



AMITY UNIVERSITY ONLINE, NOIDA, UTTAR PRADESH

In partial fulfilment of the requirement for the award of Master of Computer Applications
(Specialization: Machine Learning)

Title: Heart Disease and Heart Attack Prediction using Machine Learning

Guide Details:

Name: Mrs. Amruta Mohan Chimanna

Designation: Assistant Professor, Department of CSE at Walchand College of Engineering.

Submitted by:

Khushi Govind Upadhye

Enrolment Number: A9929722000343(el)

MCA (Machine Learning) – July 2022 Batch

Semester IV

ABSTRACT

Heart disease, also called cardiovascular disease, is the leading cause of death globally over the last several decades. It encompasses a variety of conditions that affect the heart. It links a number of heart disease risk factors and emphasizes the urgency of finding timely, accurate, and reasonable ways to diagnose the condition early on and begin treating it. The heart is an important organ in living things. The diagnosis and prognosis of heart-related disorders necessitate greater accuracy, perfection, and precision because even a small error can result in exhaustion or even death; the number of heart-related deaths is growing exponentially every day. An essential tool for solving the issue is a prediction system. In order to address the issue, a prediction system for disease awareness is vital.

Since Machine Learning (ML) is a subset of Artificial Intelligence (AI), these algorithms and techniques have been applied to different medical datasets in order to automate the analysis of huge and complex data. In this project, we use the heart.csv dataset for training and testing to calculate the accuracy of Machine learning algorithms for heart disease prediction. We have implemented different algorithms like SVM, logistic regression, k-nearest neighbor, decision tree etc and compared it to find the machine learning algorithm that provides the highest accuracy. The Anaconda (Jupyter Notebook) is one of the best tools for implementing Python programming. It has a variety of libraries and header files which improve the accuracy and work precision.

For medical professionals, using machine learning techniques for data handling and prediction can become highly efficient. Therefore, we have covered heart disease and its risk factors as well as machine learning techniques in this study. We have predicted heart disease using these machine learning techniques, and we have given a comparative analysis of the machine learning algorithms used in the prediction experiment. This research's goal or objective is entirely focused on using machine learning techniques and analysis to predict heart disease.

The use of machine learning techniques to predict cardiovascular diseases, such as heart disease, has grown. These methods evaluate patient data and forecast the cardiovascular health of the patient using algorithms. Machine learning models are capable of calculating and forecasting the risk of heart disease by integrating multiple data points, including cholesterol levels, heart rate, and ECG signals. Healthcare will be greatly impacted by the application of machine learning to the prediction of heart disease. Early detection and accurate diagnosis can result in timely disease management, which may save lives and lessen the financial burden on society. Machine learning algorithms can help with personalized diagnosis and treatment planning by giving physicians valuable insights.

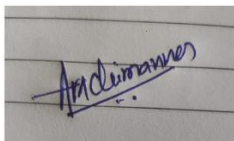
To find patterns and correlations that point to a higher risk of heart attack, machine learning algorithms can analyze clinical data, including risk factors like high blood pressure, high cholesterol, an irregular pulse rate, and diabetes. To increase the precision of heart attack prediction, machine learning models can automatically identify the most pertinent features or risk factors from the available data. The main variables that raise the risk of heart attacks are found through this process. Using historical data, which includes details about people who have and have not had heart attacks, machine learning models are trained. From this data, the models learn to identify trends and forecast outcomes based on fresh input data.

A range of machine learning algorithms, such as ensemble classification techniques, decision trees, logistic regression, and random forests, can be employed to predict heart attacks. The accuracy with which these algorithms have predicted the risk of developing heart disease is encouraging. The machine learning models must be assessed to determine their accuracy and performance after training. Metrics like area under the curve (AUC), sensitivity, specificity, and accuracy are commonly used in this assessment. Following their training and assessment, machine learning models can be used to forecast a person's risk of having a heart attack in real time for new users. Through the entry of pertinent information like medical background, vital signs, and results from diagnostic tests, the models are able to generate a risk assessment for each individual.

Keywords – Machine learning, heart attack prediction, heart attack prediction.

CERTIFICATE

This is to certify that Miss. Khushi Govind Upadhye, of Amity University Online has carried out the project work presented in the project report entitled “**Heart Attack Prediction using Machine Learning**” for the award of Master of Computer Applications (Machine Learning) under my guidance. The project report embodies results of original work, and the studies are carried out by the student herself. Certified further, that to the best of my knowledge the work reported herein does not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/ Institution.

A photograph of a handwritten signature in blue ink on lined paper. The signature is written in a cursive style and appears to read 'Amruta Mohan Chimanna'.

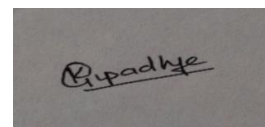
Signature

Name of the Guide – Mrs. Amruta Mohan Chimanna

Designation – Assistant Professor, Department of CSE at Walchand College of Engineering, Sangli.

DECLARATION

I, Khushi Govind Upadhye, a student pursuing Master of Computer Applications (Machine Learning), Semester IV at Amity University Online, hereby declare that the project work entitled “**Heart Disease and Heart Attack Prediction using Machine Learning**” has been prepared by me during the academic year 2023-24 under the guidance of Mrs. Amruta Mohan Chimanna. I assert that this project is a piece of original bona-fide work done by me. It is the outcome of my own effort and that it has not been submitted to any other university for the award of any degree.



Khushi Govind Upadhye

Enrolment Number - A9929722000343(el)

MCA Machine Learning, Semester IV

TABLE OF CONTENTS

Title Page

Abstract

Certificate

Declaration

List of Tables

List of Images

List of Figures

List of Abbreviations

Chapter 1 – Introduction to the topic

1.1 Introduction to Machine Learning

1.2 Types of Machine Learning

1.3 Significance of Machine Learning

1.4 Scope of Machine Learning

1.5 Machine Learning Life-Cycle

1.6 Heart Disease Overview

1.7 Heart Attack Overview

1.8 Risk Factors of Cardiovascular Diseases

1.9 Types of Cardiovascular Diseases

Chapter 2 – Review of Literature

Chapter 3 – Research Methodology and Objectives

Chapter 4 – Data Interpretation and Analysis

Chapter 5 – Findings and Inferences

Chapter 6 – Recommendations

Chapter 7 - Limitations

Chapter 8 – Conclusion

Chapter 9 - Bibliography

Appendix I – Model at Glance

Appendix II – Minutes of Meeting I

Appendix III – Minutes of Meeting II

Appendix IV – Minutes of Meeting III

Appendix V – Minutes of Meeting IV

Running head at the top of every page

LIST OF TABLES

Table 1 – Machine Learning Lifecycle

Table 2 – Machine Learning Algorithms

Table 3 – Variables

Table 4 – Summary

LIST OF IMAGES –

1. Importing Libraries
2. Reading the Dataset
3. Checking for Null Values
4. Histogram and Column Information
5. Data Normalization
6. Exploratory Data Analysis
7. Patients Suffering from Heart Problems
8. Splitting the Data into Training and Testing Data
9. Checking for the Unique Values
10. Standardization
11. Bar Plot for Heart Disease Frequency
12. Heart Disease Frequency According to Gender
13. Age wise Disease Rates
14. Feature Selection
15. Correlation

LIST OF FIGURES

Figure 1 – Types of Machine Learning

Figure 2 – Scope of Machine Learning

Figure 3 – Finding 1

Figure 4 – Finding 2

Figure 5 – Age wise Distribution

Figure 6 – Gender wise Distribution

Figure 7 – Age wise Distribution (Scatter Plot)

Figure 8 – Summary 1

Figure 9 – Summary 2

LIST OF ABBREVIATIONS –

ML – Machine Learning

AI – Artificial Intelligence

ECG – Electrocardiogram

AUC – Area under the ROC curve

CT – Computed Tomography Scan

KYC – Know Your Customer

LSTM – Long Short-Term Memory

GPS – Global Positioning System

ETL - Extract Transform and Load

EDA – Exploratory Data Analysis

ROC – Receiver Operating Characteristic Curve

API – Application Programming Interface

CVD – Cardiovascular Disease

MI – Myocardial Infarction

LDL – Low Density Lipoproteins

HDL – High Density Lipoproteins

CAD – Coronary Artery Disease

CHD – Coronary Heart Disease

CHF – Congestive Heart Failure

LR – Logistic Regression

RF – Random Forest Classifier

NB – Naïve Bayes

DTC – Decision Tree Classifier

KNN – K-Nearest Neighbor

SVC – Support Vector Classifier

CHAPTER 1 – INTRODUCTION

CHAPTER 1 – INTRODUCTION

1.1 OVERVIEW OF MACHINE LEARNING

Machine Learning (ML) is a subset of Artificial Intelligence which uses data-driven approach to enable algorithms to uncover hidden patterns within datasets, trains historical data to make predictions on new and similar data without explicitly programming for each task. Machine Learning finds patterns and insights in large datasets which can be difficult for humans to discover. It is necessary as it allows computers to learn from large and complex data to improve performance on various tasks with minimal human interaction.

In contrast to conventional computational methods, machine learning algorithms don't require explicitly coded instructions. Rather, they employ statistical analysis to produce values that fall within a predetermined range after training on data inputs. This enables computers to automate decision-making processes based on data inputs and create models from sample data. Its capacity to evaluate and decipher vast volumes of data has the potential to completely transform how companies run and how we engage with technology.

Machine learning (ML) is a branch of artificial intelligence that enables computers to "self-learn" from training data and improve over time without explicit programming. In order to generate their own predictions, machine learning algorithms are able to identify patterns in data and learn from them. To put it briefly, machine learning models and algorithms pick up new skills via experience. AI's machine learning subfield gives intelligent systems the ability to independently learn new things from data. Machine learning algorithms can be fed examples of labeled data (called training data), which enables them to process and recognize patterns automatically, in place of having to be programmed to carry out tasks.

1.2 TYPES OF MACHINE LEARNING

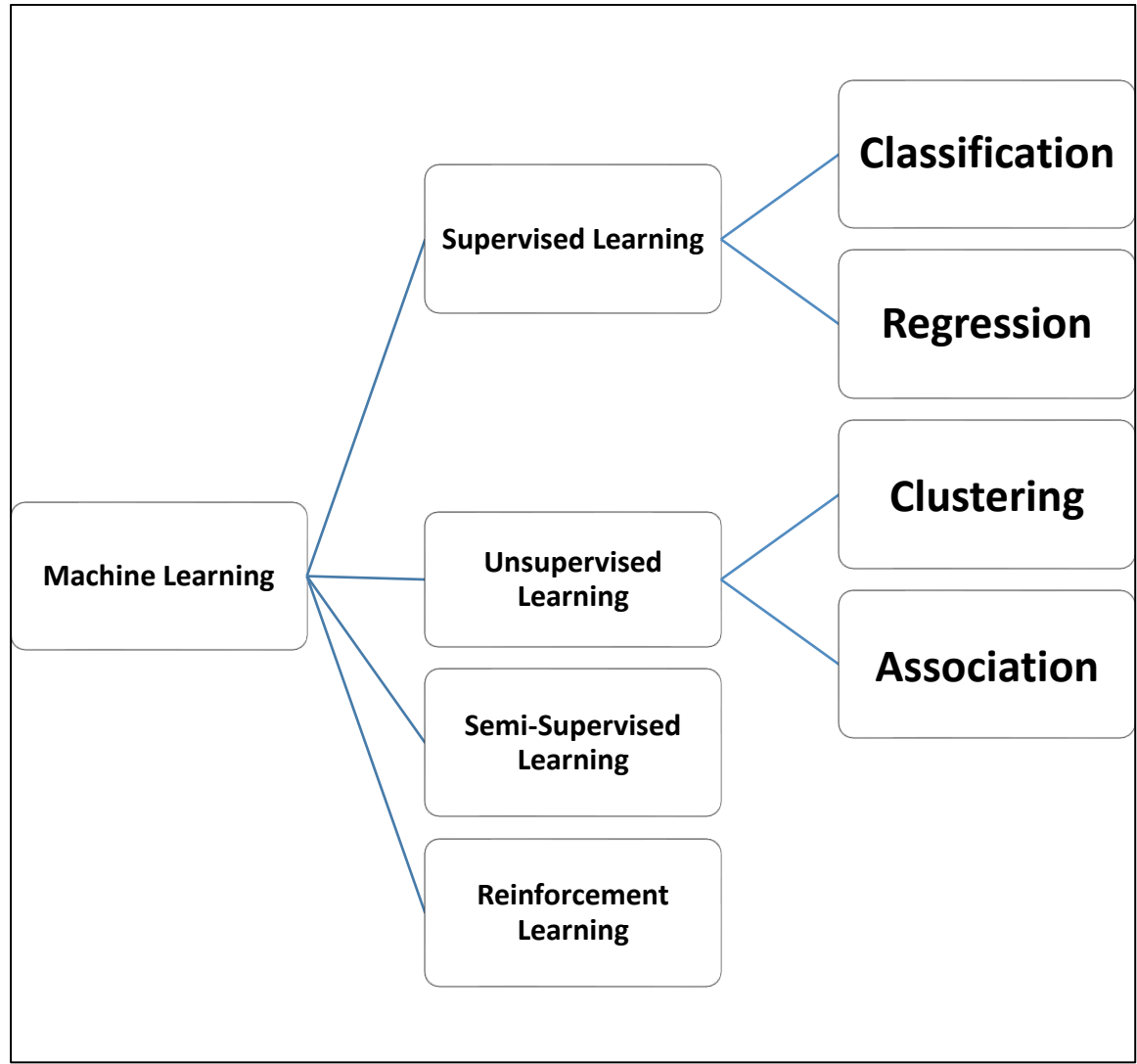


Figure 1 – Types of Machine Learning

1. SUPERVISED LEARNING –

It is based on supervision. In supervised learning, the machines are trained using 'labeled' dataset and based on the training, the output is predicted by the machines. There are parameters for both input and output in the labeled dataset. Supervised learning algorithms learn to locate the points between inputs and precise outputs. It has labeled datasets for both training and validation.

a. CLASSIFICATION –

When a classification problem involves a categorical output variable, like "Yes" or "No," Male or Female, Red or Blue, etc., classification algorithms are used to solve the problem. The categories that are present in the dataset are predicted by the classification algorithms. Examples of classification algorithms in the real world include email filtering and spam detection. Predicting categorical target variables—which stand for discrete classes or labels—is the focus of classification. For example, determining whether an email is spam or not, or determining if a patient is at high risk of heart disease. Algorithms for classification gain the ability to associate input features with one of the preset classes.

b. REGRESSION –

Regression problems involving a linear relationship between the input and output variables are solved using regression algorithms. These are employed in the prediction of continuous output variables, including market trends and weather forecasts. Conversely, regression is concerned with making predictions about continuous target variables, which are numerical values. For instance, projecting a product's sales or estimating the cost of a home based on its dimensions, location, and amenities. The ability to map input feature data to a continuous numerical value is acquired by regression algorithms.

2. UNSUPERVISED LEARNING –

Unsupervised learning is a kind of machine learning approach where an algorithm uses unlabeled data to find relationships and patterns. Unlike supervised learning, unsupervised learning does not require labeling the algorithm's target outputs, in contrast to supervised learning. Finding hidden patterns, similarities, or clusters in the data is frequently the main objective of unsupervised learning. These findings can then be applied to a variety of tasks, including dimensionality reduction, data exploration, and visualization. It uses data that isn't labeled or classified to train models. Unsupervised learning sorts out the unsorted dataset into various different groups or categories based on the similarities, patterns, and differences is the primary goal of unsupervised learning.

a. CLUSTERING –

The process of organizing data points into clusters according to how similar they are is called clustering. Without labeled examples, this method is helpful for finding patterns and relationships in data. When we wish to identify the innate groups within the data, we employ the clustering technique. It is a method of clustering the objects so that those that are most similar to one another stay in that group and are less similar to or not at all similar to those in other groups. Grouping clients based on their purchasing patterns is an illustration of the clustering algorithm in action.

b. ASSOCIATION –

Association rule is a technique to figure out the relationships between different items in the dataset. It is an unsupervised learning technique which can be used to uncover intriguing relationships between variables in a dataset. The primary goal is to identify dependencies between different data items and map those variables appropriately to maximize profit. This algorithm is used in continuous production, web usage mining, market basket analysis etc.

3. SEMI-SUPERVISED LEARNING –

Between supervised and unsupervised learning is semi-supervised learning. It uses a combination of labeled and unlabeled datasets during the training period and represents the middle ground between supervised learning (with labeled training data) and unsupervised learning (without labeled training data) algorithms. Semi-supervised learning's primary goal is to efficiently use all of the available data, as opposed to supervised learning's use of only labeled data. Both labeled and unlabeled data are used. It's especially helpful when getting labeled data requires a lot of money, time, or resources. It is quite helpful when you are dealing with expensive or time-consuming data sets. When training or learning from labeled data necessitates certain abilities and pertinent resources, semi-supervised learning is the preferred method.

4. REINFORCEMENT LEARNING –

It works on a feedback-based process that enables an AI agent (a software component) to automatically investigate its environment by hitting, trailing, acting, picking up lessons from past mistakes and enhancing its performance. Maximizing rewards is the aim of a reinforcement learning agent, as they are rewarded for good actions and punished for bad ones. Reinforcement learning exclusively uses the experiences of agents to learn, in contrast to supervised learning, which makes use of labeled data. An approach to learning that engages with the environment by generating actions and identifying mistakes is the reinforcement machine learning algorithm. The most important features of reinforcement learning are trial and error and delay. With this method, the model learns the behavior or pattern by continuously improving its performance through Reward Feedback.

1.3 SIGNIFICANCE OF MACHINE LEARNING

Machine Learning analyzes and interprets the patterns, structure of data to derive data driven recommendations solely on the input data. Organizations can make informed decisions to keep them ahead of the competition. Since ML models are iterative, they can easily adapt to new data to produce reliable and accurate results while identifying the trends in customer behaviour and operational business patterns for the development of new products. ML is used to build predictive models that predict business outcomes. It reduces costs, interprets the customer requirements, recommends best products and services for its customers according to their interest, and mitigates risks for the organizations.

The branch of computational science known as machine learning (ML) is concerned with the analysis and interpretation of data patterns and structures to facilitate learning, reasoning, and decision-making in situations where human interaction is not present. Put simply, machine learning enables the user to provide vast amounts of data to a computer algorithm, which then uses the input data alone to analyze, recommend, and make decisions. Should any adjustments be found, the algorithm can use that data to make better decisions in the future. Data-driven learning: By analyzing enormous volumes of data, machine learning algorithms allow computers to recognize patterns and extract relevant features. They can also improve the capabilities of AI systems by making precise judgments or predictions based on observed data.

Using labeled or unlabeled data, machine learning is essential to training and developing models. Additionally, it enables algorithms to modify parameters and enhance performance over time, producing AI models that are accurate and flexible. Machine learning is particularly good at automatically deriving relevant features from complex data sets, such as text, audio, images, and sensor data. It gives AI systems the data they need to perform tasks like prediction, classification, and decision-making. By using machine learning algorithms to learn from historical data, AI systems are able to classify new instances, detect anomalies, and make well-informed predictions. This helps with medical diagnosis, fraud detection, and image identification.

1.4 SCOPE OF MACHINE LEARNING

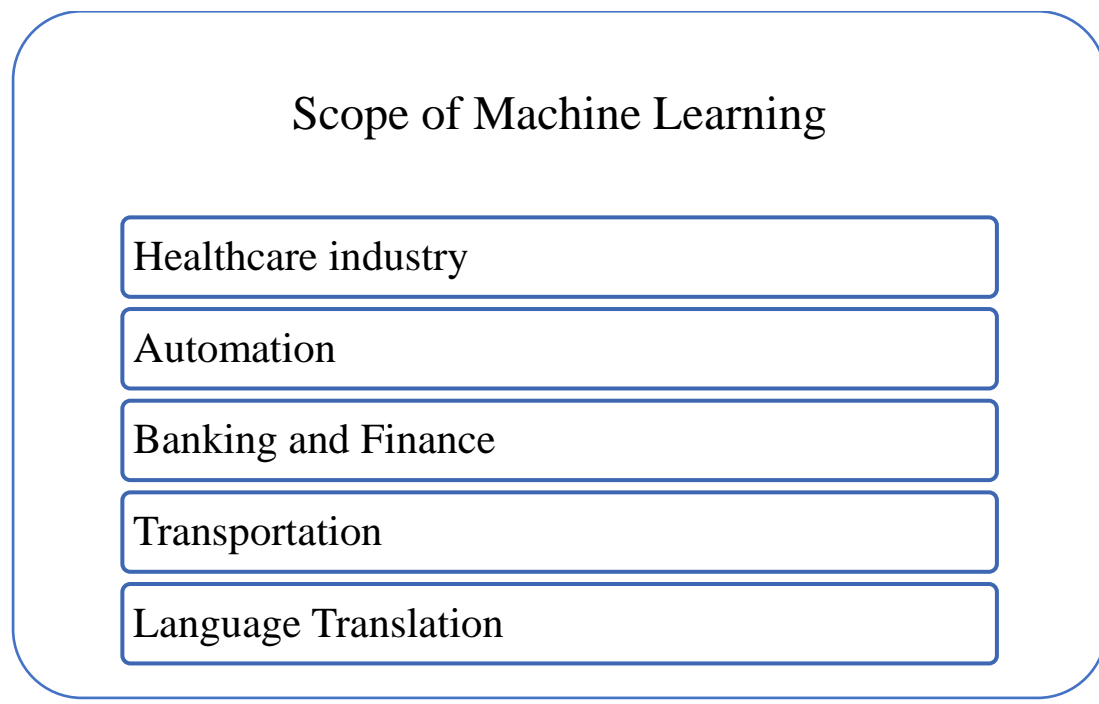


Figure 2 – Scope of Machine Learning

Machine Learning is used widely in every industry with a wide range of applications involving collecting, organizing, analyzing and responding to a large amount of data. Machine learning is widely applicable and found in a wide range of industries, such as healthcare, finance, retail, and many more. As more businesses use this technology to enhance their business procedures, its reach grows daily. All machine learning does is make predictions from past data. It allows machines to make decisions based on data, which is more effective than explicitly programming them to perform specific tasks. These algorithms' layout allows for the exposure to fresh data, which enables organizations to gain insights and enhance their tactics.

a. **HEALTHCARE INDUSTRY –**

ML is used widely by the healthcare researchers for image classification, natural language processing for better outcomes, detection of disease, its prediction and diagnosis in terms of an x-ray, ultrasound and CT-scan. Nowadays, professionals make use of robots to perform difficult operations for greater accuracy. They also implement laser technology and analyze the conditions to make better and quick decisions. Drug discovery, disease diagnosis, and medical image analysis are all aided by machine learning. It also enhances healthcare operations and allows for the personalization of treatment plans.

b. **AUTOMATION –**

It makes the system automated and performs the repetitive tasks with minimal human intervention. It performs the iterative tasks with no errors and in comparatively lesser time. Robotics and automation heavily rely on machine learning. It can be applied to the development of computer vision applications, automated surveillance systems, personal bots, and self-driving automobiles. The process of automating machine learning tasks to solve real-world problems is known as '**autoML**'. It attempts to make machine learning algorithms and methods easier for non-experts to use, enabling them to take advantage of machine learning's power without requiring in-depth expertise in the field.

c. **BANKING AND FINANCE –**

ML helps in fraud detection, risk management, document analysis, chatbots, portfolio management, anomaly detection, credit score detection, KYC processing etc. It also detects frauds in online transactions, loan eligibility and approval, predicts market risk for share market and trading using LSTM models. With algorithmic trading, credit scoring, and risk assessment, machine learning is revolutionizing the financial industry. These tools are used by financial institutions to improve their services and operations. Machine learning is used by financial institutions to identify fraudulent transactions and stop money laundering.

d. **LANGUAGE TRANSLATION –**

It is used for translation of one language into another. Similarly, it also generates captions for images, identifies the text, understands the sign languages and translates it into native languages. It can detect your voice and search content with the help of the voice of the user. Face recognition, face mask recognition can be done effectively using ML. A branch of machine learning called neural machine translation makes use of artificial intelligence to learn languages and enhance translation quality over time. Spoken or written language can be translated instantly with the use of machine learning algorithms in real-time translation applications. This has useful applications in business, travel, and international communication, where real-time translation can help people communicate effectively even when they are speaking different languages.

e. **TRANSPORTATION –**

ML is used in transportation as a navigation system which optimizes the routes by showing the shortest route to reach from destination A to destination B. It can reduce fuel consumption, improve efficiency of transportation systems. It is also used for self-driving autonomous cars like Tata and Tesla. Machine learning is used to evaluate and decipher massive amounts of data gathered from multiple sources, including sensors, cameras, and GPS devices. It supports traffic flow optimization, congestion forecasting, and real-time traffic monitoring. Route planning, traffic signal control systems, and transportation network management can all be enhanced by the use of machine learning algorithms.

1.5 MACHINE LEARNING LIFE-CYCLE

The ML Life-cycle guides the machine learning model from the problem statement to the deployment in a structured way. It can maintain scalable and sustainable solutions to deliver tangible value. It is a cyclic process that is used to build an accurate Machine Learning project. It is important to first understand the research problem, in order to find a solution.

Machine learning life-cycle is an end-to-end process which begins with problem definition to model deployment and maintenance while providing a structured framework for the development of different well-defined, data-driven machine learning models. The beginning of any machine learning model is by collecting the required data for training.

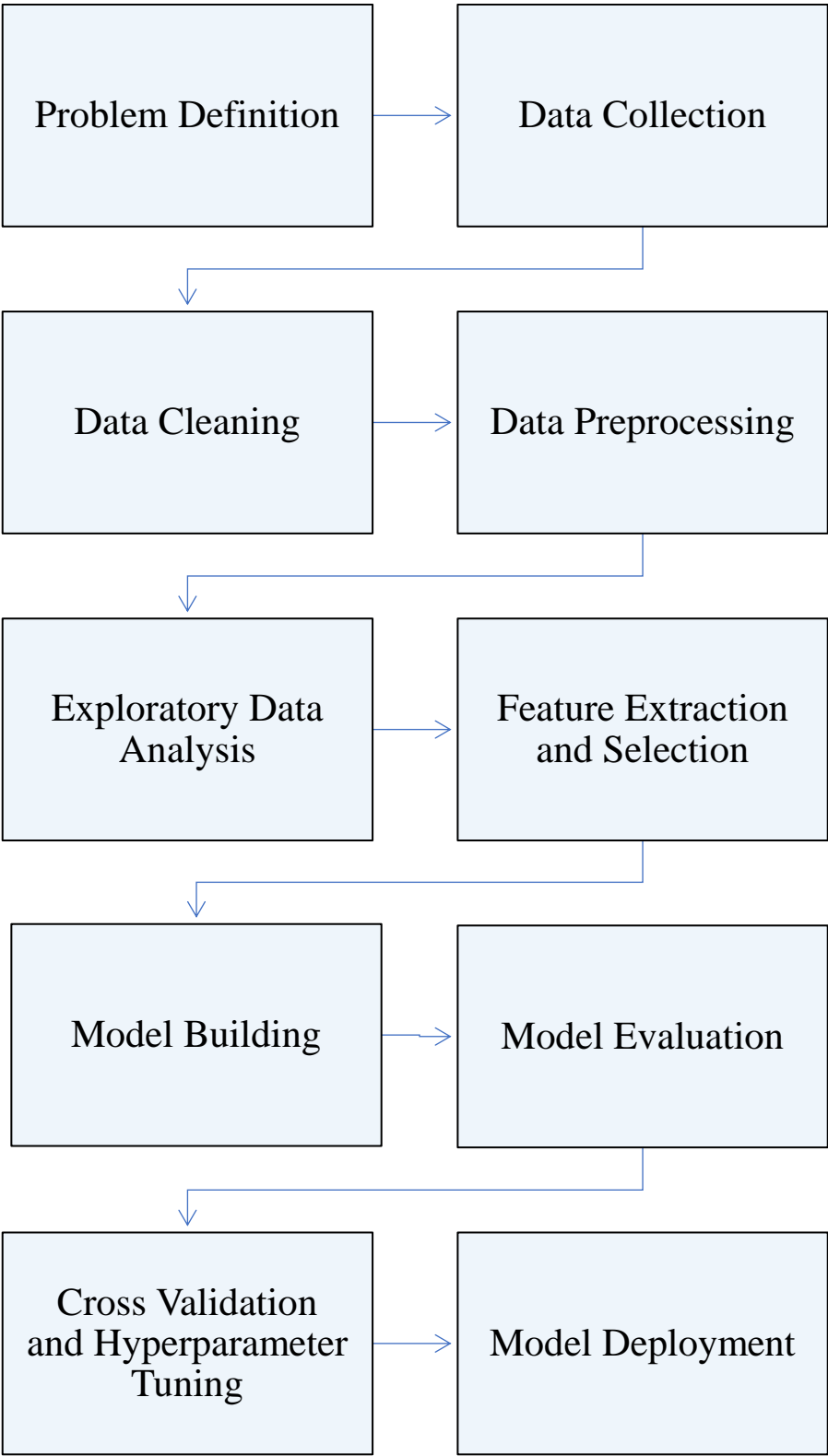


Table 1 – Machine Learning Lifecycle

1. PROBLEM DEFINITION –

In this phase, the stakeholders and the management collaborate to identify the business problems in a comprehensive manner. It is important to establish a foundation for the life-cycle. It also helps to articulate the research objectives, desired outcomes for predictive modelling or recommender systems etc and to analyze the scope of the study. It is important to understand the purpose of the problem, its nature, its sample size and research design to implement the model. It is essential to learn whether there is availability of resources like computing, storage network, and qualified professionals to carry out the study. A machine learning expert should have a clear understanding of the deep learning model process and results, its robustness, scalability and the prediction accuracy of the model. The implementation of ML model should be legal and ethical. It should ensure to solve the current problems as well as the future problems and these models should be cost-effective as well.

2. DATA COLLECTION –

The goal of Data collection is to ensure identify the target audience and obtain the internal data required to train the model. One needs to identify the different data sources like files, database, the internet, etc. The data must be relevant and must ensure the robustness and generalizability which must be complete and up to date. ML experts need to check for the data volume, shape, quality and certain ethical considerations for the subsequent phases of machine learning life-cycle. The data must be relevant to define problems and overcome it, it should be diverse. The next step is to label the collected data. When we obtain a coherent set of data, we call it as a ‘Dataset’.

3. DATA CLEANING –

Now that we have gathered all the required data, it is important to make sure that the data is clean, i.e. it must address the issues like missing values, outliers, duplicate values, unique values and several other inconsistencies of data. It should explore the nature of the data to understand the format, patterns and quality of data. Here, we check for the general trends while finding the correlations to the variables with each other or the correlations of variables with target variables. We create a data pipeline while removing the noise from data and perform quality data verification.

4. DATA PREPROCESSING –

The data preprocessing step involves standardized formats, scaled values and encoding the categorical data for consistencies. It also deals with imbalanced classes and values with data normalization, data scaling and data augmentation. It is used to clean, transform, sort the data and to integrate raw data and make it suitable for the accuracy of analysis. It uses techniques like data imputation, removal and encoding the categorical variables to combine data from different sources into one dataset. In this step we figure out the data storage solutions and store the metadata to create ETL pipelines to ensure a continuous stream for model training.

5. EXPLORATORY DATA ANALYSIS (EDA) –

Exploratory Data Analysis is a pivotal phase in order to analyze data and summarize their characteristics. It makes use of statistical graphs and data visualization methods to gain insights and understand the data beyond formal modelling and hypothesis testing. EDA can reveal patterns and detect anomalous events to identify errors.

It is the unearthing of trends, policies, challenges and context for subsequent decisions. It makes use of data visualization, summary statistics and correlation analysis for the comprehensive view of data to make informed decisions in the feature extraction process while encompassing the data intricacy.

6. FEATURE EXTRACTION AND SELECTION –

The main purpose of this step is to build an ML model which analyzes data using analytical techniques and review of outcomes. It is a transformative process which transforms raw data into predictor model. It creates new features and also transforms the existing features by using domain expertise to determine the relationship between the variables. It also identifies the subsets of feature variables which impact the performances of the models. It is important to balance the feature sets for prediction accuracy while it minimizes the computational complexity.

7. MODEL BUILDING –

The next step is to build the model and split it into training and testing datasets using different machine learning or deep learning algorithms to determine patterns, rules and features. The model needs to be exposed to the historical data to derive the relationships and dependencies with the dataset. Since model building is iterative, it adjusts the parameters in order to minimize the errors and enhance the accuracy of predictive models. The model is fine-tuned and validated rigorously to ensure that the data model generalizes well to unseen or new data to establish a foundation for the real-world values or input values entered by the clients. It helps in designing the architecture of the model and selecting appropriate frameworks on it in order to optimize its performance. We test the model by assigning the trained dataset to it, in order to derive results based on the speed, accuracy and security of the model.

8. MODEL EVALUATION –

Evaluation of the performance of the model is necessary to assess how well the model is performing and what improvements does it need. There are different evaluation metrics used to assess the performance of machine learning models that requires testing against the validation datasets to employ the accuracy, f1-score, recall, confusion matrix, area under the ROC curve and precision metrics. It is a checkpoint which provides insights into the strengths and improvements of the model. It evaluates different algorithms using same evaluation metrics to choose the best algorithm to carry out a specific task. It also helps us to determine if the model is overfitting which means that the model is performing well on training data but very poorly on the new data, or, underfitting meaning the model is performing poorly on the training data as well as the new data.

9. CROSS VALIDATION AND HYPERPARAMETER TUNING –

It is important to optimize the model performance. Cross validation is one such technique which is used to assess the performance of machine learning model on unseen data and how well does it adapt to the new data to estimate the performance of model on an independent dataset. It splits the dataset into multiple folds, trained and evaluated on the remaining validation set. It is a repetitive process with different subsets of data and multiple iterations using the k-fold cross validation. Each fold sets as a validation set once and is repeated k-number of times. Hyperparameters are the parameters which aren't learned from the data, they are set by the user before the training of model. It finds the actual values to maximize the performance by systematically searching through various combinations using a validation set on the grid search. It is a predefined set of hyperparameter value. Another approach is the randomized search where there are random combinations of hyperparameter values that are sampled from a predefined distribution.

10. MODEL DEPLOYMENT –

When the model is built and evaluated successfully, the next step is to deploy it. It involves the cumulation of machine learning models by transforming the trained model into real-world application. If the model shows accurate result with an acceptable speed, as per the client requirement, then it is deployed into real system. It is generally deployed on cloud servers, local servers, web browsers, packages as software and edge device which can be used as web API, plug-ins and dashboard in order to access the predictions. It is also important to monitor the model and making adjustments as per the necessity to maintain its effectiveness through the years.

1.6 HEART DISEASE OVERVIEW

The Cardiovascular system is a part of circulatory system that circulates blood. It also includes lymphatic system. The cardiovascular system consists of three components: the heart, blood, and blood vessels. The heart is essentially a pump that circulates blood through the vessels. It has two sides each of which has two chambers. The left side of heart pumps oxygen-rich blood to body tissues where it unloads oxygen and picks up carbon dioxide and is known as pulmonary circuit. This resulting deoxygenated blood returns to heart's right side to complete the cycle and is known as **systematic circuit**.

There are four valves which ensure one-way blood flow through heart oxygen poor blood flows from the right atrium to right ventricle to pulmonary arteries, while oxygen rich blood moves from the left atrium to left ventricle to the aorta. The heart is enclosed in a double walled protective sac called the pericardium cavity which contains a fluid serving as a lubricant and allows the heart to contract and relax within minimum friction. The heart wall has three layers – the outer layer, epicardium lines the pericardial cavity, the inner layer, endocardium, lines the heart chambers and valves and is continuous with the endothelium of blood vessels.

The thick middle layer myocardium is the muscle tissue responsible for the beating of heart. The contraction of the heart muscle is initiated by electrical impulses known as action potentials. The heart generates its own electrical stimulations. The blood contains two main components – plasma which is an extracellular fluid and formed elements such as red blood cells, white blood cells and platelets. The circulatory system of blood is closed. Cardiovascular disease (CVD) describes a range of conditions that affects the heart and/or blood vessels also known as heart disease. It includes various diseases such as coronary heart disease, cerebrovascular heart disease, peripheral arterial disease, rheumatic heart disease, congenital heart disease and pulmonary embolism. Heart attacks and heart strokes are caused by blockages that prevents blood from flowing to heart or brain. Heart failure is the problem with pumping functions which causes fluid buildup and leads to shortness of breath.

1.7 HEART ATTACK OVERVIEW

The heart is the engine of the body. It keeps us alive by pushing blood wherever it is needed. Heart attack feels like pressure or burning in the centre of the chest. It can be mild or strong. When the blood flow that brings oxygen rich blood to heart muscles is severely reduced or cut-off in a nearby artery, there is an occurrence of heart attack. It is also known as **myocardial infarction**. It might be caused due to the buildup of fat narrowed by coronary arteries.

Chest pain, pressure, tightness, shortness of breath, nausea, sweaty or clammy skin, dizziness, anxiety and heartburn are some of the symptoms to identify a heart attack. When a plaque of coronary artery breaks open a blood clot forms, which can block the blood flow to the part of heart muscle. It causes damage to the heart muscle and is determined by the size of the area supplied by the blocked artery as well as the time between injury and treatment. It should be opened as soon as possible to reduce the damage. A sudden spasm might tighten the coronary artery also result in heart attack.

A heart attack is a sudden rupture of a plaque where some sludgy areas in the arteries blocks the artery altogether and causes pain to the heart. If it is left long enough, meaning if stent isn't inserted or bypass isn't done to get rid of the clot in some way, it can lead to death. The plaque build-up blocks arteries and causes heart attacks, the plaque buildup can be prevented with the help of healthy eating and exercise. The heart attack symptoms can vary from a person to person and knowing what they look and feel like could be lifesaving.

Men and women may have different heart attack symptoms. Men experience shortness of breath, chest pain and tingling or discomfort in arms, back, neck, shoulder or jaw. Women experience extreme fatigue, severe weakness, nausea or vomiting, heart burn-like feeling, and cold sweats. **Atherosclerosis** is the leading cause of heart attacks worldwide, where **Athero** talks about the **vessels in our heart** and **sclerosis** which means the **fibrosis or hardening** of those vessels. It is among the most prevalent heart conditions worldwide.

When the heart gets clogged with the bad cholesterol, it can create a clog. The immune cells of the body come along while clearing up the fat and absorbing it into themselves trying to repair the damage. The immune cell, the macrophage takes the fat inside the body and accumulates it. It ends up looking like a cell full of fat molecules and is known as “foam cells”. This foaming process creates a backlog of fatty immune cells which clog the vessels wall with more and more cells clogging the pipe and creating a problem with the blood flow. This often presents as blood pressure, which actually means that the blood is trying to get through the small area of vessel which can lead to a rupture. Everything can be released in a rupture. It ends up in a clot and one can have a heart attack or can cause stroke. Men have a higher risk of cardiovascular disease earlier in life, while women are more at risk of cardiovascular diseases after menopause. Women may also experience different symptoms of heart attack which may go unrecognized.

1.8 RISK FACTORS OF CARDIOVASCULAR DISEASES

Cardiovascular disease can be caused by a variety of risk factors we cannot control. The major risk factors of cardiovascular diseases are unhealthy diet, physical inactivity, tobacco and alcohol use which might result into high blood pressure, raised blood glucose, raised blood lipids, and obesity.

A risk factor is something that predisposes you to something else, i.e. the stress that pathologically high blood pressure puts on your vessels would predispose you to developing coronary artery disease.

Risk factors such as high blood pressure (hypertension), high cholesterol, type II diabetes, genetic history of heart diseases, diet which is high in sodium, sugar and fat, gestational diabetes, chronic inflammatory conditions, and kidney diseases might also result in cardiovascular diseases. Lack of sleep – the people who have regular sleep are most unlikely to develop coronary heart disease.

According to the Framingham Heart Study, there are two types of risk factors. The risk factors for heart diseases are divided into two types, namely Modifiable and Non-modifiable:

Modifiable Risk Factors – As the name suggests, these factors can be modified using lifestyle changes, or nutritional changes and regular exercises while consuming medications under a doctor's observation, wherever required. Some modifiable risk factors include high blood pressure, high blood sugar, high level of fats (lipids) in the blood, high level of amino acids homocysteine in the blood, overweight, smoking, and sedentary lifestyle. It is extremely essential to quit smoking, be physically active, eat well, exercise daily and sleep well while achieving a healthy weight, manage the stress and anxiety levels. Smoking, high blood pressure, high cholesterol, high triglyceride levels, diabetes and pre-diabetes, physical inactivity, metabolic syndromes, obesity etc are some factors which can be controlled by an individual to prevent cardiovascular diseases in the future.

Non-modifiable Risk Factors – As the name suggests, these factors cannot be controlled.

Some non-modifiable risk factors are as follows:

1. **Gender** –

Men are at a higher risk of developing a cardiovascular disease at a premature age whereas women are more prone to developing cardiovascular diseases after menopause at a higher risk than men where the symptoms can be different for everyone or can be 'silent'.

2. **Age** –

As the age increases, the likelihood of developing heart diseases increases too.

3. **Family History/ Genetics** –

If a person has a family history of premature heart diseases and problems, then he is more likely to inherit from his family.

4. **Ethnicity** –

Certain ethnicities are often indirectly at a higher risk of developing cardiovascular diseases. Example – People from Africa and Asia tend to have higher rates of hypertension than people of non-African descent.

Cholesterol plays an important role in our bodies such as producing hormones and vitamin D. It is produced by the liver and also from the food that we consume on a day-to-day basis. When cholesterol is combined with the proteins, it forms a lipoprotein. There are two types of lipoprotein in your blood, **Low density lipoproteins (LDL)**, also called as bad cholesterol, can cause plaque build-up on the walls of arteries and too much of LDL can increase the risk of cardiovascular diseases. Although **High density lipoproteins (HDL)** also known as good cholesterol can help the body get rid of the bad cholesterol. Too much HDL can increase your risk of cardiovascular disease. It can block any artery in your body and limit blood flow to your heart muscle. This can cause chest pain also called angina and lead to heart attack.

As the blood circulates through our bodies, it exerts pressure against the walls of the arteries. This pressure is measured with a blood pressure device. It is measured by two numbers – the top number is called **Systolic** and measures the pressure in your arteries when your heart is pumping. The lower number is called **Diastolic** and measures the arteries when they are resting and refilling. High blood pressure is something that needs to be avoided in order to keep the arteries and the heart healthy and clear of cardiovascular diseases. There are several risks cardiovascular diseases associated with high blood pressure namely heart attack, stroke, heart failure, dementia, kidney diseases, eye problems etc.

When your body reacts to the changes physically, mentally and emotionally, these reactions are called stress. Stress can be positive or negative which keeps us alert and aware of potentially dangerous situations. Headaches, upset stomach, high blood pressure, chest pain, insomnia are all symptoms of high-stress lifestyle. It can worsen skin problems, heart conditions, diabetes, anxiety and depression.

1.8 TYPES OF CARDIOVASCULAR DISEASES

a. CORONARY ARTERY DISEASE –

Coronary Artery Disease is also known as **Coronary Heart Disease**. It is the most common type of heart disease which develops when arteries that supply blood to the heart gets clogged with plaque that contains cholesterol, thus it hardens and narrows. Less oxygen and nutrition reach the heart because of the decreased blood supply. There is a risk of heart failure as the heart muscles weaken. It affects the coronary arteries, which are the primary blood vessels that supply blood to the heart. The heart muscle receives less blood when a patient has CAD. Coronary artery disease is typically brought on by an accumulation of fats, cholesterol, and other materials in and on the artery walls, a condition known as atherosclerosis. The accumulation, known as plaque, narrows the arteries.

b. ARRHYTHMIA –

It basically refers to an **irregular heartbeat**. When the electrical impulses that instruct the heart to beat malfunction, a heart arrhythmia happens. Too fast or too slow heartbeats are possible. There may also be variations in the rhythm of the heartbeat. An arrhythmia of the heart can feel like a racing, pounding, or fluttering heartbeat. Heart arrhythmias can sometimes be benign. The heart may beat too quickly or too slowly or erratically because the electrical impulses that co-ordinate the heartbeat do not work correctly. Hence a person might notice a feeling of fluttering or a racing heart.

c. MYOCARDIAL INFARCTION –

It is also known as **heart attack**. Myocardial Infarction occurs on by a reduction in or cessation of blood flow to a section of the heart. An MI could be "silent," going unnoticed, or it could be a catastrophic event that results in hemodynamic decline and abrupt death. It involves an interruption of blood flow to the heart which damages a part of heart muscle. A plaque, blood clot or both in a coronary artery is a common cause of heart attack as an artery suddenly narrows or spasms. The fundamental mechanism of a Myocardial Infarction is typically the total blockage of a coronary artery brought on by the rupture of an atherosclerotic plaque.

d. CONGESTIVE HEART FAILURE –

The intricate clinical syndrome known as congestive heart failure (CHF) is marked by an inefficient heartbeat, which compromises the body's blood supply. Any condition affecting blood ejection from the ventricles into the systemic circulation or ventricular filling can lead to CHF. It occurs when there is a problem with pumping or relaxing function of the heart. It can occur because of untreated coronary artery disease or high blood pressure. A chronic illness known as congestive heart failure, or heart failure, occurs when the heart is unable to pump blood effectively enough to meet the demands of the body. Your heart is still functional. However, blood accumulates in other areas of your body because it is unable to handle the normal volume of blood. It typically gathers in your legs, feet, and lungs.

CHAPTER 2 – REVIEW OF LITERATURE

CHAPTER 2 – REVIEW OF LITERATURE

Cardiovascular disease (CVD) is a type of heart disease that continues to be a major cause of death accounting for over 30% of all deaths. CVDs is the primary cause of morbidity and mortality which accounts into 70% of fatalities. A study by the Global Burden of Disease Study 2017 found out that more than 43% global deaths are accounted because of CVDs.

According to World Health Organization, the overall number of deaths would rise to 23.6 million by 2030 with heart stroke and disease being the leading causes. It is important to utilize machine learning algorithms to anticipate the chances of heart attack and save lives while decreasing the cost burden on the society. Machine learning plays a vital role in the healthcare industry, since it can detect, predict and diagnose different kinds of diseases like diabetes, Parkinson's disease, autism, heart diseases etc. It predicts the likelihood of developing such diseases. ML can be used to detect and forecast disorders. It will act as a tool for the professionals to detect cardiac diseases at an early stage, while preventing damage. It will better diagnose the treatment required to the patient while avoiding serious impacts. To predict heart disease prediction, machine learning algorithms like Support Vector Machines (SVM), Random Forest Classifier, Decision Tree Classifier, Logistic Regression etc are utilized which acts as a tool for decision-making tools on individual information.

Over the world, heart disease has become one of the leading causes of death. The World Health Organization estimates that heart-related illnesses claim 17.7 million lives annually, or 31% of all deaths worldwide. Heart-related illnesses are now the main cause of death in India as well [1]. As per the September 15, 2017 release of the 2016 Global Burden of Disease Report, heart diseases claimed the lives of 1.7 million Indians in 2016. Heart-related illnesses lower an individual's productivity and raise health care expenses.

According to estimates from the World Health Organization (WHO), heart-related and cardiovascular diseases may have cost India up to \$237 billion between 2005 and 2015. Therefore, it is critical to have a practical and accurate prediction of heart-related diseases.

To create a model for the study, several machine algorithms are used here, including logistic regression (LR), Naïve Bayes (NB), Random Forest Classifier (RF), K-Neighbors Classifier (KNN), Decision Tree Classifier (DTC), and Support Vector Classifier (SVC). The projected model's capability is adequate and proficient enough to forecast the likelihood of a heart attack in a human body through the compilation of certain physiological terms. Therefore, using machine learning to predict heart attacks in advance can be a good idea so that we can take the appropriate precautions and treatments.

Since heart disease has a complicated nature, it needs to be treated carefully. Inability to do so could damage the heart or result in an early death. To identify different types of metabolic syndromes, data mining and the perspective of medical science are employed. In this project, we can observe that using the different machine learning algorithms, we can find out whether the person is suffering from a heart disease and also, we can predict if the person is likely to get heart attack.

CHAPTER 3 – RESEARCH METHODOLOGY AND OBJECTIVES

CHAPTER 3 – RESEARCH METHODOLOGY AND OBEJCTIVES

➤ RESEARCH OBJECTIVES –

1. To analyze the heart disease and heart attack prediction using six different machine learning algorithms.
2. To study different algorithms and analyze them.
3. To study the likelihood of developing heart problems based on different age, genders and total values according to the dataset.
4. To research and find out the accuracy scores of heart attack prediction models using machine learning.
5. To check the accuracy scores, precision scores and f1-score and find the algorithm which provides the best accuracy.

➤ RESEARCH PROBLEM -

Implementation and analysis of Machine Learning algorithms for Heart Disease and Heart Attack Prediction and to analyze the age, and gender wise distribution of patients suffering from heart problems.

➤ RESEARCH DESIGN –

Exploratory Research Design has been adopted in this study to explore the different algorithms for the analysis and prediction of heart attack prediction. It is a valuable preliminary research method used to gain a better understanding of a problem or issues which are yet to be investigated.

It clarifies or defines the parameters of a problem, refine a general idea into more specific research problem and determines the research priorities. Since exploratory research can be qualitative or quantitative both, in this study however, the nature of research is on the quantitative data which makes use of both the categorical as well as the continuous variables like age, gender, chest pain, fbs, etc.

➤ **TYPE OF DATA USED –**

Secondary Data was used for the study of heart attack prediction using Machine Learning. The research papers were read using Google Scholar. This data set dates from 1988 and it consists of four databases: Cleveland, Hungary, Switzerland and Long Beach V. It contains 76 attributes, including the predicted attribute, but all the published experiments refer to using a subset of 14 of them. 0 indicates no disease, and 1 indicates disease of the heart condition of the patient in the target field.

➤ **DATA COLLECTION METHOD –**

Data was collected using the internet on www.kaggle.com and using the ‘**Heart Disease Dataset**’ containing the ‘**heart.csv**’ dataset file. The dataset contains 1025 columns with 14 rows and contains some medical information of patients which tells whether the patient is getting a heart attack or how likely to get a heart attack. Using the information, we will explore the dataset while classifying the target variable using different Machine Learning models to find out which algorithm is suitable for the dataset.

➤ **SAMPLE SIZE –**

The Heart Disease Dataset contains a total of 1025 rows and 14 columns in the heart.csv file. The dataset contains the minimum age of 29 to the maximum age of 78 years of different males and females. It contains 14 columns with different parameters to detect the heart problems.

➤ **DATA ANALYSIS TOOL –**

Python version 3.12.0

Anaconda version 23.3.1

Jupyter Notebook 6.5.2

The data analysis tool will be the six machine learning algorithms used to create a prediction model. They are as follows:

Table 2 – Machine Learning Algorithms

Sr No	Machine Learning Algorithms
1	Logistic Regression
2	Naïve Bayes
3	Random Forest Classifier
4	K-Nearest Neighbors
5	Decision Tree Classifier
6	Support Vector Machines

1. Logistic Regression –

Logistic Regression is one of the most widely used machine learning algorithm that makes use of the supervised learning technique. Logistic Regression is mainly used for solving the classification problems and predicts the categorical dependent variable using the set of independent variables. It checks the probabilities and classifies the new data using continuous and discrete variables. Here, the dependent variable should be categorical in its nature and the independent variable should not have multi-collinearity. There are three types of logistic regression namely Binomial Logistic Regression, Multinomial Logistic Regression and Ordinal Logistic Regression. It delivers the binary outputs which is only limited to two possibilities. This algorithm is extensively used for predictive modelling to check whether an instance belongs to a specific category. Logistic regression uses its logistic function called as sigmoid function, which is an S shaped function. It can easily convert the real values to either 0 or 1. The key prediction is based on the maximum likelihood and it does not evaluate the co-efficient of determination.

The Logistic Regression Equation is as follows

$$y = e^{\frac{b_0 + b_1X}{1 + e^{(b_0 + b_1X)}}}$$

where, x = input value,

y = predicted output,

b₀ = bias or intercept term

b₁ = coefficient for input (x)

2. Naïve Bayes Classifier –

The Naïve Bayes algorithm, which solves text classification problems for high-dimensional data sets, is an approach to supervised learning founded in the Bayes theorem. It is called Naïve because the occurrence of a specific feature is independent of other occurrence of other features is assumed. Naïve Bayes is a probabilistic theorem that predicts the probability basis of any object. It is mainly used in sentiment analysis and spam filtration. Naïve Bayes classifiers come in three varieties: Bernoulli, Multinomial, and Gaussian.

The Bayes theorem is as follows:

$$P(A | B) = P(B | A) P(A) / P(B)$$

Where, $P(A)$ is the prior probability of hypothesis before observing evidence

$P(B)$ is marginal probability of evidence

$P(A | B)$ is posterior probability of hypothesis A on observed event B

$P(B | A)$ is likelihood probability of evidence where the probability of hypothesis is true.

3. Random Forest Classifier –

Random Forest Classifier too is a supervised learning technique. It is based on ensemble learning and can be used for both classification as well as regression. It is a classifier that contains a number of decision trees on various different subsets to improve the predictive accuracy. It does not rely on only one decision tree, instead, it takes the prediction from each tree based on majority votes and helps in the prediction of final output. The greater is the number of trees in forest, the greater is the accuracy. It also helps in prevention of overfitting.

4. K-Nearest Neighbor Algorithm –

KNN is a non-parametric algorithm which does not make any assumption in underlying data. It is quite simple to execute and robust to noisy data. It assumes the similarity between new case data and available data into similar categories. It stores the available data and classifies new point into a well suite category by using KNN. First it selects the K number of neighbors, then it calculates the Euclidean distance of K number of neighbors. According to the Euclidean distance, it takes the K nearest neighbors and counts the number of data to assign new data points to the category where the number of neighbors is maximum.

$$\text{Euclidean Distance Formula} = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$$

5. Decision Tree Classifier –

The decision tree classifier is a non-parametric supervised learning technique that is primarily used for classification problems. Each leaf node in the model represents the outcome, the internal nodes represent the dataset's features, and the branches represent the decision rules. There are two nodes namely Decision Node and Leaf Node. It is like a flowchart-like structure consisting of nodes that represents the decisions, attributes or the branches and leaf nodes representing the outcomes. The graphical representation for getting all possible solutions to a problem based on conditions. The algorithm begins at the tree's root node, compares the value of the root attribute with the record attribute, then moves on to the next node by following the branch. It is mostly used for decision solving problems and thinks about all the possible outcomes. They do not need data normalization but it requires data preparation. It can handle both categorical data, numerical data as well as multi-output problems.

6. Support Vector Machines

Support Vector Machines is a supervised learning technique which is used for linear or non-linear classification, regression, outlier detection tasks. It finds the optimal hyperplane in N-dimensional space to separate different classes in feature space. The hyperplane is the decision boundary is used to separate the data points of different classes in a feature space. SVM is of two types: Linear SVM and Non-linear SVM.

It is used for linearly separable data, which means that the dataset can be classified by using a single straight line. Non-linear separable data is used for non-linearly data which means a dataset cannot be classified using a straight line.

For SVM Linear Classifier:

$$y = \begin{cases} 1 : w^T x + b > 0 \\ 0 : w^T x + b < 0 \end{cases}$$

CHAPTER 4 – DATA INTERPRETATION AND ANALYSIS

CHAPTER 4 – DATA INTERPRETATION AND ANALYSIS

The Data Flow is as follows:

1. Importing Libraries –

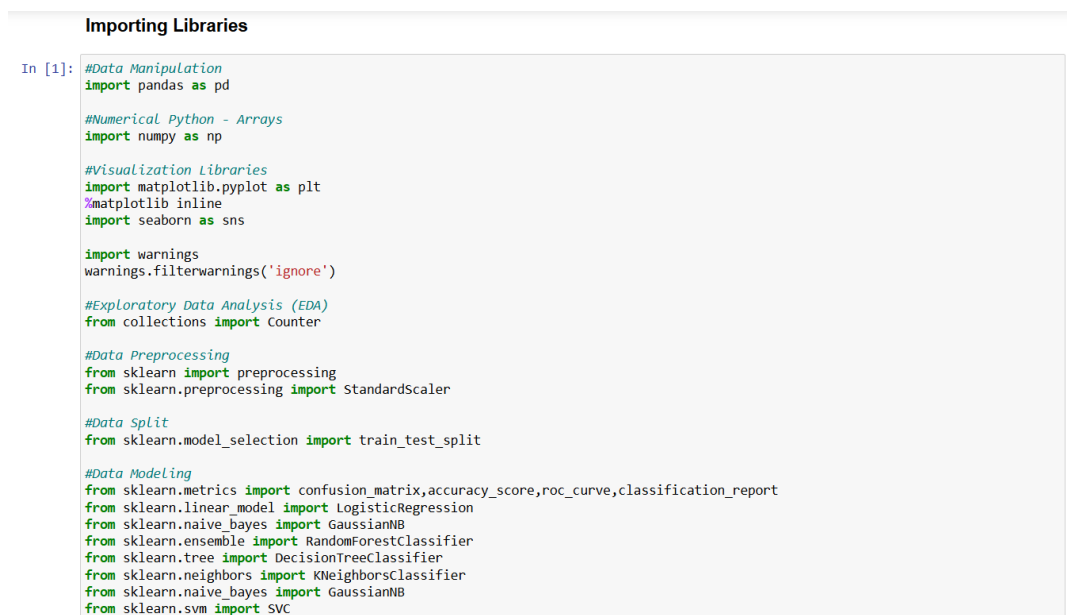


Image 1 – Importing Libraries

The first and foremost step is to import the required libraries. We imported different data manipulation libraries, data visualization libraries, warnings and filter them. Data preprocessing libraries, split data into training dataset and testing dataset. Next, we imported the Data modelling libraries for different machine learning algorithms. We imported these required libraries to analyze and predict heart attack.

1. import pandas as pd –

Pandas is a python library used to analyze data and works with the datasets. It can perform various tasks such as data cleaning, data manipulation, analysis, and exploration, and it may derive conclusions based on statistical theories.

2. import numpy as np –

Numpy is a python library to work with arrays and contains functions of linear algebra and matrices which stands for **N**umerical **P**ython. It is stored at a continuous place in memory and aims to provide an array object that is up to 50x faster than traditional python lists.

3. import matplotlib.pyplot as plt –

It is a collection of command style functions where each pyplot function makes changes to a figure. It keeps track of current plotting figure and area to current access.

4. import seaborn as sns –

A Python package called Seaborn is used to create statistical graphs. It helps to explore and understand the data whose plotting functions operates on data frames and arrays containing dataset while performing internal semantic mapping.

5. import warnings –

The warning filters determine if an error is raised (raising an exception), displayed, or ignored. The filters determine the disposition of the match. The 'ignore' warnings never prints matching warnings.

6. from collections import Counter –

The counter is a container included in collections module which holds objects and a counter is a subclass of dict which is an unordered collection where elements and their counts are stored in a dictionary.

7. from sklearn import preprocessing –

To transform unprocessed feature vectors into representations, this package offers transformer classes and utility functions. Sklearn is a preprocessing library that includes utility functions and transformer order.

8. from sklearn.preprocessing import StandardScaler –

It standardizes features by removing mean values and scaling to unit variance. It is based on the theory that dataset variables whose value lies in different ranges do not have an equal contribution to model's fit parameters and training function and may lead to bias in predictions made with the model.

9. from sklearn.model_selection import train_test_split –

It splits the arrays or matrices into random train and test subsets. We must divide our data into features (X) and labels (y), and subsequently divide the dataframe into X_train, X_test, y_train, and y_test values.

10. Data Modeling Libraries –

Data modeling libraries like confusion matrix, accuracy score, roc curve, classification report, LogisticRegression, GaussianNB (Naïve Bayes), RandomForestClassifier, DecisionTreeClassifier, KNeighborsClassifier, SVC etc are imported for different machine learning algorithm predictions.

2. Reading/Loading the Dataset –

Reading the Dataset

```
In [2]: df = pd.read_csv('D:\\Khushi MCA\\MCA Semester 4\\archive (1)\\heart.csv')
df.head()
```

Image 2 – Reading the Dataset

Now, after we have completed importing the libraries, the next step is to load the dataset containing heart attack parameters and read them. Four databases—Cleveland, Hungary, Switzerland, and Long Beach V—make up this 1988 data set. It has 76 attributes total, including the predicted attribute, but only 14 of them are used in the published experiments. 0 indicates no disease, and 1 indicates disease for the heart condition of the patient in the target field. The dataset was installed from Kaggle. It includes column values like age, sex, chest pain type (4 values), cholesterol in mg/dl, exercised induced angina (1 – yes, 0 – no), fasting blood sugar > 120 mg/dl, the maximal heart rate obtained, and resting electrocardiographic data with values 0, 1, and 2. The next step is to read the dataset through its file location in the local system and display the first five values from the heart.csv dataset. In order to read the dataset, with the help of the pandas library's **pd.read_csv** function, we can read a CSV file and use Python to create a DataFrame object. Here are some details regarding the function **pd.read_csv**. The **head()** is used to display the first few column values from the heart.csv dataset.

3. Checking Null Values, and Duplicate Values –



Image 3 – Checking for Null Values

The next step is to determine the number of rows and columns and the total values in heart.csv which is by using the shape and describe function. Using the **info()** command, we can understand the concise summary used for data analysis, including information about the index, columns, memory usage and the null values of the dataset. We can also check for the missing values (NaN) and check the boolean data using **df.isnull().any()**. In order to check for duplicate values and drop them, we will use the **drop_duplicate()** method, and eliminate the duplicate rows from the data frame, and shape again to print the values that we receive after dropping the duplicate values. Here, we made sure that we do not have any duplicate, null or missing values present in the dataset.

4. Histogram and Column Information –

```
Now lets have a look at the columns ¶

In [10]: df.hist(figsize=(15,10), bins=9)

...

In [11]: info = ["age", "0: female, 1: male", "chest pain type, 1: typical angina, 2: atypical angina, 3: non-anginal pain, 4: asymptomatic",
                "serum cholestoral in mg/dl", "fasting blood sugar > 120 mg/dl",
                "resting electrocardiographic results (values 0,1,2)",
                "maximum heart rate achieved", "exercise induced angina", "oldpeak = ST depression induced by exercise relative to rest",
                "the slope of the peak exercise ST segment",
                "number of major vessels (0-3) colored by flourosopy", "thal: 3 = normal; 6 = fixed defect; 7 = reversable defect"]

for i in range(len(info)):
    print(df.columns[i]+":\t\t\t"+info[i])

...

Searching for unique values first

In [12]: df['target'].value_counts()

...
```

Image 4 – Histogram and Column Information

In order to show all the columns present in the csv file, we plot histogram with a figure size of 15 and 10 and bin 9. The info variable stores the column information. Where age column represents the age of the patient, 1 shows male and 0 shows a female patient and is denoted by the column named as sex. Cp column denotes the chest pain type. 1 stands for ‘**typical angina**’, 2 stands for ‘**atypical angina**’, 3 stands for ‘**non-anginal pain**’, and 4 stands for ‘**asymptomatic**’. Resting blood pressure is trestbps. The serum cholesterol column is in mg/dl and fasting blood sugar is > 120 mg/dl. Rest ecg is resting electrocardiographic results with values 0,1 and 2. Thalach is the maximum heart rate achieved. Exang is exercise induced angina and old peak is ST depression induced by exercise relative to rest. The peak exercise ST statement's slope. CA is the number of major vessels (0-3) colored by fluoroscopy. Thal: 3=normal, 6=fixed defect and 7=reversable defect.

5. Data Normalization –

Data Normalization

```
In [25]: x= preprocessing.StandardScaler().fit(X).transform(X.astype(float))
x[0:5]

Out[25]: array([[ -0.26796589,  0.68265615, -0.93520799, -0.37655636, -0.66772815,
                 -0.41844626,  0.90165655,  0.80603539, -0.69834428, -0.03712404,
                  0.97951442,  1.27497996,  1.11996657],
                [ -0.15726042,  0.68265615, -0.93520799,  0.47891019, -0.84191811,
                  2.38979311, -1.0025412 ,  0.23749516,  1.43195847,  1.77395808,
                 -2.27118179, -0.71491124,  1.11996657],
                [  1.72473259,  0.68265615, -0.93520799,  0.76406571, -1.40319685,
                 -0.41844626,  0.90165655, -1.07452077,  1.43195847,  1.34274805,
                 -2.27118179, -0.71491124,  1.11996657],
                [  0.72838335,  0.68265615, -0.93520799,  0.93515902, -0.84191811,
                 -0.41844626,  0.90165655,  0.49989834, -0.69834428, -0.8995441 ,
                  0.97951442,  0.28003436,  1.11996657],
                [  0.83908882, -1.46486632, -0.93520799,  0.36484799,  0.91933586,
                  2.38979311,  0.90165655, -1.90546419, -0.69834428,  0.73905401,
                 -0.64583368,  2.26992556, -0.51399432]])
```

Image 5 – Data Normalization

Data Normalization is the process of organization and restructuration of data within a database to improve the efficiency, integrity, and consistency. It eliminates the redundant and unstructured data while standardizing the data formats and establishing relationships between various tables. Information fields like URLs, contact names, street addresses, phone numbers, can be recorded in standardized manner. Normalized data is easier to manage, query and analyze. It helps to integrate data and establish relationship between tables.

The removal of data anomalies, or inconsistent data storage, is another important benefit of data normalization. When there is a mistake when adding, updating, or removing data from a database, the structural issues with the database become apparent. Data normalization guidelines help to guarantee that new data is entered and updated accurately, avoiding duplication or false entry, and that information can be deleted without affecting related data.

6. Exploratory Data Analysis (EDA) –

Exploratory Data Analysis (EDA)

```
In [13]: categorical_val = []
          continuous_val = []
          for column in df.columns:
              print('=====')
              print(f"{column} : {df[column].unique()}")
              if len(df[column].unique()) <= 10:
                  categorical_val.append(column)
              else:
                  continuous_val.append(column)
```

Image 6 – Exploratory Data Analysis

We create two variables and name them continuous values and categorical values. The categorical data is known as qualitative data, wherein the variables can take on a fixed number of possible values. The categorical values are also called as discrete values. Every category, individual or a unit is based on qualitative properties. It is represented using bar graphs, pie charts etc. Categorical variables are descriptive. A continuous value can have an upper and lower bound but it can be any value in range. They contain a finite number of categories or distinct groups as they contain a finite number of categories or distinct groups. It might not have a logical order. They have mutually exclusive levels. It is presented according to its division into certain groups, values are sorted into predefined categories according to design. It ensures control and establishing relevance.

Continuous data describes information that can take heights, weights, age, temperature or numerical measurement. It has the biggest benefit of accuracy and can be denoted graphically. It is the standard format for industries in order to quantify and understand how the information is implied. It can virtually take any value and can be measured through real numbers. It does a precise and detailed interpretation. It can predict continuous output variable based on the set of input features. Continuous variables are quite different from discrete variables since they can hold any value between a theoretical minimum and maximum.

7. Patients suffering from Heart Problems –

```
In [14]: plt.figure(figsize=(15, 15))

for i, column in enumerate(categorical_val, 1):
    plt.subplot(3, 3, i)
    df[df["target"] == 0][column].hist(bins=35, color='blue', label='Does not have Heart Disease', alpha=0.6)
    df[df["target"] == 1][column].hist(bins=35, color='red', label='Has Heart Disease', alpha=0.6)
    plt.legend()
    plt.xlabel(column)
    plt.savefig('HeartDisease1.png')

...

In [15]: plt.figure(figsize=(15, 15))

for i, column in enumerate(continous_val, 1):
    plt.subplot(3, 2, i)
    df[df["target"] == 0][column].hist(bins=35, color='blue', label='Does not have Heart Disease', alpha=0.6)
    df[df["target"] == 1][column].hist(bins=35, color='red', label='Has Heart Disease', alpha=0.6)
    plt.legend()
    plt.xlabel(column)
    plt.savefig('HeartDisease2.png')
```

Image 7 – Patients suffering from heart problems

We plot the 15*15 figure in which we use for loop to plot whether the patient suffers from heart disease or not. We use two target parameters, where 0 is for females and 1 is for males. It checks for the values from each column like sex, chest pain, fbs, restecg, exang, slope, ca, thal, and target. According to all the parameters, it analyzes the patients suffering from heart problems, and those patients who suffer from heart disease is represented using blue color and those patients who do not suffer from heart diseases is shown using red color. We plot one more similar plot to determine the target variables of the remaining parameters i.e. age, trestbps, chol, thalach and oldpeak columns. Those patients who do not suffer from heart problems are shown using blue color and red color for the patients who do not suffer from heart diseases.

8. Splitting the data into Training and Testing Dataset –

Splitting the Data into Training and Testing Dataset

```
In [16]: y = df["target"]
X = df.drop('target', axis=1)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=0)

print("X_Train Shape is: ", X_train)
print("X_Test Shape is: ", X_test)

print("y_Train Shape is: ", y_train)
print("y_Test Shape is: ", y_test)
```


Image 8 – Splitting the data into training and testing dataset

Now we will split the data into two different datasets in order to assess the model's ability to generalize and make predictions which are accurate – Training set which is used to train the machine learning model and testing set is used to test and evaluate the model on new and unseen data. We make use of train test split in order to estimate the model performance while working with large datasets. The dataset is split into two parts in the ratio 80:20 – The trained data is split into 80% and the testing data is the remaining 20%. Then we print the shape of X_train, X_test and y_train and y_test data which means the total number of X_train, y_train and X_test, y_test data using the shape attribute of the pandas data frame.

9. Counting the Unique Values –

Checking for Unique Values

```
In [17]: print(y_test.unique())  
Counter(y_train)
```

Image 9 – Checking the unique values

The Counter class is a part of specialized container data type provided by collections module in Python. It is used to count hashable objects and implicitly creates a hash table of an iterable when invoked. It is employed to tally the instances of each element in a set. Since the Counter class creates a dictionary like object where the elements are stored as keys and their counts are stored as values. Now let us test the y_test Counter values.

10. Standard Scaler –

Standardizing

```
In [18]: scaler = StandardScaler()
         X_train = scaler.fit_transform(X_train)
         X_test = scaler.transform(X_test)
```

Image 10 - Standardization

StandardScaler is a preprocessing technique in scikit-learn used for standardizing the data which is used to ensure that the features have a same scale and are centred around zero. The Standard Scaler normalizes the features individually to ensure that each feature has a mean of 0 and a standard deviation of 1 and is sensitive to outliers. Here, we standardize the X_train and X_test data.

11. Bar Plot for Heart Disease Frequency for Different Ages –

Heart Disease Frequency for Ages

```
In [19]: pd.crosstab(df.age,df.target).plot(kind="bar",figsize=(20,6))
         plt.title('Heart Disease Frequency for Ages')
         plt.xlabel('Age')
         plt.ylabel('Frequency')
         plt.savefig('heartDiseaseAndAges.png')
         plt.show()
```

Image 11 – Bar Plot for Heart Disease Frequency

We will use the `pd.crosstab` function in pandas to compute a simple cross-tabulation of two or more factors which allows to create a table showing frequency distribution of variables and their relationships. We use this function for data analysis and visualization and summarizing data frequencies, percentage and other aggregations. Here we use age and target columns from the variable `df` titled 'Heart Disease Frequency for different ages' where we plot Age on the x axis and Frequency on the y axis where 0 stands for female frequencies of all ages represented using the blue color and 1 that stands for male frequencies of all ages which is represented using orange color. We will save this figure as a .PNG image using `plt.savefig()`

12. Heart Disease Frequency according to the Genders

Heart Disease Frequency according to the Genders - i.e Males and Female Frequencies

```
In [20]: pd.crosstab(df.sex,df.target).plot(kind="bar",figsize=(15,6),color=['yellow', 'green' ])
plt.title('Heart Disease Frequency for Sex')
plt.xlabel('Sex (0 = Female, 1 = Male)')
plt.xticks(rotation=0)
plt.legend(["Don't Have Heart Disease", "Have Heart Disease"])
plt.ylabel('Frequency')
plt.savefig('HeartDiseaseFrequency.png')
plt.show()
```

Image 12 – Heart Disease Frequency according to the gender

We will use the `pd.crosstab` again to plot the Heart Disease Frequency of both the genders, male and female which has the values 0 and 1 respectively. The **yellow** colored columns denote the female frequencies and male frequencies having heart disease, whereas, the **green** colored columns represent the male and female patients having heart disease.

13. Age wise heart Disease Rate –

Age wise Heart Disease Rate

```
In [21]: plt.scatter(x=df.age[df.target==1], y=df.thalach[(df.target==1)], c="purple")
plt.scatter(x=df.age[df.target==0], y=df.thalach[(df.target==0)])
plt.legend(["Has Disease", "Does Not Have Disease"])
plt.xlabel("Age")
plt.ylabel("Maximum Heart Rate")
plt.savefig('HeartDiseaseRate.png')
plt.show()
```

Image 13 – Age wise heart disease rate

Let us now plot a scatter plot with two parameters, patients having heart problems in purple color and patients not having heart problems in blue color. It shows the age wise maximum heart disease rate among all the ages starting from 29 years to 70 years. We plot the age of the patient on the X axis and Maximum Heart Rate on the Y axis.

14. Feature Selection –

Feature Selection

```
In [22]: names=['age','restecg','chol','thalach','trestbps','exang']

#Set the width and height of the plot
f, ax = plt.subplots(figsize=(7, 5))

#Correlation plot
corr = df.loc[:,names]
#Generate correlation matrix
correlation = corr.corr()

#Plot using seaborn library
sns.heatmap(correlation, annot = True, cmap='coolwarm',linewidths=.1)
plt.savefig('FeatureSelection.png')
plt.show()
```

```
In [23]: corr
```

```
In [24]: df.drop('target', axis=1).corrwith(df.target).plot(kind='bar', grid=True, figsize=(12, 8),
title="Correlation with target")
plt.savefig('CorrwithTarget.png')
```

Image 14 – Feature Selection

Feature Selection is the process of selecting a sub-set of input features from the dataset heart.csv to use in a model construction. It improves the performance of predictive model and reduces the computational modelling cost. It narrows down the set of features that are most important to the machine learning model. It reduces number of input variables by eliminating the redundant values.

Here we take the column names, age, restecg, chol, thalach, trestbps, and exang as an array to store in names variable. We will generate the correlation plot using .loc method, which is used in the pandas library in Python for label based indexing and selecting from a data frame. It is primarily used for label indexing and can access multiple columns. It is used to filter or slice a data frame based on specific conditions. We have created a plot using seaborn library.

15. Correlation –

```
In [23]: corr
...

In [24]: df.drop('target', axis=1).corrwith(df.target).plot(kind='bar', grid=True, figsize=(12, 8),
                                                    title="Correlation with target")
plt.savefig('CorrwithTarget.png')
```

Image 15 - Correlation

Now let us find the correlation with the target variables and plot a bar graph including the columns – age, sex, cp, testbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal columns.

16. Testing Different Machine Learning Algorithms to Predict Heart Attack –

a. Logistic Regression –

Now lets test different Machine Learning Models to Predict Heart Attack

First Model - Logistic Regression

```
In [26]: model1 = 'Logistic Regression'
         lr = LogisticRegression()
         model = lr.fit(X_train, y_train)
         lr_predict = lr.predict(X_test)
         lr_conf_matrix = confusion_matrix(y_test, lr_predict)
         lr_acc_score = accuracy_score(y_test, lr_predict)
         print("Confusion Matrix")
         print(lr_conf_matrix)
         print("\n")
         print("Accuracy of Logistic Regression:",lr_acc_score*100,'\n')
         print(classification_report(y_test,lr_predict))
```

Image 16 – Logistic Regression

The first model to predict heart attack, we will make use of Logistic Regression. It is a supervised learning technique which is used for binary tasks to predict the probability that an instance belongs to a given class or not. It also analyzes the relationship between two factors. When the target prediction is binary, meaning it has two possible outcomes, such as yes/no or 0/1 or true/false which should be categorical or discrete value, but it gives the probabilistic output. Logistic Regression is used to predict the categorical dependent variable using independent values. It is widely used for solving classification problems. Logistic Regression is used to predict continuous outcomes and predicts the likelihood of an observation falling into specific category and employs S shaped logistic function.

Evaluation of Logistic Regression Model. Here, we apply the logistic regression function and store it in lr variable. We will first fit the X_train and y_train values and predict the X_test values using lr.predict method and storing it in the model variable and lr_predict variable. lr_conf_matrix variable stores the confusion matrix of the y_test values and lr_predict values. Now, let us take a look on the accuracy score too, in order to store it, we created a variable called lr_acc_score to store the accuracy score obtained with the help of y_test and lr_predict values. Then we print the classification report and the accuracy score of Logistic Regression.

b. Naïve Bayes Classifier –

Second Model - Naive Bayes

```
In [27]: model2 = 'Naive Bayes'
nb = GaussianNB()
nb.fit(X_train,y_train)
nbpred = nb.predict(X_test)
nb_conf_matrix = confusion_matrix(y_test, nbpred)
nb_acc_score = accuracy_score(y_test, nbpred)
print("Confusion Matrix")
print(nb_conf_matrix)
print("\n")
print("Accuracy of Naive Bayes model:",nb_acc_score*100,'\n')
print(classification_report(y_test,nbpred))
```

Image 17 – Naïve Bayes Classifier

It is the second model to study the prediction model. The Gaussian Naïve Bayes (GNB) is the founded on the Bayes theorem basically with a strong (naïve) assumption that every feature in dataset is unrelated to every feature. Here, we created a variable named nb and fit it into X_train and y_train values using the nb.fit method. To predict the accuracy score, we created a variable named nbpred and used the predict method on the X_train data.

We created a confusion matrix on `y_test` and `nbpred` data and print the confusion matrix array. We print the accuracy score and multiply it by 100 and print the classification report. The Gaussian distribution calculates variance and mean of each feature for every class training. The Gaussian distributions represent the probability distribution of features with each class by estimating the μ and variance σ^2 for every feature in every class is a part of representation for a dataset.

$$-\frac{(x - \mu)^2}{2\sigma^2}$$

$$P(X_i | C_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x - \mu)^2}{2\sigma^2}}$$

c. Random Forest Classifier –

Third Model - Random Forest Classifier

```
In [28]: model3 = 'Random Forest Classifier'
rf = RandomForestClassifier(n_estimators=20, random_state=12, max_depth=5)
rf.fit(X_train, y_train)
rf_predicted = rf.predict(X_test)
rf_conf_matrix = confusion_matrix(y_test, rf_predicted)
rf_acc_score = accuracy_score(y_test, rf_predicted)
print("Confusion Matrix")
print(rf_conf_matrix)
print("\n")
print("Accuracy of Random Forest:", rf_acc_score*100, '\n')
print(classification_report(y_test, rf_predicted))
```

Image 18 – Random Forest Classifier

Random Forest Classifier is also known as '**Random Decision Tree**'. They work together to gather one single output. It can handle complex data while reducing the risk of overfitting and predict accurate forecasts. Random Forests leverages ensemble learning to construct decision trees. The next step is random feature selection to ensure the unique perspective after which we use bagging technique used for training strategy involving multiple bootstrap sample from original dataset.

Each decision casts its vote so the final prediction is determined by mode across all the trees when it comes to the classification problems, whereas, in regression tasks, the average of individual tree is taken. Random forest provides high accuracy and resistance to overfitting while handling large and complex datasets. It also fosters robustness and accuracy while handling the missing values. It also provides built-in resistance for the variable importance and interpretation of influential factors. To predict the random forest classifier model values, we store the random forest classifier with 20 estimators to avoid noise, random state 12 and depth to be 5. We used the fit method to fit X_train and y_train values and predict the X_test values to store it in rf_predicted variable. We created a confusion matrix of y_test and rf_predicted variables and calculate the accuracy score on rf_predicted and y_test values. We print the accuracy score and generate the classification reports.

d. K-Nearest Neighbors Algorithm –

Fourth Model - K-Neighbors Classifier

```
In [29]: model4 = 'K-NeighborsClassifier'
knn = KNeighborsClassifier(n_neighbors=10)
knn.fit(X_train, y_train)
knn_predicted = knn.predict(X_test)
knn_conf_matrix = confusion_matrix(y_test, knn_predicted)
knn_acc_score = accuracy_score(y_test, knn_predicted)
print("Confusion Matrix")
print(knn_conf_matrix)
print("\n")
print("Accuracy of K-NeighborsClassifier:", knn_acc_score*100, '\n')
print(classification_report(y_test, knn_predicted))
```

Image 19 – K Nearest Neighbors

It was developed in 1951 by Evelyn Fix and Joseph Hodges and was expanded by Thomas Cover. KNN is a versatile and widely used machine learning algorithm designed for its simplicity. It is quite flexible as it can work with numerical data as well as categorical data.

It is non-parametric method that makes predictions based on similarity of data points. KNN algorithm helps to identify nearest points or group for query points and to determine the query points we use Euclidean distance, Manhattan distance and Minkowski Distance. The value of k is essential to define the number of neighbors and should be chosen based on the input data. When there are outliers or noise, higher value of k is preferred. KNN operates on principles of similarity. Here, we decide the value of n neighbors to be 10 and fit the model using fit method on the values X_train and y_train. Then we predict the X_test values and store the variable in knn_predicted variable. We print the y_test and knn_predicted values using confusion matrix and print the accuracy score. We generated the classification result and printed it.

e. Decision Tree Classifier –

Fifth Model - Decision Tree Classifier

```
In [30]: model5 = 'DecisionTreeClassifier'
dt = DecisionTreeClassifier(criterion = 'entropy',random_state=0,max_depth = 6)
dt.fit(X_train, y_train)
dt_predicted = dt.predict(X_test)
dt_conf_matrix = confusion_matrix(y_test, dt_predicted)
dt_acc_score = accuracy_score(y_test, dt_predicted)
print("Confusion Matrix")
print(dt_conf_matrix)
print("\n")
print("Accuracy of DecisionTreeClassifier:",dt_acc_score*100,'\n')
print(classification_report(y_test,dt_predicted))
```

Image 20 – Decision Tree Classifier

Decision trees provide interpretable models to visualize flowcharts for understanding and analyzing the data. It is structured like a tree where internal nodes denote the dataset features, branches denote the decision rules and leaf node denotes the outcomes. It is a illustration of all potential solutions to a problem or decision given certain conditions.

Here, we store the Decision Tree Classifier values in a variable called 'dt' with random state 0 and maximum depth 6. We used the fit method to fit X_train and y_train values. We will predict the X_test values of decision tree and store it in dt_predicted variable. We will also analyze the y_test, dt_predicted values to store the values in the accuracy score variable named dt_acc_score and multiply it by 100 to find the accuracy of decision tree classifier. We will print the confusion matrix and generate the classification report.

f. Support Vector Machines –

Sixth Model - Support Vector Classifier

```
In [31]: model6 = 'Support Vector Classifier'
svc = SVC(kernel='rbf', C=2)
svc.fit(X_train, y_train)
svc_predicted = svc.predict(X_test)
svc_conf_matrix = confusion_matrix(y_test, svc_predicted)
svc_acc_score = accuracy_score(y_test, svc_predicted)
print("Confusion matrix")
print(svc_conf_matrix)
print("\n")
print("Accuracy of Support Vector Classifier:", svc_acc_score*100, '\n')
print(classification_report(y_test, svc_predicted))
```

Image 21 – Support Vector Classifier

The last algorithm to predict heart attack is support vector machines. When the SVM algorithm finds the best line or decision boundary, it is known as hyperplane and the closest point of lines is known as vector. The distance between vectors and hyperplane is called **margin**. The hyperplane with maximum number of margins is known as **optimal hyperplane**. The vectors which are closest to hyperplane affecting the position of hyperplane is termed as support vector. Here, we create a variable named as svc where, we fit the svc model values of X_test using fit method and then create a confusion matrix with a variable name svc_conf_matrix on y_test, svc_predicted. We shall print the confusion matrix and generate the classification report of SVC model.

CHAPTER 5 – FINDINGS AND INFERENCES

CHAPTER 5 – FINDINGS AND INFERENCES

From the above data interpretation, we found out the following:

- Originally, there are total 14 columns in the heart.csv file and 1025 rows before dropping the duplicate values. The above image shows the total number of columns with their description.

Total Number of Rows and Columns in the Dataset

```
In [3]: print("Number of Rows in the Dataset: ",df.shape[0])  
        print("Number of Columns in the Dataset: ",df.shape[1])
```

```
Number of Rows in the Dataset:  1025  
Number of Columns in the Dataset:  14
```

- After dropping the duplicate values using drop_duplicates() method, the dataset heart.csv contains 302 rows and 14 columns.

```
In [7]: df_dup = df.duplicated().any()  
        print(df_dup)
```

```
True
```

```
In [8]: df = df.drop_duplicates()
```

```
In [9]: print("Number of Rows in the Dataset after Dropping Duplicate Values: ",df.shape[0])  
        print("Number of Columns in the Dataset after Dropping Duplicate Values: ",df.shape[1])
```

```
Number of Rows in the Dataset after Dropping Duplicate Values:  302  
Number of Columns in the Dataset after Dropping Duplicate Values:  14
```

Running head at the top of every page

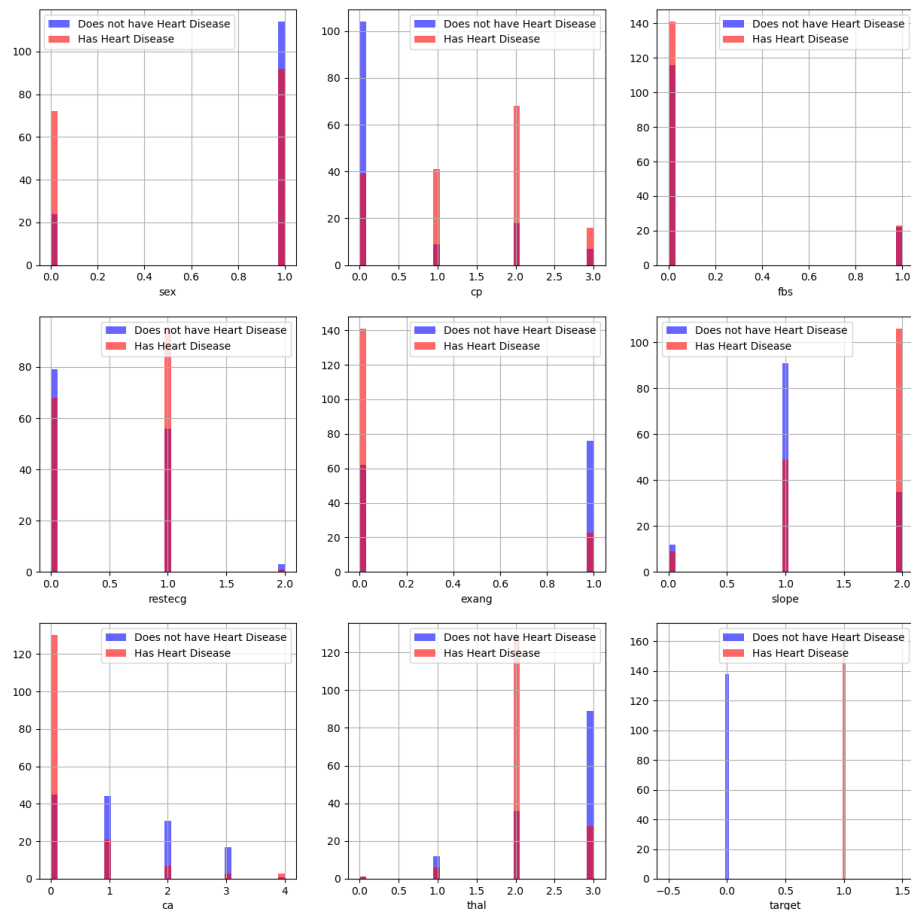
- Nine columns namely cp, sex, fbs, restecg, exang, slope, ca, thal and target are categorical values and five columns including age, trestbps, chol, thalach, and oldpeak are defined as numerical data. Table 3 represents the column names.

Table 3 – Variables

Categorical Columns	Continuous Columns
• Cp (Chest Pain)	• Age
• Sex (0 – female, 1- male)	• Trestbps (Resting Blood Pressure)
• Fbs (Fasting Blood Sugar)	• Chol (Serum Cholestrol)
• Restecg (Resting Electrocardiographic Results)	• Thalach (Maximum Heart Rate)
• Exang (Exercise induced Angina)	• Oldpeak (ST depression induced by exercise relative)
• Slope (Slope of ST segment)	
• Ca (Number of vessels colored by fluoroscopy)	
• Thal (thal, 3 = normal, 6 = fixed defect and 7 = reversible defect)	
• Target	

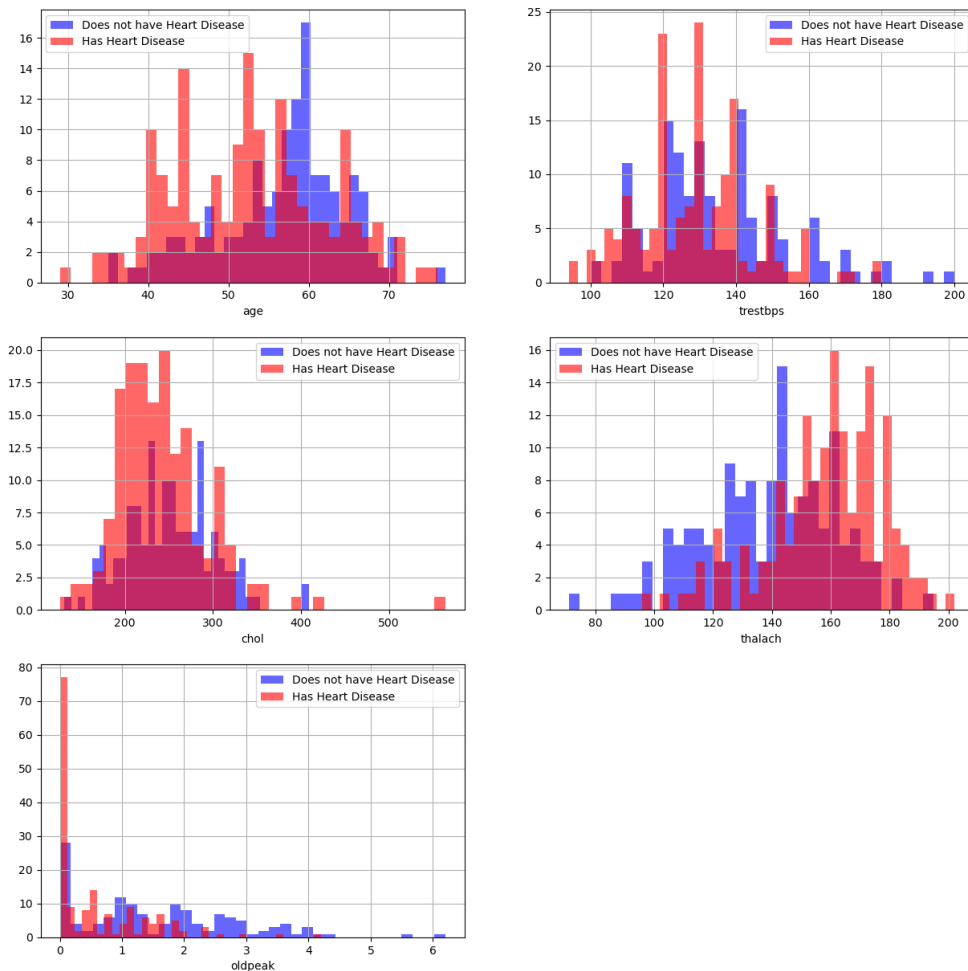
- According to the above table, the findings include the patients who are most likely to develop a heart disease based on the different parameters such as fbs, cp, exang, slope, sex, restecg, exang, slope, ca, thal, target, age, trestbps, chol, thalach and oldpeak. The blue color represents the patients who do not suffer from heart disease, whereas the orange color represents the patients suffering from heart diseases.

Figure 3 – Finding 1



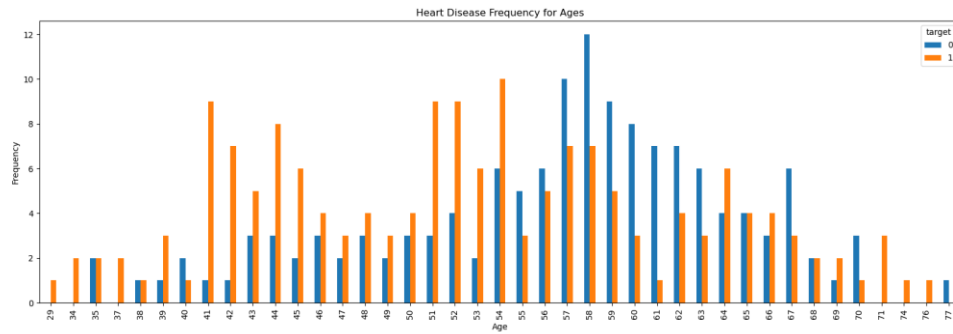
- Finding 1 - The Patients with Chest Pain (cp) 1, 2 and 3 are more likely to have a heart disease with chest pain 0.
- Finding 2 - The Patients with resting ECG results with a value 1 are more likely to have heart disease.
- Finding 3 - The Patients with Exercise Induced Angina are more likely to have a heart disease than people with value 1.
- Finding 4 - The Patients with incline value of 2 slop peak exercise are more likely to develop heart disease than people with inclination value 2 or 1.

Figure 4 – Finding 2



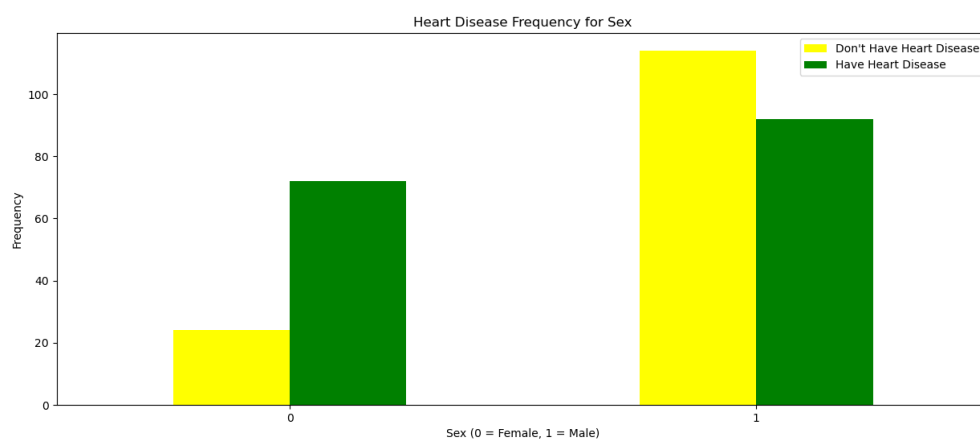
- Finding 5 – The Patients with thal (corrected disease) value 2 are more likely to develop heart problems.
- Finding 6 – Resting Blood Pressure around 130 to 140 is a major cause for heart attack.
- Finding 7 – Cholesterol over 200 is a matter of concern.
- Finding 8 – The Patients with a maximum of 140 are likely to develop heart disease.
- Finding 9 – The former peak of rest vs induced ST depression looks at cardiac stress during exercises and unhealthy heart are the most stressed.

Figure 5 – Age wise Distribution



- Finding 10 – The ages of the patients range from **29** years to **77** years. The total numbers of rows were 1025, out of which, 723 rows were found to be duplicate, in order to make the data complete, we segregated it and dropped the duplicate values from the dataset by using the drop_duplicates() method. Hence, we found **302** rows to be accurate out of which,

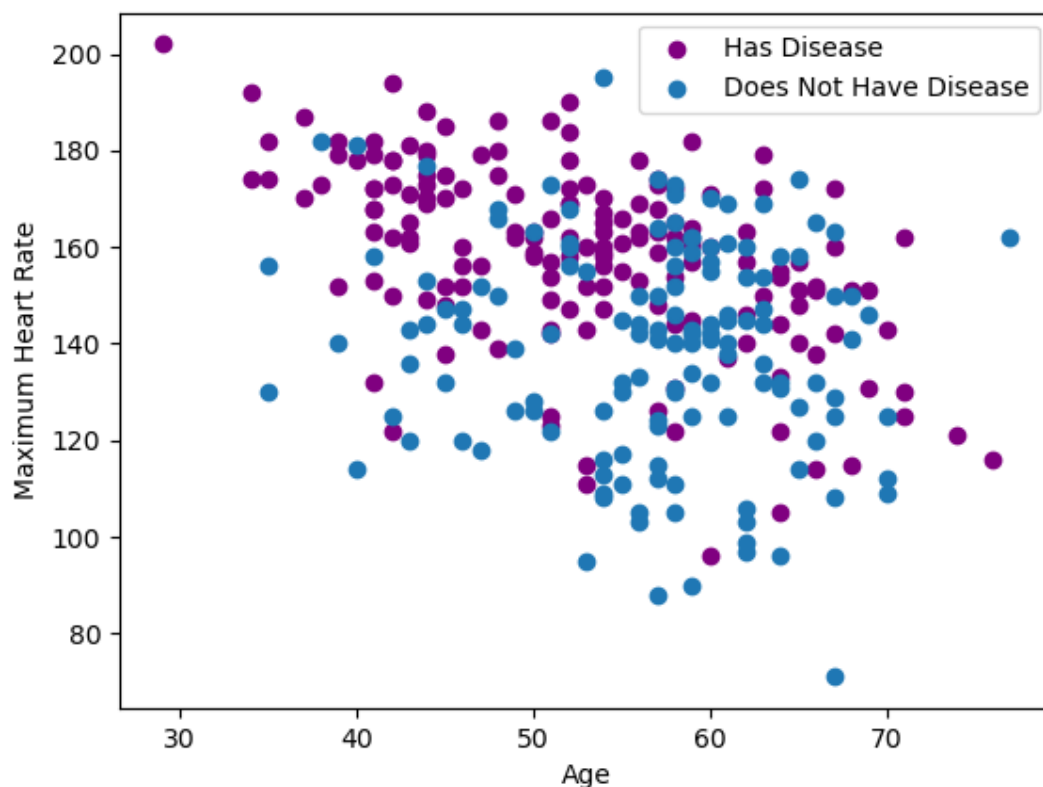
Figure 6 – Gender wise Distribution



0 stands for **female** and **1** stands for **male** patients and the yellow portion of the bar graphs represents the persons who do not suffer from heart disease, whereas the green portion of bar graphs denotes the patients suffering from heart disease. Approximately **75 to 78%** females suffer from heart diseases while around 90% males suffer from heart diseases.

To infer, we see that there are only 20% females who are healthy while compared to men who are healthy. **It means that females are at more risk of developing heart diseases or suffer from heart attacks than men.**

Figure 7 – Age wise Distribution



Your pulse gives you your heart rate, which is a crucial indicator of your general health and level of fitness. It may indicate the need to address certain medical conditions or modify lifestyle choices that cause your heart rate to rise above the age-appropriate normal range.

The heart.csv dataset contains values ranging from minimum age **29** years to maximum age of **77** years. In this scatter plot, we can see the age wise distribution of the patients of all ages. The **blue** color represents the patients who are **not suffering from any heart disease** whereas the **violet** color shows the patients who **are suffering from heart diseases** or are more likely to suffer from heart diseases. When you're not exercising, your heart pumps the least amount of blood necessary for your body, which is known as your resting heart rate. When you're sitting or lying down, feeling at ease, and not unwell, your pulse can be used to determine your resting heart rate.

When engaging in moderate-intensity physical activity, such as walking, your target heart rate should be between 50% and 70% of your maximum heart rate. Your target heart rate for more strenuous activities, like running, weightlifting, or exercising, should be between 70% and 85% of your maximum heart rate. Studies reveal a connection between an elevated resting heart rate and increased blood pressure, body weight, and decreased physical fitness.

According to our analysis on age-wise heart rate distribution, we can observe that the youngest patient who is at the risk of developing a heart disease is about 29 years old with the maximum heart rate of 200, which is extremely alarming. While a majority of patients who suffer from heart diseases are from the ages 40 to 60 and the average heart rate is around 120 to 180. The target heart rate for a 20-year-old should be in between 100 to 170 bpm, whereas the maximum heart rate should be 200 bpm. A 45-year-old should have a target heart rate within 88 to 149 bpm and maximum heart rate at around 175 bpm. As we check for older ages, the target heart rate for 65 years old would be around 75 to 128 bpm and the maximum heart rate around 150 bpm. So, in case of our study, the 29 years old patient has a maximum heart rate of slightly above 200, which is concerning. Most patients suffering from heart attack are under the ages of 35 to 65 years.

We can have a look at the Hypothesis with the help of the following table

Figure 8 – Summary 1

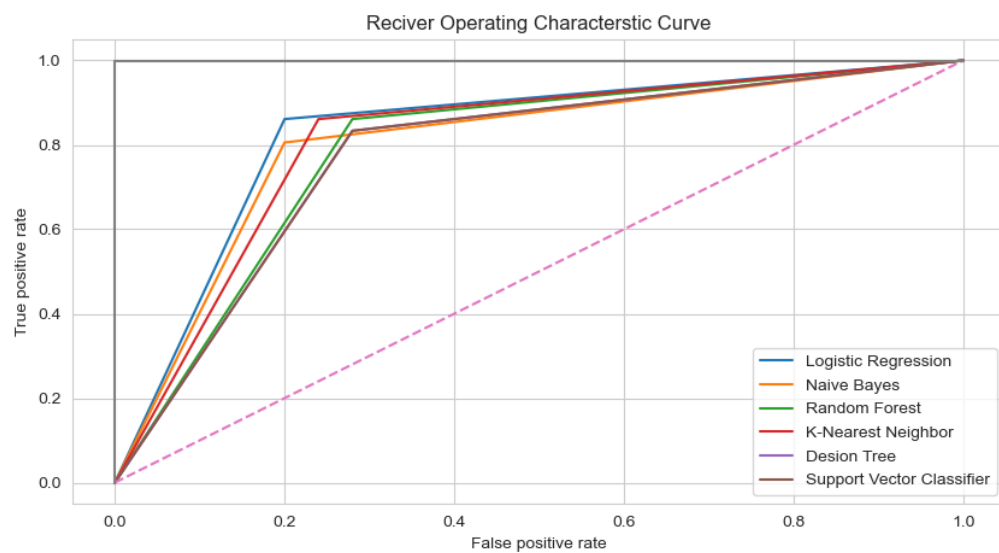
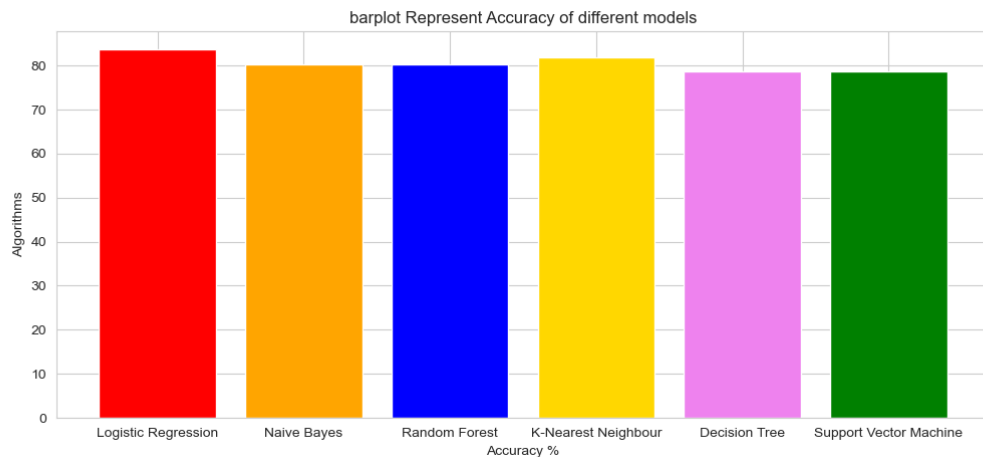


Table 4 – Summary

Hypothesis/Algorithm	LR	NB	RF	KNN	DT	SVC
True Positive	0.85	0.8	0.83	0.84	-	0.81
False Positive	0.2	0.2	0.3	0.3	0.3	0.3
Total Accuracy	83.60	80.32	80.32	81.96	78.68	78.68

Figure 9 – Summary 2



Following is the bar plot for representing the accuracy scores of all the six algorithms.

1. The bar plot in **red** color shows the accuracy score of the **Logistic Regression Model** with the total accuracy of **83.60%**.
2. The bar plot in **orange** color shows the accuracy score of **Naïve Bayes Classifier Model** with a total accuracy of **80.32%**.
3. The bar plot in **blue** color shows the accuracy score of **Random Forest Classifier Model** with a total accuracy of **80.32%**.
4. The bar plot in **yellow** color shows the accuracy score of **K-Nearest Neighbor Model** with a total accuracy of **81.96%**.
5. The bar plot in **purple** color shows the accuracy score of **Decision Tree Classifier Model** with a total accuracy of **78.68%**.
6. The bar plot in **green** color shows the accuracy score of **Support Vector Classifier Model** with a total accuracy of **78.68%**.

Therefore, out of the six machine learning algorithms that we have used to perform the heart disease and heart attack predictions on the heart.csv dataset, we can infer that the Logistic Regression model in red color provides the **highest accuracy of 83.60%** out of the other five machine learning algorithms. The second accurate machine learning algorithm is K-nearest neighbor algorithm in the yellow color with an accuracy of **81.96%**. Naïve Bayes Classifier and Random Forest Classifier both represent the same level of accuracy i.e. **80.32%** and the last two machine learning algorithms are in purple and green colors, they are Decision Tree Classifier and Support Vector Classifier models with the same level of accuracy – **78.68%**.

CHAPTER 6 – RECOMMENDATIONS AND LIMITATIONS

CHAPTER 6 – RECOMMENDATIONS

The most basic yet an important recommendation is to modify the lifestyle and change the following factors in your life:

1. SMOKING:

Smoking is a significant risk factor for developing heart diseases. It causes damage to blood vessels, raises blood pressure, lowers the heart's oxygen delivery, and encourages blood clot formation. One of the best strategies to reduce the risk of heart disease is to **stop** smoking.

2. PHYSICAL ACTIVITY:

Heart health benefits from regular physical activity. Moderate-intensity aerobic exercises can help control weight, lower blood pressure, improve cholesterol, and lower the risk of heart disease. Examples of these exercises include brisk walking, swimming, and cycling. Every week, at least **150 minutes** of moderate-intensity exercise are advised by the American Heart Association.

3. HEALTHY DIET:

Preserving heart health requires a well-balanced diet. Lowering cholesterol, controlling blood pressure, managing weight, and lowering the risk of heart disease can all be achieved by eating a diet high in fruits, vegetables, whole grains, lean proteins, and healthy fats. It's also critical to restrict sodium, added sugars, and saturated and trans fats.

4. WEIGHT CONTROL:

Preventing heart disease requires maintaining a healthy weight. Being overweight, particularly in the waist area, raises the risk of diabetes, high blood pressure, and high cholesterol—all of which are heart disease risk factors. A healthy weight can be attained and maintained with the help of a balanced diet and frequent exercise.

5. STRESS MANAGEMENT:

High levels of anxiety and prolonged stress can raise the risk of heart disease. Reducing the negative effects of stress on heart health can be achieved by adopting healthy coping mechanisms like relaxation exercises, hobbies, getting enough sleep, and asking for support from loved ones.

6. ALCOHOL USE:

Drinking too much alcohol can lead to weight gain, elevated blood pressure, and an increased risk of heart disease. To preserve heart health, alcohol should be consumed in moderation or abstained from completely.

7. FREQUENT MEDICAL EXAMINATIONS:

Frequent medical examinations, which include blood pressure, cholesterol, and blood sugar tests, can assist in determining and controlling heart disease risk factors. To keep an eye out for any possible issues and take appropriate action, close collaboration with medical professionals is very essential.

8. COMPREHENSIVE DATASETS:

Accurately predicting heart disease is essential to early detection and successful treatment. For research predicting heart disease, use comprehensive and dependable datasets. Research on heart disease frequently makes use of the Cleveland database, which is accessible through the UCI machine learning repository. Make sure the dataset contains pertinent characteristics and attributes that are linked to heart disease. Examine the application of hybrid models, which integrate various machine learning methods. For instance, a hybrid model that combined neural networks and genetic algorithms was very accurate in predicting heart disease. Examine the possible advantages of integrating various algorithms or methods to enhance the precision of your predictions.

9. ACTIVE LEARNING:

According to certain studies, it can be difficult to forecast the likelihood that a disease will progress in the future. Examine how active learning strategies can be used to predict heart disease. With fewer labeled samples, prediction accuracy can be increased through active learning, which enables the model to choose the most informative samples for labeling. Active learning can greatly minimize the quantity of labeled data required for model training by choosing the most informative samples for labeling. Active learning can assist the model in concentrating on regions of the data distribution where errors are likely to occur by deliberately choosing difficult or ambiguous samples.

It's crucial to remember that a number of variables, such as the representativeness and quality of the data, the active learning strategy selected, and the accessibility of expert labeling, affect how effective active learning is at predicting heart attacks. Furthermore, active learning ought to be combined with other machine learning best practices, like appropriate feature selection, algorithm selection, and model evaluation.

10. THE USE OF ARTIFICIAL INTELLIGENCE AND INTERNET OF THINGS IN HEART ATTACK PREDICTION:

The use of Artificial Intelligence and Internet of Things - The field of heart attack prediction has shown promise with the integration of Internet of Things (IoT) and Artificial Intelligence (AI) technologies. With the use of these technologies, real-time data from multiple sources can be gathered and analyzed, making predictions more precise. IoT-based systems employ sensor data to precisely categorize illnesses, such as heart disease. Systems for remote monitoring based on the Internet of Things have been developed to track patients with heart failure and forecast heart disease. These systems use IoT and AI-based technologies to track and report on heart patients' activities, making precise predictions and enhancing patient care.

11. PERSONALIZED UPGRADATION AND MAINTENANCE OF THE PATIENT DATA IN REAL-TIME –

By evaluating specific risk factors and producing risk assessment graphs, artificial intelligence (AI) tools can offer individualized predictions of heart health. These graphs are useful for tracking changes in risk over time and identifying specific risk factors for patients as well as medical professionals. Individuals' health parameters can be continuously monitored by IoT devices, which can then transfer the data to AI-powered systems. When these systems identify early warning signs of heart disease, they can analyze the data in real time and produce alerts or notifications. This makes prompt action and preventative measures possible.

CHAPTER 7 - LIMITATIONS OF MACHINE LEARNING –

1. INCOMPLETE DATA –

An incomplete, biased, or homogeneous training set of data may hinder the model's capacity to predict outcomes accurately when faced with new data.

2. CANNOT BE GENERALIZED -

It might not translate well to broader demographics or diverse healthcare environments. While some studies have found that machine learning techniques can predict heart attacks with high accuracy rates, it's possible that these results won't always translate to real-world scenarios with distinct patient populations and healthcare systems.

3. PRIVACY CONCERNS –

Research on ensuring the models' interpretability and explainability is still ongoing. Due to privacy concerns and data availability, it can be difficult to access large and diverse datasets for the purpose of training machine learning models. Medical imaging and patient records are examples of sensitive healthcare data that must adhere to stringent privacy laws.

4. SUPERVISION OF CARDIOLOGISTS AND EXPERTS –

The development of machine learning models for heart attack prediction ought to be done in conjunction with subject matter experts, like cardiologists or other medical specialists. Their knowledge is essential for identifying pertinent features, deciphering the model's predictions, and guaranteeing the validity and clinical relevance of the outcomes.

CHAPTER 8 – CONCLUSION AND BIBLIOGRAPHY

CHAPTER 8 - CONCLUSION

Understanding how to process unprocessed medical data about the heart can save lives in the long run and aid in the early identification of irregularities in heart conditions. In this project, raw data was processed using machine learning techniques to produce a new and novel diagnosis of heart disease. Predicting heart disease is a difficult but crucial task in medicine. However, if the disease is identified early and preventative measures are implemented as soon as possible, the death rate can be significantly reduced. In order to focus the investigations on real-world datasets rather than merely theoretical approaches and simulations, it would be highly desirable to extend this study further.

Regarding heart attack prediction, the following inferences can be made based on the search results:

High levels of efficiency have been demonstrated by machine learning (ML) models in recognizing the signs of a heart attack and predicting the disease's risk factors. In particular, improving algorithms have shown promise in terms of forecasting the symptoms of heart disease.

While inaccurate heart disease predictions can be fatal, accurate predictions can avert life-threatening situations. Promising results have been obtained by analyzing datasets related to heart disease using a variety of machine learning algorithms and deep learning techniques. Neural networks and data mining techniques are employed in heart disease prediction systems (EHDPS) to estimate the risk of heart disease. Medical parameters like age, sex, blood pressure, cholesterol, and obesity are utilized by these systems.

It has been discovered that ensemble classification methods, like bagging and boosting, increase the prediction accuracy of weak classifiers in the heart disease prediction process. These methods have demonstrated adequate efficacy in the early detection of heart disease.

Coronary heart disease (CHD) risk is accurately predicted by predictive models that include categorical variables, such as blood pressure and cholesterol classifications. Doctors can use these models to predict multivariate CHD risk in patients who do not have overt CHD.

Machine learning classifiers that have demonstrated high accuracy in predicting the presence of coronary heart disease include Random Forest, Multilayer Perceptron, and Gradient Boosted Tree. Particularly Random Forest obtained the highest classification accuracy of 96.28%.

Heart disease must be detected early in order to be prevented and lives saved. In order to accurately classify and predict cases of heart disease with minimal attributes, comparative analysis of various classifiers has been carried out. In conclusion, promising results in heart attack prediction have been demonstrated by machine learning models, ensemble classification techniques, and predictive models incorporating categorical variables. These methods can aid medical professionals in interpreting patient situations and analyzing the disease's risk factors, which will ultimately improve heart disease management and prevention.

In conclusion, the benefits of machine learning include enhanced customer experience, fraud detection, personalized analytics, automation, personalization, data-driven decision making, advancements in healthcare, and industry optimization. Its capacity to evaluate and decipher vast volumes of data has the potential to completely transform how companies run and how we engage with technology. Thus, the machine learning models Logistic Regression and K-Nearest Neighbors provide the highest accuracy in prediction of heart diseases and heart attack.

CHAPTER 9 - BIBLIOGRAPHY

- [1] Felman, A. (2023, October 26). *Everything you need to know about heart disease*.
<https://www.medicalnewstoday.com/articles/237191#causes-and-risk-factors>
- [2] Ramalingam, V. V., Dandapath, A., & Raja, M. K. (2018). Heart disease prediction using machine learning techniques : a survey. *International Journal of Engineering & Technology*, 7(2.8), 684. <https://doi.org/10.14419/ijet.v7i2.8.10557>
- [3] Nandal, N., Goel, L., & Tanwar, R. (2022). Machine learning-based heart attack prediction: A symptomatic heart attack prediction method and exploratory analysis. *F1000Research*, 11, 1126. <https://doi.org/10.12688/f1000research.123776.1>
- [4] *Effective heart disease prediction using hybrid machine learning techniques*. (2019).
IEEE Journals & Magazine | IEEE Xplore.
<https://ieeexplore.ieee.org/abstract/document/8740989>
- [5] Alshraideh, M., Alshraideh, N., Alshraideh, A., Alkayed, Y., Trabsheh, Y. A., & Alshraideh, B. (2024). Enhancing Heart Attack Prediction with Machine Learning: A Study at Jordan University Hospital. *Applied Computational Intelligence and Soft Computing*, 2024, 1–16. <https://doi.org/10.1155/2024/5080332>

- [6] Kannan, D., Boxi, A., & IJARCCCE. (2022). Heart disease prediction using machine learning algorithms. In *International Journal of Advanced Research in Computer and Communication Engineering* (Vol. 11, Issue 6, p. 495) [Journal-article].
<https://doi.org/10.17148/IJARCCCE.2022.11696>
- [7] *What is a Heart Attack?* (2024, January 9). www.heart.org.
<https://www.heart.org/en/health-topics/heart-attack/about-heart-attacks>
- [8] Website, N. (2023, July 31). *Heart attack*. nhs.uk. <https://www.nhs.uk/conditions/heart-attack/>
- [9] Bhatt, C. M., Patel, P., Ghetia, T., & Mazzeo, P. L. (2023). Effective heart disease prediction using machine learning techniques. *Algorithms*, 16(2), 88.
<https://doi.org/10.3390/a16020088>
- [10] GeeksforGeeks. (2023, December 17). *Gaussian Naive Bayes using Sklearn*.
GeeksforGeeks. <https://www.geeksforgeeks.org/gaussian-naive-bayes-using-sklearn/>
- [11] Ahmad, A. A., & Polat, H. (2023). Prediction of heart disease based on machine learning using Jellyfish optimization algorithm. *Diagnostics*, 13(14), 2392.
<https://doi.org/10.3390/diagnostics13142392>

- [12] Sabay, A., Harris, L., Bejugama, V., & Jaceldo-Siegl, K. (n.d.). *Overcoming small data limitations in heart disease prediction by using surrogate data*. SMU Scholar.

https://scholar.smu.edu/datasciencereview/vol1/iss3/12/?utm_source=scholar.smu.edu%2Fdatasciencereview%2Fvol1%2Fiss3%2F12&utm_medium=PDF&utm_campaign=PDFCoverPages

Heart Disease Prediction Dataset Link –

- [13] *Heart disease dataset*. (2019, June 6). Kaggle.

<https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset>

APPENDIX I – MODEL AT GLANCE

Importing Libraries

```
In [1]: #Data Manipulation
import pandas as pd

#Numerical Python - Arrays
import numpy as np

#Visualization Libraries
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')

#Exploratory Data Analysis (EDA)
from collections import Counter
#Data Preprocessing
from sklearn import preprocessing
from sklearn.preprocessing import StandardScaler

#Data Split
from sklearn.model_selection import train_test_split
#Data Modeling
from sklearn.metrics import confusion_matrix, accuracy_score, roc_curve, classification_report
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
```

Reading the Dataset

```
In [2]: df = pd.read_csv('D:\\Khushi MCA\\MCA Semester 4\\archive (1)\\heart.csv')
df.head()
```

```
Out[2]:
```

	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

Total Number of Rows and Columns in the Dataset

```
In [3]: print("Number of Rows in the Dataset: ",df.shape[0])
print("Number of Columns in the Dataset: ",df.shape[1])

Number of Rows in the Dataset: 1025
Number of Columns in the Dataset: 14
```

Running head at the top of every page

Description of Columns

In [4]: `df.describe()`

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
count	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000	1025.000000
mean	54.434146	0.695610	0.942439	131.611707	246.000000	0.149268	0.529756	149.114146	0.336585	1.071512	1.385366	0.754146	2.323902	0.513171
std	9.072290	0.460373	1.029641	17.516718	51.59251	0.356527	0.527878	23.005724	0.472772	1.175053	0.617755	1.030798	0.620660	0.500070
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%	48.000000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	132.000000	0.000000	0.000000	1.000000	0.000000	2.000000	0.000000
50%	56.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	152.000000	0.000000	0.800000	1.000000	0.000000	2.000000	1.000000
75%	61.000000	1.000000	2.000000	140.000000	275.000000	0.000000	1.000000	166.000000	1.000000	1.800000	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

In [5]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1025 entries, 0 to 1024
Data columns (total 14 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   age             1025 non-null   int64
1   sex             1025 non-null   int64
2   cp              1025 non-null   int64
3   trestbps        1025 non-null   int64
4   chol            1025 non-null   int64
5   fbs             1025 non-null   int64
6   restecg         1025 non-null   int64
7   thalach         1025 non-null   int64
8   exang           1025 non-null   int64
9   oldpeak         1025 non-null   float64
10  slope           1025 non-null   int64
11  ca              1025 non-null   int64
12  thal            1025 non-null   int64
13  target          1025 non-null   int64
dtypes: float64(1), int64(13)
memory usage: 112.2 KB
```

Searching for Null and Duplicate Values

In [6]: `df.isnull().sum()`

```
Out[6]: age           0
sex           0
cp            0
trestbps      0
chol          0
fbs           0
restecg       0
thalach       0
exang         0
oldpeak       0
slope         0
ca            0
thal          0
target        0
dtype: int64
```

Running head at the top of every page

```
upper = df.duplicated().any()
```

```
In [7]: df_dup = df.duplicated().any()
print(df_dup)
```

True

```
In [8]: df = df.drop_duplicates()
```

```
In [9]: print("Number of Rows in the Dataset after Dropping Duplicate Values: ",df.shape[0])
print("Number of Columns in the Dataset after Dropping Duplicate Values: ",df.shape[1])
```

Number of Rows in the Dataset after Dropping Duplicate Values: 302

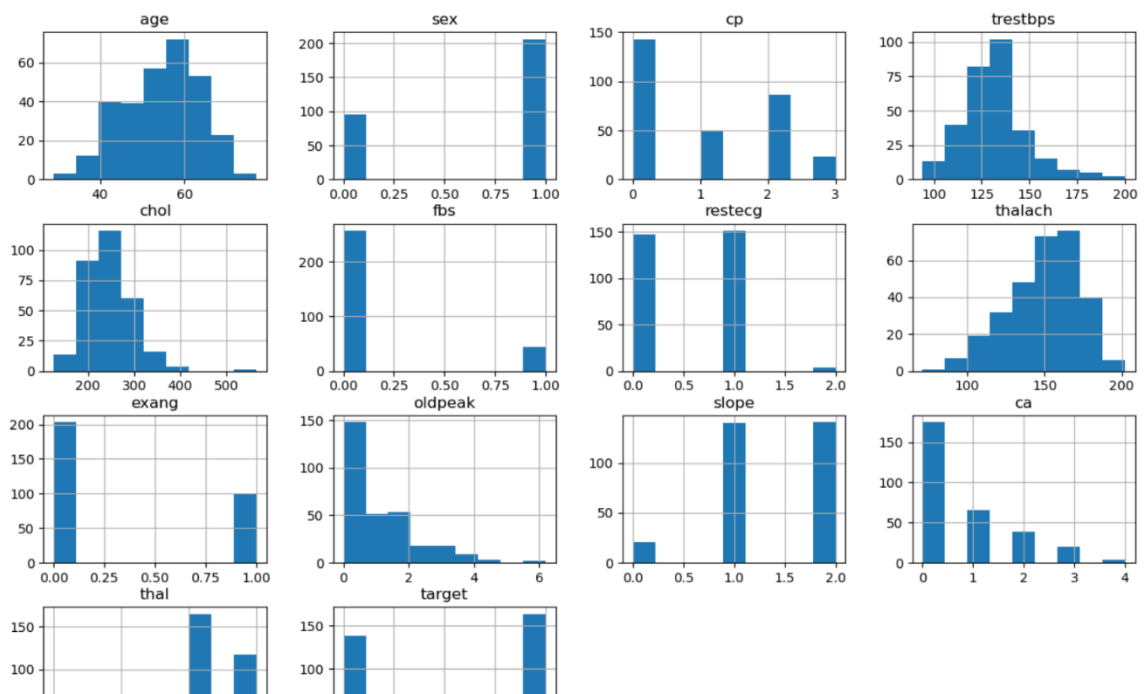
Number of Columns in the Dataset after Dropping Duplicate Values: 14

We do not have any missing values

Now lets have a look at the columns

```
In [10]: df.hist(figsize=(15,10), bins=9)
```

```
Out[10]: array([[<Axes: title={'center': 'age'}>, <Axes: title={'center': 'sex'}>,
<Axes: title={'center': 'cp'}>,
<Axes: title={'center': 'trestbps'}>],
[<Axes: title={'center': 'chol'}>,
<Axes: title={'center': 'fbs'}>,
<Axes: title={'center': 'restecg'}>,
<Axes: title={'center': 'thalach'}>],
[<Axes: title={'center': 'exang'}>,
<Axes: title={'center': 'oldpeak'}>,
<Axes: title={'center': 'slope'}>,
<Axes: title={'center': 'ca'}>],
[<Axes: title={'center': 'thal'}>,
<Axes: title={'center': 'target'}>, <Axes: >, <Axes: >]],
dtype=object)
```



Running head at the top of every page

```
In [11]: info = ["age","0: female, 1: male","chest pain type, 1: typical angina, 2: atypical angina, 3: non-anginal pain, 4: asymptomatic","resting blood pressure",
               "serum cholestoral in mg/dl","fasting blood sugar > 120 mg/dl",
               "resting electrocardiographic results (values 0,1,2)",
               "maximum heart rate achieved","exercise induced angina","oldpeak = ST depression induced by exercise relative to rest",
               "the slope of the peak exercise ST segment",
               "number of major vessels (0-3) colored by flourosopy","thal: 3 = normal; 6 = fixed defect; 7 = reversable defect"]

for i in range(len(info)):
    print(df.columns[i]+":\t\t\t"+info[i])

age:          age
sex:          0: female, 1: male
cp:          chest pain type, 1: typical angina, 2: atypical angina, 3: non-anginal pain, 4: asymptomatic
trestbps:     resting blood pressure
chol:         serum cholestoral in mg/dl
fbs:          fasting blood sugar > 120 mg/dl
restecg:      resting electrocardiographic results (values 0,1,2)
thalach:      maximum heart rate achieved
exang:        exercise induced angina
oldpeak:      oldpeak = ST depression induced by exercise relative to rest
slope:        the slope of the peak exercise ST segment
ca:           number of major vessels (0-3) colored by flourosopy
thal:         thal: 3 = normal; 6 = fixed defect; 7 = reversable defect
```

Searching for unique values first

```
In [12]: df['target'].value_counts()

Out[12]:
1    164
0    138
Name: target, dtype: int64
```

Searching for unique values first

```
In [12]: df['target'].value_counts()

Out[12]:
1    164
0    138
Name: target, dtype: int64
```

Exploratory Data Analysis (EDA)

```
In [13]: categorical_val = []
         continuous_val = []
         for column in df.columns:
             print('=====')
             print(f"{column} : {df[column].unique()}")
             if len(df[column].unique()) <= 10:
                 categorical_val.append(column)
             else:
                 continuous_val.append(column)

=====
age : [52 53 70 61 62 58 55 46 54 71 43 34 51 50 60 67 45 63 42 44 56 57 59 64
      65 41 66 38 49 48 29 37 47 68 76 40 39 77 69 35 74]
=====
sex : [1 0]
=====
cp : [0 1 2 3]
=====
trestbps : [125 140 145 148 138 100 114 160 120 122 112 132 118 128 124 106 104 135
           130 136 180 129 150 178 146 117 152 154 170 134 174 144 108 123 110 142
           126 192 115 94 200 165 102 105 155 172 164 156 101]
=====
chol : [212 203 174 294 248 318 289 249 286 149 341 210 298 204 308 266 244 211
       185 223 208 252 209 307 233 319 256 327 169 131 269 196 231 213 271 263
       229 360 258 330 342 226 228 278 230 283 241 175 188 217 193 245 232 299
       288 197 315 215 164 326 207 177 257 255 187 201 220 268 267 236 303 282
       126 309 186 275 281 206 335 218 254 295 417 260 240 302 192 225 325 235
       274 234 182 167 172 321 300 199 564 157 304 222 184 354 160 247 239 246
       409 293 180 250 221 200 227 243 311 261 242 205 306 219 353 198 394 183
       237 224 265 313 340 259 270 216 264 276 322 214 273 253 176 284 305 168
       407 290 277 262 195 166 178 141]
=====
fbs : [0 1]
=====
```

Running head at the top of every page

```

=====
fbs : [0 1]
=====
restecg : [1 0 2]
=====
thalach : [168 155 125 161 106 122 140 145 144 116 136 192 156 142 109 162 165 148
172 173 146 179 152 117 115 112 163 147 182 105 150 151 169 166 178 132
160 123 139 111 180 164 202 157 159 170 138 175 158 126 143 141 167 95
190 118 103 181 108 177 134 120 171 149 154 153 88 174 114 195 133 96
124 131 185 194 128 127 186 184 188 130 71 137 99 121 187 97 90 129
113]
=====
exang : [0 1]
=====
oldpeak : [1. 3.1 2.6 0. 1.9 4.4 0.8 3.2 1.6 3. 0.7 4.2 1.5 2.2 1.1 0.3 0.4 0.6
3.4 2.8 1.2 2.9 3.6 1.4 0.2 2. 5.6 0.9 1.8 6.2 4. 2.5 0.5 0.1 2.1 2.4
3.8 2.3 1.3 3.5]
=====
slope : [2 0 1]
=====
ca : [2 0 1 3 4]
=====
thal : [3 2 1 0]
=====
target : [0 1]

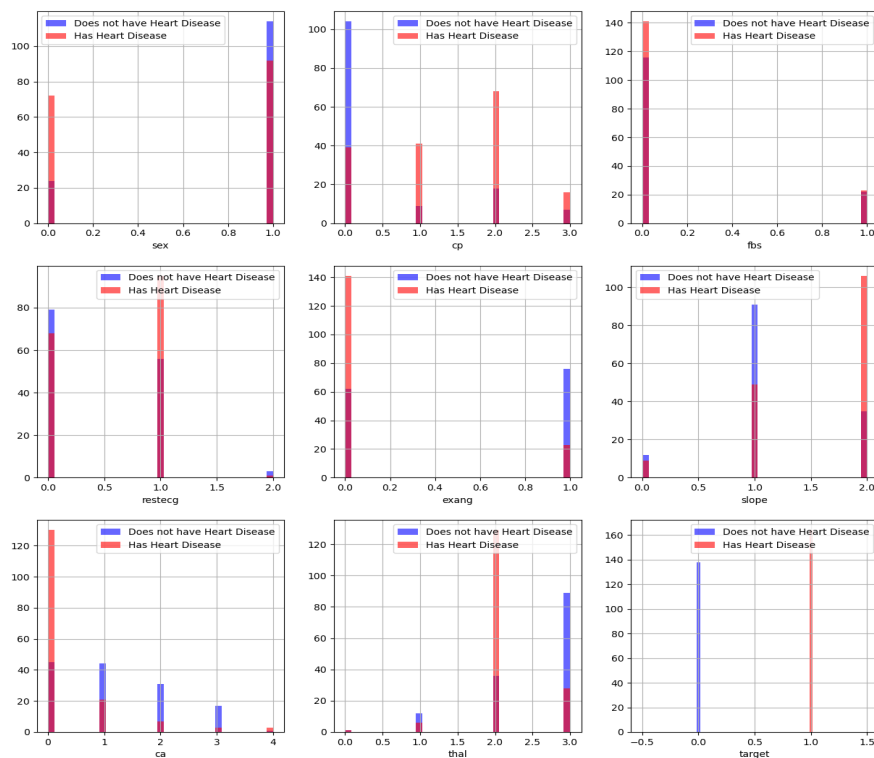
```

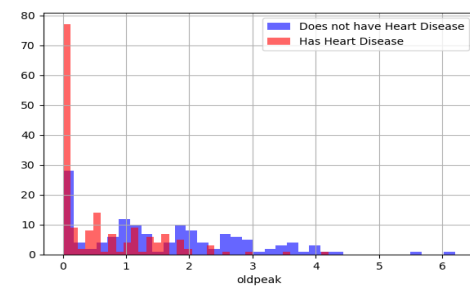
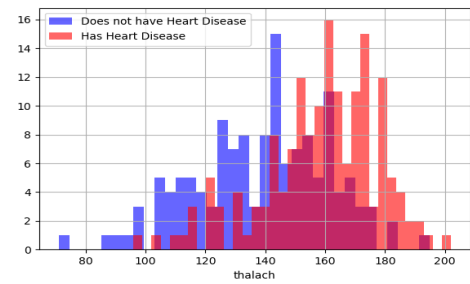
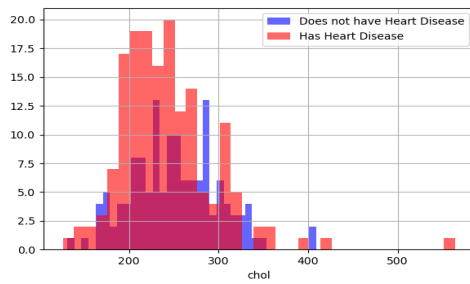
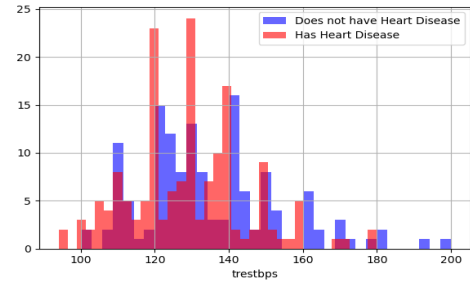
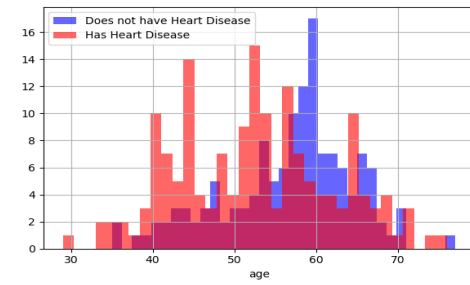
```

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

for i, column in enumerate(categorical_val, 1):
    plt.subplot(3, 3, i)
    df[df["target"] == 0][column].hist(bins=35, color='blue', label='Does not have Heart Disease', alpha=0.6)
    df[df["target"] == 1][column].hist(bins=35, color='red', label='Has Heart Disease', alpha=0.6)
    plt.legend()
    plt.xlabel(column)
    plt.savefig('HeartDisease1.png')

```





Splitting the Data into Training and Testing Dataset

```
n [16]: y = df["target"]
X = df.drop('target', axis=1)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=0)

print("X_Train Shape is: ",X_train)
print("X_Test Shape is: ", X_test)

print("y_Train Shape is: ", y_train)
print("y_Test Shape is: ", y_test)
```

```
X_Train Shape is:      age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
81    49    1    2      118   149    0         0    126      0      0.8
193   69    1    3      160   234    1         0    131      0      0.1
70    59    1    0      170   326    0         0    140      1      3.4
719   52    1    0      108   233    1         1    147      0      0.1
628   69    0    3      140   239    0         1    151      0      1.8
..    ...    ...    ..      ...    ...    ...      ...    ...    ...    ...
425   51    0    0      130   305    0         1    142      1      1.2
271   44    1    1      120   263    0         1    173      0      0.0
143   34    1    3      118   182    0         0    174      0      0.0
50    58    0    3      150   283    1         0    162      0      1.0
232   60    1    0      125   258    0         0    141      1      2.8
```

```
      slope  ca  thal
81         2   3    2
193        1   1    2
70         0   0    3
719        2   3    3
628        2   2    2
..        ...  ..   ...
425        1   0    3
271        2   0    3
143        2   0    2
50         2   0    2
232        1   1    3
```

[241 rows x 13 columns]

```
X_Test Shape is:      age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
342   65    0    2      155   269    0         1    148      0      0.8
191   56    1    1      130   221    0         0    163      0      0.0
349   62    0    2      130   263    0         1     97      0      1.2
288   58    0    2      120   340    0         1    172      0      0.0
```

Running head at the top of every page

```
[241 rows x 13 columns]
X_Test Shape is:      age  sex  cp  trestbps  chol  fbs  restecg  thalach  exang  oldpeak  \
342  65    0  2      155  269    0        1   148    0    0.8
191  56    1  1      130  221    0        0   163    0    0.0
349  62    0  2      130  263    0        1    97    0    1.2
288  58    0  2      120  340    0        1   172    0    0.0
56   56    1  3      120  193    0        0   162    0    1.9
..   ...  ...  ..      ...  ...  ...      ...  ...  ...  ...
182  60    1  0      140  293    0        0   170    0    1.2
878  54    1  0      120  188    0        1   113    0    1.4
27   58    0  1      136  319    1        0   152    0    0.0
128  52    1  2      138  223    0        1   169    0    0.0
102  54    1  1      108  309    0        1   156    0    0.0

      slope  ca  thal
342      2    0    2
191      2    0    3
349      1    1    3
288      2    0    2
56       1    0    3
..   ...  ...  ...
182      1    2    3
878      1    1    3
27       2    2    2
128      2    4    2
102      2    0    3
```

```
[61 rows x 13 columns]
y_Train Shape is:  81    0
193    1
70     0
719    1
628    1
..
425    0
271    1
143    1
50     1
232    0
Name: target, Length: 241, dtype: int64
y_Test Shape is:  342    1
191    1
349    0
288    1
56     1
```

Checking for Unique Values

```
[17]: print(y_test.unique())
Counter(y_train)

[1 0]
it[17]: Counter({0: 113, 1: 128})
```

Standardizing

```
[18]: scaler = StandardScaler()

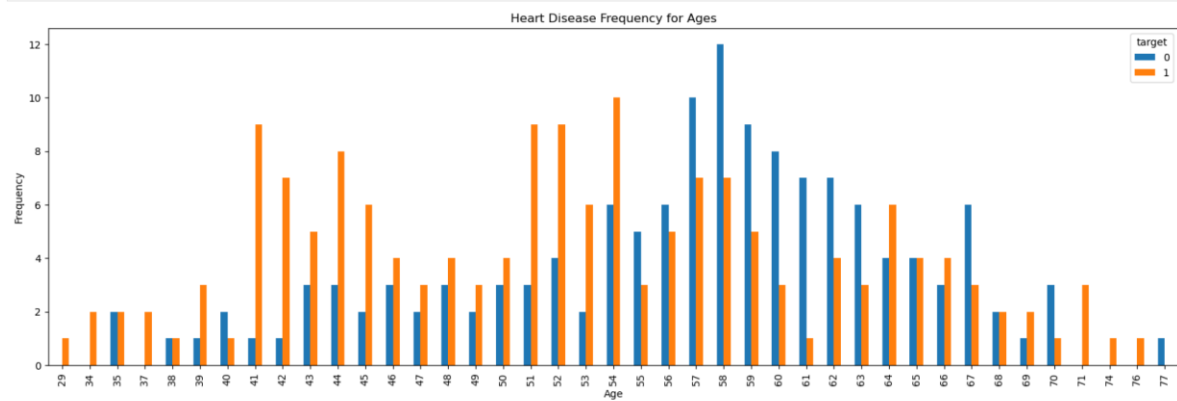
X_train = scaler.fit_transform(X_train)

X_test = scaler.transform(X_test)
```

Heart Disease Frequency for Ages

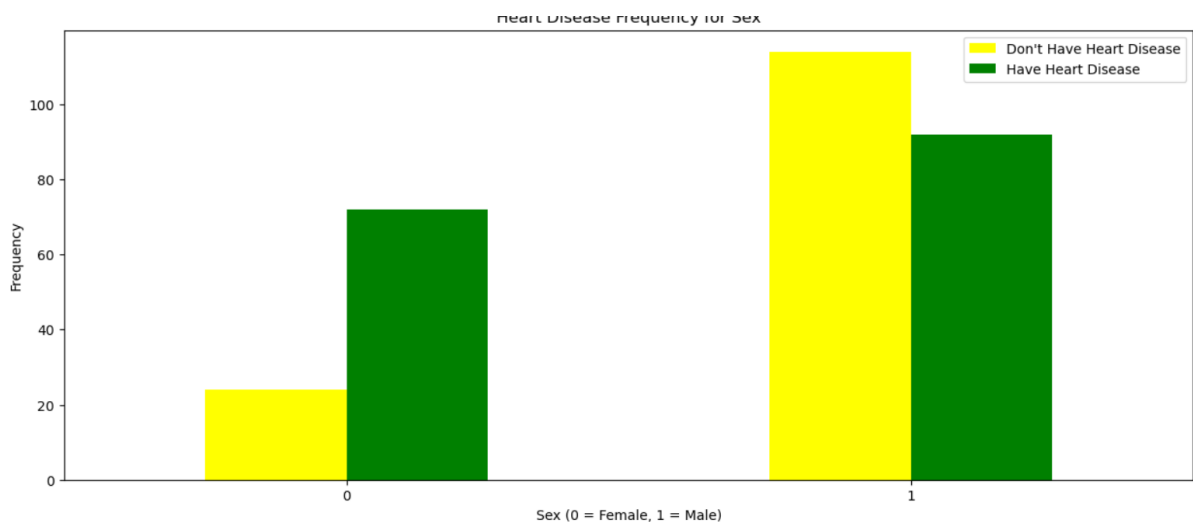
```
[19]: pd.crosstab(df.age,df.target).plot(kind="bar",figsize=(20,6))
plt.title('Heart Disease Frequency for Ages')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.savefig('heartDiseaseAndAges.png')
plt.show()
```

Running head at the top of every page



Heart Disease Frequency according to the Genders - i.e Males and Female Frequencies

```
[10]: pd.crosstab(df.sex,df.target).plot(kind="bar",figsize=(15,6),color=['yellow','green' ])
plt.title('Heart Disease Frequency for Sex')
plt.xlabel('Sex (0 = Female, 1 = Male)')
plt.xticks(rotation=90)
plt.legend(["Don't Have Heart Disease", "Have Heart Disease"])
plt.ylabel('Frequency')
plt.savefig('HeartDiseaseFrequency.png')
plt.show()
```

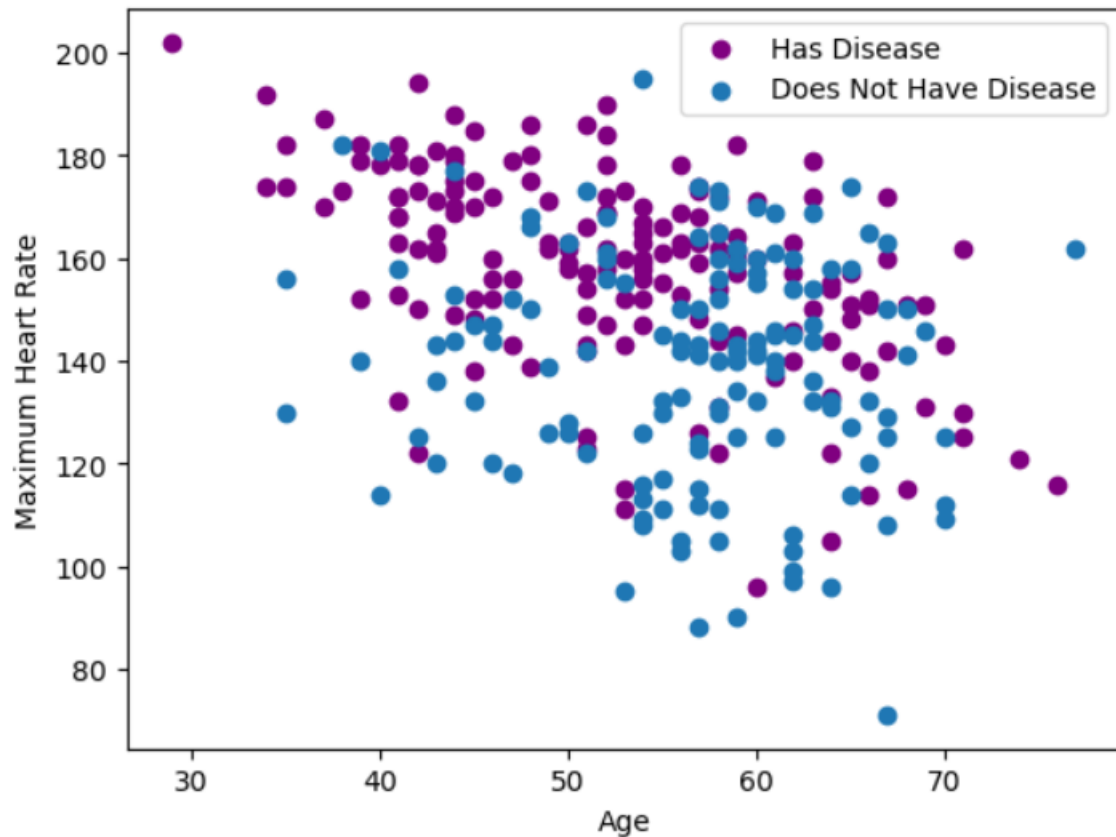


Age wise Heart Disease Rate

```
plt.scatter(x=df.age[df.target==1], y=df.thalach[(df.target==1)], c="purple")
plt.scatter(x=df.age[df.target==0], y=df.thalach[(df.target==0)])
plt.legend(["Has Disease", "Does Not Have Disease"])
plt.xlabel("Age")
plt.ylabel("Maximum Heart Rate")
plt.savefig('HeartDiseaseRate.png')
plt.show()
```

Age wise Heart Disease Rate

```
In [21]: plt.scatter(x=df.age[df.target==1], y=df.thalach[(df.target==1)], c="purple")
plt.scatter(x=df.age[df.target==0], y=df.thalach[(df.target==0)])
plt.legend(["Has Disease", "Does Not Have Disease"])
plt.xlabel("Age")
plt.ylabel("Maximum Heart Rate")
plt.savefig('HeartDiseaseRate.png')
plt.show()
```



Feature Selection

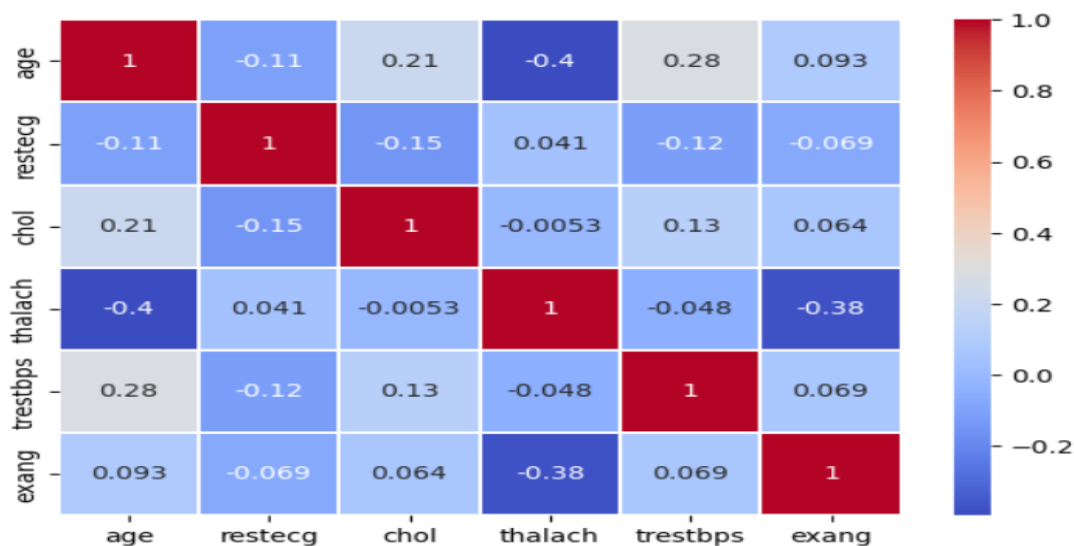
```
2]: names=['age','restecg','chol','thalach','trestbps','exang']

#Set the width and height of the plot
f, ax = plt.subplots(figsize=(7, 5))

#Correlation plot
corr = df.loc[:,names]
#Generate correlation matrix
correlation = corr.corr()

#Plot using seaborn library
sns.heatmap(correlation, annot = True, cmap='coolwarm',linewidths=.1)
plt.savefig('FeatureSelection.png')
```

Running head at the top of every page



In [23]: corr

```
Out[23]:
```

	age	restecg	chol	thalach	trestbps	exang
0	52	1	212	168	125	0
1	53	0	203	155	140	1
2	70	1	174	125	145	1
3	61	1	203	161	148	0
4	62	1	294	106	138	0
...
723	68	0	211	115	120	0
733	44	1	141	175	108	0
739	52	1	255	161	128	1
843	59	0	273	125	160	0
878	54	1	188	113	120	0

302 rows × 6 columns

```
In [24]: df.drop('target', axis=1).corrwith(df.target).plot(kind='bar', grid=True, figsize=(12, 8),  
title="Correlation with target")  
plt.savefig('CorrwithTarget.png')
```

Data Normalization

```
In [25]: x= preprocessing.StandardScaler().fit(X).transform(X.astype(float))
x[0:5]
```

```
Out[25]: array([[ -0.26796589,  0.68265615, -0.93520799, -0.37655636, -0.66772815,
        -0.41844626,  0.90165655,  0.80603539, -0.69834428, -0.03712404,
         0.97951442,  1.27497996,  1.11996657],
       [ -0.15726042,  0.68265615, -0.93520799,  0.47891019, -0.84191811,
         2.38979311, -1.0025412 ,  0.23749516,  1.43195847,  1.77395808,
        -2.27118179, -0.71491124,  1.11996657],
       [ 1.72473259,  0.68265615, -0.93520799,  0.76406571, -1.40319685,
        -0.41844626,  0.90165655, -1.07452077,  1.43195847,  1.34274805,
        -2.27118179, -0.71491124,  1.11996657],
       [ 0.72838335,  0.68265615, -0.93520799,  0.93515902, -0.84191811,
        -0.41844626,  0.90165655,  0.49989834, -0.69834428, -0.8995441 ,
         0.97951442,  0.28003436,  1.11996657],
       [ 0.83908882, -1.46486632, -0.93520799,  0.36484799,  0.91933586,
         2.38979311,  0.90165655, -1.90546419, -0.69834428,  0.73905401,
        -0.64583368,  2.26992556, -0.51399432]])
```

First Model - Logistic Regression

```
|: model1 = 'Logistic Regression'
lr = LogisticRegression()
model = lr.fit(X_train, y_train)
lr_predict = lr.predict(X_test)
lr_conf_matrix = confusion_matrix(y_test, lr_predict)
lr_acc_score = accuracy_score(y_test, lr_predict)
print("Confusion Matrix")
print(lr_conf_matrix)
print("\n")
print("Accuracy of Logistic Regression:",lr_acc_score*100,'\n')
print(classification_report(y_test,lr_predict))
```

Confusion Matrix

```
[[20  5]
 [ 5 31]]
```

Accuracy of Logistic Regression: 83.60655737704919

	precision	recall	f1-score	support
0	0.80	0.80	0.80	25
1	0.86	0.86	0.86	36
accuracy			0.84	61
macro avg	0.83	0.83	0.83	61
weighted avg	0.84	0.84	0.84	61

Second Model - Naive Bayes

```
model2 = 'Naive Bayes'
nb = GaussianNB()
nb.fit(X_train,y_train)
nbpred = nb.predict(X_test)
nb_conf_matrix = confusion_matrix(y_test, nbpred)
nb_acc_score = accuracy_score(y_test, nbpred)
print("Confusion Matrix")
print(nb_conf_matrix)
print("\n")
print("Accuracy of Naive Bayes model:",nb_acc_score*100,'\n')
print(classification_report(y_test,nbpred))
```

Confusion Matrix

```
[[20  5]
 [ 7 29]]
```

Accuracy of Naive Bayes model: 80.32786885245902

	precision	recall	f1-score	support
0	0.74	0.80	0.77	25
1	0.85	0.81	0.83	36
accuracy			0.80	61
macro avg	0.80	0.80	0.80	61
weighted avg	0.81	0.80	0.80	61

Third Model - Random Forest Classifier

```
model3 = 'Random Forest Classifier'
rf = RandomForestClassifier(n_estimators=20, random_state=12,max_depth=5)
rf.fit(X_train,y_train)
rf_predicted = rf.predict(X_test)
rf_conf_matrix = confusion_matrix(y_test, rf_predicted)
rf_acc_score = accuracy_score(y_test, rf_predicted)
print("Confusion Matrix")
print(rf_conf_matrix)
print("\n")
print("Accuracy of Random Forest:",rf_acc_score*100,'\n')
print(classification_report(y_test,rf_predicted))
```

Confusion Matrix

```
[[18  7]
 [ 5 31]]
```

Accuracy of Random Forest: 80.32786885245902

	precision	recall	f1-score	support
0	0.78	0.72	0.75	25
1	0.82	0.86	0.84	36
accuracy			0.80	61
macro avg	0.80	0.79	0.79	61
weighted avg	0.80	0.80	0.80	61

Fourth Model - K-Neighbors Classifier

```
model4 = 'K-NeighborsClassifier'  
knn = KNeighborsClassifier(n_neighbors=10)  
knn.fit(X_train, y_train)  
knn_predicted = knn.predict(X_test)  
knn_conf_matrix = confusion_matrix(y_test, knn_predicted)  
knn_acc_score = accuracy_score(y_test, knn_predicted)  
print("Confusion Matrix")  
print(knn_conf_matrix)  
print("\n")  
print("Accuracy of K-NeighborsClassifier:",knn_acc_score*100,'\n')  
print(classification_report(y_test,knn_predicted))
```

Confusion Matrix

```
[[19  6]  
 [ 5 31]]
```

Accuracy of K-NeighborsClassifier: 81.9672131147541

	precision	recall	f1-score	support
0	0.79	0.76	0.78	25
1	0.84	0.86	0.85	36
accuracy			0.82	61
macro avg	0.81	0.81	0.81	61
weighted avg	0.82	0.82	0.82	61

Fifth Model - Decision Tree Classifier

```
model5 = 'DecisionTreeClassifier'  
dt = DecisionTreeClassifier(criterion = 'entropy',random_state=0,max_depth = 6)  
dt.fit(X_train, y_train)  
dt_predicted = dt.predict(X_test)  
dt_conf_matrix = confusion_matrix(y_test, dt_predicted)  
dt_acc_score = accuracy_score(y_test, dt_predicted)  
print("Confusion Matrix")  
print(dt_conf_matrix)  
print("\n")  
print("Accuracy of DecisionTreeClassifier:",dt_acc_score*100,'\n')  
print(classification_report(y_test,dt_predicted))
```

Confusion Matrix

```
[[18  7]  
 [ 6 30]]
```

Accuracy of DecisionTreeClassifier: 78.68852459016394

	precision	recall	f1-score	support
0	0.75	0.72	0.73	25
1	0.81	0.83	0.82	36
accuracy			0.79	61
macro avg	0.78	0.78	0.78	61
weighted avg	0.79	0.79	0.79	61

Sixth Model - Support Vector Classifier

```
model6 = 'Support Vector Classifier'
svc = SVC(kernel='rbf', C=2)
svc.fit(X_train, y_train)
svc_predicted = svc.predict(X_test)
svc_conf_matrix = confusion_matrix(y_test, svc_predicted)
svc_acc_score = accuracy_score(y_test, svc_predicted)
print("Confusion matrix")
print(svc_conf_matrix)
print("\n")
print("Accuracy of Support Vector Classifier:", svc_acc_score*100, '\n')
print(classification_report(y_test, svc_predicted))
```

Confusion matrix

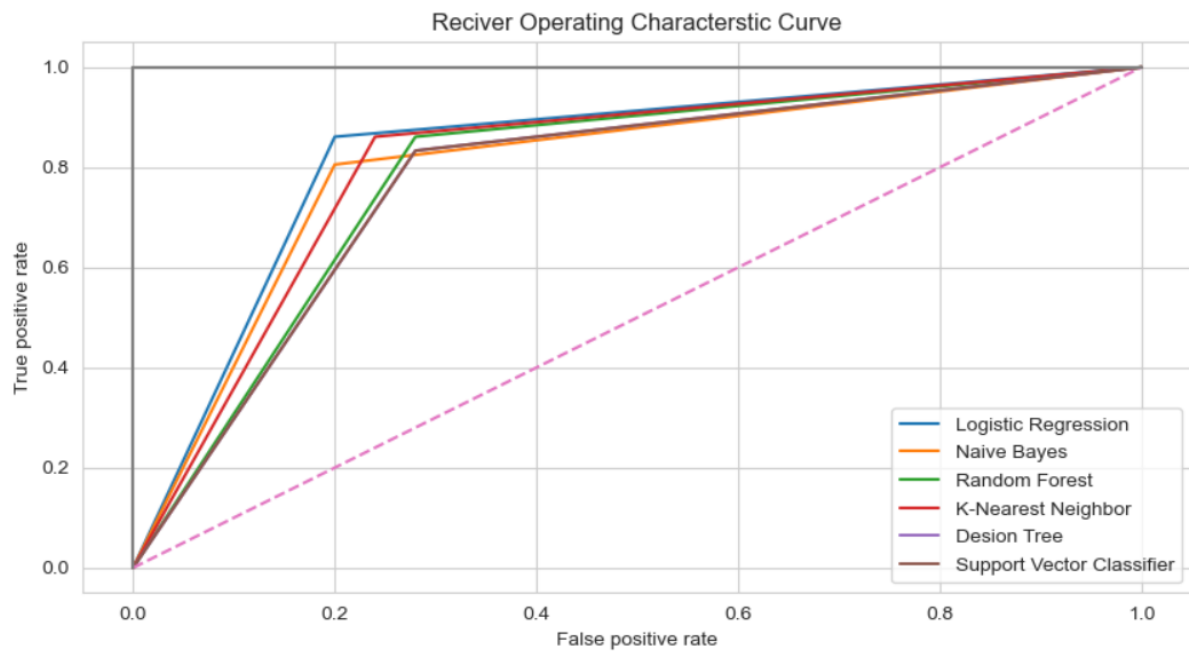
```
[[18  7]
 [ 6 30]]
```

Accuracy of Support Vector Classifier: 78.68852459016394

	precision	recall	f1-score	support
0	0.75	0.72	0.73	25
1	0.81	0.83	0.82	36
accuracy			0.79	61
macro avg	0.78	0.78	0.78	61
weighted avg	0.79	0.79	0.79	61

```
lr_false_positive_rate,lr_true_positive_rate,lr_threshold = roc_curve(y_test,lr_predict)
nb_false_positive_rate,nb_true_positive_rate,nb_threshold = roc_curve(y_test,nbpred)
rf_false_positive_rate,rf_true_positive_rate,rf_threshold = roc_curve(y_test,rf_predicted)
knn_false_positive_rate,knn_true_positive_rate,knn_threshold = roc_curve(y_test,knn_predicted)
dt_false_positive_rate,dt_true_positive_rate,dt_threshold = roc_curve(y_test,dt_predicted)
svc_false_positive_rate,svc_true_positive_rate,svc_threshold = roc_curve(y_test,svc_predicted)
```

```
sns.set_style('whitegrid')
plt.figure(figsize=(10,5))
plt.title('Reciver Operating Characterstic Curve')
plt.plot(lr_false_positive_rate,lr_true_positive_rate,label='Logistic Regression')
plt.plot(nb_false_positive_rate,nb_true_positive_rate,label='Naive Bayes')
plt.plot(rf_false_positive_rate,rf_true_positive_rate,label='Random Forest')
plt.plot(knn_false_positive_rate,knn_true_positive_rate,label='K-Nearest Neighbor')
plt.plot(dt_false_positive_rate,dt_true_positive_rate,label='Desion Tree')
plt.plot(svc_false_positive_rate,svc_true_positive_rate,label='Support Vector Classifier')
plt.plot([0,1],ls='--')
plt.plot([0,0],[1,0],c='.5')
plt.plot([1,1],c='.5')
plt.ylabel('True positive rate')
plt.xlabel('False positive rate')
plt.legend()
plt.savefig('Output1.png')
plt.show()
```

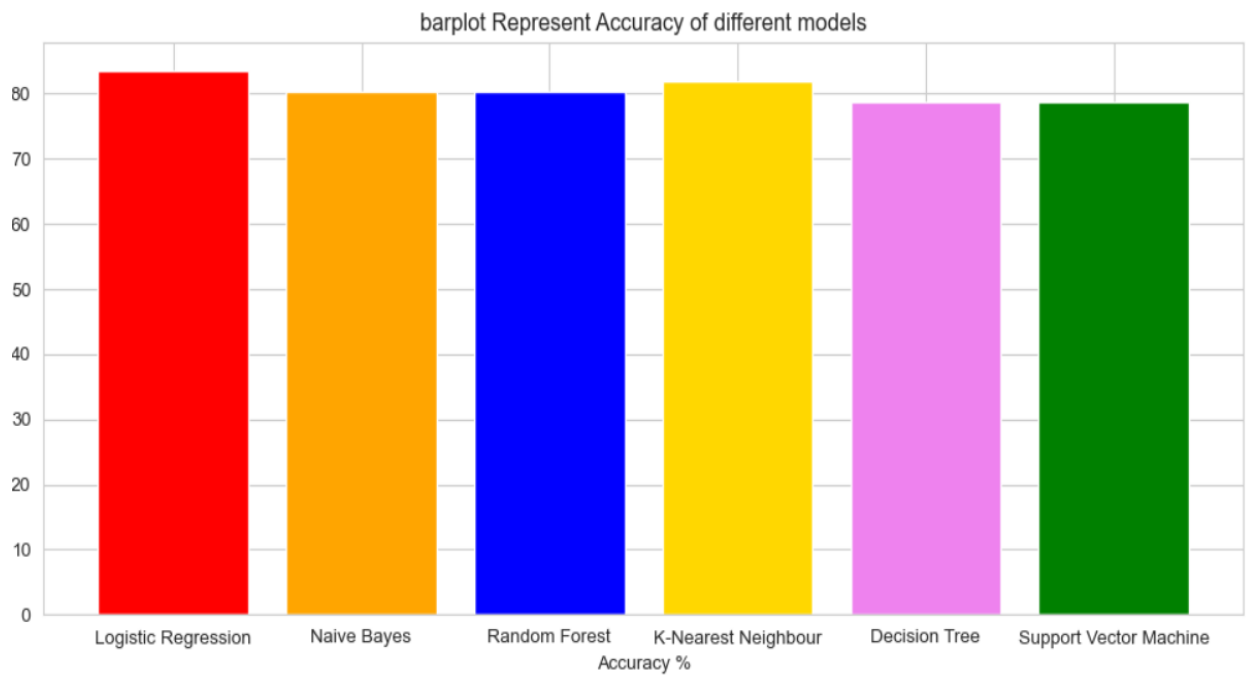


Model Summary

```
model_summary = pd.DataFrame({'Model': ['Logistic Regression','Naive Bayes','Random Forest',
    'K-Nearest Neighbour','Decision Tree','Support Vector Machine'], 'Accuracy': [lr_acc_score*100,
    nb_acc_score*100,rf_acc_score*100,knn_acc_score*100,dt_acc_score*100,svc_acc_score*100]})
model_summary
```

	Model	Accuracy
0	Logistic Regression	83.606557
1	Naive Bayes	80.327869
2	Random Forest	80.327869
3	K-Nearest Neighbour	81.967213
4	Decision Tree	78.688525
5	Support Vector Machine	78.688525

```
colors = ['red','orange','blue','gold','violet','green',]
plt.figure(figsize=(12,5))
plt.title("barplot Represent Accuracy of different models")
plt.xlabel("Accuracy %")
plt.ylabel("Algorithms")
plt.bar(model_summary['Model'],model_summary['Accuracy'],color = colors)
plt.savefig('Output2.png')
plt.show()
```



APPENDIX II – MINUTES OF MEETING

MINUTES OF MEETING OF FIRST GOOGLE MEET

Date – 15th May 2024

Time – 9:30 am to 10:15 am

Participants –

1. Miss Khushi Govind Upadhye
2. Mrs Amruta Mohan Chimanna

Subject – Finalizing the Major Project Topic for Machine Learning Project.

Summary –

I arranged a Google Meet on 11th May 2024 to suggest the topics which were chosen by me to my project guide, Mrs. Amruta Chimanna who is an assistant professor at Walchand College. I showed her the format and guidelines which were received by me from Amity University Online, Noida, Uttar Pradesh.

I presented a few project topics like recommender systems, disease prediction, and sentiment analysis projects. She advised me to go with the disease prediction models where I could choose any disease like diabetes prediction, autism prediction, heart attack prediction, Parkinson's disease prediction, cancer detection and prediction etc. She explained me that disease prediction will be a good topic so I moved forward with it. She explained me the basic flow of the model, and told me to finalize a topic and start reading research journals, articles and available literature on the internet.

Completed on - 16th May 2024.

APPENDIX III – MINUTES OF MEETING II

MINUTES OF MEETING OF SECOND GOOGLE MEET -

Date – 2nd June 2024

Time - 4:20 pm to 5:30 pm

Participants -

1. Khushi Govind Upadhye
2. Amruta Mohan Chimanna

Subject – Model Basics

Summary –

I shared the Google Meet link and shared her a demo model which I had prepared on the heart disease prediction. I asked her to go through the code once and tell me if there were any improvements required. Initially I had included three Machine Learning Algorithms like Logistic Regression, Support Vector Classifier and KNN algorithm. She advised me to include three more, so I added the Random Forest Classifier, Decision Tree Algorithm and Naïve Bayes Algorithm to my model.

Completed on – 12th June 2024

APPENDIX IV – MINUTES OF MEETING III -

MINUTES OF MEETING OF THIRD GOOGLE MEET –

Date – 13th June 2024

Time – 10:35 am to 10:58 am

Participants –

1. Miss. Khushi Govind Upadhye
2. Mrs. Amruta Mohan Chimanna

Subject – Coding portion completion and approval.

Summary –

I sent her my project report and asked her if there were any changes required in the prediction model coding part. She told me that it was perfectly fine and I could move ahead with the project report. My project report was partially ready, the findings and the recommendations were left. So, I sent her whatever was completed at that time frame.

Completed on – 16th June 2024

Running head at the top of every page

APPENDIX V – MINUTES OF MEETING IV –

MINUTES OF MEETING OF FOURTH GOOGLE MEET –

Date – 18th June 2024

Time – 2:20 pm

Participants –

1. Miss. Khushi Govind Upadhye
2. Mrs. Amruta Mohan Chimanna

Subject – Improvements given by the project guide.

Summary –

I sent her the complete project report after which she suggested me a few corrections – to include all the figure labels in the center, add the recommendations and highlighting the main headings. She had approved for the rest of the project report and informed me that there were no further corrections required.

Completed on – 20th June 2024.