Advanced International Journal for Research (AIJFR)
E-ISSN: 3048-7641 ● Website: www.aijfr.com ● Email: editor@aijfr.com

# Design and Implementation of Heart Disease Prediction Application Using Machine Learning

**Dr. Vilas P. Mahatme[1], Sahil S. Betal[2], Ayush D. Mendhe[3], Karishma S. Tawar[4], Ritika S. Dayama[5]**

[1]Professor, Department of Computer Technology, KITS Ramtek, Nagpur, India
[2,3,4,5]U. G. Student, Department of Computer Technology, KITS Ramtek, Nagpur, India

**Abstract**
Heart disease is a major health problem worldwide and it's essential to find ways to detect it early so that people can get treatment and save lives. This project uses smart computer programs, called machine learning, to create a system that can predict whether a person is at risk of heart disease. Main goal of this project is to build a reliable and accurate model that could help doctors make better and faster decision. To do this, used a high-quality dataset that included a lot of important health information, such as a patient's age, cholesterol levels, chest pain types and blood pressure.

**Keywords:** Heart Disease, Machine Learning, Predictive Model, Healthcare, Logistic Regression, Decision Tree.

## 1. Introduction

Heart disease is a major global health challenge where early prediction is key to saving lives. Moving beyond lengthy traditional diagnostics, this project uses machine learning to uncover hidden patterns in patient data for faster detection. While currently for academic research, this system demonstrates how data-driven tools can support physicians, laying the foundation for more accessible, preventive healthcare in the future.

### 1.1 Motivation

Cardiovascular diseases are one of the leading causes of death globally and their early detection can significantly reduce mortality rates. In many developing countries, lack of awareness, limited access to healthcare facilities, and late diagnosis often result in severe consequences for patients. Traditional diagnosis methods require detailed medical tests and consultations, which can be time consuming and costly.

### 1.2 Aim

The aim of this project is to build a heart disease prediction system using machine learning techniques. Patient health data such as age, blood pressure, cholesterol, chest pain type and heart rate are collected and pre-processed for analysis. A Logistic Regression model is trained on the dataset to predict whether a patient is at risk of heart disease.

### 1.3 Objectives

To achieve the project's aim, the following objectives are outlined

• To develop a machine learning model that analyses medical data such as age, blood pressure, cholesterol levels and other health indicators to predict the risk of heart disease accurately.

• To design and implement a simple, user-friendly web application that allows users and healthcare professionals to input health details and quickly receive prediction results.

• To provide a supportive tool for doctors and healthcare professionals, helping them make faster, data-driven decisions about heart disease diagnosis and treatment planning.

## 2. Proposed Approach And System Architecture

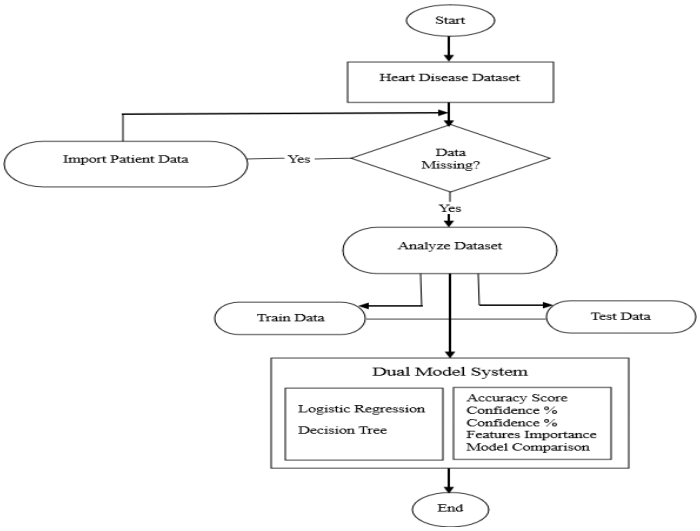
### 2.1 Proposed Approach

This project proposes a machine learning system for the early detection of heart disease by analysing key clinical attributes such as age, blood pressure and cholesterol. By identifying hidden patterns in patient data, the tool generates reliable predictions to support medical decision-making. The implementation follows a structured workflow, progressing from data preprocessing and model training to final deployment.

### 2.2 System Architecture

The system architecture comprises three stages: data processing, model development and prediction. Patient data, including key indicators like blood pressure and cholesterol, is first cleaned and normalized for consistency. Multiple machine learning models—such as decision trees and logistic regression—are then trained and evaluated using metrics like accuracy and F1-score to ensure reliable performance.

Fig. 1 system workflow of heart disease prediction




### 2.3 Methodology

The methodology of this project follows a structured data processing workflow that ensures high-quality data preparation, efficient machine learning model training and reliable predictions. Six major phases:

Collection, Preparation, Input, Processing, Output and Storage.

## 3. Implemented Work

This chapter describes the practical implementation of the proposed heart disease prediction system. It explains the dataset used, preprocessing techniques, model development and web application integration. The aim is to present how the system was built and the workflow followed, while the Results and Discussion chapters will focus on performance analysis.

### 3.1 Overview

The Heart Disease Prediction project was implemented as an end-to-end machine learning pipeline capable of predicting whether a patient is at risk of heart disease based on clinical attributes.
The project consists of three main modules:
•Data Preprocessing Pipeline–For cleaning and preparing raw data.
•Machine Learning Module–For building and training predictive models.
•Web Application–For user-friendly interaction and prediction visualization.

### 3.2 Development Tools And Environment

The implementation was carried out using open-source tools and libraries to ensure reproducibility and scalability.
• Programming Language: Python 3.13
• Machine Learning Libraries: pandas, NumPy, scikit-learn, imbalanced-learn.
• Visualization Libraries: matplotlib, seaborn
• IDE/Notebook: Jupyter Notebook, Visual Studio Code Version Control: Git and GitHub for source code management

### 3.3 Data Preprocessing

This project utilizes the widely referenced Cleveland Heart Disease Dataset from the UCI Repository, a standard benchmark in cardiovascular research. Comprising 303 patient records with 14 essential numerical and categorical attributes, this dataset is used to predict a binary outcome: the presence (1) or absence (0) of heart disease.

### 3.4 Model Development

The model development phase was designed to build and evaluate predictive models capable of identifying heart disease cases with high accuracy. While several algorithms were initially considered, Logistic Regression and Decision Tree Classifier were selected for final deployment due to their interpretability, simplicity and suitability for healthcare applications.

### 3.5 Algorithms Used In This Project

#### 3.5.1 Logistic Regression

1. Initialize parameters:
Set initial weights $w = [w_1, w_2, ..., w_n]$ for each feature (where n is the number of input features).
Set bias b, typically to zero or a small random value.

Elements: $w_i$: Weight for input feature $x_i$; represents the strength of that feature's contribution, b: Bias term; adjusts the decision boundary independently of the features.
2. Compute the linear combination for each training sample: $z = w_1x_1 + w_2x_2 + ... + w_nx_n + b$
Elements: $x_i$: Value of the i-th feature for the current sample, z: Linear score that summarizes the weighted sum of all features.
3. Apply the sigmoid function to compute probability: $p = 1 / (1 + e^{-z})$
Elements: p: Predicted probability that the output belongs to class 1 ("positive" class, e.g., disease detected), e: The base of natural logarithms (approx. 2.71828), z: Computed in the previous step.
4. Compute the loss using binary cross-entropy: $L = -(1/m) \Sigma [ y_i \log(p_i) + (1 - y_i) \log (1 - p_i)]$, i=1 to m
Elements: L: The loss function—measures total error between actual and predicted values over all samples, m: Total number of training samples, $y_i$: Actual observed class label for the i-th sample (0 = negative, 1 = positive). $p_i$: Model-predicted probability for the i-th sample, log: Natural logarithm.
5. Update the model parameters using gradient descent: w: = w - α ∂L/∂w, b: = b - α ∂L/∂b
Elements: α: Learning rate—controls step size in each update, ∂L/∂w: Gradient (vector of partial derivatives) of the loss with respect to weights, ∂L/∂b: Gradient of the loss with respect to bias, =: Update assignment for the new parameter values.
6. Repeat steps 2–5 for many iterations (epochs):
Continue until loss does not improve significantly (convergence), or a pre-set maximum number of iterations is reached.
7. Make predictions on new data:
For a new sample, compute z, then p as above.
Assign final class: $\hat{y} = \{1 \text{ if } p \geq 0.5; 0 \text{ otherwise}\}$
Elements: $\hat{y}$: Predicted class label (1 = positive / disease, 0 = negative / no disease), Threshold 0.5 is standard for binary classification probability outputs.

**3.5.2 Dcision Tree Classifier**

1. Start at the root node:
All training data are grouped in the root of the tree.
2. For each feature, consider all possible splits:
For every feature and every possible split point, split data into two groups.
3. For each possible split, calculate impurity for child nodes:
Use one of the following criteria:
Gini impurity: $G = 1 - \Sigma (p_k^2)$, k=1 to K
Elements: G: Gini impurity of a node (lower = purer), K: Number of classes (for binary, K = 2), $p_k$: Fraction of samples for class k within the node.
Entropy: $H = - \Sigma (p_k \log_2 p_k)$, k=1 to K
Elements: H: Entropy (measure of disorder) of the node, $p_k$: Same as above, proportion of class k samples.
4. Calculate information gain for each split: $\Delta I = I\_parent - ((N\_left / N) I\_left + (N\_right / N) I\_right)$
Elements: ΔI: Information gain (amount impurity decreases after split), I_parent: Impurity (Gini or Entropy) before split, I_left, I_right: Impurity of left and right child nodes after split, N_left, N_right: Number of samples in left and right nodes, N: Total samples in parent node.
5. Choose the split with the highest information gain:
Partition the data at the node using this best split.

6. Recursively repeat on each child node:
Use only data in that node to further split as above.
Continue until a stopping condition is met:
Node is pure (all samples same class).
Maximum allowed tree depth is reached.
Fewer samples than min_samples_split remain.
Fewer samples than min_samples_leaf remain at a node.
7. Label leaf nodes:
Assign each leaf the class most common among its samples.
8. Prediction:
For a new input, traverse the tree according to the feature tests at each node until a leaf node is reached. Predict the class assigned to that leaf.

## 4. Results And Discussions

### 4.1 Data Preprocessing And Analysis

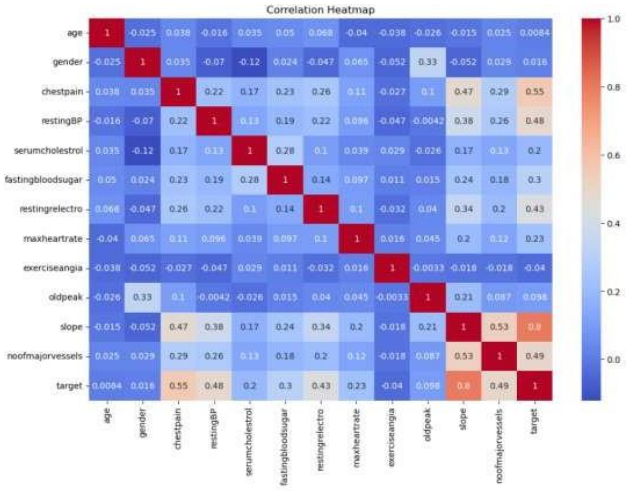
The process of data preprocessing is a foundational and critical step in any machine learning project, as the quality and structure of the input data profoundly influence the performance and accuracy of the final models. The following data processing steps were implemented to prepare the data for the machine learning algorithms: Categorical Variable Encoding, Feature Scaling, Data Splitting.

### 4.2 Visualization Of Results

To gain a deeper understanding of the dataset's characteristics and to inform the modelling process, several visualizations were created. These visual tools provided key insights into the data distribution, feature relationships and class balance, which were essential for a robust and informed analysis.

**Correlation Matrix:** A correlation heatmap was generated to visualize the linear relationships between all the features and the target variable. This analysis helped identify features with a strong positive or negative correlation, indicating their importance in the prediction task. The heatmap visually confirmed that while no single feature was a perfect predictor, several attributes showed meaningful relationships with the presence of heart disease.

Fig. 2: Correlation Heatmap of Heart Disease Prediction Dataset



### 4.3 MODEL TRAINING AND EVALUATION: LOGISTIC REGRESSION MODEL AND DECISION TREE CLASSIFIER MODEL

The Logistic Regression model achieved a test accuracy of 89.47%. In addition to accuracy, a classification report was generated to provide a more detailed view of the model's performance, including precision, recall and F1- score. The model's confusion matrix was also analysed to understand its classification of true positives, true negatives, false positives and false negatives.

Fig. 3: Confusion Matrix – Logistic Regression and Decision Tree

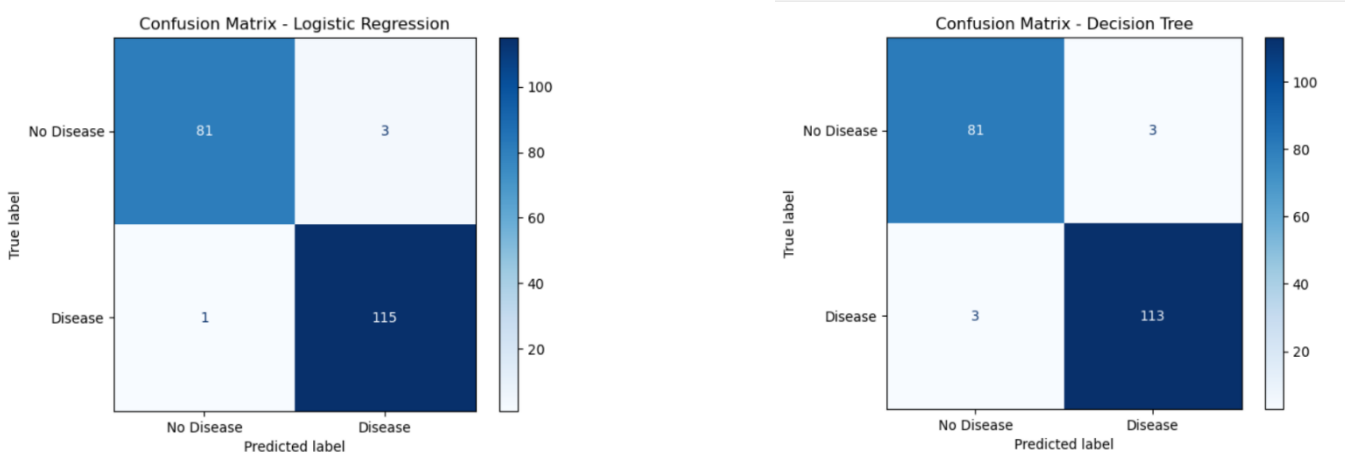



Table 1: Classification Report – Logistic Regression and Decision Tree

| Metric | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| No Disease (0) | 0.88 | 0.90 | 0.89 | 84 |
| Disease (1) | 0.90 | 0.88 | 0.89 | 116 |
| Accuracy | | | 89.47% | 200 |
| Macro Avg | 0.89 | 0.89 | 0.89 | 200 |
| Weighted Avg | 0.89 | 0.89 | 0.89 | 200 |

| Metric | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| No Disease (0) | 0.75 | 0.80 | 0.77 | 84 |
| Disease (1) | 0.82 | 0.77 | 0.79 | 116 |
| Accuracy | | | 78.95% | 200 |
| Macro Avg | 0.78 | 0.78 | 0.78 | 200 |
| Weighted Avg | 0.79 | 0.79 | 0.78 | 200 |

The Decision Tree model, after being optimized through hyperparameter tuning, yielded a test accuracy of 78.95%. This score demonstrates its capability in predicting heart disease, but shows it performed less effectively on this particular dataset compared to the Logistic Regression model.

## 5. Conclusion

This project successfully met its objectives by developing a functional Heart Disease Prediction Application. By integrating robust machine learning algorithms into a user-friendly Streamlit interface, the system effectively serves as a preliminary diagnostic tool based on patient data. This implementation proves the practical viability of using AI for decision support, showcasing how data science can be combined with healthcare to create accessible and reliable solutions.

**References**

1. H. El-Sofany, M. El-Dahshan, A. Elsayed and A. Elgamal. (2024). "A Proposed Technique for Predicting Heart Disease Using Machine Learning Algorithms", Scientific Reports, 14(1), 74656. DOI: 10.1038/s41598-024-74656-2.
2. Abdul Saboor, Muhammad Usman, Sikandar Ali, Ali Samad, Muhammad Faisal Abrar and Najeeb Ullah. (2022). "A Method for Improving Prediction of Human Heart Disease Using Machine Learning Algorithms", Mobile Information Systems, 2022(1), 1410169. DOI: 10.1155/2022/1410169.
3. J. P. Li, A. U. Haq, S. U. Din, J. Khan, A. Khan and A. Saboor. (2020). "Heart disease identification method using machine learning classification in E healthcare", IEEE Access, 8(1), 107562-107582. DOI: 10.1109/ACCESS.2020.3001479.
4. U. N. Dulhare. (2018). "Prediction system for heart disease using naïve bayes and particle swarm optimization", Biomedical Research, 29(12), 2646-2649. DOI: 10.4066/biomedicalresearch.29-18-1249.
5. S. Mohan, C. Thirumalai and G. Srivastava. (2019). "Effective heart disease prediction using hybrid machine learning techniques", IEEE Access, 7(1), 81542 81554. DOI: 10.1109/ACCESS.2019.2923707.
6. S. Nashif, Md. R. Raihan, Md. R. Islam and M. H. Imam. (2018). "Heart Disease Detection by Using Machine Learning Algorithms and a Real-Time Cardiovascular Health Monitoring System", World Journal of Engineering and Technology, 6(4), 854-873. DOI: 10.4236/wjet.2018.64057.
7. A. K. Khan, R. K. Singh and M. A. Khan. (2022). "Heart disease prediction using ensemble learning", Journal of Medical Systems, 46(3), 20. DOI: 10.1007/s10916 022-01804-0.
8. I. M. Nasr, E. S. Darwish and N. F. Ibraheem. (2021). "Heart disease prediction based on hybrid machine learning techniques", Computers and Electrical Engineering, 91(1), 107055. DOI: 10.1016/j.compeleceng.2021.107055.