Project Report

On

DISEASE PREDICTION USING MACHINE LEARNING

Submitted

in partial fulfilment of the requirements for the degree of

BACHELOR OF SCIENCE

in

MATHEMATICS

by

SNEHA THOMAS (AB21BMAT054)

Under the Supervision of

DR. ELIZABETH RESHMA M T

THE FEAR OF THE LORD
IS THE BEGINNING OF WISDOM

ST. TERESA'S COLLEGE
(AUTONOMOUS)
ESTD 1925

DEPARTMENT OF MATHEMATICS

ST. TERESA'S COLLEGE (AUTONOMOUS)

ERNAKULAM, KOCHI – 682011

APRIL 2024

ST. TERESA'S COLLEGE (AUTONOMOUS), ERNAKULAM

THE FEAR OF THE LORD IS THE BEGINNING OF WISDOM

ST. TERESA'S COLLEGE
(AUTONOMOUS)
ESTD 1925

CERTIFICATE

This is to certify that the dissertation entitled, **DISEASE PREDICTION USING MACHINE LEARNING** is a bonafide record of the work done by **Ms. SNEHA THOMAS** under my guidance as partial fulfillment of the award of the degree of **Bachelor of Science in Mathematics** at St. Teresa's College (Autonomous), Ernakulam affiliated to Mahatma Gandhi University, Kottayam. No part of this work has been submitted for any other degree elsewhere.

Date: 16/02/2024
Place: Ernakulam

Reshma

DR. ELIZABETH RESHMA M T
Assistant Professor,
Department of Mathematics,
St. Teresa's College (Autonomous),
Ernakulam.

ST. TERESA'S COLLEGE (AUTONOMOUS)
DEPARTMENT OF MATHEMATICS AND STATISTICS
ERNAKULAM

Ursala

DR. URSALA PAUL
Assistant Professor and Head,
Department of Mathematics,
St. Teresa's College (Autonomous),
Ernakulam.

External Examiners

1:………………………… 30.04.24

2:…………………………

DECLARATION

I hereby declare that the work presented in this project is based on the original work done by me under the guidance of Dr. ELIZABETH RESHMA M T, Assistant Professor, Department of Mathematics and Statistics, St. Teresa's College (Autonomous) , Ernakulam and has not been included in any other project submitted previously for the award of any degree.

Ernakulam **SNEHA THOMAS**

Date:16/02/2024 **AB21BMAT054**

ACKNOWLEDGEMENT

I want to express my deep gratitude to Mrs. Dr. Elizabeth Reshma M T, Assistant Professor at St. Teresa's College, for her exceptional coordination and guidance throughout the various stages of the project. I also want to acknowledge the valuable contributions of my group members during the engaging defense sessions and for their insightful inputs. Special thanks are also due to Dr. Ursala Paul, Dr. Elizabeth Reshma M T, and divine guidance for their invaluable support in navigating through challenges, ultimately leading to the successful completion of this project.

Ernakulam **SNEHA THOMAS**

Date: 16/02/2024 **AB21BMAT054**

CONTENTS

CHAPTER 1

INTRODUCTION

Nowadays, humans face various diseases due to the current environmental condition and their living habits. The identification and prediction of such diseases at their earlier stages are much important, so as to prevent the extremity of it. Hence the disease prediction system plays an important role as it predicts the disease based on symptoms. This could be achieved by using machine learning techniques.

1.1 MACHINE LEARNING

In the modern era, humans are experiencing exponential growth of data like never before. With the availability of online data and inexpensive computational computer power, machine learning algorithms can learn and develop models without human intervention. Machine learning, a subset of artificial intelligence, can collect meaningful knowledge from its training data and automatically improve through exposure without having to be programmed. The goal of machine learning generally is to understand the structure of data and fit that data into models that can be understood and utilized by people.

Machine learning algorithms can be classified into three : supervised, unsupervised, and reinforcement learning.

Supervised learning :

Supervised learning is a category of machine learning that uses labeled datasets to train algorithms to predict outcomes and recognize patterns. Supervised learning algorithms are given labeled training to learn the relationship between the input and the outputs. Supervised machine learning algorithms make it easier for organizations to create complex models that can make accurate predictions. As a result, they are widely used across various industries and fields, including healthcare, marketing, financial services, and more. Supervised learning can be further classified into two categories : classification and regression.

Classification uses an algorithm to accurately assign test data into specific categories. It recognizes specific entities within the dataset and attempts to draw some conclusions on how those entities should be labeled or defined. Common classification algorithms are linear classifiers, support vector machines (SVM), decision trees, k-nearest neighbor, and random forest.

Regression algorithms are used to predict a real or continuous value, where the algorithm detects a relationship between two or more variables. Linear regression, logistical regression, and polynomial regression are popular regression algorithms.

Unsupervised learning :

Unsupervised learning is a type of machine learning that learns from data without human supervision. Unlike supervised learning, the model is given raw, unlabeled data and has to infer its own rules and structure the information based on similarities, differences, and patterns without explicit instructions on how to work with each piece of data. Unsupervised learning algorithms are better suited for more complex processing tasks, such as organizing large datasets into clusters. They are useful for identifying previously undetected patterns in data and can help identify features useful for categorizing data. Unsupervised learning can be classified into two categories: clustering and association.

Clustering is a technique for exploring raw, unlabeled data and breaking down into groups based on similarities or differences. Clustering algorithms split data into natural groups by finding similar structures or patterns in uncategorized data. There are several types of unsupervised learning algorithms that are used for clustering, which include exclusive, overlapping, hierarchical, and probabilistic clustering.

Association rule mining is a rule-based approach to reveal interesting relationships between data points in large datasets. Unsupervised learning algorithms search for frequent if-then associations-also called rules-to discover correlations and co-occurrences within the data and the different connections between data objects.

Reinforcement learning:

Reinforcement learning refers to a sub-field of machine learning that enables AI-based systems to take actions in a dynamic environment through trial and error to maximize the collective rewards based on the feedback generated for individual activities. In the RL context, feedback refers to a positive or negative notion reflected through rewards or punishments. RL optimizes AI-driven systems by imitating natural intelligence that emulates human cognition.

1.2 MEDICAL BIG DATA

Big data in healthcare and medicine refers to these various large and complex data, which they are difficult to analyze and manage with traditional software or hardware. Big data analytics covers integration of heterogeneous data, data quality control, analysis, modeling, interpretation and validation. Application of big data analytics provides comprehensive knowledge discovering from the available huge amount of data.

Particularly, big data analytics in medicine and healthcare enables analysis of the large datasets from thousands of patients, identifying clusters and correlation between datasets, as well as developing predictive models using machine learning techniques. Big data analytics in medicine and healthcare integrates analysis of several scientific areas such as bioinformatics, medical imaging, sensor informatics, medical informatics and health informatics. Medical big data is widely used to improve healthcare quality. Such data include audio, lab tests, previous diagnostic reports, clinical records, research, and images. ML plays vital roles in analysis of different data in medical centers. The techniques and algorithms can be directly used on a data set for creating some models or to draw vital conclusions from the dataset.

1.3 RISE OF MACHINE LEARNING IN HEALTHCARE SETTINGS

As technology expands, machine learning provides an exciting opportunity in healthcare to improve the accuracy of diagnoses, personalized healthcare, and find solutions to decades old problems. You can use machine learning to programmed computers to make connections and predictions and discover critical insights from large amounts of data that healthcare providers may otherwise miss-all of this can add up to a direct impact on the health of your community.

The goal of machine learning is to improve patient outcomes and produce medical insights that were previously unavailable. It provides a way to validate doctor’s reasoning and decisions through predictive algorithms. For the healthcare industry, machine learning algorithms are particularly valuable because they can help us make sense of the massive amounts of healthcare data that is generated every day within electronic health records. Using machine learning in healthcare, like machine learning algorithms can help us find patterns and insights in medical data that would be impossible to find manually.

1.4 LITERATURE SURVEY

Research conducted by Aishwarya Mujumdara and Dr. Vaidehi V titled “Diabetes Prediction using Machine Learning Algorithms” [5] explores the application of diverse machine learning algorithms on a dataset. Among these algorithms, Logistic Regression achieved the highest accuracy at 96% for classification. Furthermore, employing a pipeline revealed the AdaBoost classifier as the top-performing model, achieving an accuracy of 98.8%.

A research paper on diagnosis of heart disease by data mining techniques by Shra Bahrami and Mirsaeid Hosseini Shirvani (2015) [4] evaluated various classification methods such as, Decision Tree, K-Nearest Neighbors (KNN), SMO (used to train Support Vector Machines). On their dataset, they used feature selection techniques to only select the important attributes and achieved the highest accuracy of 83.732% with Decision Trees.

The main objective of the research conducted by Apoorva V, Yogish H K, and Chayadevi M L, titled “Breast Cancer Prediction using Machine Learning Techniques,” [1] is to develop and implement an innovative computational approach for interpreting and managing breast cancer data derived from mammography and pathology results of patient scans obtained from UCI repositories cloud. The study indicates that the proposed CNN surpasses existing methods in accurately identifying and diagnosing breast cancer within image datasets. Additionally, Support Vector Machine (SVM) outperforms Classification and Regression Trees (CART), Naïve Bayes (NB), and K-Nearest Neighbors (KNN) in the analysis and prediction of cancer using numerical datasets.

A research paper presented by Dr S Seema and Kumari Deepika on Predictive Analysis to Prevent and Control Chronic diseases [3] used two separate datasets from the UCI machine learning repository for the diagnosis of heart disease and diabetes. All the patient’s data are trained by using different classifiers such as Naïve Bayes, SVM, Decision Tree and Artificial Neural Networks. From experiment, it is been found that SVM gives highest accuracy rate of 95.556% in case of heart disease and in case of diabetes Naïve Bayes classifier gives highest accuracy of 73.588%.

In a study conducted by D Shah, Sameer Patel [2] on heart disease prediction using machine learning techniques, the authors aimed to develop a model for predicting cardiovascular disease using machine learning techniques. The data used for this purpose were obtained from the Cleveland heart disease dataset, which consisted of 303 instances and 17 attributes, and were sourced from the UCI machine learning repository. The authors employed a variety of supervised classification methods, including naïve Bayes, decision tree, random forest, and k-nearest neighbor (KNN). The results of the study indicated that the KNN model exhibited the highest level of accuracy, at 90.8%. The study highlights the potential utility of machine learning techniques in predicting cardiovascular disease, and emphasizes the importance of selecting appropriate models and techniques to achieve optimal results.

CHAPTER 2

MACHINE LEARNING ALGORITHMS FOR DISEASE PREDICTION

In recent years, the integration of machine learning algorithms into healthcare has significantly transformed medical diagnosis, treatment, and patient care. Capitalizing on the wealth of healthcare data and the advancements in machine learning techniques, healthcare professionals now have the capability to harness these algorithms for disease prediction, treatment personalization, and streamlining clinical workflows. Below are several key algorithms that exemplify this integration:

2.1 DECISION TREE

A decision tree in machine learning is a versatile, interpretable algorithm used for predictive modelling. It structures decisions based on input data, making it suitable for both classification and regression tasks. A decision tree is a type of supervised learning algorithm that is commonly used in machine learning to model and predict outcomes based on input data. It is a tree-like structure where each internal node tests on attribute, each branch corresponds to attribute value and each leaf node represents the final decision or prediction. The decision tree algorithm falls under the category of supervised learning. They can be used to solve both regression and classification problems.

Decision Tree Terminologies:

There are specialized terms associated with decision trees that denote various components and facets of the tree structure and decision-making procedure:

Root Node: A decision tree’s root node, which represents the original choice or feature from which the tree branches, is the highest node.

Internal Nodes (Decision Nodes): Nodes in the tree whose choices are determined by the values of particular attributes. There are branches on these nodes that go to other nodes.

Leaf Nodes (Terminal Nodes): A leaf node in a decision tree is a terminal node that doesn't split further. It represents the final outcome or decision in the tree's classification or regression process.

Branches (Edges): Links between nodes that show how decisions are made in response to particular circumstances.

Splitting: The process of dividing a node into two or more sub-nodes based on a decision criterion. It involves selecting a feature and a threshold to create subsets of data.

Parent Node: A node that is split into child nodes. The original node from which a split originates.

Child Node: Nodes created as a result of a split from a parent node.

Decision Criterion: The rule or condition used to determine how the data should be split at a decision node. It involves comparing feature values against a threshold.

Pruning: The process of removing branches or nodes from a decision tree to improve its generalization and prevent overfitting.

The decision tree algorithm can be summarized in the following steps:

1. Selecting the Best Feature: Identify the feature that best splits the data into subsets, aiming to maximize information gain (for classification) or minimize impurity (for regression).
2. Splitting the Data: Divide the dataset into subsets based on the chosen feature.
3. Recursive Process: Repeat steps 1 and 2 for each subset until a stopping criterion is met (e.g., maximum depth, minimum samples per leaf).
4. Leaf Node Prediction: Assign a class label (for classification) or a predicted value (for regression) to each leaf node. The resulting tree is used to make predictions by traversing from the root to a leaf node based on the values of features for a given input. The tree structure as in **fig 1** facilitates interpretable decision-making.

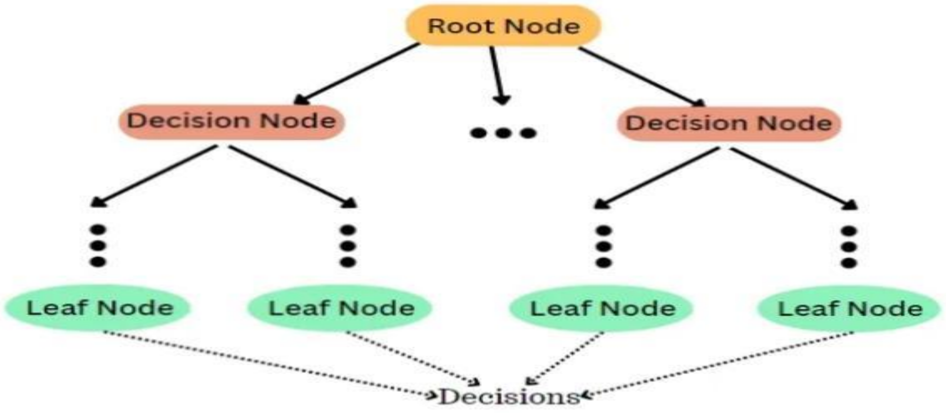


Fig 1

2.2 RANDOM FOREST

Random forest is a commonly-used machine learning algorithm trademarked by Leo Breiman and Adele Cutler, which combines the output of multiple decision trees to reach a single result. Its ease of use and flexibility have fueled its adoption, as it handles both classification and regression problems. Random forest algorithms have three main hyper parameters, which need to be set before training. These include node size, the number of trees, and the number of features sampled. From there, the random forest classifier can be used to solve for regression or classification problems.

The random forest algorithm is made up of a collection of decision trees (**fig 2**), and each tree in the ensemble is comprised of a data sample drawn from a training set with replacement, called the bootstrap sample. Of that training sample, one-third of it is set aside as test data, known as the out-of-bag sample. Another instance of randomness is then injected through feature bagging, adding more diversity to the dataset and reducing the correlation among decision trees. For a regression task, the individual decision trees will be averaged, and for a classification task, a majority vote-i.e. the most frequent categorical variable—will yield the predicted class. Finally, the out of bag sample is then used for cross-validation, finalizing that prediction.

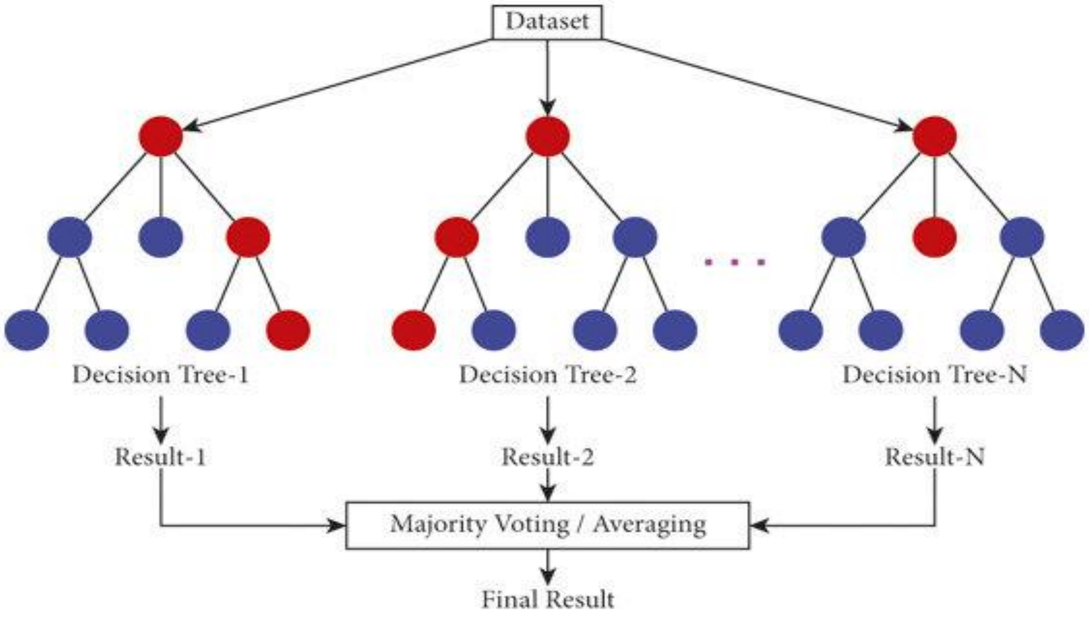


Fig 2

The Working process can be explained in the below steps

Step-1: Select random K data points from the training set.
Step-2: Build the decision trees associated with the selected data points (Subsets).
Step-3: Choose the number N for decision trees that you want to build.
Step-4: Repeat Step 1 & 2.
Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

2.3 SUPPORT VECTOR MACHINE (SVM)

A support vector machine (SVM) is a machine learning algorithm that uses supervised learning models to solve complex classification, regression, and outlier detection problems by performing optimal data transformations that determine boundaries between data points based on predefined classes, labels, or outputs. SVMs are widely adopted across disciplines such as healthcare, natural language processing, signal processing applications, and speech & image recognition fields.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

In the mathematical context, an SVM refers to a set of ML algorithms that use kernel methods to transform data features by employing kernel functions. Kernel functions rely on the process of mapping complex datasets to higher dimensions in a manner that makes data point separation easier. The function simplifies the data boundaries for non-linear problems by adding higher dimensions to map complex data points.

The Fig 3 shows the graph of SVM in accordance with some of the basic terminologies that we went through.

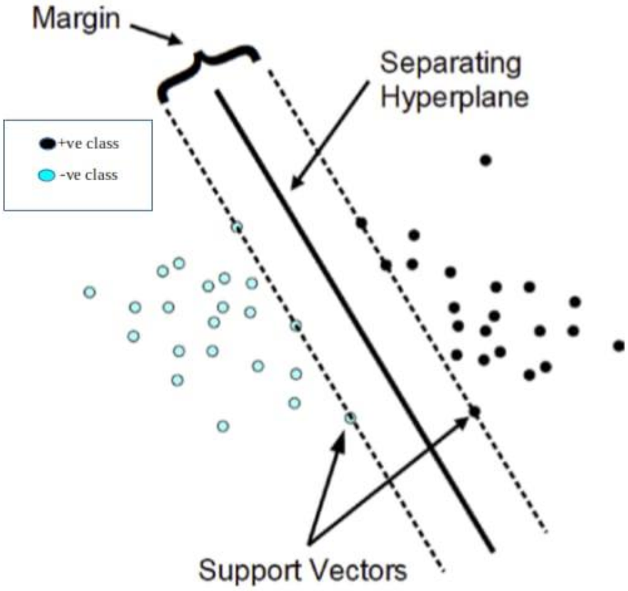


Fig 3

2.4 LOGISTIC REGRESSION

Logistic regression is a supervised machine learning algorithm that accomplishes binary classification tasks by predicting the probability of an outcome, event, or observation. The model delivers a binary or dichotomous outcome limited to two possible outcomes: yes/no, 0/1, or true/false. Logical regression analyzes the relationship between one or more independent variables and classifies data into discrete classes. It is extensively used in predictive modeling, where the model estimates the mathematical probability of whether an instance belongs to a specific category or not. For example, 0 – represents a negative class; 1 – represents a positive class. Logistic regression is commonly used in binary classification problems where the outcome variable reveals either of the two categories (0 and 1).

In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1). The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc. Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets. Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification.

Logistic Function (Sigmoid Function):

The sigmoid function is a mathematical function used to map the predicted values to probabilities. It maps any real value into another value within a range of 0 and 1. The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the “S” form. The S-form curve is called the Sigmoid function or the logistic function. In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

2.5 K – MEANS CLUSTERING

K-means is an unsupervised learning algorithm which is used to classify the given dataset that is unlabeled. The goal of this algorithm is to find similar groups represented by variable k. Here k is the number of clusters, so k centroids are defined one for each cluster. Now, the Euclidean distance is calculated from each data point to the centroid, assignment of data points to the centroid depends upon the minimum Euclidean from that centroid. When no point is left un-assigned, an early grouping is done. Now, k new centroids are re-calculated, as a result iteration continues until the k centroids stop changing their position.

The Working process can be explained in the below steps

Let Y = { x_1, x_2, x_3, …, x_n } be the set of data points and Z = { z_1, z_2, …, z_c } be the set of centers.

1. Randomly select ‘c’ cluster centres.
2. Calculate the distance between each data point and cluster centers.
3. Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers.
4. Recalculate the new cluster center using: Z_i=(1/c_i)
5. Where, ‘C_i' represents the number of data points in i^{th} cluster.
6. Recalculate the distance between each data point and new obtained cluster centers.

7. If no data point was reassigned then stop, otherwise repeat from step 3.

2.6. PRINCIPAL COMPONENT ANALYSIS

Principal Component Analysis (PCA) technique was introduced by the mathematician Karl Pearson in 1901. It works on the condition that while the data in a higher dimensional space is mapped to data in a lower dimension space, the variance of the data in the lower dimensional space should be maximum.

Principal Component Analysis (PCA) is an unsupervised learning algorithm technique used to examine the interrelations among a set of variables. It is also known as a general factor analysis where regression determines a line of best fit. The main goal of Principal Component Analysis (PCA) is to reduce the dimensionality of a dataset while preserving the most important patterns or relationships between the variables without any prior knowledge of the target variables. Principal Component Analysis (PCA) is used to reduce the dimensionality of a data set by finding a new set of variables, smaller than the original set of variables, retaining most of the sample's information, and useful for the regression and classification of data.

Steps for PCA algorithm:

1.**Getting the dataset:** Firstly, we need to take the input dataset and divide it into two subparts X and Y, where X is the training set, and Y is the validation set.

2.**Representing data into a structure**: Now we will represent our dataset into a structure. Such as we will represent the two-dimensional matrix of independent variable X. Here each row corresponds to the data items, and the column corresponds to the Features. The number of columns is the dimensions of the dataset.

3.**Standardizing the data:** In this step, we will standardize our dataset. Such as in a particular column, the features with high variance are more important compared to the features with lower variance. If the importance of features is independent of the variance of the feature, then we will divide each data item in a column with the standard deviation of the column. Here we will name the matrix as Z.

4.**Calculating the Covariance of Z:** we will take the matrix Z, and will transpose it. After transpose, we will multiply it by Z. The output matrix will be the Covariance matrix of Z.

5.**Calculating the Eigen Values and Eigen Vectors:** Eigenvectors or the covariance matrix are the directions of the axes with high information. And the coefficients of these eigenvectors are defined as the eigenvalues.

6.**Sorting the Eigen Vectors:** In this step, we will take all the eigenvalues and will sort them in decreasing order, which means from largest to smallest. And simultaneously sort the eigenvectors accordingly in matrix P of eigenvalues. The resultant matrix will be named as P*.

7.**Calculating the new features Or Principal Components:** Here we will calculate the new features. To do this, we will multiply the P* matrix to the Z. In the resultant matrix Z*, each observation is the linear combination of original features. Each column of the Z* matrix is independent of each other.

8.Remove less or unimportant features from the new dataset.

2.7 ISOLATION TREE ANOMALY DETECTION

An Isolation Tree or Isolation Forest, is an unsupervised machine learning algorithm designed for anomaly detection. It works by isolating instances (anomalies) in a dataset by constructing a set of binary trees. Isolation Forests (IF), similar to Random Forests, are build based on decision trees. And since there are no pre-defined labels here, it is an unsupervised model. In an Isolation Forest, randomly sub-sampled data is processed in a tree structure based on randomly selected features. The samples that travel deeper into the tree are less likely to be anomalies as they required more cuts to isolate them. Similarly, the samples which end up in shorter branches indicate anomalies as it was easier for the tree to separate them from other observations.

Step for Isolation tree anomaly detection algorithm:

1.When given a dataset, a random sub-sample of the data is selected and assigned to a binary tree.

2.Branching of the tree starts by selecting a random feature first. And then branching is done on a random threshold.

3.If the value of a data point is less than the selected threshold, it goes to the left branch else to the right. And thus, a node is split into left and right branches.

4.This process from step 2 is continued recursively till each data point is completely isolated or till max depth (if defined) is reached.

5.The above steps are repeated to construct random binary trees.

6.After an ensemble of i trees (Isolation Forest) is created, model training is complete. During scoring, a data point is traversed through all the trees which were trained earlier. Now, an ‘anomaly score’ is assigned to each of the data points based on the depth of the tree required to arrive at that point. This score is an aggregation of the depth obtained from each of the i trees. An anomaly score of -1 is assigned to anomalies and 1 to normal points based on the contamination (percentage of anomalies present in the data) parameter provided.

2.8 HIERARCHICAL CLUSTERING

Hierarchical clustering is another unsupervised machine learning algorithm, which is used to group the unlabeled datasets into a cluster and also known as hierarchical cluster analysis or HCA. In this algorithm, we develop the hierarchy of clusters in the form of a tree, and this tree-shaped structure is known as the dendrogram.

Two approaches of hierarchical clustering

Agglomerative: Agglomerative is a bottom-up approach, in which the algorithm starts with taking all data points as single clusters and merging them until one cluster is left.

Divisive: Divisive algorithm is the reverse of the agglomerative algorithm as it is a top-down approach.

Steps of working of hierarchical clustering:

- **Start with individual data points**: - Treat each data point as a single cluster.
- **Calculate distances**: - Compute the pairwise distances between all clusters. Common distance measures include Euclidean distance or Manhattan distance.
- **Merge closest clusters**: - Combine the two clusters with the smallest distance. This creates a new cluster.
- **Update distance matrix**: - Recalculate distances between the new cluster and the remaining clusters.

- **Repeat**: - Steps 3 and 4 are repeated until all data points are in a single cluster or until a predefined number of clusters is reached
- **Dendrogram**: - Visualize the clustering process with a dendrogram. It's a tree-like diagram where the leaves represent individual data points, and the branches represent the merging of clusters.
- **Choose the no. of clusters**: - Determine the optimal number of clusters by interpreting the dendrogram or using criteria like the "elbow method" on the resulting clusters.
- **Assign clusters**: - Based on the chosen number of clusters, cut the dendrogram to form distinct clusters and assign each data point to its final cluster.

CHAPTER 3

CHALLENGES OF USING MACHINE LEARNING IN DISEASE PREDICTION

3.1. CHALLENGES

Despite the promise of ML research in the field of precision medicine, many challenges still need to be addressed to ensure the further development and acceptance of ML approaches. Some of the challenges are:

Data Quality

Being a data-driven approach, the performance of the ML model depends heavily on the quality of the data that it builds on. Data Needs to have a sufficient sample size and quality in order to represent the target population in the clinical application. In General, a larger sample size is essential for the development of a more robust ML model, which allows accurate prediction for supporting clinical decisions. ML models trained by small sample sizes often suffer from the problem of "overfitting", where the model over relies on characteristics from the under-represented training data and loses the ability to effectively perform in practice. Similar to the multiple testing issue in conventional statistics, ML models with small sample size might cause false significant discoveries due to random variation under numerous repetitions. For example, one can generate 1,000 different splits of train/test data and evaluate performance. If the performance based on splits shows a great variance, this might indicate an "unstable" model. One way to improve model reliability due to small sample size is by reducing the model variance, as low variance algorithms are less influenced by the specificity of the training data. However, model variance reduction often results in an increase in model biased error, leading to a weakened predictive performance of models. Meanwhile, obtaining a larger sample size often requires more resources (time, funding, access to large patient populations and computer power etc.). One way to ensure the appropriateness of study design for the research outcome investigated is by having universal guidance of the adequate sample size required for the ML model training for researchers to follow.

Studies have already attempted to develop tools to assist decision making in study design. For example, an r package “pmsampsize” was developed to calculate the minimum sample size for the predictive model development to avoid model overfitting, taking into account the number of participants, outcome events and predictive variables. However, the use of a limited sample size can be sometimes inevitable due to the rare nature of certain diseases. To overcome the limitation of small sample size, more comprehensive procedures and careful considerations are necessary for generating reliable results. One example is juvenile-onset SLE(JSLE) – a rare ARD. In one study, researchers applied a ML model to stratify JSLE patients based on their immune profile. Only 67 JSLE patients and 39 healthy controls with 28 immune cell predictors were included in the analysis. A random forest algorithm was selected as it was less likely to overfit the data due to an implanted bagging method and random feature selection in the model ensembled by a large number of decision trees. The results of this model were combined with additional analysis such as the sparse-PLS-DA and univariate logistic regression and were further validated by 10-fold cross-validation. Although the lack of an external validation dataset meant there was still risks for overfitting and not being able to extrapolate the results, the study shows the potential for applying a ML-based pipeline to other rare and heterogeneous immune-mediated inflammatory conditions. Another challenge in the development of ML models is access to high quality and well-defined datasets, needed for algorithm training and evaluation. In recent years there has been a big push to make research data FAIR (Findable, Accessible, Interoperable and Reusable) Datasets generated in research studies should collect enough machine-readable metadata to allow for discovery and searches. Ideally, clear rules for data access and use should be available, as well as use of domain-specific ontologies to describe the data. There should also be enough information available describing how the acquisition of data was carried out, enabling re-use of data

Model Implementation in Clinical Practice

Transforming a well-performed model into an actual clinical application associated with improvement in patient outcomes can be challenging; the term “AI Chasm” describes the discrepancy between the model development and translation of models to real-world applications. The clinical impact of potentially promising ML models requires careful evaluation before considering implementation in clinical settings. For example, a wide range of performance metrics (accuracy, AUC, precision, sensitivity, specificity etc.) are applied to represent the predictive efficacy of ML models in clinical studies. However, most of the metrics do not directly affirm the clinical applicability and can be difficult to evaluate with limited interdisciplinary knowledge. Another common obstacle for the clinical translatability of ML data arrives where emerging ML studies that stratify patients with novel signatures suffer from the lack of effective drugs for the newly identified targets. Furthermore, the reported predictive model needs to provide clinically meaningful advantages over traditional approaches, such as significantly outperform the existing standard statistical approach in relevant fields. To help address these questions, standard practice guidance is necessary. Reporting of a multivariable prediction model for Individual Prognosis or Diagnosis (TRIPOD) guideline is an internationally accepted reporting guideline developed to improve the reliability and value of prediction models for diagnostic or prognostic purposes. TRIPOD-ML focuses on the standardized methodology of ML model development, which together with the interdisciplinary effort from trained experts in different clinical and technology areas of expertise, can ensure that ML applications maximize their chance to translate into precision medicine approaches associated with patient benefit.

Ethical Concerns

The upsurge of ML applications in personalized medicine has raised potential ethical concern regarding data privacy, as a wide range of big datasets including personal information from genetics data, femographic data and medication history are stored and used in various studies. Anonymization is the most straightforward and common way for privacy protection of medical datasets by removing personal data for re-identification purposes. However, advanced re-identification techniques were developed and used to target the vulnerability of the anonymization system by data mining companies, and data were then exploited by health insurance companies. Thus, more rigorous data handling methods such as data decentralization (storing data in separate locations) and federated machine learning (training algorithm across different decentralized local data) are necessary for institutes and companies dealing with large-scale personal data. From patients and the general public perspective, there is an innate Scepticism related to the use of AI for clinical applications, especially with limited understanding about how ML and personal data are used in medical research. Face-to-face communication between specialists and patients is effective in conveying the scope of ML applications and addressing questions and concerns in terms of patient satisfaction. Public education events such as interactive Patient and Public Involvement and Engagement (PPIE) activities can inform patients about how AI and ML research can lead to better disease management and how data are handled within a secured framework. With a better understanding of ML approaches and how personal data are stored, used and protected, patients are more likely to engage with such research. The phenomenon of ML algorithm-driven discriminating decisions has been well-observed in other areas of research using AI, such as racial discrimination in criminal charge, facial recognition technology and gender discrimination in job recruitment algorithms. Algorithm discrimination is not exempt in the clinical world. For example, an implemented algorithm in the US healthcare system for future health care needs prediction is heavily biased against black patients because of the lack of data on these patients. This algorithm-intrinsic bias is inherited from existing inequality in society as black patients are generally less accessible to the healthcare system. Another study showed that the predicted hospital mortality of patients in critical care can vary by up to 20% according to their ethnic group. Many inflammatory diseases are independently associated with demographic variables such as age, sex and ethnicity. For example, autoimmune diseases are more frequent in the female population, which sometimes, for practical reasons, promotes research only within the most represented groups of patients, discriminating against the under-represented ones. Moreover, model development is highly data-driven with low tolerance to missing values in model training, which can also lead to potential bias by not capturing the real-life patient population of interest. For example, previous studies showed that vulnerable populations are less likely to attend the same clinic regularly due to limited access to healthcare, including diagnostic testing and medicine. Unintentionally excluding these incomplete datasets will lead to development of models that are less effective in populations with existing disadvantages. Thus, it is important for researchers and data scientists representing the diversity of the human condition to have opportunities to participate in the decision making and algorithm supervision process, assessment of the underlying biases associated with AI and ML and implementation of regulatory adjustments. This will avoid the development of discriminating decision-aiding algorithms.

The Future of Personalized Medicine

With such challenges evident at every possible step during the application of ML approaches, ambition of personalized medicine to ensure that every individual receives an optimal treatment decision guided by their disease particularities and individual risk becomes uncertain. To warrant a future for ML applications in the clinical field, it is crucial to have universal procedure guidelines from data collection, data processing to model training, validation, and implementation. By ensuring the standardization of ML applications, research study design can be optimized to facilitate granular and relevant data collection, as well as the use of an adequate sample size in relation to data multidimensionality to minimize the risk of significant data redundancy which can hamper the relevant patient identification. In addition, identification of reproducible biomarkers associated with response to therapy is one of the key requirements for personalized medicine approaches and we advocate for the use of truly independent data sets for validation. Although in theory, personalized medicine could be advanced by the use of ML algorithms for individual disease risk identification and prognostic, as well as therapy selection, its implementation in large health systems poses the ethical challenges of reconciling health risk inequalities with finite health care resources and standardized taxpayer or health insurance contributions. Future research should provide answers regarding the advantages of ML-driven personalized medicine strategies for long-term outcomes of patients in real-life.

3.2. HOW TO OVERCOME THE CHALLENGES

Overcoming challenges in using machine learning for disease prediction involves employing various strategies to address data limitations, ensure privacy, enhance model interpretability, and adapt to dynamic healthcare environments.

Data Quality and Quantity:

- Collaborate with healthcare institutions to access larger and more diverse datasets.
- Implement data augmentation techniques to artificially increase the size of the dataset.

- Utilize transfer learning approaches to leverage pre-trained models on related tasks.

Privacy Concerns:

- Implement robust data anonymization and de-identification techniques to protect patient privacy.
- Use federated learning, where models are trained on decentralized data, keeping sensitive information on local servers.

Model Interpretability:

- Focus on using interpretable models, such as decision trees or rule-based models, especially in critical healthcare applications.
- Employ model-agnostic interpretability techniques, such as LIME (Local Interpretable Model-Agnostic Explanations), to explain predictions.

Multifactorial Disease Dynamics:

- Develop ensemble models that combine the strengths of different algorithms to capture diverse aspects of disease complexity.
- Incorporate domain knowledge and collaborate with healthcare experts to identify relevant features and relationships.

Continuous Model Adaptation:

- Implement a system for regular model updates to stay current with evolving healthcare standards and knowledge.
- Utilize transfer learning or online learning techniques to adapt models to changes in disease dynamics.

Interdisciplinary Collaboration:

- Foster collaboration between data scientists, clinicians, and domain experts to ensure a comprehensive understanding of both data and medical context.
- Encourage joint research initiatives to bridge the gap between machine learning and healthcare expertise.

Ethical Considerations:

- Adhere to ethical guidelines and regulatory frameworks governing the use of healthcare data.
- Foster transparency and open communication about the ethical considerations associated with Machine learning applications in healthcare.

CHAPTER 4

CASE STUDY

The application of machine learning algorithms in predicting heart disease has emerged as a pivotal area in healthcare, offering the potential to enhance diagnostic accuracy and improve patient outcomes. This case study focuses on the prediction of heart disease using machine learning algorithms, utilizing a dataset sourced from kaggle. The dataset has 303 instances and 14 attributes. The datasets description is given below:

1. Age: age of the patient [in years]
2. Sex: sex of the patient [1: Male, 0: Female]
3. cp: chest pain type [0: Typical Angina, 1: Atypical Angina, 2: Non-Anginal Pain, 3: Asymptomatic]
4. Trestbps: resting blood pressure [mm Hg on admission to the hospital]
5. Chol: serum cholesterol [in mg/dl]
6. fbs: fasting blood sugar [1: if FastingBS > 120 mg/dl, 0: otherwise]
7. Restecg: resting electrocardiogram results
8. Thalach : maximum heart rate achieved
9. Exang: exercise-induced angina [1=yes; 0=no]
10. Oldpeak: oldpeak = ST depression induced by exercise relative to rest
11. Slope: the slope of the peak exercise ST segment [Up: 0, Flat: 1, Down:2]
12. Ca: number of major vessels (0-3) colored by fiourusopy
13. Thal: 3 normal,6 fixed defect,7 reversable defect
14. target: have disease or not [1: yes, 0: no]

4.1 HEART DISEASE PREDICTION MODEL USING RANDOM FOREST ALGORITHM

Random Forest is a popular choice for heart disease prediction due to its several advantages:

• Accuracy: It often performs well in predictive tasks, offering high accuracy by reducing overfitting and handling noise in the data.

• Feature Importance: Random Forest provides a measure of feature importance, helping identify key factors contributing to heart disease prediction.

• Robustness: It is less prone to overfitting compared to individual decision trees, making it robust against noise in the data and variations in the dataset.

• Handles Missing Values: Random Forest can handle missing values in the dataset, which is common in medical data.

• Non-linear Relationships: It can capture non-linear relationships between variables, allowing for a more comprehensive analysis of complex interactions in heart disease prediction.

• Reduced Variance: By aggregating predictions from multiple trees, Random Forest reduces variance and enhances generalization performance.

Python Coding Procedures:

Data Loading: The code starts by loading a dataset named 'heart (1).csv' into a pandas DataFrame (df).

Data Splitting: The dataset is split into features (x) and the target variable (y). It then further splits the data into training and testing sets using the train_test_split function from scikit-learn.

Model Training: A Random Forest Classifier is initialized with 100 trees and then trained on the training data (x_train and y_train) using the fit method.

Model Prediction: The trained model is used to make predictions on the test data (x_test), and the predicted values are stored in y_pred.

Training Data: Features (x_train): A subset of the original dataset, used for training the model.

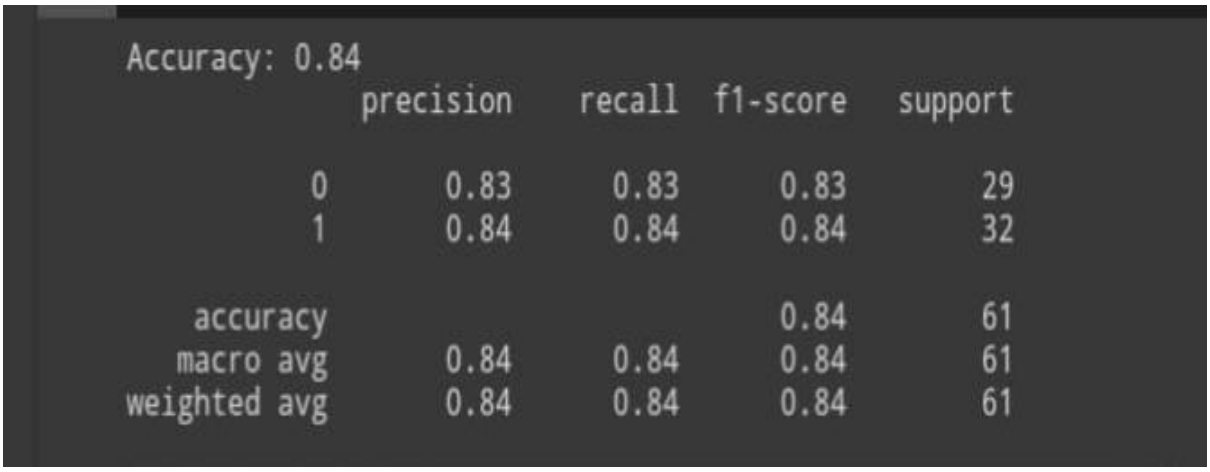
Target variable (y_train): Corresponding labels indicating the presence or absence of heart disease.

Test Data: Features (x_test): Another subset of the dataset, reserved for evaluating the model's performance.

Target variable (y_test): Actual labels for the test set.

Result: The overall accuracy of the model on the test set is 84%

Classification Report: Additional metrics (precision, recall, F1-score, and support) for each class are printed, providing a more detailed evaluation.

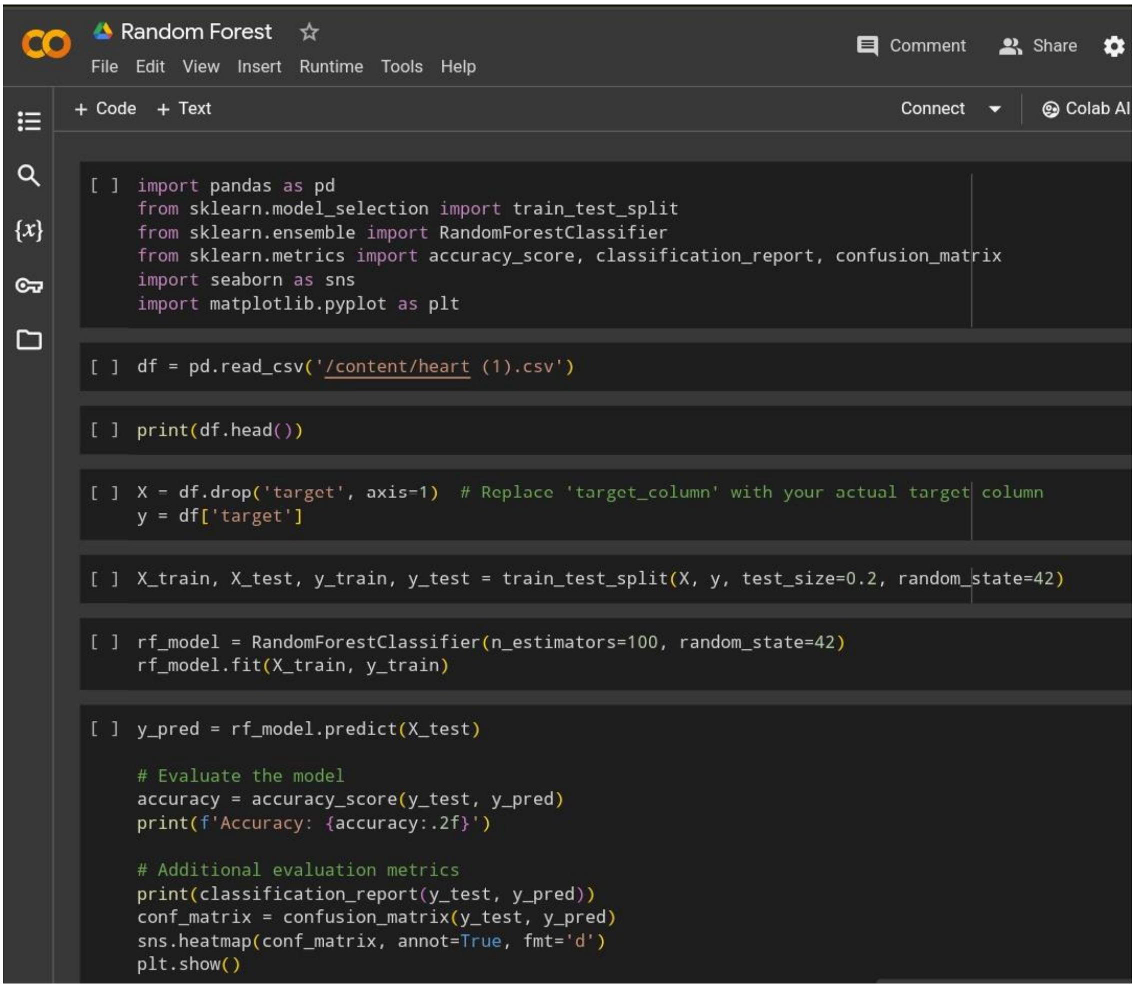


```
Accuracy: 0.84
              precision    recall  f1-score   support

           0       0.83      0.83      0.83        29
           1       0.84      0.84      0.84        32

    accuracy                           0.84        61
   macro avg       0.84      0.84      0.84        61
weighted avg       0.84      0.84      0.84        61
```

Fig 4



```
[ ] import pandas as pd
    from sklearn.model_selection import train_test_split
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
    import seaborn as sns
    import matplotlib.pyplot as plt

[ ] df = pd.read_csv('/content/heart (1).csv')

[ ] print(df.head())

[ ] X = df.drop('target', axis=1)  # Replace 'target_column' with your actual target column
    y = df['target']

[ ] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

[ ] rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
    rf_model.fit(X_train, y_train)

[ ] y_pred = rf_model.predict(X_test)

    # Evaluate the model
    accuracy = accuracy_score(y_test, y_pred)
    print(f'Accuracy: {accuracy:.2f}')

    # Additional evaluation metrics
    print(classification_report(y_test, y_pred))
    conf_matrix = confusion_matrix(y_test, y_pred)
    sns.heatmap(conf_matrix, annot=True, fmt='d')
    plt.show()
```

Fig 5

This code uses a Random Forest Classifier to build a predictive model for heart disease based on a dataset. It then evaluates the model's performance using various metrics and visualizations.

Confusion Matrix

A heatmap visualization of the confusion matrix is shown, offering insights into true positives, true negatives, false positives, and false negatives.

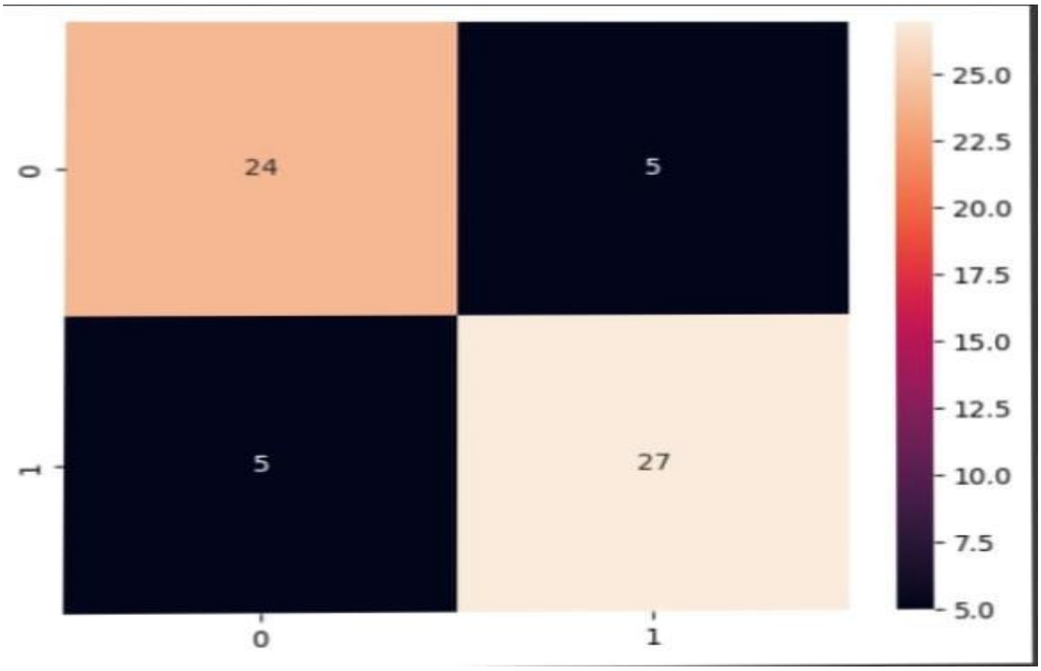


Fig 6

4.2 HEART DISEASE PREDICTION MODEL USING LOGISTIC REGRESSION ALGORITHM

Logistic regression is a suitable choice for heart disease prediction models due to several reasons:

• Binary Classification: Logistic regression is well-suited for binary classification problems, making it appropriate for predicting whether a patient is at risk of heart disease or not.

• Interpretability: Logistic regression provides interpretable results, allowing you to understand the impact of each predictor variable on the likelihood of heart disease. This transparency is crucial in a medical context for decision-making and trust.

• Efficiency: Logistic regression is computationally efficient, especially when dealing with large datasets. This makes it practical for analyzing medical data, which often involves a substantial amount of information.

• Assumption of Linearity: While logistic regression assumes a linear relationship between the log odds of the outcome and predictor variables, it can be effective in capturing complex relationships when used in conjunction with feature engineering or polynomial terms.

• Scalability: Logistic regression is scalable to handle a moderate number of predictors, making it feasible for modeling with a reasonable number of relevant features in heart disease prediction.

Python Coding Procedures :

Loading the Dataset: The code uses pandas to load a dataset from a CSV file.

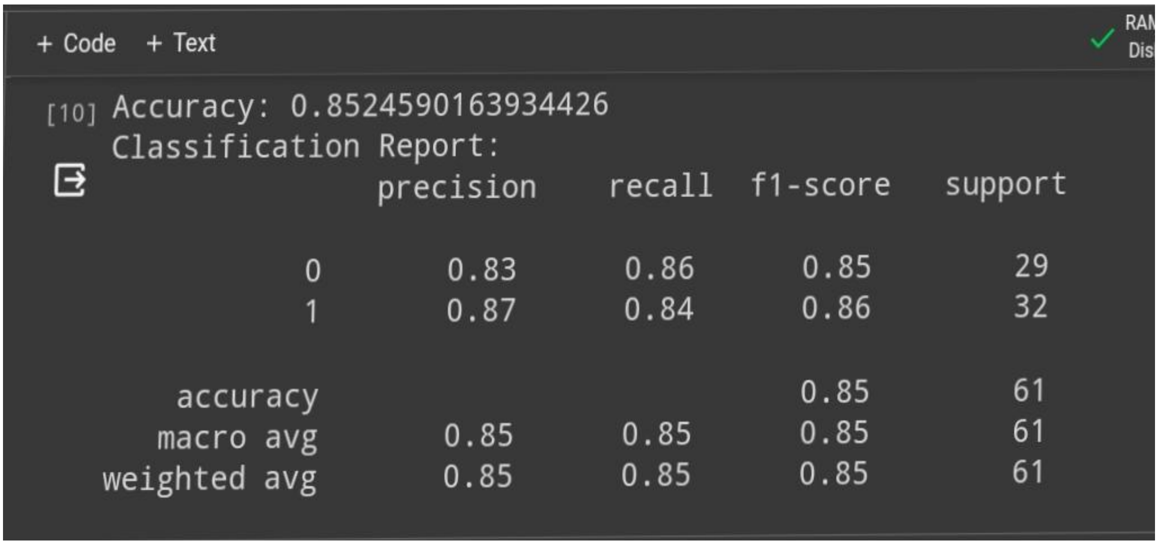
Splitting the Data: It splits the dataset into training and testing sets using `train_test_split` from scikit-learn. The test set size is set to 20%, and a random seed (42) is provided for reproducibility.

Standardizing the Features: It uses `StandardScaler` to standardize the features. Standardization ensures that all features have the same scale, which is important for some machine learning algorithms, including logistic regression.

Creating and Training the Logistic Regression Model: A logistic regression model is created using `LogisticRegression()` from scikit-learn. The model is then trained on the standardized training data using `fit`.

Making Predictions and Evaluating the Model: The model is used to make predictions on the standardized test set. Accuracy is calculated using `accuracy_score`. A detailed classification report is generated using `classification_report`, including precision, recall, and F1-score for each class.

Results: The final step prints the accuracy and the classification report to the console.

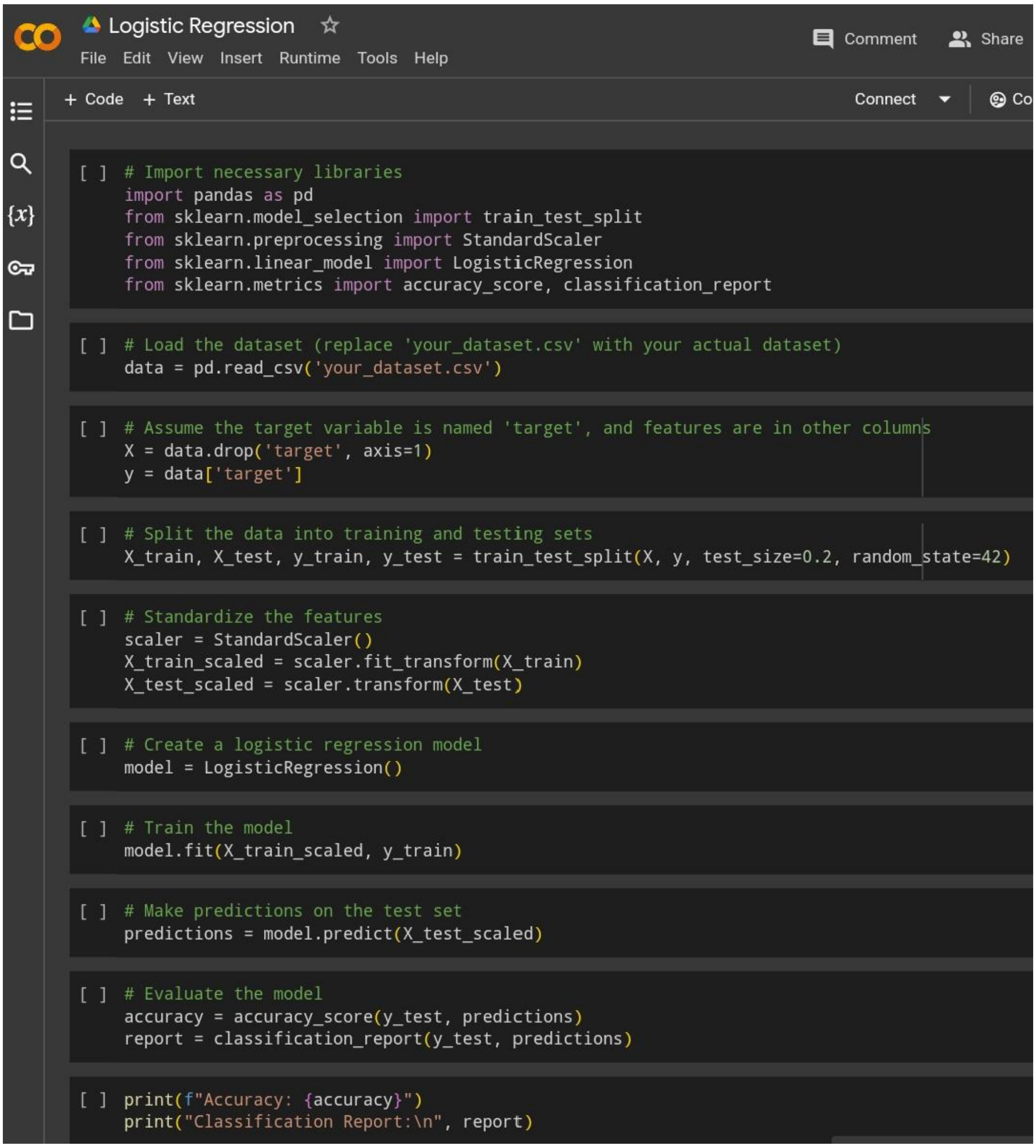


```
Accuracy: 0.8524590163934426
Classification Report:
              precision    recall  f1-score   support

           0       0.83      0.86      0.85        29
           1       0.87      0.84      0.86        32

    accuracy                           0.85        61
   macro avg       0.85      0.85      0.85        61
weighted avg       0.85      0.85      0.85        61
```

Fig 7



```
[ ] # Import necessary libraries
    import pandas as pd
    from sklearn.model_selection import train_test_split
    from sklearn.preprocessing import StandardScaler
    from sklearn.linear_model import LogisticRegression
    from sklearn.metrics import accuracy_score, classification_report

[ ] # Load the dataset (replace 'your_dataset.csv' with your actual dataset)
    data = pd.read_csv('your_dataset.csv')

[ ] # Assume the target variable is named 'target', and features are in other columns
    X = data.drop('target', axis=1)
    y = data['target']

[ ] # Split the data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

[ ] # Standardize the features
    scaler = StandardScaler()
    X_train_scaled = scaler.fit_transform(X_train)
    X_test_scaled = scaler.transform(X_test)

[ ] # Create a logistic regression model
    model = LogisticRegression()

[ ] # Train the model
    model.fit(X_train_scaled, y_train)

[ ] # Make predictions on the test set
    predictions = model.predict(X_test_scaled)

[ ] # Evaluate the model
    accuracy = accuracy_score(y_test, predictions)
    report = classification_report(y_test, predictions)

[ ] print(f"Accuracy: {accuracy}")
    print("Classification Report:\n", report)
```

Fig 8

This code uses a Logistic Regression to build a predictive model for heart disease based on a dataset. It then evaluates the model’s performance using various metrics and visualizations.

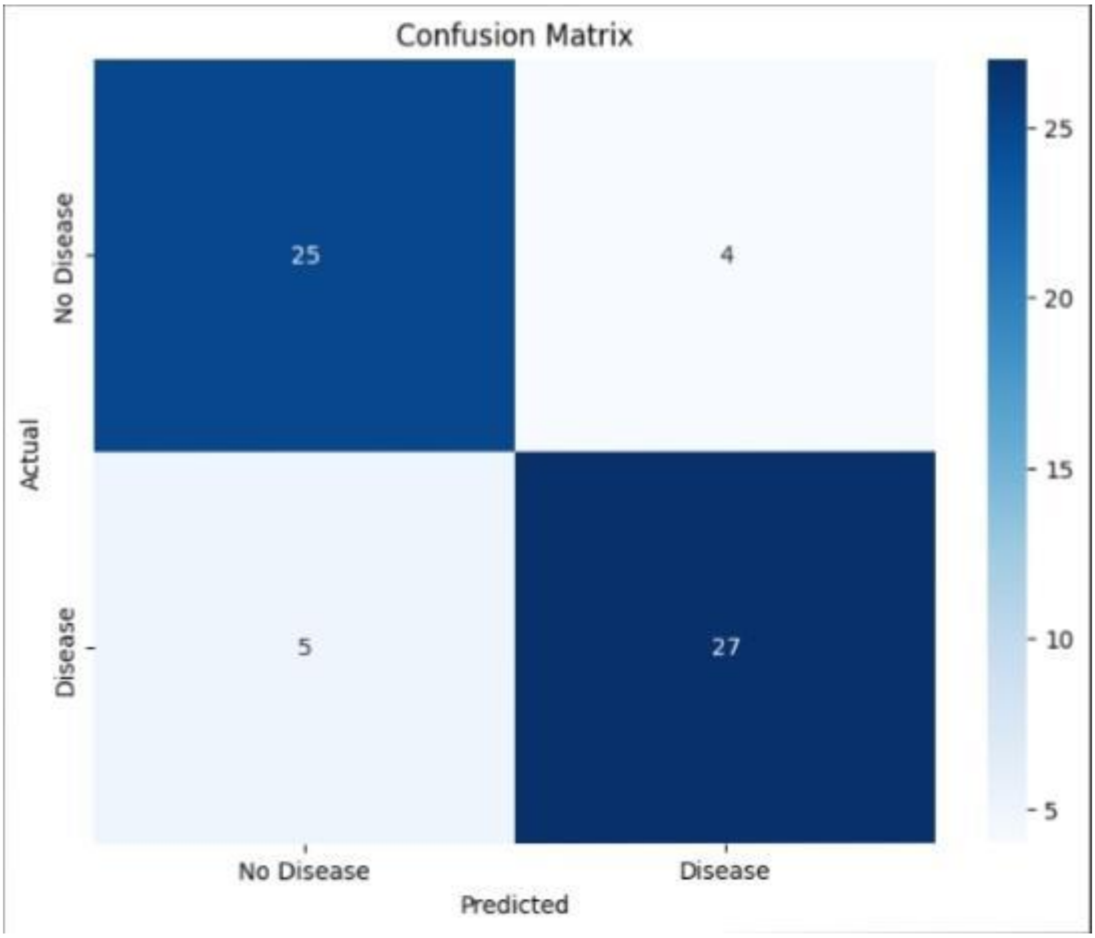


Fig 9

4.3 HEART DISEASE PREDICTION USING SUPPORT VECTOR MACHINE

SVM model is a popular choice to heart disease prediction due to,

- Non-Linearity: SVMs can effectively model non-linear relationships in data, which is valuable in medical dataset where the relationship between features and the presence of heart disease may not be linear.

- High-Dimensional Data: SVMs perform well in high-dimensional spaces, making them suitable for datasets with numerous features, common in medical applications where various health metrics are considered.

- Feature Interactions: SVMs can capture complex interactions between features, which is essential in understanding how different factors may jointly contribute to heart disease prediction.

• Robust to Overfitting: SVMs are less prone to overfitting compared to some other algorithms, crucial in medical contexts where ensuring generalization to new cases is essential.

• Versatility: SVMs can be adapted to different kernel functions, allowing flexibility in capturing different types of patterns in the data, whether linear or non-linear.

• Binary Classification: SVMs are inherently binary classifiers, making them suitable for tasks like predicting the presence or absence of heart disease.

Python Coding Procedures :

Loading Data: The dataset is loaded from a CSV file named ‘your_dataset.csv’.

Data Preparation: Features (x) are obtained by excluding the ‘target’ column. The target variable (y) is defined as the ‘target' column.

Train-Test Split: The data is split into training and testing sets using the train_test_split function. 80% of the data is used for training, and 20% is reserved for testing.

Feature Standardization: Features are standardized using StandardScaler. This ensures that each feature has a mean of 0 and a standard deviation of 1, improving the performance of some machine learning algorithms, including SVM.

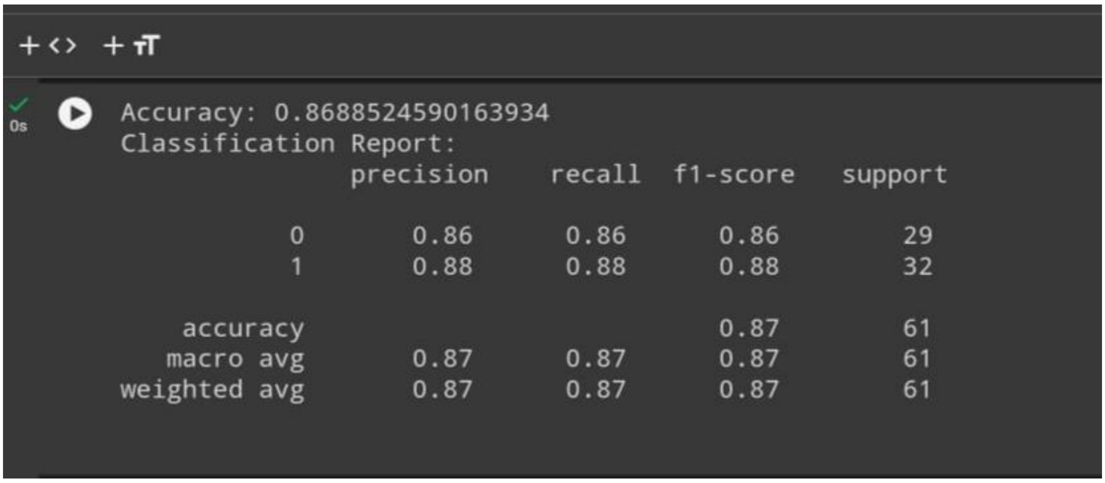
SVM Model Creation: A Support Vector Machine (SVM) model with a linear kernel is created and trained on the standardized training data.

Prediction: The trained SVM model is used to make predictions on the standardized test set.

Training Data: The training data (x_train, y_train) is used to train the SVM model. The features (x) are standardized, and the SVM model is fitted to this data.

Test Data: The test data (x_test, y_test) is used to assess the generalization performance of the trained SVM model. Features are standardized using the same scaling parameters obtained from the training set.

Result: The accuracy of the model on the test set is printed, providing a measure of overall correctness. The classification report gives detailed metrics such as precision, recall, and F1-score for each class (assumed to be binary in this case).

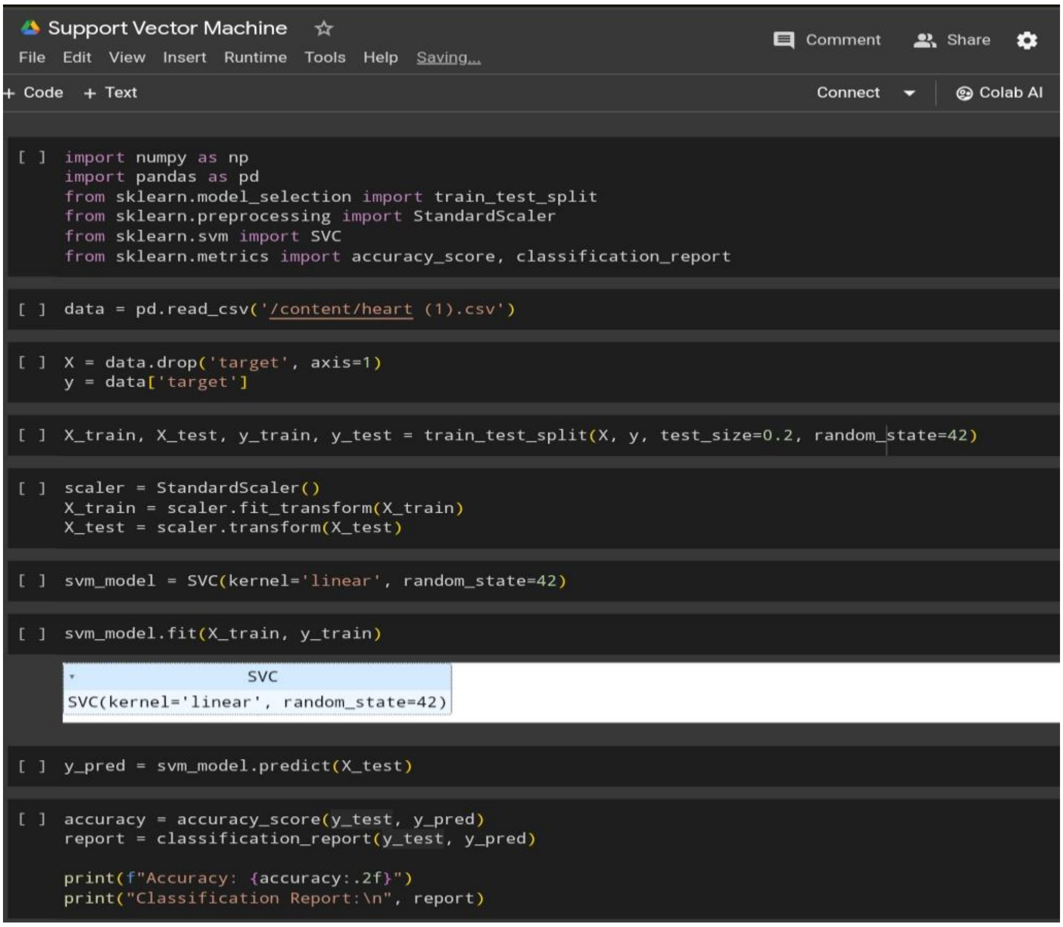


```
Accuracy: 0.8688524590163934
Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.86      0.86        29
           1       0.88      0.88      0.88        32

    accuracy                           0.87        61
   macro avg       0.87      0.87      0.87        61
weighted avg       0.87      0.87      0.87        61
```

Fig 10



```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report
```

```
data = pd.read_csv('/content/heart (1).csv')
```

```
X = data.drop('target', axis=1)
y = data['target']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

```
svm_model = SVC(kernel='linear', random_state=42)
```

```
svm_model.fit(X_train, y_train)
```

```
SVC
SVC(kernel='linear', random_state=42)
```

```
y_pred = svm_model.predict(X_test)
```

```
accuracy = accuracy_score(y_test, y_pred)
report = classification_report(y_test, y_pred)

print(f"Accuracy: {accuracy:.2f}")
print("Classification Report:\n", report)
```

Fig11

This code demonstrates the usage of Support Vector Machine (SVM), a popular machine learning algorithm, for a binary classification task.

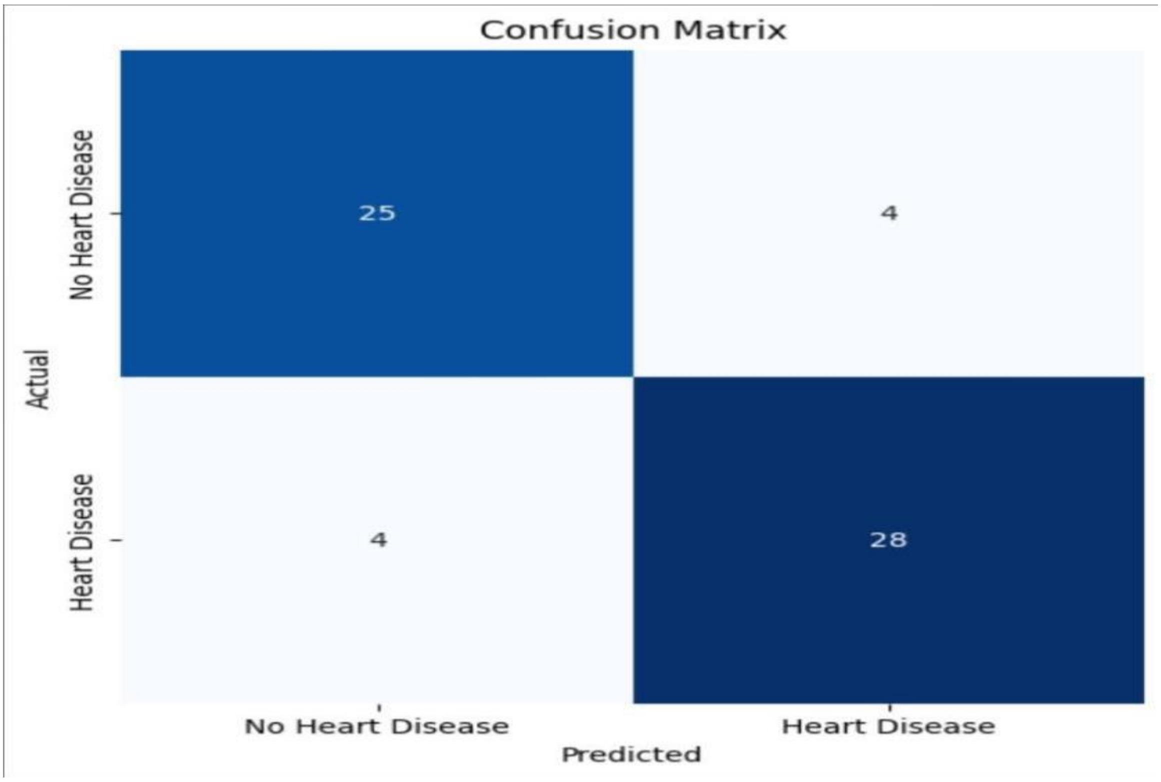

Confusion Matrix
Actual
No Heart Disease
Heart Disease
25
4
4
28
No Heart Disease
Heart Disease
Predicted

Fig 12

CHAPTER 5

CONCLUSION

In conclusion, the implementation of machine learning algorithms for the heart disease prediction model has shown promising results. The model's ability to analyze diverse data sets and make accurate predictions contributes significantly to early detection and prevention. Further refinement and validation could enhance its reliability, making it a valuable tool in the realm of cardiovascular health.

With the help of the Random Forest Classifier, Logistic regression algorithm and support vector machine, we were able to build a machine-learning models. Our model was trained and tested by a dataset from the Kaggle. The dataset consisted of labeled 303 patients, it included both diagnosed heart disease patients and normal patients. After the model was trained and then tested, we achieved an accuracy of 84% in Random Forest, 85.2% in Logistic regression, 87% in support vector machine.

We can conclude that machine learning can play an important role in our healthcare system. Traditionally, diagnosis of the disease was performed by standard procedures and doctor's intuitions which had limitations and led to costly expenses, but with machine learning models, diagnosis can be done on large datasets cost-effectively.

BIBLIOGRAPHY

[1] Apoorva V., Yogish H., Chayadevi M., “Breast Cancer Prediction Using Machine Learning Techniques”. Proceedings of the 3rd International Conference on Integrated Intelligent Computing Communication & Security (ICIIC), 2021

[2] Devansh S., Sameer P., Santhosh. K., “Heart Disease prediction using machine learning techniques”, *SN Computer Science*, vol. 1, article no. 345, pp. 1-6, 2020.

[3] Kumari D., Seema, “Predictive analytics to prevent and control chronic diseases”, *2nd international conference on applied and theoretical computing and communication technology (iCATccT)*, pp. 381-386, 2016.

[4] Shra B., Mirsaeid H., “Prediction and Diagnosis of Heart Disease by Data Mining Techniques”, *Journal of Multidisciplinary Engineering Science and Technology (JMEST)*, ISSN:3159-0040, Vol. 2, Issue 2, pp. 164-168, 2015.

[5] Vaidehi V., Aishwarya M., “Diabetes Prediction using Machine Learning Algorithms”, *International Conference on Recent Trends in Advanced Computing (ICRTAC)*, 2019.