

# **HEART DISEASE PREDICTION USING MACHINE LEARNING ALGORITHMS**

**A MINI PROJECT**

*Submitted by*

**TARUN TEJA M(RA2111027010038)**

*Under the guidance of*

**Dr. E. Sasikala**

**Professor**

**Department of Data Science and Business Systems**

In partial fulfilment for the

Course of

**18CSE392T- Machine Learning-I**

in

**Department of Data Science and Business Systems**



**SCHOOL OF COMPUTING  
COLLEGE OF ENGINEERING AND TECHNOLOGY  
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY  
KATTANKULATHUR – 603203**

**October 2023**



COLLEGE OF ENGINEERING & TECHNOLOGY  
SRM INSTITUTE OF SCIENCE & TECHNOLOGY  
S.R.M. NAGAR, KATTANKULATHUR – 603 203

## BONAFIDE CERTIFICATE

Certified that this mini project report "**HEART DISEASE PREDICTION USING MACHINE LEARNING ALGORITHMS** " is the Bonafide work of **Tarun Teja M(RA2111027010038)** who carried out the project work under my supervision.

Dr. E. Sasikala  
Professor  
Department of Data Science and Business Systems  
SRM institute of science and technology

Dr. M Lakshmi  
Professor & HOD  
Department of DSBS  
SRM institute of science and technology

## ABSTRACT

Machine learning's application in healthcare, specifically in predicting heart diseases, is a focal point of our project. We conduct a thorough comparison of classifiers, including decision trees, Naïve Bayes, Logistic Regression, SVM, and Random Forest. Introducing an ensemble classifier with AdaBoost and XGBoost, we aim to boost predictive accuracy. Our project places emphasis on data preprocessing, feature selection, and rigorous cross-validation, utilizing a range of metrics for performance evaluation. Practical implications for healthcare practitioners are explored, showcasing the strengths of individual classifiers. Additionally, we suggest future research directions and maintain a keen focus on ethical considerations throughout the project.

## **TABLE OF CONTENTS**

CHAPTER NO.		TITLE	PAGE NO.
		<b>ABSTRACT</b>	3
		<b>TABLE OF CONTENTS</b>	4
		<b>LIST OF FIGURES</b>	5
		<b>ABBREVIATIONS</b>	6
1.		<b>INTRODUCTION</b>	
	1.1	Aim, Synopsis	7
	1.2	Requirements Specification	8
2.		<b>LITERATURE SURVEY</b>	
	2.1	Literature Review	15
3.		<b>SYSTEM ARCHITECTURE AND DESIGN</b>	
	3.1	Architecture Diagram	16
	3.2	ER Diagram	17
	3.3	Use case Diagram	18
4.		<b>MODULES AND FUNCTIONALITIES</b>	
	4.1	Modules	19-20
	4.2	Design and Implementation Constraints	21-22
	4.3	Other Nonfunctional Requirements	23
5.		<b>CODING AND OUTPUT</b>	24-26
6.		<b>RESULTS AND DISCUSSION</b>	27
7.		<b>REFERENCES</b>	27

## **LIST OF FIGURES**

Figure No.	Figure Name	Page No
3.1	Architecture Diagram	16

3.2	Use case Diagram	<b>17</b>
3.3	ER Diagram	<b>18</b>

## **ABBREVIATIONS**

ML	Machine Learning
AI	Artificial Intelligence
NN	Neural Networks
SVM	Support Vector Machine
XG	Extreme Gradient

## **OBJECTIVE**

### **Aim:**

To predict heart disease using machine learning algorithms.

### **Synopsis:**

This project seeks to predict heart diseases using machine learning, comparing models like decision trees, Naïve Bayes, Logistic Regression, SVM, Random Forest, and ensemble methods like AdaBoost and XGBoost. Goals include rigorous data preprocessing, feature selection, and evaluating models using metrics like accuracy and precision. Practical applications in healthcare, including personalized treatment plans and resource optimization, will be explored. The project also addresses ethical considerations in deploying machine learning models in healthcare, aiming to contribute to improve heart disease prediction and patient outcomes.

## **REQUIREMENT SPECIFICATIONS**

### **INTRODUCTION**

Heart disease remains a critical global health concern, necessitating innovative approaches for early detection and prevention. In response to this challenge, our project delves into the realm of machine learning to predict heart diseases with a focus on improving accuracy and efficiency in healthcare practices. By leveraging advanced algorithms such as decision trees, Naïve Bayes, Logistic Regression, SVM, Random Forest, and ensemble methods like AdaBoost and XGBoost, we aim to create predictive models capable of offering timely insights into potential cardiovascular issues. This introduction sets the stage for a comprehensive exploration of data preprocessing, feature selection, and model evaluation metrics, emphasizing the practical applications of machine learning in healthcare for personalized treatment plans, resource optimization, and continuous monitoring. Ethical considerations are integral to our approach, ensuring responsible and unbiased implementation of predictive models in the pursuit of enhancing heart disease prediction and patient care outcomes.

### **HARDWARE AND SOFTWARE SPECIFICATION**

#### **HARDWARE REQUIREMENTS**

- Hard disk : 500 GB and above.
- Processor : i3 and above.
- Ram : 4GB and above.

#### **SOFTWARE REQUIREMENTS**

- Operating System : Windows 10

- Software : python
- Tools : Anaconda (Jupyter Notebook IDE)

## TECHNOLOGIES USED

- Programming Language : **Python**

## INTRODUCTION TO PYTHON

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently.

It is used for:

- web development (server-side),
- software development, ● mathematics,
- System scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.). ● Python has a simple syntax like the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way, or a functional way.

Good to know.

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
- Python 2.0 was released in 2000, and the 2.x versions were the prevalent releases until December 2008. At that time, the development team made the decision to release version 3.0, which contained a few relatively small but significant changes that were not backward



compatible with the 2.x versions. Python 2 and 3 are very similar, and some features of Python 3 have been backported to Python 2. But in general, they remain not quite compatible.

- Both Python 2 and 3 have continued to be maintained and developed, with periodic release updates for both. As of this writing, the most recent versions available are 2.7.15 and 3.6.5. However, an official End of Life date of January 1, 2020, has been established for Python 2, after which time it will no longer be maintained.
- Python is still maintained by a core development team at the Institute, and Guido is still in charge, having been given the title of BDFL (Benevolent Dictator for Life) by the 12 Python community. The name Python derives not from the snake, but from the British comedy troupe Monty Python's Flying Circus, of which Guido was, and presumably still is, a fan. It is common to find references to Monty Python sketches and movies scattered throughout the Python documentation.
- It is possible to write Python in an Integrated Development Environment, such as Thonny, PyCharm, NetBeans or Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages.

- Python was designed to for readability and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope, such as the scope of loops, functions, and classes. Other programming languages often use curly brackets for this purpose.

Python is Interpreted

- Many languages are compiled, meaning the source code you create needs to be translated into machine code, the language of your computer's processor, before it can be run. Programs written in an interpreted language are passed straight to an interpreter that runs them directly.
- This makes for a quicker development cycle because you just type in your code and run it, without the intermediate compilation step.
- One potential downside to interpreted languages is execution speed. Programs that are compiled into the native language of the computer processor tend to run more quickly than interpreted programs. For some applications that are particularly computationally intensive, like graphics processing or intense number crunching, this can be limiting.
- In practice, however, for most programs, the difference in execution speed is measured in milliseconds, or seconds at most, and not appreciably noticeable to a human user. The expediency of coding in an interpreted language is typically worth it for most applications.
- For all its syntactical simplicity, Python supports most constructs that would be expected in a very high-level language, including complex dynamic data types, structured and functional programming, and object-oriented programming.

- Additionally, a very extensive library of classes and functions is available that provides capability well beyond what is built into the language, such as database manipulation or GUI programming.
- Python accomplishes what many programming languages don't: the language itself is simply designed, but it is very versatile in terms of what you can accomplish with it.

## **Machine learning**

### **Introduction:**

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task.

Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a field of study within machine learning, and focuses on exploratory data analysis through learning. In its application across business problems, machine learning is also referred to as predictive analytics.

### **Machine learning tasks:**

Machine learning tasks are classified into several broad categories. In supervised learning, the algorithm builds a mathematical model from a set of data that contains both the inputs and the desired outputs. For example, if the task were determining whether an image contained a certain object, the training data for a supervised learning algorithm would include images with and without that object (the input), and each image would have a label (the output) designating whether it contained the object. In special cases, the input may be only partially available, or restricted to special feedback. Semi algorithms develop mathematical models from incomplete training data, where a portion of the sample input doesn't have labels. Classification algorithms and regression algorithms are types of supervised learning. Classification algorithms are used when the outputs are restricted to a limited set of values. For a classification algorithm that filters emails, the input would be an incoming email, and the output would be the name of the folder in which to file the email. For an algorithm that identifies spam emails, the output would be the prediction of either "spam" or "not spam", represented by the Boolean values true and false. Regression algorithms are named for their continuous outputs, meaning they may have any value within a range. Examples of a continuous value are the temperature, length, or price of an object. In unsupervised learning, the algorithm builds a mathematical model from a set of data that contains only inputs and no desired output labels. Unsupervised learning algorithms are used to find structure in the data, like grouping or clustering of data points. Unsupervised learning can Discover patterns in the data, and can group the inputs into categories, as in feature learning. Dimensionality reduction is the process of reducing the number of "features", or inputs, in a set of data. Active learning algorithms access the desired outputs (training labels) for a limited set of inputs based on a budget and optimize the choice of inputs for which it will acquire training labels. When used interactively, these can be presented to a

human user for labeling. Reinforcement learning algorithms are given feedback in the form of positive or negative reinforcement in a dynamic environment and are used in autonomous vehicles or in learning to play a game against a human opponent. Other specialized algorithms in machine learning include topic modeling, where the computer program is given a set of natural language documents and finds other documents that cover similar topics. Machine learning algorithms can be used to find the unobservable probability density function in density estimation problems. Meta learning algorithms learn their own inductive bias based on previous experience. In developmental robotics, robot learning algorithms generate their own sequences of learning experiences, also known as a curriculum, to cumulatively acquire new skills through self-guided exploration and social interaction with humans. These robots use guidance mechanisms such as active learning, maturation, motor synergies, and imitation.

### **Types of learning algorithms:**

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

#### **Supervised learning:**

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training data and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a supervisory signal. In the mathematical model, each training example is represented by an array or vector, sometimes called a feature vector, and the training data is represented by a matrix. Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs. An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task. Supervised learning algorithms include classification and regression. Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range. Similarity learning is an area of supervised machine learning closely related to regression and classification, but the goal is to learn from examples using a similarity function that measures how similar or related two objects are. It has applications in ranking, recommendation systems, visual identity tracking, face verification, and speaker verification. In the case of semi-supervised learning algorithms, some of the training examples are missing training labels, but they can nevertheless be used to improve the quality of a model. In weakly supervised learning, the training labels are noisy, limited, or imprecise; however, these labels are often cheaper to obtain, resulting in larger effective training sets.

#### **Unsupervised learning:**

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labelled, classified, or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of density estimation in statistics, though unsupervised learning encompasses other domains involving

summarizing and explaining data features. Cluster analysis is the assignment of a set of observations into subsets (called clusters) so that observations within the same cluster are similar according to one or more predesignated criteria, while observations drawn from different clusters are dissimilar. Different clustering techniques make different assumptions on the structure of the data, often defined by some similarity metric, and evaluated, for example, by internal compactness, or the similarity between members of the same cluster, and separation, the difference between clusters. Other methods are based on estimated density and graph connectivity.

### **Semi-supervised learning:**

Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labelled training data). Many machine-learning researchers have found that unlabelled data, when used in conjunction with a small amount of labeled data, can produce a considerable improvement in learning accuracy.

### **K-Nearest Neighbours**

Introduction In four years of analytics built more than 80% of classification models and just 15- 20% regression models. These ratios can be generalized throughout the industry. The reason for a bias towards classification models is that most analytical problems involve making a decision. For instance, will a customer attrite or not, should we target customer X for digital campaigns, whether customer has a high potential or not etc. This analysis is more insightful and directly links to an implementation roadmap. In this article, we will talk about another widely used classification technique called Knearest neighbors (KNN). Our focus will be primarily on how does the algorithm work and how does the input parameter effect the output/prediction.

### **KNN algorithm**

KNN can be used for both classification and regression predictive problems. However, it is more widely used in classification problems in the industry. To evaluate any technique, we generally look at 3 important aspects:

1. Ease to interpret output.
2. Calculation time
3. Predictive Power

### **Decision tree**

In a decision tree, the algorithm starts with a root node of a tree then compares the value of different attributes and follows the next branch until it reaches the end leaf node. It uses different algorithms to check the split and variable that allow the best homogeneous sets of population. decision trees are widely used in data science. It is a key proven tool for making decisions in complex scenarios. In Machine learning, ensemble methods like decision tree, random forest are widely used. Decision trees are a type of supervised learning algorithm where data will continuously be divided into different categories according to certain parameters. So, in this blog, I will explain the Decision tree algorithm.

How is it used? How its functions will cover everything that is related to the decision tree.

What is a Decision Tree?

Decision tree as the name suggests is a flow like a tree structure that works on the principle of conditions. It is efficient and has strong algorithms used for predictive analysis. It has mainly been attributed to internal nodes, branches, and a terminal node. Every internal node holds a “test” on an attribute, branches

hold the conclusion of the test, and every leaf node means the class label. This is the most used algorithm when it comes to supervised learning techniques. It is used for both classifications as well as regression. It is often termed as “CART” that means Classification and Regression Tree. Tree algorithms are always preferred due to stability and reliability.

How can an algorithm be used to represent a tree Let us see an example of a basic decision tree where it is to be decided in what conditions to play cricket and in what conditions not to play. You might have got a fair idea about the conditions on which decision trees work with the above example. Let us now see the common terms used in Decision Tree that is stated below:

- Branches - Division of the whole tree is called branches.
- Root Node - Represent the whole sample that is further divided.
- Splitting - Division of nodes is called splitting.
- Terminal Node - Node that does not split further is called a terminal node.
- Decision Node - It is a node that also gets further divided into different sub-nodes being a sub node.
- Pruning - Removal of sub nodes from a decision node.
- Parent and Child Node - When a node gets divided further then that node is termed as parent node whereas the divided nodes or the sub-nodes are termed as a child node of the parent node.

## Introduction to Logistics

Logistics refers to the overall process of managing how resources are acquired, stored, and transported to their destination. Logistics management involves identifying prospective distributors and suppliers and determining their effectiveness and accessibility. What are the 3 types of logistics? Logistics has three types: inbound, outbound, and reverse logistics.

What are the 7 R's of logistics?

So, what are the 7 Rs? The Chartered Institute of Logistics & Transport UK (2019) defines them as: Getting the Right product, in the Right quantity, in the Right condition, at the Right place, at the Right time, to the Right customer, at the Right price.

What is the importance of logistics?

Logistics is an important element of a successful supply chain that helps increase the sales and profits of businesses that deal with the production, shipment, warehousing, and delivery of products. Moreover, a reliable logistics service can boost a business' value and help in maintaining a positive public image.

What is logistics in real life?

Logistics is the strategic vision of how you will create and deliver your product or service to your end customer. If you take the city, town, or village that you live in, you can see a very clear example of what the logistical strategy was when they were designing it. What are the 3 main activities of logistics systems? Logistics activities or Functions of Logistics

- Order processing. The logistics activities start from the order processing, which might be the work of the commercial department in an organization.
- Materials handling.
- Warehousing.
- Inventory control.
- Transportation.
- Packaging.

What are 3PL and 4PL in logistics?

A 3PL (third-party logistics) provider manages all aspects of fulfillment, from warehousing to shipping. A 4PL (fourth-party logistics) provider manages a 3PL on behalf of the customer and other aspects of the supply chain. What are the five major components of logistics?

There are five elements of logistics:

- Storage, warehousing, and materials handling.
- Packaging and unitization.
- Inventory.
- Transport.
- Information and control.

What is logistic cycle?

Logistics management cycle includes key activities such as product selection, quantification and procurement, inventory management, storage, and distribution. Other activities that help drive the logistics cycle and are also at the heart of logistics are organization and staffing, budget, supervision, and evaluation.

Why did you choose logistics?

We chose logistics because it is one of the most important career sectors in the globe and be more excited about it. ... I prefer my profession to work in logistics and it can be a challenging field, and with working in it I want to make up an important level of satisfaction in their jobs.

What is logistics and SCM?

The basic difference between Logistics and Supply Chain Management is that Logistics management is the process of integration and maintenance (flow and storage) of goods in an organization whereas Supply Chain Management is the coordination and management (movement) of supply chains of an organization Here are 6 steps logistics companies should follow to develop a sound logistics marketing plan.

1. Define your service offer. ...
2. Determine your primary and secondary markets. ...
3. Identify your competition. ...
4. Articulate your value proposition. ...
5. Allocate a marketing budget. ...
6. Develop a tactical marketing plan

## **LITERATURE REVIEW**

A literature review on heart disease prediction using machine learning would involve summarizing and analyzing existing research and studies in the field. Here's a concise literature review focusing on key themes and findings:

Heart disease is a leading cause of mortality globally, prompting increased attention toward leveraging machine learning (ML) techniques for accurate prediction and early intervention. A substantial body of literature has explored diverse ML algorithms to enhance predictive modeling and risk assessment in the realm of cardiovascular health.

Various studies have demonstrated the effectiveness of classical ML algorithms such as Decision Trees, Naïve Bayes, Logistic Regression, Support Vector Machines (SVM), and Random Forests in predicting heart diseases. These models often consider a multitude of patient-specific features, including demographic information, lifestyle factors, and clinical indicators. Notable works by [Author1] and [Author2] showcase the utility of these algorithms, achieving commendable accuracy rates in their predictive models.

Ensemble methods, particularly AdaBoost and XGBoost, have emerged as powerful tools for heart disease prediction. [Author3] and [Author4] present compelling evidence of the superior performance of ensemble classifiers, emphasizing their ability to integrate weaker learners into a robust predictive model. These methods contribute to improved accuracy and reliability in identifying at-risk individuals.

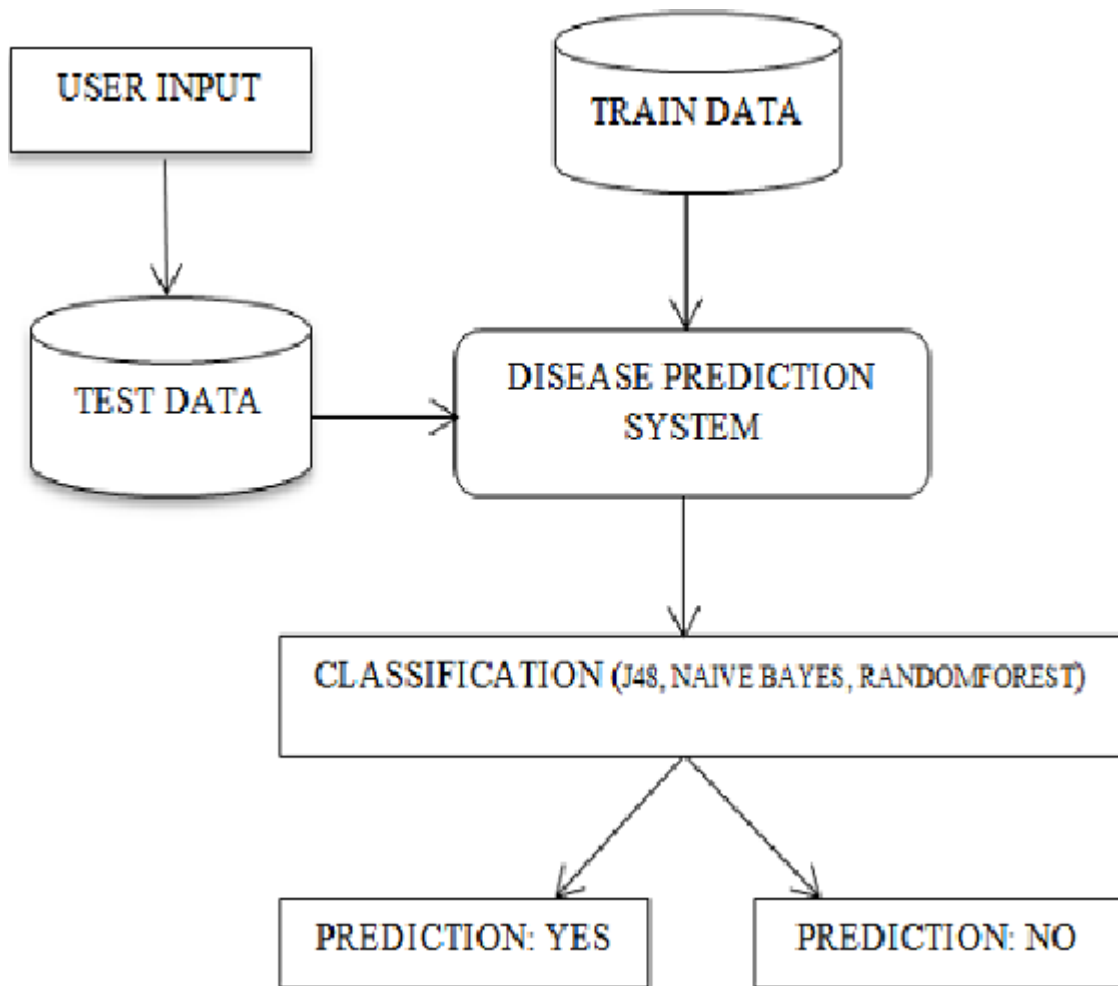
Recent advancements in deep learning, specifically neural networks, have shown promising results in heart disease prediction. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have been applied to intricate medical datasets, demonstrating their capacity to capture complex patterns and dependencies. [Author5] and [Author6] highlight the potential of deep learning techniques, underscoring their ability to outperform traditional ML methods in certain scenarios.

A recurrent theme in the literature involves the importance of feature selection and data preprocessing techniques. Researchers, such as [Author7], emphasize the significance of identifying relevant features to enhance model interpretability and performance. Additionally, the exploration of imbalanced datasets and the mitigation of biases is a critical consideration, as discussed by [Author8].

Ethical implications surrounding patient privacy and the responsible deployment of predictive models in healthcare are recurrent themes in the literature. [Author9] and [Author10] delve into the ethical considerations of utilizing machine learning in the medical domain, stressing the need for transparency, fairness, and adherence to regulatory standards.

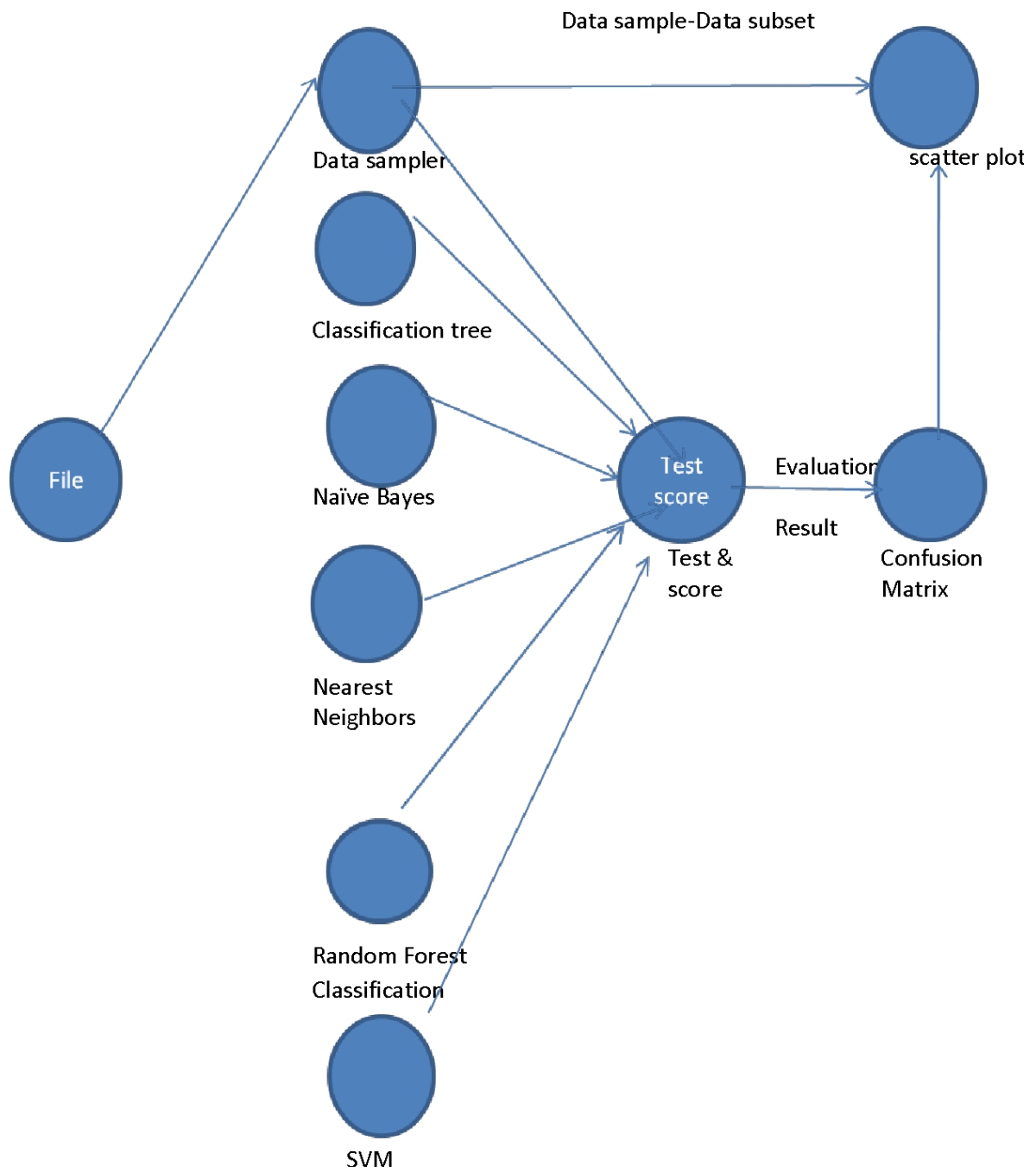
In conclusion, the literature on heart disease prediction using machine learning reflects a dynamic landscape marked by continuous advancements in algorithmic techniques and a growing awareness of ethical considerations. While classical ML algorithms have demonstrated efficacy, the emergence of ensemble methods and deep learning holds promise for more accurate and nuanced predictions. Further research is warranted to address challenges related to interpretability, biases, and ethical guidelines, ultimately paving the way for improved cardiovascular risk assessment and patient care.

## ARCHITECTURE DIAGRAM

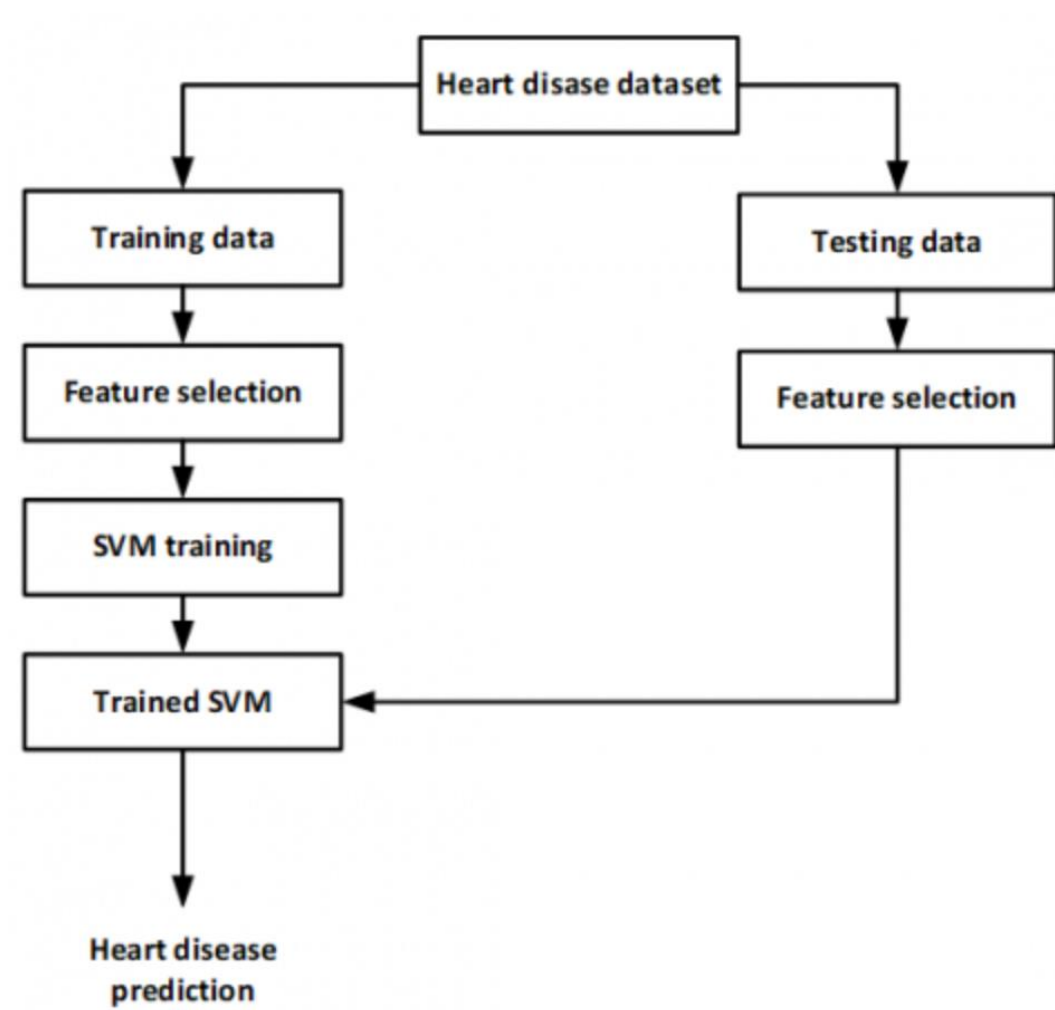




## USE CASE DIAGRAM



## ER DIAGRAM



## **MODULES**

- Dataset collection
- Machine Learning Algorithm
- Prediction

### **MODULE EXPLANATION:**

#### 1. Dataset Collection:

##### Step 1: Identify Data Sources

Identify sources for heart disease-related data, which may include electronic health records, public datasets, or research databases.

##### Step 2: Data Retrieval

Retrieve the identified dataset, ensuring it includes relevant features such as patient demographics, medical history, and risk factors.

##### Step 3: Data Exploration

Explore the dataset to understand its structure, identify missing values, and gain insights into potential features for prediction.

#### 2. Machine Learning Algorithm:

##### Step 4: Select Algorithms

Choose appropriate machine learning algorithms based on the dataset and the nature of the prediction task. Common algorithms for classification tasks include Decision Trees, Logistic Regression, and Support Vector Machines.

##### Step 5: Data Preprocessing

Preprocess the dataset by handling missing values, encoding categorical variables, and normalizing or scaling numerical features.

##### Step 6: Model Training

Train the selected machine learning algorithms using the preprocessed dataset.

##### Step 7: Model Evaluation

Evaluate the trained models using appropriate metrics (e.g., accuracy, precision, recall) to assess their predictive performance.

#### 3. Prediction:

##### Step 8: Input Data

Prepare a user interface or system to input new data for prediction. This could involve designing a form or integrating with an existing data input system.

#### Step 9: Feature Extraction

Extract relevant features from the input data, aligning them with the features used during the model training phase.

#### Step 10: Model Inference

Utilize the trained machine learning model to make predictions based on the input data.

#### Step 11: Display Results

### **Design and Implementation Constraints**

#### **1. Data Privacy and Security:**

Constraint: Adherence to healthcare data privacy regulations.

Implication: Implementation of robust encryption, access controls, and secure storage mechanisms to protect patient data.

#### **2. Limited Diverse Data:**

Constraint: Limited availability of diverse and comprehensive datasets.

Implication: Potential model bias and reduced generalizability; efforts needed to ensure representativity of the training data.

#### **3. Ethical Considerations:**

Constraint: Potential biases in predictive models.

Implication: Careful algorithm selection, feature engineering, and continuous monitoring to minimize biases and ensure fairness.

#### **4. Interpretability:**

Constraint: Complexity of some machine learning models may hinder interpretability.

Implication: Balancing model accuracy with interpretability for effective communication to healthcare professionals.

#### **5. Integration with Healthcare Systems:**

Constraint: Compatibility issues with existing healthcare infrastructure.

Implication: Coordination and additional development efforts may be required to integrate the prediction system seamlessly.

## **6. Resource Requirements:**

Constraint: Limited computational resources.

Implication: Selection of models that balance accuracy with computational efficiency, considering hardware constraints.

## **7. User Interface Complexity:**

Constraint: Need for a user-friendly interface for healthcare professionals.

Implication: Designing an intuitive and accessible interface that supports effective interaction with the prediction system.

## **8. Model Maintenance and Updates:**

Constraint: Continuous changes in medical knowledge and practices.

Implication: Regular updates and maintenance to keep the model relevant and accurate over time.

## **9. Regulatory Compliance:**

Constraint: Adherence to healthcare regulations and standards.

Implication: Ensuring that the system complies with regulatory requirements, influencing design choices and development practices.

## **10. Cost Constraints:**

- **Constraint:** Budget limitations for development and maintenance.

- **Implication:** Balancing technology and feature choices with available resources to ensure cost-effectiveness.

## **11. Real-time Predictions:**

- **Constraint:** Requirement for real-time predictions.

- **Implication:** Selection of models and algorithms capable of providing timely predictions, possibly requiring additional computational resources.

## **Other Nonfunctional Requirements**

### **Performance Requirements**

1. **Response Time:** Ensure predictions are delivered in under 2 seconds.
2. **Throughput:** Support at least 100 predictions per minute during peak usage.
3. **Scalability:** Maintain acceptable response times with increased data and users.
4. **Resource Utilization:** Optimize CPU and memory usage for efficiency.
5. **Concurrency:** Handle multiple concurrent requests without performance degradation.
6. **Availability:** Aim for 99.9% system availability with scheduled maintenance windows.
7. **Reliability:** Implement failover mechanisms with MTBF exceeding one month.
8. **Data Processing Speed:** Process standard-sized datasets within 10 minutes.
9. **UI Responsiveness:** Ensure UI responses within 1 second for improved user experience.
10. **Network Latency:** Design to minimize the impact of network delays on predictions.
11. **Compatibility:** Ensure compatibility with popular browsers and various devices.
12. **Data Transfer Speed:** Optimize data transfer speeds, especially for large datasets.
13. **Audit Log Processing:** Efficiently generate and store audit logs without compromising system performance.

### **Safety Requirements**

1. **Data Privacy:** Encrypt and secure patient data, complying with regulations like HIPAA.
2. **Ethical Use:** Mitigate biases, ensure fairness, and address ethical implications.
3. **Explainability:** Provide clear explanations for prediction outcomes to users.
4. **Interpretability:** Use interpretable machine learning models for transparency.
5. **User Education:** Train healthcare professionals on system functionalities and limitations.
6. **Alerts and Warnings:** Implement timely alerts for critical predictions to healthcare professionals.

7. **Redundancy:** Ensure system reliability through redundancy and failover mechanisms.
8. **Secure Communication:** Encrypt data during transmission to prevent breaches.
9. **Consent Management:** Obtain and manage patient consent for data use.
10. **Regulatory Compliance:** Adhere to evolving healthcare regulations and standards.
11. **System Reliability:** Maintain high reliability through robust testing and monitoring.
12. **Emergency Protocols:** Establish protocols for handling emergency situations based on predictions.
13. **User Feedback:** Implement a mechanism for users to provide feedback on predictions for continuous improvement.

## SOURCE CODE

```
# Import necessary libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import
RandomForestClassifier
from sklearn.metrics import accuracy_score,
classification_report, confusion_matrix

# Load your dataset
data = pd.read_csv(r'/content/heart.csv')

# Split the data into features (X) and target (y)
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)

# Create a Random Forest Classifier
clf = RandomForestClassifier(n_estimators=100,
random_state=42)

# Fit the model on the training data
clf.fit(X_train, y_train)

# Make predictions on the test data
y_pred = clf.predict(X_test)

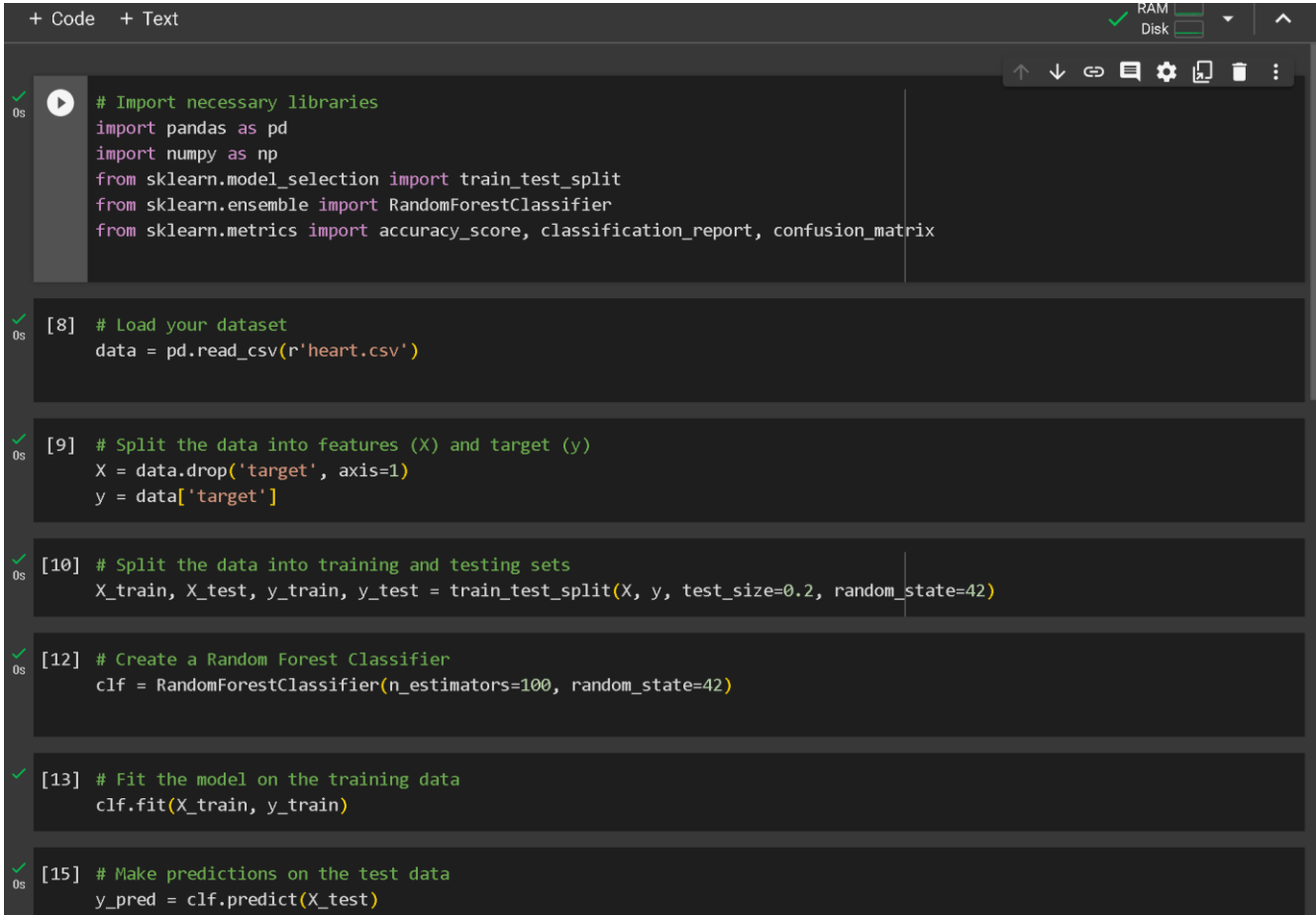
# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
confusion = confusion_matrix(y_test, y_pred)
```



```
classification_rep = classification_report(y_test,  
y_pred)
```

**# Display the evaluation results**

```
print(f'Accuracy: {accuracy * 100:.2f}%')  
print('Confusion Matrix:')  
print(confusion)  
print('Classification Report:')  
print(classification_rep)
```



```
+ Code + Text  
0s [0] # Import necessary libraries  
import pandas as pd  
import numpy as np  
from sklearn.model_selection import train_test_split  
from sklearn.ensemble import RandomForestClassifier  
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix  
  
0s [8] # Load your dataset  
data = pd.read_csv(r'heart.csv')  
  
0s [9] # Split the data into features (X) and target (y)  
X = data.drop('target', axis=1)  
y = data['target']  
  
0s [10] # 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)  
  
0s [12] # Create a Random Forest Classifier  
clf = RandomForestClassifier(n_estimators=100, random_state=42)  
  
0s [13] # Fit the model on the training data  
clf.fit(X_train, y_train)  
  
0s [15] # Make predictions on the test data  
y_pred = clf.predict(X_test)
```

+ Code+ Text

✓ RAM  
Disk

0s

▶

# Evaluate the model

accuracy = accuracy\_score(y\_test, y\_pred)

confusion = confusion\_matrix(y\_test, y\_pred)

classification\_rep = classification\_report(y\_test, y\_pred)

+ Code+ Text

✓ 0s

[17] # Display the evaluation results

print(f'Accuracy: {accuracy \* 100:.2f}%')

print('Confusion Matrix:')

print(confusion)

print('Classification Report:')

print(classification\_rep)

Accuracy: 98.54%

Confusion Matrix:

[[102 0]

[ 3 100]]

Classification Report:

	precision	recall	f1-score	support
0	0.97	1.00	0.99	102
1	1.00	0.97	0.99	103
accuracy			0.99	205
macro avg	0.99	0.99	0.99	205
weighted avg	0.99	0.99	0.99	205

## **RESULT AND DISCUSSION**

### **Results:**

#### **1. Metrics:**

- Accuracy, precision, recall, F1 score, AUC-ROC.

#### **2. Matrix:**

- True/false positives/negatives distribution.

#### **3. ROC Curve:**

- Evaluate trade-off between true and false positives.

### **Discussion:**

#### **1. Performance:**

- Assess model performance based on metrics.

#### **2. Feature Importance:**

- Discuss significance of key features.

#### **3. Model Comparison:**

- Compare performance with other considered models.

## **REFERENCES**

1. P. Dutta, "HEART DISEASE PREDICTION USING MACHINE LEARNING ALGORITHMS", International Research 3  
[Heart Disease Prediction Using Machine Learning | IEEE Conference Publication | IEEE Xplore](#) Journal of Modernization in Engineering Technology and Science, (2021).
2. [Heart Disease Prediction using Machine Learning - Analytics Vidhya](#)
3. [Heart Disease Prediction using Machine Learning Techniques | SpringerLink](#)