**A REAL-TIME / FIELD-BASED RESEARCH PROJECT REPORT ON**

Heart Disease Prediction

*in the partial fulfillment of the requirements for the award of the degree of*

**BACHELOR OF TECHNOLOGY**

in

**DEPARTMENT OF CSE (DATA SCIENCE)**

**Submitted by**

# B CHAKRIKA 23B81A6709

**A ESHWAR 23B81A6712 S MADHU SUDHAN 23B81A6720**

Under the guidance of

# Dr. S.V. Suryanarayana

**PROFESSOR & HEAD OF CSE(DS)**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)**

**CVR COLLEGE OF ENGINEERING**

**(*An Autonomous institution, NAAC Accredited and Affiliated to JNTUH, Hyderabad*)**

Vastunagar, Mangalpalli (V), Ibrahimpatnam (M), Rangareddy (D), Telangana- 501 510

**APRIL 2025**

CVR COLLEGE OF ENGINEERING

**(*An Autonomous institution, NAAC Accredited and Affiliated to JNTUH, Hyderabad*)**

Vastunagar, Mangalpalli (V), Ibrahimpatnam (M), Rangareddy (D), Telangana- 501 510

**DEPARTMENT OF CSE (DATA SCIENCE)**

****

**CERTIFICATE**

This is to certify that the Real-Time / Field-Based Research Project report entitled **“Heart Disease Prediction”** bonafide record of work carried out by **Chakrika Bandi(23B81A6709), Amaravadi Eshwar(23B81A6712), Shevva Madhu Sudhan Reddy(23B81A6720)** submitted to **Dr. S.V. Suryanarayana, Professor & Head of CSE (DS)** for the requirement of the award of **Bachelor of Technology** in **Department of Computer Science and Engineering (Data Science)** to the CVR College of Engineering, affiliated to Jawaharlal Nehru Technological University, Hyderabad during the year 2024-2025.

### Project Guide Project Coordinator

**Dr. S.V. Suryanarayana Dr. Shaik Janbhasha**

Professor & Head Associate Professor

Department of CSE (DS) Department of CSE (DS)

### Head of the Department External Examiner Dr. S.V. Suryanarayana

Professor & Head Department of CSE (DS)

 **CVR COLLEGE OF ENGINEERING**

**(*An Autonomous institution, NAAC Accredited and Affiliated to JNTUH, Hyderabad*)**

Vastunagar, Mangalpalli (V), Ibrahimpatnam (M) Rangareddy (D), Telangana- 501 510

# DECLARATION

We hereby declare that the Real-Time / Field-Based Research Project report entitled **“Heart Disease Prediction”** is an original work done and submitted to Computer Science & DataScience Department, CVR College of Engineering, affiliated to Jawaharlal Nehru Technological University Hyderabad in partial fulfilment for the requirement of the award of Bachelor of Technology in Computer Science and Engineering(Data Science) and it is a record of bonafide project work carried out by us under the guidance of **Dr. S.V. Suryanarayana, Professor & Head,** Department of CSE (Data Science).

We further declare that the work reported in this project has not been submitted, either in part or in full, for the award of any other degree in this Institute or any other Institute or University.

Signature of the Student

**B CHAKRIKA**

Signature of the Student

**A ESHWAR**

### Date:

**Place**: Hyderabad

Signature of the Student

**S MADHU SUDHAN REDDY**

**ACKNOWLEDGEMENT**

We are thankful and fortunate enough to get constant encouragement, support, and guidance from all the Teaching staff of **the CSE (Data Science) Department,** who helped us in successfully completing this Real-Time / Field-Based Research Project.

We are extremely thankful to our internal guide, **Dr. S.V. Suryanarayana, Professor, Head,** Department of CSE (Data Science) for providing support and guidance, which enabled us to complete the Project on time.

We thank **Dr.Shaik Janbhasha, Dr.Afreen Fatima Mohammed**, and the Project Review Committee members for their valuable guidance and support, which helped us to complete the Real-Time / Field-Based Research Project Work successfully.

We thank Professor, Head of the Department CSE (Data Science), **Dr. S. V. Suryanarayana,** for giving us all the support and guidance, which made us complete the Real- Time / Field-Based Research Project duly.

Our gratitude also extends to **Dr. Lakshmi H. N**, Associate Dean, Emerging Technologies, for her encouragement to complete the project work.

We thank our Vice-Principal, **Prof. L. C. Siva Reddy,** for providing excellent computing facilities and a disciplined atmosphere for doing our work.

We wish a deep sense of gratitude and heartfelt thanks to our Principal**, Dr.**

**Ramamohan Reddy,** and the **Management** for providing excellent lab facilities and tools.

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **Chapter No.** | | **Contents** | **Page No.** |
|  |  | List of Figures | i |
|  |  | List of Tables | ii |
|  |  | List of Symbols | iii |
|  |  | Abbreviations | iv |
|  |  | Abstract | 1 |
| 1 |  | **Introduction** | **2-4** |
|  | 1.1 | Motivation | 2 |
|  | 1.2 | Problem Statement | 3 |
|  | 1.3 | Project Objectives | 3-4 |
|  | 1.4 | Project Report Organization | 4 |
| 2 |  | **Literature Survey** | **5-7** |
|  | 2.1 | Existing work | 5-6 |
|  | 2.2 | Limitations of Existing Work | 7 |
| 3 |  | **Software & Hardware specifications** | **8-9** |
|  | 3.1 | Software requirements | 8 |
|  | 3.2 | Python Requirements & Libraries | 8 |
|  | 3.3 | Hardware requirements | 9 |
| 4 |  | **Proposed System Design** | **10-18** |
|  | 4.1 | Proposed method | 10-11 |
|  | 4.2 | System Architecture | 12 |
|  | 4.3 | Class Diagram | 13 |
|  | 4.4 | Use Case Diagram | 14 |
|  | 4.5 | Activity Diagram | 15 |
|  | 4.6 | Sequence Diagram | 16 |
|  | 4.7 | Technology Description | 17-18 |
| 5 |  | **Implementation & Testing** | **19-30** |
|  | 5.1 | Frontend Implementation | 19 |
|  | 5.2 | Data Preparation | 20-22 |
|  | 5.3 | Data Attributes | 22-23 |
|  | 5.4 | Model Selection and Training | 23-30 |
| 6 |  | **Conclusion & Future Scope** | **31-32** |
|  |  | **References** | **33** |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Title** | **Page No.** |
| 4.2 | System Architecture | 12 |
| 4.3 | Class Diagram | 13 |
| 4.4 | Use Case Diagram | 14 |
| 4.5 | Activity Diagram | 15 |
| 4.6 | Sequence Diagram | 16 |
| 5.1 | Frontend Implementation | 19 |
| 5.2.1 | Naming Columns | 20 |
| 5.2.2 | Feature Selection | 21 |
| 5.2.3 | Encoding & Scaling | 21 |
| 5.2.4 | Training & Testing | 22 |
| 5.4.1 | Logistic Regression | 24 |
| 5.4.2 | Random Forest | 25 |
| 5.4.3 | Ada Boosting | 26 |
| 5.4.4 | Stacking Classifier | 27 |
| 5.4.5 | User Interface Code | 28 |
| 5.4.6 | Test Input 1 | 29 |
| 5.4.7 | Test Input 2 | 30 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Title** | **Page No.** |
| 3.1 | Software Requirements | 8 |
| 3.2 | Python Environment & Libraries | 8 |
| 3.3 | Hardware Requirements | 9 |
| 5.3 | Dataset Attributes | 22-23 |

**LIST OF SYMBOLS**

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Meaning** | **Context** |
| x̄ | Mean | Used in standardization (e.g., scaling  features like age, thalach) |
| σ | Standard Deviation | Scaling factor in StandardScaler during  normalization |
| 0 / 1 | Binary Values | Represented in features like sex (0=Female,  1=Male), target (0=No, 1=Yes) |
| ? | Missing Value Placeholder | Marker in original dataset (replaced with  NaN during preprocessing) |
| C | Inverse Regularization Strength | Hyperparameter in Logistic Regression  (default: C = 1.0) |

**ABBREVIATIONS**

|  |  |
| --- | --- |
| **Abbreviation** | **Full Form** |
| ML | Machine Learning |
| AI | Artificial Intelligence |
| UCI | University of California, Irvine |
| ANN | Artificial Neural Network |
| GUI | Graphical User Interface |

**ABSTRACT**

This project investigates the application of machine learning techniques to predict the risk of heart disease by integrating structured clinical data with real-time user input. The primary dataset used is the well-known UCI Cleveland Heart Disease Dataset, which encompasses key clinical features such as age, gender, blood pressure, cholesterol levels, maximum heart rate, chest pain type, fasting blood sugar, ECG results, exercise-induced angina, and ST depression metrics.

In contrast to traditional models that rely on predefined rules or binary classifications (e.g., "high risk" or "low risk"), this system generates a personalized probability score, providing a more nuanced and accurate assessment of cardiovascular risk. Although the model does not incorporate lifestyle factors or family medical history, it emphasizes clinically significant variables directly associated with heart disease outcomes.

The system produces an output indicating whether the user is at risk of heart disease, thereby enabling early detection and facilitating personalized healthcare decisions. By offering accessible, data-driven insights through a user-friendly interface, this project contributes to preventive healthcare efforts, empowers individuals to make informed choices, and supports the broader goal of reducing the prevalence and impact of heart disease through early intervention and awareness.

**CHAPTER 1 INTRODUCTION**

# MOTIVATION:

The rising global burden of cardiovascular disease underscores the need for more accurate, accessible, and personalized heart disease prediction tools. Despite technological advancements, many existing models rely on outdated or overly simplistic methods that fail to reflect the complexities of individual health profiles.

Traditional systems often use broad classifications and lack the ability to deliver personalized risk assessments based on a user's unique clinical indicators. Moreover, their results can be difficult for non-medical users to interpret, limiting their practical use for early intervention.

This project seeks to address these challenges by developing a machine learning-based prediction system trained on the UCI Cleveland Heart Disease Dataset. It focuses on objective clinical features such as age, cholesterol, blood pressure, heart rate, and ECG results to produce a clear, probability-based risk score that is both accurate and easy to understand.

By combining clinical precision with user-friendly design, the system aims to empower individuals with meaningful insights into their heart health—supporting early detection, informed decision-making, and broader preventive healthcare efforts.

# PROBLEM STATEMENT:

Cardiovascular disease remains a leading cause of death worldwide, highlighting the need for early and accurate risk assessment to improve prevention and timely treatment. While several heart disease prediction models exist, many rely on generalized assumptions and fail to adapt to individual health profiles, limiting their accuracy and usefulness in personalized care.

Traditional models often use basic clinical indicators like age, cholesterol, blood pressure, and heart rate, but lack the flexibility of data-driven approaches that can provide tailored insights. Additionally, many tools present risk in broad categories such as "low" or "high," rather than offering clear probability scores that help users better understand their specific risk.

This project addresses these gaps by developing a machine learning-based model trained on the UCI Cleveland Heart Disease Dataset. It uses detailed clinical features such as ECG results, exercise-induced angina, and ST depression to generate personalized, probability-based risk scores. The goal is to deliver an intuitive and reliable tool that supports early detection, informed decisions, and improved preventive care.

# PROJECT OBJECTIVES:

The primary objective of this project is to design and develop a robust machine learning-based application that can accurately predict the likelihood of heart disease in individuals based on clinical health indicators. By leveraging real-world medical data and advanced predictive algorithms, the project seeks to create a reliable, interpretable, and user-friendly tool that facilitates early detection and informed health decisions. The specific goals of the project are outlined in detail below:

* + - To develop a heart disease risk prediction system that takes basic health-related inputs from the user and provides a clear output indicating the presence or absence of heart disease risk.
    - To deliver results in an intuitive format, clearly stating whether the user is at risk or not based on predefined data-driven logic.
    - To assist users in understanding their heart health status easily, without needing complex medical interpretation.
    - To support early awareness and health monitoring by offering a simple, user-focused interface.
    - To demonstrate the application of machine learning in healthcare prediction through a straightforward, input-output-based system.

# PROJECT REPORT ORGANISATION:

The project report is structured into well-defined chapters to present the development process, methodology, and evaluation of the heart disease prediction system. The structure is as follows:

* + - **Chapter 1 – Introduction:** Outlines the background, motivation, problem statement, and objectives of the heart disease prediction project.
    - **Chapter 2 – Literature Review:** Discusses existing heart disease prediction models, their limitations, and the need for a more personalized, machine learning-based approach.
    - **Chapter 3 – Software and Hardware Specifications:** Lists the tools, technologies, programming languages, and system requirements used during development.
    - **Chapter 4 – System Design:** Describes the overall system architecture, prediction methodology, and includes relevant diagrams such as use case, class, activity, and sequence.
    - **Chapter 5 – Implementation and Testing:** Explains how the system was developed and tested, including user interface screenshots and model performance metrics like accuracy and precision.
    - **Chapter 6 – Conclusion and Future Scope:** Summarizes the key outcomes of the project and suggests future improvements, such as incorporating lifestyle data and enhancing prediction accuracy.

This structured organization helps readers follow the complete development process— from identifying the problem to building and evaluating the final system.

**CHAPTER 2 LITERATURE SURVEY**

# EXISTING WORK:

### Framingham Risk Score (FRS)

The Framingham Risk Score is one of the earliest and most widely used tools for estimating the 10-year risk of developing cardiovascular disease. Developed from data obtained during the long-term Framingham Heart Study, this model considers factors such as age, sex, total and HDL cholesterol, blood pressure, smoking status, and diabetes. It calculates a risk percentage that helps clinicians classify patients into low, intermediate, or high-risk categories. While clinically useful, its generalization across diverse populations has been criticized. The original methodology and validation of this risk score were documented in the paper titled *“General cardiovascular risk profile for use in primary care: the Framingham Heart Study”* by D’Agostino et al. (2008), published in *Circulation*. This study helped establish a foundation for statistical modelling in preventive cardiology, although it lacks consideration of ethnicity, menopause, and other nuanced risk factors.

### Heart Disease Prediction Using Machine Learning

The study focuses on applying machine learning techniques to predict heart disease based on user diagnostic inputs, aiming for early detection to reduce mortality rates. It uses structured clinical features like age, sex, chest pain type, cholesterol, and maximum heart rate to predict the presence of heart disease. Three key machine learning algorithms were implemented and evaluated: Decision Tree, Random Forest, and K-Nearest Neighbors (KNN), achieving maximum model accuracy of 86.9% with KNN. The project involved building a web-based system using Python, Flask, and SQLite, allowing users to register, input their diagnostic values, and receive instant predictive results. Comprehensive software engineering practices were employed, including system architecture diagrams, sequence diagrams, and thorough validation testing.

### MyHeartDiseaseRisk.com

MyHeartDiseaseRisk.com is a publicly accessible online tool developed by researchers at the Harvard T.H. Chan School of Public Health to estimate an individual's risk of developing heart disease. Designed with a focus on public health education, the tool allows users to input basic health and lifestyle information such as age, cholesterol levels, blood pressure, smoking habits, and physical activity. It then calculates a simplified risk estimate along with lifestyle recommendations for prevention. While the interface is intuitive and informative, the model behind it has significant limitations. It does not consider granular clinical inputs such as electrocardiogram (ECG) readings, exercise-induced angina, or other advanced diagnostic data. Moreover, it operates independently of real-time patient records or electronic health records (EHRs), restricting its practical use in medical settings. The tool’s generalized nature makes it useful for awareness but less suitable for accurate clinical diagnosis or personalized risk prediction, especially across diverse populations or those with complex medical histories.

### The NHS Heart Age Tool

It is an initiative developed by Public Health England in collaboration with the British Heart Foundation and other UK health organizations. Aimed at raising awareness about cardiovascular health, the tool calculates a user’s “heart age” by analyzing factors such as age, sex, weight, blood pressure, cholesterol levels, and lifestyle habits like smoking. While it serves as an engaging and educational resource to encourage preventive health measures among the general public, the tool has notable limitations. It does not offer detailed medical analysis or clinical decision-making support. The absence of integration with electronic health records (EHRs) and lack of real-time data processing restrict its use to surface-level awareness rather than personalized care. Additionally, it does not recommend tailored medical interventions or consider more complex factors such as family history, comorbidities, or genetic predispositions. As a result, the NHS Heart Age Tool remains a public health resource rather than a diagnostic tool, with limited clinical applicability in personalized treatment planning.

# LIMITATIONS OF EXISTING WORK:

### Binary Classification Output

Most traditional models provide a simple "Yes" or "No" result for heart disease presence. This lacks nuance and does not help users understand the severity or likelihood of risk.

### Outdated Machine Learning Techniques

These models primarily rely on basic statistical methods rather than advanced machine learning algorithms, limiting their ability to capture complex patterns in data.

### Dataset Bias and Limited Diversity

Datasets used are often skewed towards specific ethnicities or demographics, making them less effective for diverse or global populations.

### Categorical Risk Levels

Instead of giving a personalized risk percentage, many systems classify users into rigid categories like “low,” “medium,” or “high” risk, which may not reflect the true health condition.

### Static Rule-Based Models

Fixed clinical rules prevent these models from evolving over time. They cannot learn from new patient data or adapt to emerging health patterns.

### Missing Diagnostic Features

Critical indicators such as ECG results, chest pain types, or ST-segment abnormalities are often excluded, which may weaken diagnostic capability.

### Outdated Data Sources

Many datasets were collected decades ago and may not reflect current medical standards, lifestyle habits, or treatment methods.

* + 1. **Poor Accuracy**

Several early machine learning approaches showed limited predictive accuracy, reducing their clinical usefulness and reliability for decision-making.

**CHAPTER 3**

**SOFTWARE AND HARDWARE SPECIFICATIONS**

# SOFTWARE REQUIREMENTS

**Table 3.1: Software Requirements**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Category** |  | **Technology/Component** | **Version** | **Purpose** |
| **Backend** |  | Python | 3.8+ | Core programming language for  ML and backend logic |
|  |  | Gradio | ≥3.0 | Rapid UI development for ML model deployment and testing |
| **Model Artifacts** |  | UCI Cleveland Dataset | - | Heart disease dataset used for  training/testing |
| **Dev Tools** |  | Jupyter Notebook / VS  Code | Latest | IDEs used for code writing,  execution, and testing |
|  |  | Google Colab | - | Cloud-based environment with  GPU support |

# PYTHON REQUIREMENTS & LIBRARIES

**Table 3.2: Python Requirements & Libraries**

|  |  |  |
| --- | --- | --- |
| **Library** | **Version** | **Purpose** |
| **NumPy** | ≥1.18 | Numerical operations |
| **Pandas** | ≥1.1 | Data handling and manipulation |
| **Scikit-learn** | ≥0.24 | Machine learning algorithms and evaluation |
| **Matplotlib** | ≥3.3 | Basic plotting and charts |
| **Gradio** | ≥3.0 | Build interactive interfaces for model inference |

# Key Notes:

* + - Python Environment: Python 3.x with essential libraries (Pandas, NumPy, scikit-learn).
    - Dependencies:
      * pandas for data manipulation.
      * scikit-learn for preprocessing (LabelEncoder, StandardScaler).
      * joblib for saving/loading scaler models.
    - Google Colab/Notebook: The code is designed for Jupyter/Colab environments (Google Drive integration for file I/O).
    - Data Format: CSV files with specific column names (e.g., age, thalach, target).

# HARDWARE REQUIREMENTS

**Table 3.3: Hardware Requirements**

|  |  |  |
| --- | --- | --- |
| **Component** | **Specification** | **Purpose** |
| **Processor (CPU)** | Intel i5 / AMD Ryzen 5 or higher | For efficient data processing and ML model execution |
| **RAM** | Minimum 8 GB  (Recommended: 16 GB) | Smooth handling of datasets, model training, and app runtime |
| **Storage** | Minimum 500 MB free (Recommended: SSD) | To store datasets, models, and environment files |
| **GPU**  **(Optional)** | NVIDIA GPU with CUDA  support | Accelerates model training (if using deep learning later) |
| **Display** | 1080p resolution or higher | For better visual rendering of Streamlit app and plots |
| **Internet Access** | Required (for package installations/API) | To install dependencies and optionally deploy the web app |

# Key Notes:

Storage: Sufficient space to store datasets (e.g., cleveland\_read.csv, final\_cleveland.csv). Memory: Minimum 4GB RAM for handling moderate-sized datasets (297 samples post- cleaning).

CPU: Standard multi-core processor for preprocessing tasks (scaling, encoding).

GPU (Optional): Not mandatory but beneficial for large-scale ML tasks (not heavily utilized in this notebook).

**CHAPTER 4 PROPOSED SYSTEM DESIGN**

# PROPOSED METHOD:

The system takes user inputs related to basic health parameters and uses a pre-trained machine learning model to determine whether the user is at risk of heart disease or not. The output is displayed as either "Risk Detected" or "No Risk," making it easy to understand.

### Data Utilization and Integration

The core of the system’s predictive capabilities is based on the UCI Cleveland Heart Disease Dataset, a widely recognized benchmark in medical data science. The dataset includes vital clinical attributes such as age, gender, resting blood pressure, serum cholesterol, maximum heart rate achieved, chest pain type, fasting blood sugar, resting ECG results, exercise-induced angina, ST depression, and slope of peak exercise. This structured data is pre-processed and normalized to ensure high-quality input to the machine learning model. Additionally, users can manually input their health parameters through a web interface, which feeds the model with real-time values for on-demand predictions.

### Modules and Functional Features

* + - 1. **User Authentication & Data Input**

A secure and simple authentication module allows users to log in and manage their profiles. After logging in, users input their medical parameters through a guided form. The interface ensures all necessary fields are filled and validated before processing.

### Machine Learning-Based Risk Prediction

A supervised machine learning model, trained on the UCI dataset, forms the backbone of the system. Algorithms such as **Logistic Regression**, **Random Forest**, **AdaBoosting**, and **Stacking** are employed to calculate the probability of heart disease occurrence. The model predicts a risk score in percentage format, rather than assigning rigid categories like "low", "moderate", or "high".

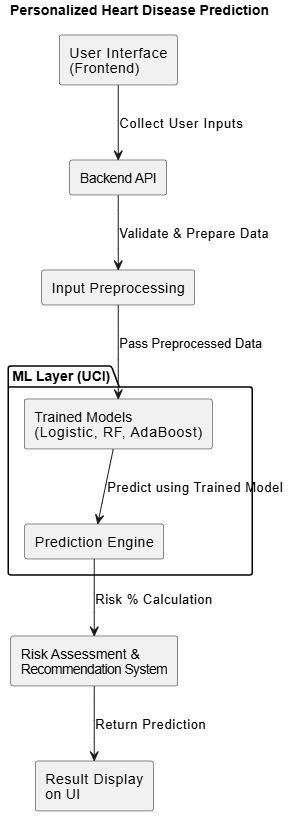
### Personalized Risk Assessment Interface

To improve user engagement, results are presented through interactive charts and visual indicators (e.g., risk meters or coloured bars) that visually communicate how close a user might be to a high-risk zone. This makes medical predictions more accessible to the public.

### Advantages of the Proposed System

* Personalized Risk Output: Provides users with a numerical probability instead of a vague classification, allowing better interpretation of health status.
* Improved Accuracy: The use of a real-world clinical dataset and standard ML algorithms ensures credible and statistically sound predictions.
* User-Centric Visualization: The risk is presented through easy-to-understand visuals, improving user comprehension, especially for non-technical individuals.
* Web-Based Platform: Built using Gradio, the interface ensures ease of use across various devices with no need for complex installation or technical setup.
* Scalability and Modularity: The system’s modular design allows for easy enhancement, including future integration of additional datasets or health parameters.

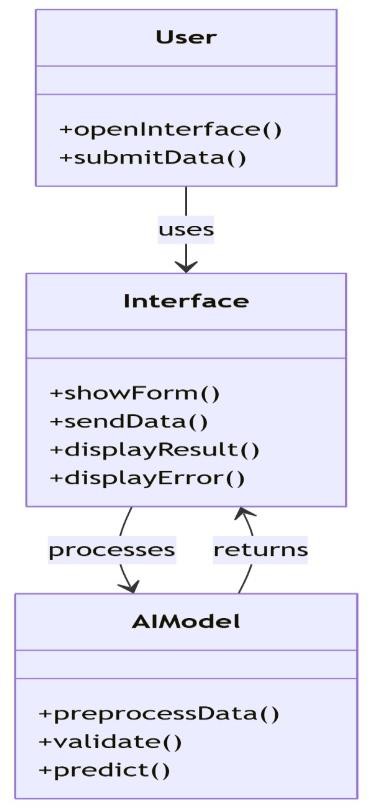
# SYSTEM ARCHITECTURE:

****

## Figure 4.2: System Architecture

**Description**: A System Architecture diagram provides a high-level view of the components of a system and how they interact with each other. It outlines the structure of the system, including hardware, software, and network components, and illustrates how they work together to fulfill the system's requirements. For the heart disease prediction system, the diagram highlights how user input, machine learning algorithms, and backend processes work together to generate an accurate risk assessment. It visualizes the flow of data from user inputs to prediction outputs, ensuring seamless interaction between each component.

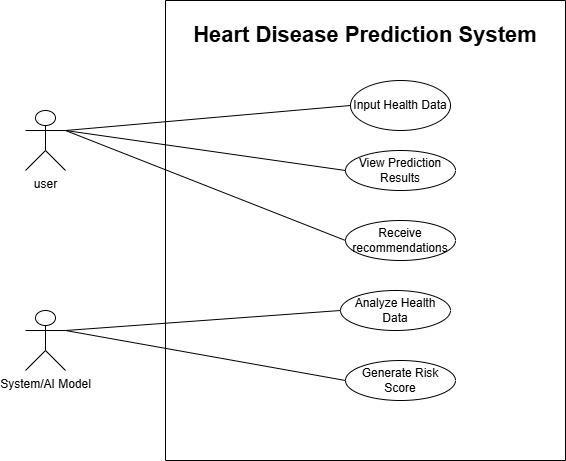
# CLASS DIAGRAM:

****

## Figure 4.3: Class Diagram

**Description:** A class diagram is a type of static structure diagram that represents the classes in a system and their relationships. The Heart Disease Prediction System utilizes a streamlined class structure with three core components: the User initiates interactions through opening interfaces and submitting health data, the Interface handles form presentation, data transmission, and result/error display, while the AI Model manages data preprocessing, validation, and predictive analysis. This architecture enables efficient data flow from user input collection through predictive processing, ultimately delivering risk assessment results back to the user interface. The system's design emphasizes clear separation of concerns, with the Interface mediating between user interactions and the AI Model's computational capabilities for heart disease risk prediction.

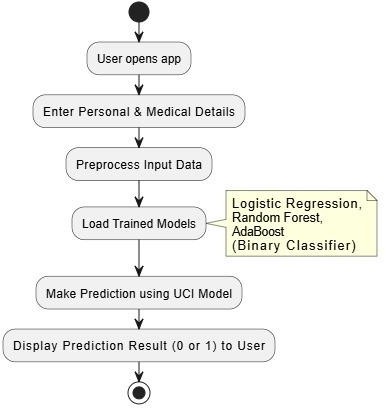
# USE CASE DIAGRAM:

****

## Figure 4.4: Use Case Diagram

**Description:** A use case diagram is a visual representation of the interactions between users (actors) and the system. It helps to identify the functional requirements of the system by illustrating the various ways users can interact with it. Above is a detailed explanation of the components and structure of a use case diagram for the Heart Disease Prediction System in Python.

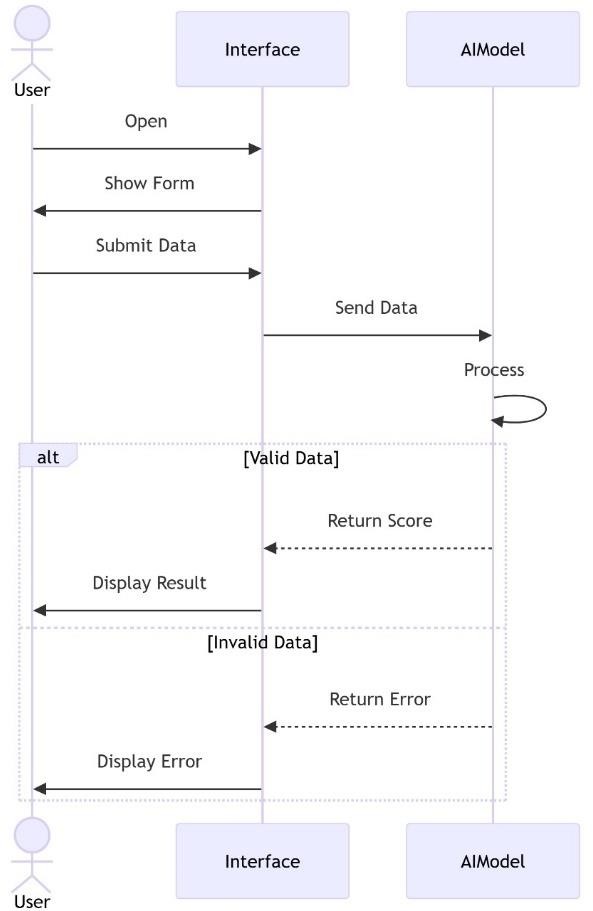
# ACTIVITY DIAGRAM:

****

## Figure 4.5: Activity Diagram

**Description:** An activity diagram is a type of UML (Unified Modeling Language) diagram that represents the flow of activities or actions in a system. It is particularly useful for modeling the dynamic aspects of a system, such as workflows and processes. In the context of a heart disease prediction system in Python using machine learning, the activity diagram illustrates how users input health data, which is processed by ML models to generate risk scores. It visualizes the workflow from data collection through analysis to result display, showing the system's predictive pipeline.

# SEQUENCE DIAGRAM:

****

## Figure 4.6: Sequence Diagram

**Description**: A sequence diagram is a type of UML (Unified Modeling Language) diagram that illustrates how objects interact in a particular scenario of a use case. It shows the sequence of messages exchanged between objects over time, providing a clear view of the flow of control and data in a system. In the context of a heart disease prediction system using Python/ML, a sequence diagram illustrates the interaction flow between the user interface, backend API, and ML model to process health inputs and generate risk predictions. It captures the data journey from submission through preprocessing, model inference, to displaying the final risk assessment.

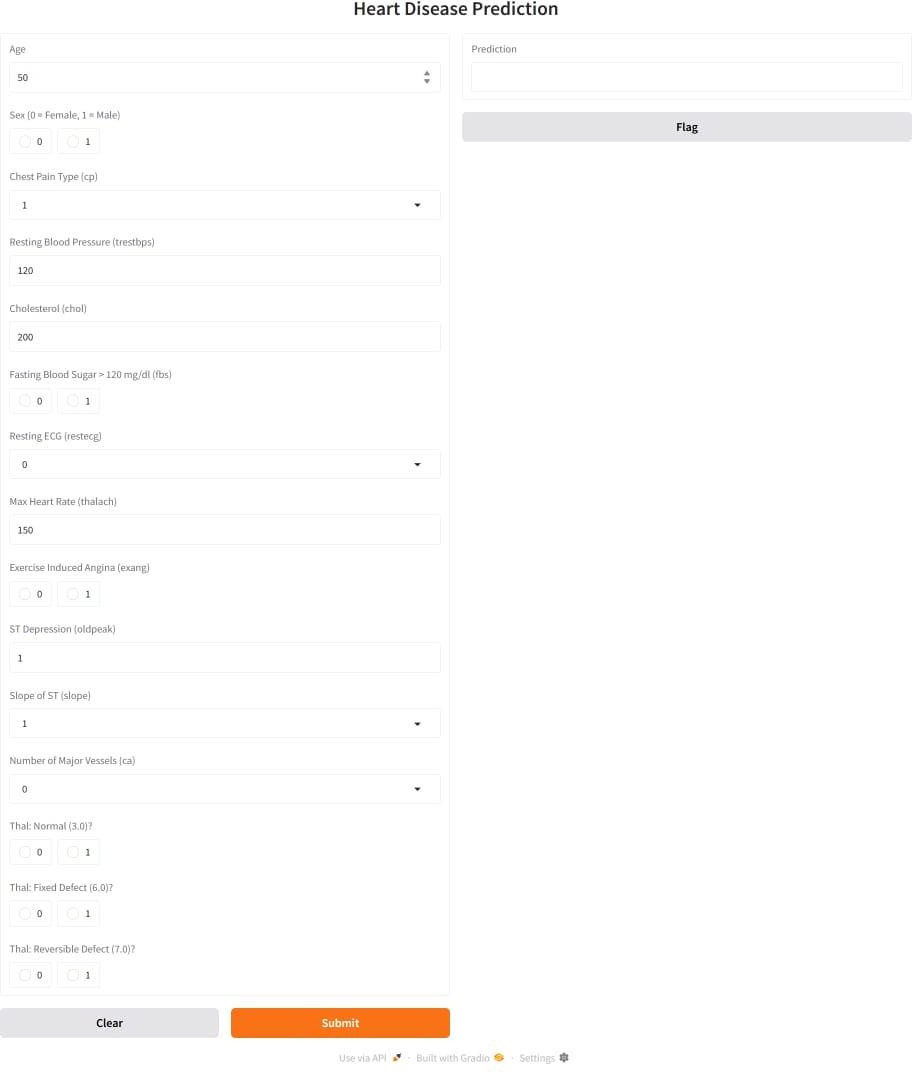
* 1. **TECHNOLOGY DESCRIPTION:**

|  |  |
| --- | --- |
| Python | Core programming language used throughout the project for data preprocessing, machine learning, and deployment logic. |
| Google Colab | Cloud-based development platform used for writing, testing, and running Python notebooks with free GPU/TPU access. |
| Pandas | Used for reading, transforming, and exporting the dataset during preprocessing. |
| NumPy | Supports numerical operations and assists in handling data arrays within Pandas dataframes. |
| Scikit-learn | Provides comprehensive tools for ML modeling, feature transformation, model selection, and evaluation. |
| LabelEncoder | Encodes categorical variables (e.g., slope, ca) into numerical values for model compatibility. |
| One-Hot Encoding | Converts multi-category variables (e.g., cp, thal) into binary columns using pd.get\_dummies(). |
| StandardScaler | Standardizes continuous features (e.g.,  age, thalach, chol, oldpeak) to improve model performance. |

|  |  |
| --- | --- |
| Missing Value Handling | Cleans the dataset by identifying and removing or replacing missing or invalid values (e.g., ?). |
| Logistic Regression | A simple, interpretable model used for binary classification of heart disease presence. |
| Random Forest | An ensemble learning method using decision trees to improve prediction accuracy and handle overfitting. |
| AdaBoosting | A boosting technique that adjusts weights on misclassified examples to build a stronger classifier. |
| Stacking | Combines multiple ML models (base learners) and a meta-learner to enhance overall predictive performance. |
| Joblib | Serializes trained objects (like scalers and models) into .pkl files for future prediction without retraining. |
| Gradio | Web interface framework used to create an interactive, real-time prediction UI for end users. |
| CSV Dataset (UCI Cleveland) | Publicly available dataset containing clinical attributes; used for training, testing, and evaluating heart disease risk  prediction. |

**CHAPTER 5 IMPLEMENTATION & TESTING**

# FRONTEND IMPLEMENTATION

****

## Figure 5.1: Frontend Implementation

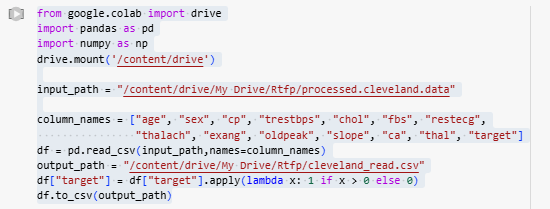
This Gradio-based interface allows users to input key health metrics such as age, cholesterol, blood pressure, and more to assess heart disease risk. Upon submission, the system processes the inputs through a trained ML model and displays the probability (0–100%) of heart disease.

# DATA PREPARATION

The dataset used for this project was imported from a CSV file named heart\_disease\_data.csv. It contains several key medical and demographic attributes, including age, sex, chest pain type, cholesterol level, blood pressure, and more. Before training the machine learning model, the following preprocessing steps were performed:

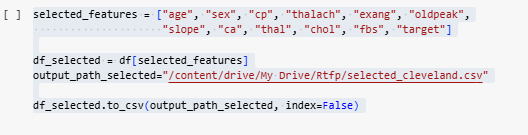
* + - **Naming Columns**: The Cleveland Heart Disease dataset was loaded from a data file using Pandas. As the dataset had no header row, appropriate column names were manually assigned. These names reflect key medical attributes relevant to heart disease diagnosis. To simplify analysis, the target column was converted into a binary format

— 1 for the presence and 0 for the absence of heart disease. This transformation enables the use of binary classification models. The cleaned and labelled dataset was then saved for the next preprocessing steps.



## Figure 5.2.1: Naming Columns

* + - **Feature Selection**: A subset of key features was manually selected based on domain knowledge and relevance to heart disease prediction.

These selected features were saved as a new dataset for focused preprocessing and model training.

## Figure 5.2.2: Feature Selection

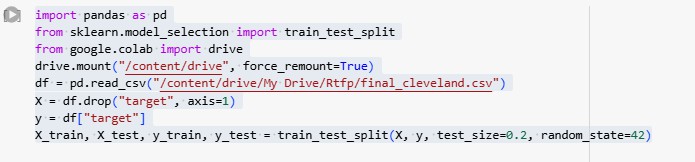
* + - **Encoding and Scaling :** To prepare the data for modeling, categorical and numerical features were processed. One-hot encoding was applied to cp and thal to convert them into binary columns. Label Encoder was used on slope and ca if they were detected as object types. Numerical features like age, thalach, oldpeak, and chol were scaled using StandardScaler.This normalization ensures all features contribute equally during training.A check was added to remove any ambiguous columns like thal\_? if present. The cleaned and transformed dataset was saved as final\_cleveland.csv.This step finalized the dataset for machine learning model training.



## Figure 5.2.3: Encoding & Scaling

The dataset was split into training and testing sets using an 80:20 ratio to evaluate model performance.

Features (X) and target (y) were separated, and train\_test\_split ensured reproducibility with random\_state=42.



**Figure 5.2.4: Training & Testing**

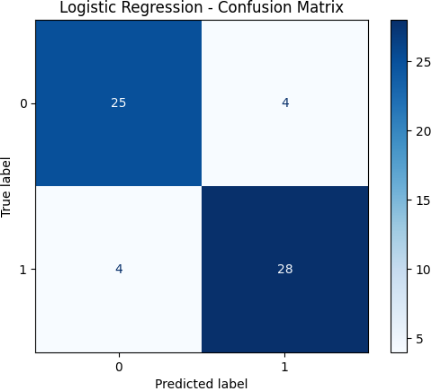
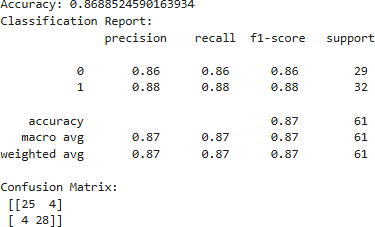
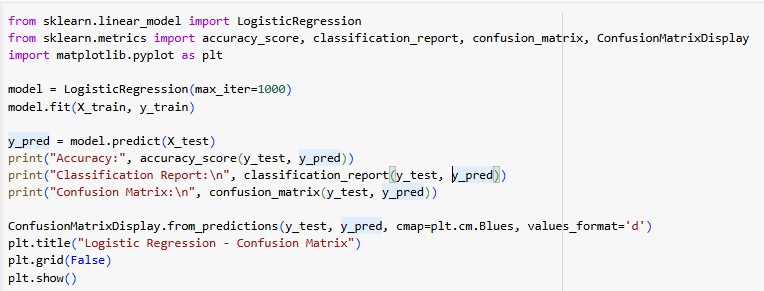
# DATA ATTRIBUTES:

**TABLE 5.3: Data Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Description** | **Preprocessing Applied** |
| **age** | Numerical (float) | Age of the patient | Standardized (Scaled) |
| **sex** | Binary (0/1) | Gender (0: Female, 1: Male) | No transformation |
| **cp\_1.0** | Binary (0/1) | Chest pain type (Typical angina) | One-Hot Encoded (from cp) |
| **cp\_2.0** | Binary (0/1) | Chest pain type (Atypical angina) | One-Hot Encoded (from cp) |
| **cp\_3.0** | Binary (0/1) | Chest pain type (Non-anginal pain) | One-Hot Encoded (from cp) |
| **cp\_4.0** | Binary (0/1) | Chest pain type (Asymptomatic) | One-Hot Encoded (from cp) |
| **thalach** | Numerical (float) | Maximum heart rate achieved | Standardized (Scaled) |

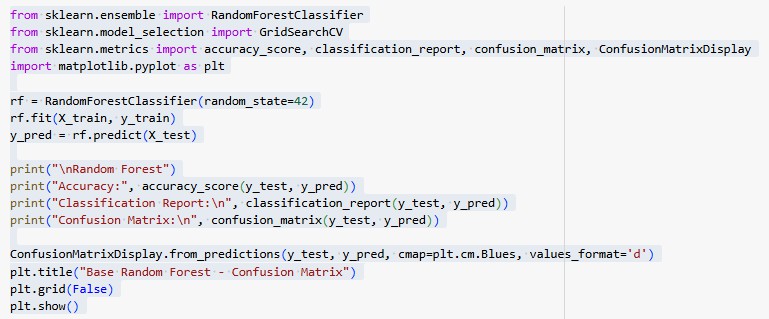
|  |  |  |  |
| --- | --- | --- | --- |
| **exang** | Binary (0/1) | Exercise-induced angina (0:  No, 1: Yes) | No transformation |
| **oldpeak** | Numerical (float) | ST depression induced by exercise | Standardized (Scaled) |
| **slope** | Categorical (int) | Slope of the peak exercise ST segment | Label Encoded |
| **ca** | Categorical (int) | Number of major vessels (0– 3) colored by fluoroscopy | Label Encoded |
| **thal\_3.0** | Binary (0/1) | Thalassemia type (Normal) | One-Hot Encoded (from thal) |
| **thal\_6.0** | Binary (0/1) | Thalassemia type (Fixed defect) | One-Hot Encoded (from thal) |
| **thal\_7.0** | Binary (0/1) | Thalassemia type (Reversible defect) | One-Hot Encoded (from thal) |
| **chol** | Numerical (float) | Serum cholesterol (mg/dL) | Standardized (Scaled) |
| **fbs** | Binary (0/1) | Fasting blood sugar > 120 mg/dL (0: No, 1: Yes) | No transformation |

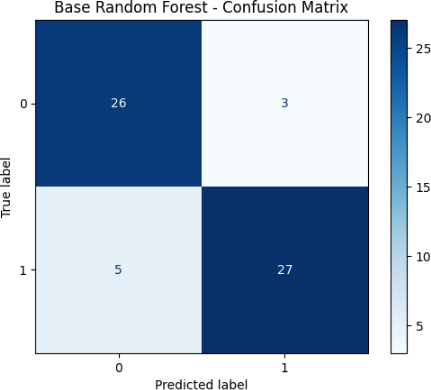
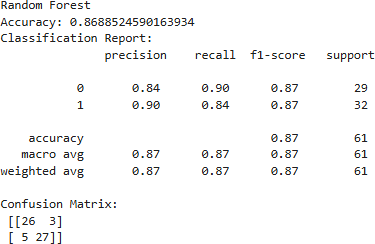
# MODEL SELECTION AND TRAINING

To predict heart disease, multiple models were evaluated for their accuracy and performance.

## Figure 5.4.1: Logistic Regression

**Logistic Regression** was first applied as a baseline due to its simplicity and interpretability. It provides a solid foundation for binary classification tasks and helps benchmark the performance of more complex models. Additionally, it offers easily explainable results through coefficients and statistical metrics like accuracy, classification report, and confusion matrix.

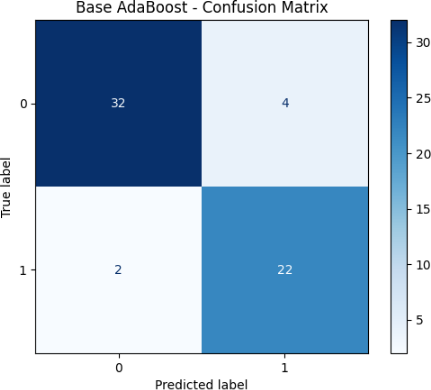
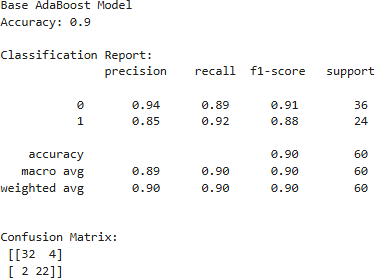




## Figure 5.4.2: Random Forest

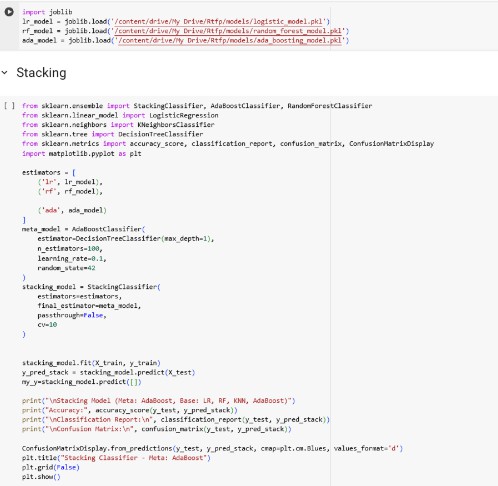
**Random Forest** Classifier was chosen for its robustness and ability to handle feature interactions and non-linearity in the data. It leverages an ensemble of decision trees, reducing the risk of overfitting and improving generalization. Its performance was evaluated using accuracy, classification report, and a confusion matrix to validate its effectiveness.

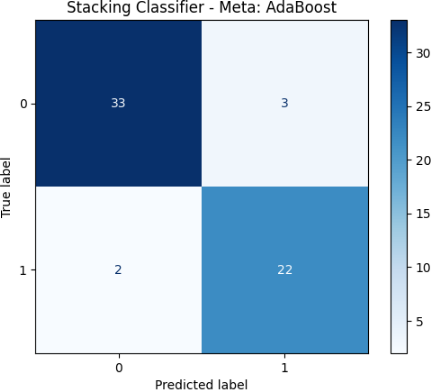
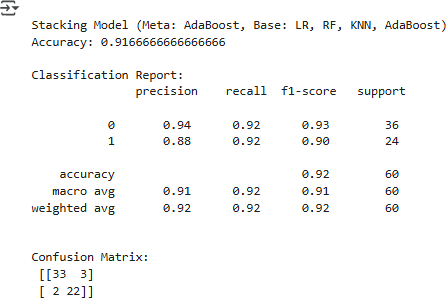




## Figure 5.4.3: Ada Boosting

**AdaBoost Classifier** with shallow decision trees as base estimators was then used to boost model accuracy by combining multiple weak learners. This technique emphasizes harder-to- classify instances by adjusting weights iteratively, making it effective for imbalanced datasets. The model’s performance was validated through accuracy, a detailed classification report, and a confusion matrix to assess predictive reliability.





## Figure 5.4.4: Stacking Classifier

**A Stacking Classifier** was implemented to enhance prediction by combining the strengths of multiple models. Logistic Regression, Random Forest, and AdaBoost were used as the base learners. An additional AdaBoost model acted as the meta-learner, learning from the outputs of the base models. This ensemble approach aimed at improving generalization and overall accuracy. Cross-validation was used to ensure robustness during stacking.

A screen shot of a computer

AI-generated content may be incorrect.

## Figure 5.4.5: User Interface Code

A web-based heart disease prediction tool is developed using Gradio, allowing users to input medical details and receive real-time risk predictions. It uses a pre-trained stacked machine learning model and a scaler to process inputs like age, sex, cholesterol, and heart rate. User inputs are collected via interactive widgets, scaled, and fed into the model. The predicted probability is then used to display a message indicating high or low heart disease risk offering a simple and effective way to apply ML in healthcare.

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 5.4.6: Test Input 1**

A screenshot of a computer

AI-generated content may be incorrect.

**Figure 5.4.7: Test Input 2**

**CHAPTER 6 CONCLUSION & FUTURE SCOPE**

# CONCLUSION

The Heart Disease Prediction system utilizes machine learning models to assess a user’s risk of heart disease based on clinical input parameters. Built with a Python backend, the system incorporates data preprocessing, model training (including Logistic Regression, Random Forest, and AdaBoost), and performance evaluation through accuracy, classification reports, and confusion matrices. With a user-friendly interface built using Gradio, the application provides an interactive way for individuals to receive personalized risk predictions. This project effectively combines healthcare data science, interpretability, and web deployment to support early detection and awareness.

This project showcases how machine learning can be harnessed to deliver impactful solutions in the healthcare domain. By training and evaluating multiple models on medical data, it demonstrates the potential of ML to detect patterns and predict health risks with high accuracy. The use of supervised learning techniques provides a reliable framework for risk assessment and early intervention.

# FUTURE SCOPE

### Integration with Wearable Devices

Connect with fitness trackers and smartwatches to collect real-time health data like heart rate, blood pressure, and physical activity for more dynamic predictions.

### Patient Health Record Integration

Integrate with Electronic Health Record (EHR) systems to allow seamless access to medical history, improving prediction accuracy through richer data inputs.

### Real-Time Monitoring and Alerts

Enable continuous monitoring of patient vitals with automated alerts to notify users or healthcare providers when risk levels are elevated**.**

### Mobile Application Development

Develop a cross-platform mobile app using frameworks like Flutter to increase accessibility and enable health monitoring on the go**.**

### Explainable AI Integration

Implement explainability techniques (like SHAP or LIME) to help users and clinicians understand how each feature contributes to the risk prediction.

### Multi-Disease Prediction Capability

Extend the system to predict other chronic conditions (e.g., diabetes, hypertension) using a shared health profile, offering a comprehensive health risk assessment tool.

### Voice-Based Health Query System

Introduce a voice interface using speech recognition and NLP to allow users to interact with the system hands-free for a more inclusive experience.

# References:

### UCI Machine Learning Repository

Dua, D., & Graff, C. (2019). UCI Machine Learning Repository: Heart Disease Dataset. University of California, Irvine.

Available at: [https://archive.ics.uci.edu/ml/datasets/heart+Disease](https://archive.ics.uci.edu/ml/datasets/heart%2BDisease)

* + - * Efficient Prediction of Cardiovascular Disease Using Machine Learning Algorithms: Authored by Pronab Ghosh, Sami Azam, Mirjam Jonkman
      * An Efficient Computational Risk Prediction Model of Heart Diseases Based on Dual- Stage Stacked Machine Learning Approaches: Authored by Subhash Mondal, Ranjan Maity, Yachang Omo, Soumadip Ghosh, and Amitava Nag

### Gradio Interface Library

Abid, A. et al. Gradio: Create Machine Learning Web Apps in Python. Available at: [https://www.gradio.app](https://www.gradio.app/)

### Pandas: Data Analysis Library

The Pandas Development Team. Pandas Documentation. Available at: https://pandas.pydata.org/docs/