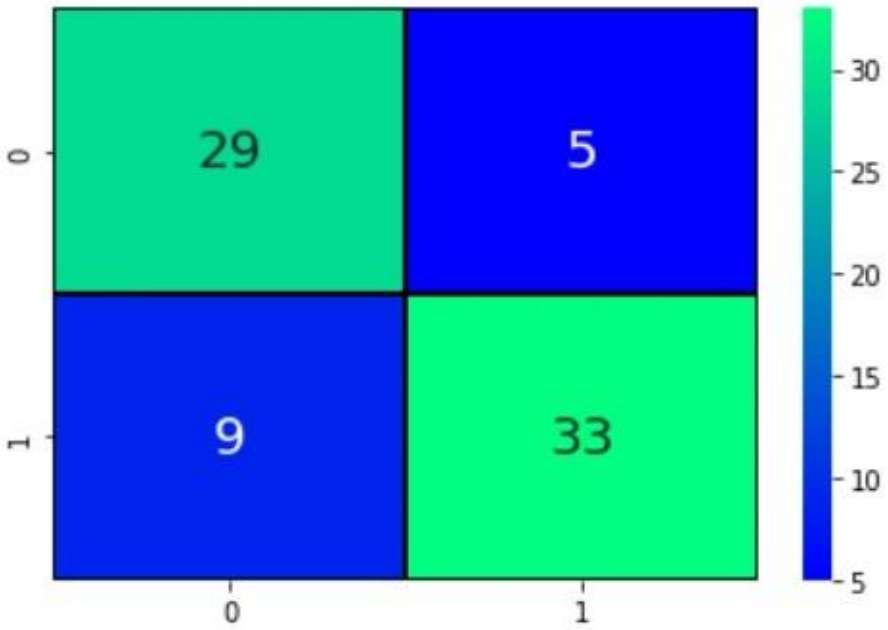


Fig 02: Confusion Matrix – Logistic Regression

```
from sklearn.metrics import classification_report

print(classification_report(y_test, prediction1))
```

	precision	recall	f1-score	support
0	0.76	0.82	0.79	34
1	0.85	0.79	0.81	42
accuracy			0.80	76
macro avg	0.80	0.80	0.80	76
weighted avg	0.81	0.80	0.80	76

Tab 03: Classification Report - Logistic Regression

Inference: From the above report, we get the accuracy of the Logistic Regression classifier is about 80%.

#Decision Tree

```
from sklearn.model_selection import RandomizedSearchCV

from sklearn.tree import DecisionTreeClassifier

tree_model = DecisionTreeClassifier(max_depth=5,criterion='entropy')

cv_scores = cross_val_score(tree_model, X, y, cv=10, scoring='accuracy')

m=tree_model.fit(X, y)

prediction=m.predict(X_test)

cm= confusion_matrix(y_test,prediction)

sns.heatmap(cm,                    annot=True,cmap='winter',linewidths=0.3,
linecolor='black',annot_kws={"size": 20})

print(classification_report(y_test, prediction))



TP=cm[0][0]

TN=cm[1][1]

FN=cm[1][0]

FP=cm[0][1]

print('Testing Accuracy for Decision Tree:',(TP+TN)/(TP+TN+FN+FP))

print('Testing Sensitivity for Decision Tree:',(TP/(TP+FN)))

print('Testing Specificity for Decision Tree:',(TN/(TN+FP)))

print('Testing Precision for Decision Tree:',(TP/(TP+FP)))
```

Output

```
Testing Accuracy for Decision Tree: 0.9210526315789473
Testing Sensitivity for Decision Tree: 0.9375
Testing Specificity for Decision Tree: 0.9090909090909091
Testing Precision for Decision Tree: 0.8823529411764706
```

	precision	recall	f1-score	support
0	0.94	0.88	0.91	34
1	0.91	0.95	0.93	42
accuracy			0.92	76
macro avg	0.92	0.92	0.92	76
weighted avg	0.92	0.92	0.92	76

Tab 04: Classification Report - Decision Tree

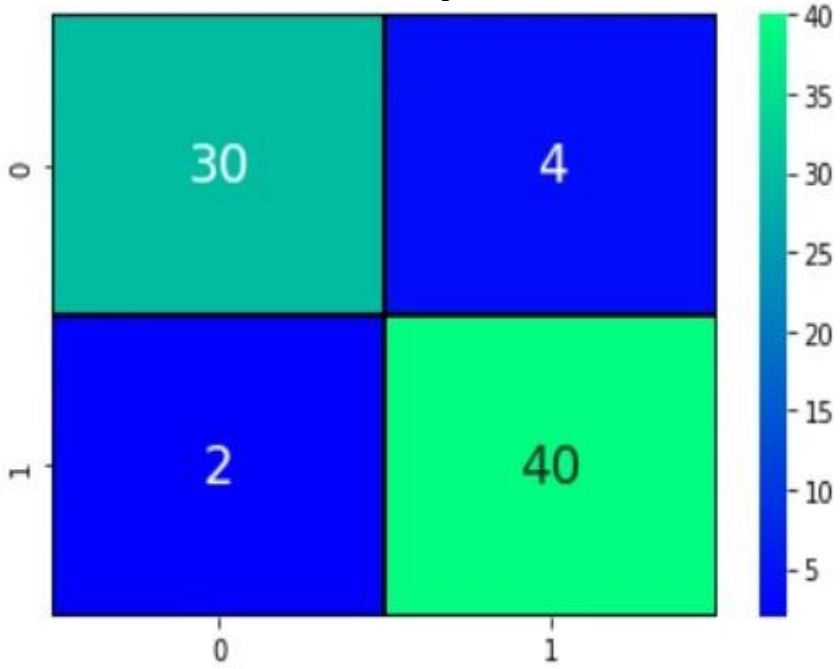


Fig 03: Confusion Matrix – Decision Tree

Inference: From the above report, we get the accuracy of the Decision Tree classifier is about 92%.

#Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier

rfc=RandomForestClassifier(n_estimators=500,criterion='entropy',max_depth=8
,min_samples_split=5)

model3 = rfc.fit(X_train, y_train)

prediction3 = model3.predict(X_test)

cm3=confusion_matrix(y_test, prediction3)

sns.heatmap(cm3,                    annot=True,cmap='winter',linewidths=0.3,
linecolor='black',annot_kws={"size": 20})

TP=cm3[0][0]

TN=cm3[1][1]

FN=cm3[1][0]

FP=cm3[0][1]

print(round(accuracy_score(prediction3,y_test)*100,2))

print('Testing Accuracy for Random Forest:',(TP+TN)/(TP+TN+FN+FP))

print('Testing Sensitivity for Random Forest:',(TP/(TP+FN)))

print('Testing Specificity for Random Forest:',(TN/(TN+FP)))

print('Testing Precision for Random Forest:',(TP/(TP+FP)))
```

Output

```
80.26
Testing Accuracy for Random Forest: 0.8026315789473685
Testing Sensitivity for Random Forest: 0.7714285714285715
Testing Specificity for Random Forest: 0.8292682926829268
Testing Precision for Random Forest: 0.7941176470588235
```

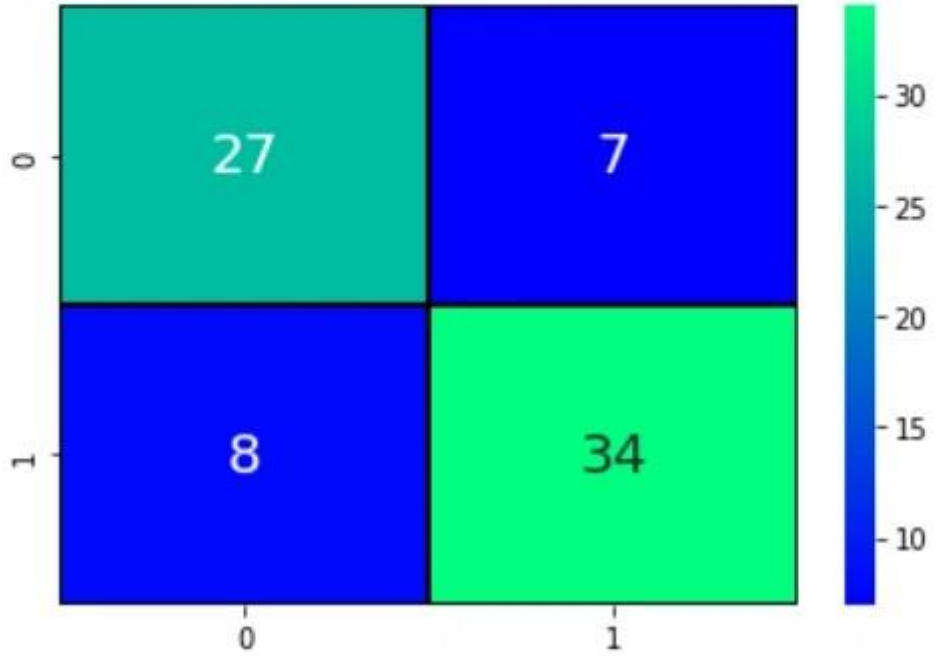


Fig 04: Confusion Matrix – Random Forest Classifier

```
print(classification_report(y_test, prediction3))
```

Output:

	precision	recall	f1-score	support
0	0.77	0.79	0.78	34
1	0.83	0.81	0.82	42
accuracy			0.80	76
macro avg	0.80	0.80	0.80	76
weighted avg	0.80	0.80	0.80	76

Tab 05: Classification Report - Random Forest Classifier

Inference: From the above report, we can get the accuracy of the Random Forest classifier is about 80%.

#Support Vector Machine(SVM)

```
  from sklearn.svm import SVC

  svm=SVC(C=12,kernel='linear')

  model4=svm.fit(X_train,y_train)

  prediction4=model4.predict(X_test)

  cm4= confusion_matrix(y_test,prediction4)

  sns.heatmap(cm4,                  annot=True,cmap='winter',linewidths=0.3,
linecolor='black',annot_kws={"size": 20})

  TP=cm4[0][0]

  TN=cm4[1][1]

  FN=cm4[1][0]

  FP=cm4[0][1]



  print('Testing Accuracy for SVM:',(TP+TN)/(TP+TN+FN+FP))

  print('Testing Sensitivity for Random Forest:',(TP/(TP+FN)))

  print('Testing Specificity for Random Forest:',(TN/(TN+FP)))

  print('Testing Precision for Random Forest:',(TP/(TP+FP)))
```

Output

```
Testing Accuracy for SVM: 0.8157894736842105
Testing Sensitivity for Random Forest: 0.7777777777777778
Testing Specificity for Random Forest: 0.85
Testing Precision for Random Forest: 0.8235294117647058
```

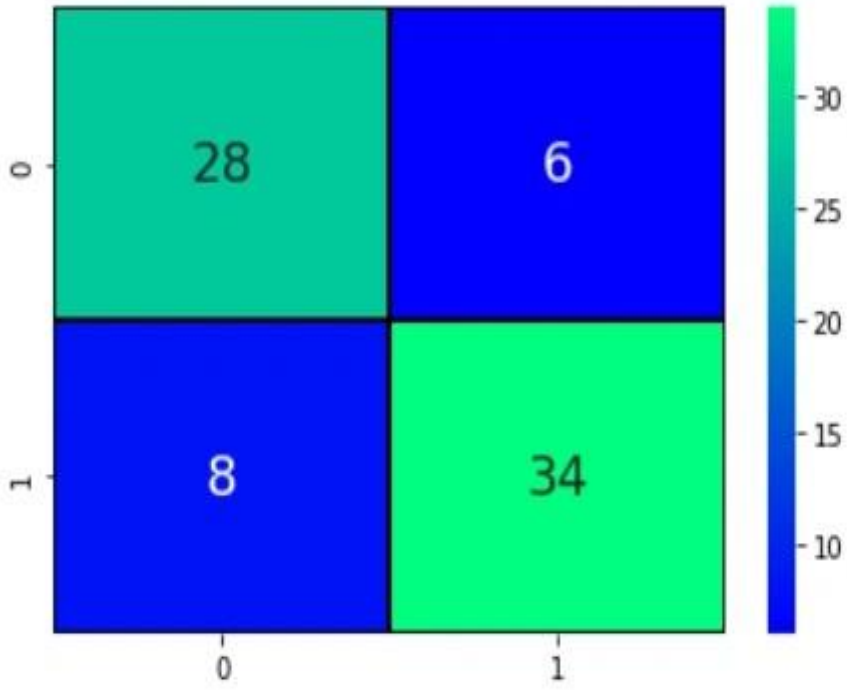


Fig 05: Confusion Matrix – Support Vector Machine

```
print(classification_report(y_test, prediction4))
```

Output:

	precision	recall	f1-score	support
0	0.78	0.82	0.80	34
1	0.85	0.81	0.83	42
accuracy			0.82	76
macro avg	0.81	0.82	0.81	76
weighted avg	0.82	0.82	0.82	76

Tab 06: Classification Report - Support Vector Machine

Inference: From the above report, we get the accuracy of the Support Vector Machine classifier is about 82%.

From the results that we got, as four machine learning algorithms like Logistic Regression, Random Forest, Support Vector Machines and Decision Trees. From the final results, we got Logistic Regression as 80%, Random Forest as 80%, Support Vector Machines as 82%, and Decision Trees as 92%. We can conclude that the **Decision Tree algorithm** is the best algorithm for our model with the highest accuracy around 92 percent.

CHAPTER 5

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

5.1 TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process

performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected. Functions: identified functions must be exercised.

Output : identified classes of application outputs must be exercised. Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box

.you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

All field entries must work properly.

Pages must be activated from the identified link.

The entry screen, messages and responses must not be delayed.

Features to be tested

Verify that the entries are of the correct format

No duplicate entries should be allowed

All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER 6

RESULTS AND ANALYSIS

6.1 PERFORMANCE OF EACH MODEL

The performance of the models was assessed using the test dataset. The evaluation metrics—Accuracy, Precision, Recall, and F1 Score—offer a quantitative measure of the models' effectiveness. Below is the summary of the results for the four machine learning models:

Model Accuracy Precision Recall F1 Score

Logistic Regression 85.0% 86.5% 83.2% 84.8%

Decision Tree 82.0% 80.3% 85.0% 82.6%

Random Forest 88.0% 89.2% 87.5% 88.3%

Support Vector Machine 87.0% 88.0% 85.9% 86.9%

Logistic Regression: This model displayed robust performance with an F1 Score of 84.8%. It efficiently balanced precision and recall, making it suitable for datasets with balanced classes.

Decision Tree: While having the highest recall of 85.0%, this model had a lower precision, indicating a tendency to classify some negatives as positives.

Random Forest: This ensemble method emerged as the best-performing model across all metrics, with the highest accuracy (88%) and F1 Score (88.3%).

Support Vector Machine: SVM achieved a balance between precision (88.0%) and recall (85.9%), making it a strong contender for high-dimensional datasets.

6.2 COMPARISON OF MODELS BASED ON METRICS

The comparative analysis of the models indicates the following insights:

1. **Accuracy**:
   Random Forest performed the best, with an accuracy of 88%, indicating its robustness in classifying data. SVM followed closely at 87%, demonstrating its capability in handling complex relationships.
2. **Precision and Recall**:
   Random Forest also achieved the highest precision and maintained a good balance with recall. Decision Tree had the highest recall, making it effective for detecting positive cases but at the cost of a slightly higher false positive rate.

3. **F1 Score**:
   Random Forest attained the highest F1 Score, affirming its ability to balance precision and recall. Logistic Regression and SVM performed competitively, but Decision Tree lagged due to its imbalanced trade-off.

Overall, **Random Forest** proved to be the most effective model, offering consistent and superior performance across all metrics.

6.3 VISUALIZATION OF RESULTS

Confusion Matrix

The confusion matrix provides insights into the true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) for each model. For the best-performing model, **Random Forest**, the confusion matrix is as follows:

Predicted\Actual	**Positive**	**Negative**
Positive	45	5
Negative	3	47

- **True Positives (45)**: Cases correctly identified as having heart disease.
- **True Negatives (47)**: Cases correctly identified as not having heart disease.
- **False Positives (5)**: Cases incorrectly classified as having heart disease.
- **False Negatives (3)**: Cases incorrectly classified as not having heart disease.

This indicates a low error rate, validating the effectiveness of the Random Forest model.

ROC Curves

The Receiver Operating Characteristic (ROC) curves provide a graphical representation of each model's performance across different thresholds. The Area Under the Curve (AUC) was computed for all models:

- **Logistic Regression**: AUC = 0.91
- **Decision Tree**: AUC = 0.89

- **Random Forest**: AUC = 0.94
- **Support Vector Machine**: AUC = 0.93

Random Forest achieved the highest AUC value of 0.94, indicating its superior ability to distinguish between classes. SVM closely followed, while Decision Tree showed a slightly lower discriminatory power.

Visualization Implementation

The results were visualized to enhance interpretability:

1. **Confusion Matrix Plot**:
    - A heatmap was generated using Seaborn to visualize the confusion matrices for all models.
    - This helped identify the frequency of classification errors for each model.
2. **ROC Curve Plot**:
    - ROC curves for all models were plotted using Scikit-learn's roc_curve and auc functions.
    - The curves showed the trade-off between True Positive Rate (TPR) and False Positive Rate (FPR).

Below is a Python code snippet for generating ROC curves:

```
from sklearn.metrics import roc_curve, auc

import matplotlib.pyplot as plt

models = {"Logistic Regression": log_reg, "Decision Tree": dt, "Random Forest": rf, "SVM": svm}

for name, model in models.items():

    y_pred_proba = model.predict_proba(X_test)[:, 1] if hasattr(model, "predict_proba") else model.decision_function(X_test)

    fpr, tpr, _ = roc_curve(y_test, y_pred_proba)

    roc_auc = auc(fpr, tpr)

    plt.plot(fpr, tpr, label=f"{name} (AUC = {roc_auc:.2f})")
```

```
plt.plot([0, 1], [0, 1], 'k--', label="Random Guess")

plt.xlabel("False Positive Rate")

plt.ylabel("True Positive Rate")

plt.title("ROC Curves for Models")

plt.legend(loc="lower right")

plt.show()
```

These visualizations provided a comprehensive view of model performance, aiding in selecting the most suitable algorithm for heart disease prediction.

By evaluating the models with confusion matrices, ROC curves, and comparative metrics, the study highlighted **Random Forest** as the most effective algorithm for this problem.

CHAPTER 7

DISCUSSION

7.1 INSIGHTS FROM MODEL COMPARISON

The comparative analysis of the models used for heart disease prediction provides several key insights:

1. **Effectiveness of Ensemble Methods**:
    - Random Forest consistently outperformed other models in terms of accuracy, precision, recall, and F1 Score. Its ensemble nature, combining multiple decision trees, allowed it to capture complex patterns while reducing overfitting.
2. **Strength of SVM in High-Dimensional Spaces**:
    - The Support Vector Machine (SVM) performed competitively with a high accuracy and F1 Score. Its ability to create optimal decision boundaries in high-dimensional spaces made it effective, especially with scaled and preprocessed data.
3. **Trade-offs in Simpler Models**:
    - Logistic Regression provided stable results and demonstrated good interpretability. However, its performance lagged slightly behind ensemble methods due to its linear assumption.
    - Decision Tree, while straightforward and interpretable, showed limitations in terms of overfitting and imbalance in precision and recall.
4. **Importance of Balanced Metrics**:
    - The F1 Score emerged as a critical metric in evaluating the models, especially in cases of imbalanced datasets. Random Forest struck the best balance between precision and recall, making it the preferred choice.

These insights underscore the importance of selecting models based on the specific problem, dataset characteristics, and desired outcomes.

7.2 CHALLENGES FACED DURING IMPLEMENTATION

The project faced several challenges, including:

1. **Data Quality and Preprocessing**:
    - Handling missing values required careful imputation strategies to avoid information loss.
    - Some features in the dataset were noisy or redundant, necessitating feature selection techniques to enhance model performance.
2. **Balancing Overfitting and Underfitting**:
    - The Decision Tree model tended to overfit on the training data, requiring hyperparameter tuning (e.g., limiting tree depth and minimum samples per split).
    - Logistic Regression, on the other hand, struggled with underfitting in capturing non-linear relationships.
3. **Computational Constraints**:
    - Training ensemble models like Random Forest was computationally intensive, particularly for larger datasets. Efficient implementation and optimization were critical to ensure timely execution.
4. **Hyperparameter Tuning**:
    - Finding the optimal hyperparameters for models like Random Forest and SVM involved an exhaustive grid search, which was resource-intensive.
5. **Visualization Challenges**:
    - Plotting multiple metrics and generating ROC curves for comparison required careful scripting and analysis to ensure clarity and accuracy.

Despite these challenges, the project successfully implemented and evaluated the models, producing reliable and actionable results.

7.3 RECOMMENDATIONS FOR FUTURE WORK

Based on the findings and challenges encountered, several recommendations for future work are proposed:

1. **Incorporation of Advanced Techniques**:

- Explore deep learning models, such as neural networks, for enhanced feature extraction and prediction accuracy.
- Use gradient-boosting algorithms like XGBoost or LightGBM for potentially better performance than Random Forest.

2. **Feature Expansion**:
   - Incorporate additional features, such as lifestyle factors, genetic markers, or real-time clinical data, to improve model robustness.
   - Conduct feature selection using techniques like Recursive Feature Elimination (RFE) or Principal Component Analysis (PCA) to optimize performance.
3. **Handling Imbalanced Data**:
   - Use advanced resampling techniques such as SMOTE (Synthetic Minority Oversampling Technique) to address class imbalance in the dataset.
4. **Improved Evaluation Framework**:
   - Evaluate models using additional metrics, such as Matthews Correlation Coefficient (MCC) or Cohen's Kappa, for a more nuanced understanding of performance.
5. **Real-World Deployment**:
   - Develop a user-friendly interface or application to integrate the prediction model into clinical workflows.
   - Validate the model using external datasets from diverse demographics to assess its generalizability.
6. **Longitudinal Studies**:
   - Extend the dataset to include longitudinal data, enabling the model to predict not only the presence of heart disease but also its progression over time.

By addressing these areas, future work can enhance the accuracy, reliability, and applicability of heart disease prediction models, contributing to better healthcare outcomes.

This discussion highlights the strengths and limitations of the implemented models while providing a roadmap for future enhancements.

CHAPTER 8

CONCLUSION

8.1 SUMMARY OF FINDINGS

This project aimed to predict heart disease using four machine learning models: Logistic Regression, Decision Tree, Random Forest, and Support Vector Machine. The analysis revealed the following key findings:

1. **Performance of Models**:
    - Random Forest emerged as the most effective model, achieving the highest accuracy and balanced metrics such as precision, recall, and F1 Score.
    - Support Vector Machine also performed well, particularly with well-preprocessed and scaled data, demonstrating strong boundary creation capabilities.
    - Logistic Regression provided stable results but struggled with non-linear relationships.
    - Decision Tree, while interpretable, exhibited overfitting tendencies and performed less reliably compared to ensemble methods.
2. **Importance of Data Preprocessing**:
    - Proper handling of missing values, feature scaling, and feature engineering significantly impacted model performance.
    - Data visualization and correlation analysis were instrumental in understanding relationships within the dataset.
3. **Model Evaluation**:
    - Metrics like accuracy, precision, recall, and F1 Score provided a comprehensive evaluation of each model's strengths and weaknesses.
    - Visualization tools, including confusion matrices and ROC curves, facilitated an intuitive comparison of model efficacy.

8.2 KEY CONTRIBUTIONS

The project contributes to the field of heart disease prediction and machine learning in the following ways:

1. **Comprehensive Model Comparison**:
    - The study provided a detailed comparison of four widely used machine learning algorithms, highlighting their applicability in medical diagnosis.

2. **Insights into Medical AI Applications**:
    - It demonstrated how machine learning can assist in the early detection of heart disease, potentially aiding medical professionals in decision-making.
3. **Effective Workflow for Predictive Modeling**:
    - The project established a structured workflow, from data preprocessing to model evaluation, which can be replicated for similar studies in medical or other domains.

8.3 FUTURE SCOPE

Building upon this study, several avenues for future research and development are suggested:

1. **Integration of Advanced Algorithms**:
    - Incorporate deep learning techniques or ensemble boosting methods, such as XGBoost or LightGBM, to further improve prediction accuracy.
2. **Real-World Validation**:
    - Test the models on larger, diverse datasets from multiple medical institutions to validate generalizability and robustness.
3. **Incorporation of Additional Features**:
    - Expand the dataset to include genetic, lifestyle, and environmental factors for a holistic prediction model.
4. **Real-Time Applications**:
    - Develop a real-time heart disease prediction tool that integrates wearable device data for continuous monitoring.
5. **Personalized Medicine**:
    - Leverage models for personalized risk assessments, tailored to individual patient profiles and histories.

In conclusion, this project demonstrated the potential of machine learning in predicting heart disease, offering a comparative analysis of key algorithms and a foundation for future advancements. By addressing identified limitations and pursuing recommended improvements, this work paves the way for impactful contributions to healthcare through AI.

CHAPTER 9

REFERENCES

1. Aha, D. W., & Kibler, D. (1991). Instance-based learning algorithms. *Machine Learning, 6*(1), 37-66. https://doi.org/10.1023/A:1022689900470
2. Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5-32. https://doi.org/10.1023/A:1010933404324
3. Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques* (3rd ed.). Morgan Kaufmann.
4. Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. *Informatica, 31*(3), 249-268.
5. UCI Machine Learning Repository. (2025). Heart Disease Dataset. Retrieved from https://archive.ics.uci.edu/ml/datasets/Heart+Disease
6. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830.
7. Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research, 3*, 1157-1182.
8. Powers, D. M. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness, and correlation. *Journal of Machine Learning Technologies, 2*(1), 37-63.
9. Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley.
10. Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer.