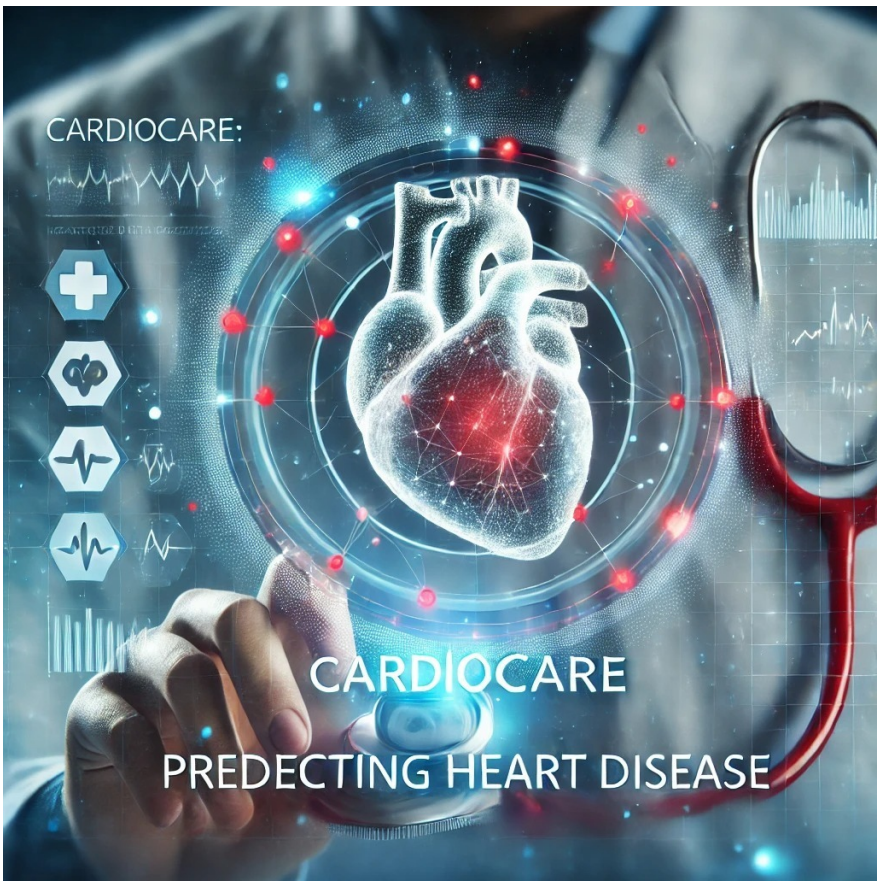


AI CardioCare: Intelligent Heart Disease Prediction and Diagnosis System

Submitted To: **Sir Samyan Qayyum Wahla**

Tayyaba Afzal (2022-CS-134)

Raveeha Mohsin (2022-CS-149)

DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF ENGINEERING & TECHNOLOGY
Lahore, Pakistan

AI Project Report 2024

AI CardioCare: Intelligent Heart Disease Prediction and Diagnosis System

Utilizing Machine Learning for Early Detection and Risk Assessment

Artificial Intelligence Project

University of Engineering & Technology
Department of Computer Science
Lahore, Pakistan

AI CardioCare: Intelligent Heart Disease Prediction and Diagnosis System
Tayyaba Afzal 2022-CS-134
Raveeha Mohsin 2022-CS-149
Department of Computer Science
University of Engineering & Technology, Lahore

Abstract

The "**AI CardioCare:** Intelligent Heart Disease Prediction and Diagnosis System" is an advanced healthcare solution designed to predict and diagnose heart disease while providing personalized recommendations for treatment and care. This project leverages the power of Artificial Intelligence (AI) to analyze both textual patient data and report imaging for accurate and early detection of heart conditions. The primary objective of the system is to classify patients into three categories: Yes (heart disease present), No (no heart disease), and Maybe (further investigation required). The project utilizes a neural network model for multi-class classification, ensuring accurate predictions based on the data collected from various hospitals across Pakistan. By integrating AI models, Optical Character Recognition (OCR) for document processing, and deep learning techniques like Neural Networks, the system ensures precise analysis and diagnosis. Key outcomes include enhanced healthcare accessibility, improved early detection rates, and actionable insights for medical professionals and patients. Aligned with Sustainable Development Goal 3, this project aims to transform heart disease diagnosis by promoting preventative care and reducing healthcare inequalities.

Keywords: Heart disease prediction, multi-class classification, neural networks, AI in healthcare, early disease detection, patient data analysis, SDG 3, medical data from Pakistani hospitals, AI-driven diagnosis.

Acknowledgements

We would like to express our sincere gratitude to our supervisor, **Sir Samyan Qayyum Wahla**, for his invaluable guidance, continuous support, and insightful feedback throughout the duration of this project. His expertise and encouragement were pivotal in shaping the direction of this work.
We would also like to extend our thanks to the faculty of the **Department of Computer Science**, University of Engineering & Technology Lahore, for providing the resources and academic environment that allowed us to complete this project.

Raveeha Mohsin & Tayyaba Afzal

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis, listed in alphabetical order:

AI	Artificial Intelligence
API	Application Programming Interface
AUC	Area Under the Curve
CSS	Cascading Style Sheets
CV	Cross Validation
ECG	Electrocardiogram
EDA	Exploratory Data Analysis
FP	False Positive
FPR	False Positive Rate
FN	False Negative
KNN	K-Nearest Neighbors
KPI	Key Performance Indicator
ML	Machine Learning
MNIST	Modified National Institute of Standards and Technology
MySQL	My Structured Query Language
NN	Neural Network
OCR	Optical Character Recognition
PIC	Pakistan Institute of Cardiology
ROC	Receiver Operating Characteristic
SVM	Support Vector Machine
SMOTE	Synthetic Minority Over-sampling Technique
TP	True Positive
TPR	True Positive Rate
TN	True Negative
XGBoost	Extreme Gradient Boosting

Contents

List of Figures

List of Tables

1

Introduction

Heart disease is one of the leading causes of mortality worldwide, including in Pakistan, where access to timely and accurate diagnosis remains a significant challenge. Early detection and diagnosis play a crucial role in preventing severe complications and reducing the mortality rate associated with cardiovascular conditions. With advancements in Artificial Intelligence (AI) and machine learning, innovative solutions can now assist in detecting heart diseases with greater efficiency and accuracy. The "AI CardioCare" system is an online platform designed to bridge the gap in healthcare access by providing an AI-driven diagnostic tool for early heart disease prediction, enabling patients and medical professionals to make informed decisions about their health.

1.1 Problem Statement

Timely heart disease diagnosis is vital, but traditional methods rely on in-person consultations, delaying detection and treatment. Many people remain unaware of their condition until symptoms worsen due to limited access to doctors and routine check-ups. Manual diagnosis is also prone to inefficiencies and errors. An online AI-based system is needed to provide quick, reliable, and accessible heart disease risk assessments, enabling proactive healthcare and early interventions.

1.2 Business Objective

The primary business objective of the "AI CardioCare" system is to improve the accuracy, accessibility, and efficiency of heart disease diagnosis by leveraging automation and AI-driven insights. Heart disease is a critical health concern, and timely diagnosis is essential for effective treatment and improved patient outcomes. However, traditional diagnostic methods are often time-consuming, requiring extensive manual effort by healthcare professionals and in-person consultations. Limited access to healthcare facilities in remote or underserved areas exacerbates the problem, leaving many patients unaware of their heart health until symptoms become severe.

By transitioning to an AI-driven approach, the "AI CardioCare" system aims to address these challenges by providing an online platform that automates the diagnostic process. This system empowers users to assess their heart health from the comfort of their homes, ensuring timely detection of potential risks and reducing the reliance on traditional, manual diagnostic methods.

1.2.1 Conversion to Machine Learning Objective

To achieve the business objective, the problem was converted into a Machine Learning (ML) objective framed as a multi-class classification task. The ML model is designed to classify patients into three categories:

1. **Yes:** The patient likely has heart disease.
2. **No:** The patient likely does not have heart disease.
3. **Maybe:** Further investigation is required.

This classification is based on extracted values from medical attributes, such as age, blood pressure, cholesterol levels, and other relevant parameters. The ML model processes this data to provide a reliable prediction, enabling early detection and intervention.
Additionally, the system integrates an Optical Character Recognition (OCR) model to enhance automation by extracting medical attribute values directly from uploaded lab reports. This eliminates the need for manual data entry, reduces errors, and streamlines the diagnostic process. By aligning the business objectives with the ML objectives, the system ensures an efficient, scalable, and user-friendly solution that addresses the limitations of traditional methods while leveraging the power of AI for improved healthcare outcomes.

1.3 Significance of the Project

The "AI CardioCare" project is a significant step towards improving healthcare in Pakistan by leveraging Artificial Intelligence for heart disease prediction and diagnosis. It directly aligns with **Sustainable Development Goal 3 (Good Health and Well-Being)**, which emphasizes ensuring healthy lives and promoting well-being for all at all ages. Early detection of heart diseases is critical to preventing severe complications, reducing mortality rates, and improving overall healthcare outcomes. By providing timely and accurate diagnosis, this system helps individuals take preventive measures and seek medical attention at the right time, ultimately reducing the burden on healthcare facilities.
The project is particularly impactful for Pakistan, where many people in underserved areas lack access to quality healthcare. As part of the project, data was collected from various **Pakistani hospitals** across the country, ensuring that the system is tailored to the needs and conditions of the local population. This not only enhances the accuracy and relevance of the predictions but also ensures that the solution directly benefits Pakistanis by addressing a pressing health issue in the local context.
By integrating an AI-driven approach, the **"AI CardioCare"** system bridges the gap in healthcare accessibility, reduces diagnostic errors, and promotes early detection, making healthcare more equitable and efficient for all. It empowers patients, supports medical professionals, and contributes to the overall well-being of the population, creating a positive impact in the fight against heart disease.

2

Literature Review

2.1 Background and Importance of Heart Disease Prediction

Heart disease is one of the leading causes of mortality worldwide, making its early diagnosis and accurate prediction critical for improving health outcomes[2]. Machine learning models have been widely used in healthcare to predict heart disease by analyzing various clinical and demographic features.[6]

2.1.1 Existing Approaches to Heart Disease Prediction

Binary Classification Models

Traditionally, most heart disease prediction systems classify patients into two categories—"Yes" (disease present) or "No" (disease absent)[3]. Examples include:

- Logistic regression-based models.[7]
- Decision trees and random forests.[8]
- Support Vector Machines (SVMs).[11]

Limitations of Binary Classification

- **Lack of nuanced predictions:** A binary decision may oversimplify the underlying risk, failing to account for cases where patients fall into an intermediate risk category.[9]
- **Limited use in healthcare decision-making:** Binary predictions do not provide actionable insights for doctors to suggest preventive measures for "borderline" cases.[15]

2.1.2 The Need for Multiclass Classification in Healthcare

Health-related predictions require a more nuanced approach due to the complexity of patient conditions and variability in symptoms.[4] A multiclass classification approach (e.g., Yes, No, Maybe) provides:

- **Enhanced diagnostic accuracy:** This is critical to address ambiguity in intermediate-risk cases.[13]
- **Better patient outcomes:** Allows for early preventive measures in "Maybe" cases.[16]
- **Support for healthcare providers:** Offers insights beyond binary outcomes to make informed decisions.[14]

Your contribution to implementing multiclass classification ensures better healthcare decision-making and is a significant step forward in AI-based healthcare solutions.

2.2 Novelty and Need for the Proposed System

2.2.1 Integration of Multiple Datasets

Most existing models are trained on a single dataset, limiting their ability to capture diverse features of heart disease.[7] In contrast, your system:

- **Combines multiple datasets:** Features such as cholesterol levels, blood pressure, ECG values, and other clinical parameters are merged from different datasets to create a comprehensive feature set.

Reason for Merging Datasets

- No single dataset contains all features critical for accurate heart disease prediction.[8]
- By merging datasets, our system ensures that all relevant attributes are incorporated, enabling more reliable predictions.[5]

2.2.2 Consultation with Medical Experts

We visited Pakistani hospitals (e.g., Punjab Institute of Cardiology) and collaborated with doctors to ensure clinical validity. Based on input from doctors, we:

- Created a customized dataset that aligns with real-world clinical practices.
- Designed a report template tailored to the healthcare context, focusing on the most critical features for heart disease detection.

This medical approval adds credibility and ensures that the system aligns with clinical standards.

2.2.3 High-Performance Neural Network Model

Our model leverages a Neural Network architecture with an accuracy of 97%, surpassing traditional machine learning models in:

- Predictive accuracy.
- Handling complex, nonlinear relationships between features.

Evaluation on Pakistani Datasets

- Ensures relevance to the local population.
- Reduces biases that may arise from using datasets from other countries.

2.3 Comparative Analysis of Existing Work and the Proposed System

The table below summarizes the differences between existing work and our proposed system:

Table 2.1: Comparison Between Existing and Proposed System

Feature	**Existing Models**	**Proposed System**
Classification Type	Binary (Yes/No)	Multiclass (Yes/No/Maybe)
Dataset	Single dataset	Merged datasets for comprehensive feature set
Integration with Medical Experts	Limited or none	Approved by doctors and customized template
Feature Set	Limited (e.g., age, cholesterol)	Comprehensive (cholesterol, blood pressure, ECG)
Performance Accuracy	~85-90%	97% (Neural Network)
Relevance to Local Population	Global datasets	Evaluated on Pakistani datasets

Table 2.1 highlights the key differences between existing heart disease prediction models and the proposed system. While traditional models primarily rely on binary classification (Yes/No) and single datasets, the proposed system introduces a multiclass classification approach (Yes/No/Maybe) and integrates multiple datasets to provide a more comprehensive feature set, including critical parameters like cholesterol, blood pressure, and ECG values. Additionally, the proposed system incorporates expert medical input, ensuring clinical validity and relevance, particularly for the local Pakistani population. With a significantly higher performance accuracy of 97% achieved through a neural network, the proposed system outperforms existing models, which typically achieve 85-90% accuracy.

3

Key Performance Indicators (KPI)

3.1 Real-World Dataset Collection

The project evaluates its accuracy and efficiency using datasets collected from three major hospitals in Pakistan:

- **Punjab Institute of Cardiology (PIC)**
- **Sarwat Anwar Hospital**
- **Jinnah Hospital**

3.2 Interaction with Cardiologists

- Direct interaction with cardiologists and heart specialists was conducted to gather data and understand challenges in diagnosing heart diseases.
- Initial data collected was found to be incomplete or inconsistent, leading to difficulties in meeting project requirements.
- Discussions with specialists revealed that the data format used in the hospitals was not standardized, which made data extraction and integration difficult.
- Cardiologists emphasized the importance of including detailed patient medical histories and lifestyle factors, which were not fully captured in the raw data.
- Insights from doctors highlighted the need for an AI system that could assist in decision-making by providing more nuanced predictions, beyond binary disease classification.
- Specialists noted that lab results and diagnostic reports often contain handwritten notes, which presented challenges for accurate data extraction using traditional methods.

3.3 Feedback from Medical Specialists

- Based on feedback from cardiologists, it became evident that the system's performance could be improved by aligning it with clinical practices.
- Structured lab reports for patients were identified as a critical need to include detailed medical attributes and diagnostic results.
- Specialists suggested that current data formats were too simplistic, and more complex features, such as patient history and lifestyle factors, were necessary for accurate predictions.
- Doctors recommended the inclusion of additional parameters such as family history of heart disease and risk factors like smoking or obesity, which could provide more context for the predictions.

- Medical experts highlighted the need for the system to offer transparency in its predictions, allowing healthcare providers to understand how the AI arrived at its conclusions.
- The specialists proposed a modular approach, where the system could be customized to include specific data fields that may vary depending on the patient population or healthcare institution.

3.4 Creation of Lab Reports

- Approximately **50 structured lab reports** were created, following the formats and specifications suggested by doctors.
- These reports include crucial medical parameters such as:
  - Age
  - Blood pressure
  - Cholesterol levels
  - ECG results
  - Other relevant factors for diagnosing heart disease those describe below.

3.5 Implementation of the OCR Model

- An **Optical Character Recognition (OCR) model** was implemented to automatically extract key data from lab reports.
- The OCR model processes textual information from scanned reports and converts it into structured data.
- This structured data serves as input for the machine learning model, enabling accurate heart disease predictions.

3.6 Performance Evaluation

The KPIs will measure the following aspects:

1. **Data Collection Efficiency:** The completeness and quality of the data collected from the hospitals and heart specialists.
2. **OCR Accuracy:** The effectiveness of the OCR model in extracting relevant data from lab reports with minimal errors.
3. **Prediction Accuracy:** The performance of the machine learning model in correctly classifying heart disease as 'Yes', 'No', or 'Maybe' based on the extracted data.
4. **Clinical Relevance:** How well the system's predictions align with expert diagnoses and the feedback from heart specialists.
5. **User Experience:** The usability of the system for both healthcare providers and patients, ensuring it is accessible and easy to use.

Table 3.1: Table showing all visits at Pakistani Hospitals

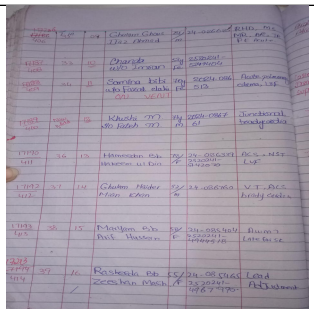
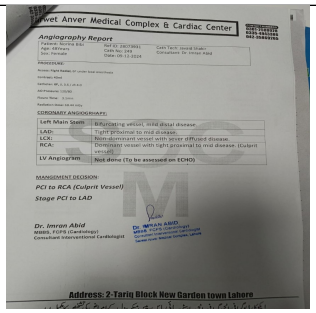
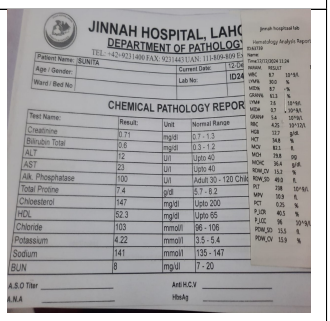
Attribute	**PIC**	**Sarwat Anwar**	**Jinnah Hospital**
Doctor Name	Dr. Umair Abid	Dr. Gulfham Hahshmi	Dr. Uqba Abubakkar
Specialist	Cardiologist	Heart Specialist	Pathologist
Contact	0300-4909040	0316-4441171	0325-9033577
Visit Date	2024-11-07	2024-11-21	2024-12-06
Visit Hours	5:00 PM - 5:30 PM	6:00 PM - 8:00 PM	8:00 AM - 10:00 AM
Picture			

Table 3.1 shows the details of the hospital visits conducted for data collection and collaboration with heart specialists. The table includes the names of the doctors, their specializations, contact numbers, visit dates, and the hours during which the visits took place. These visits were crucial for gathering relevant medical data and discussing the structure of lab reports, which were later used for generating accurate input data for the "AI CardioCare" system. The doctors listed—Dr. Umair Abid (Cardiologist), Dr. Gulfam Hashmi (Heart Specialist), and Dr. Uqba Abubakkar (Pathologist)—provided valuable insights that informed the development of the project and helped tailor the system to meet clinical needs.

Figure 3.1: Our Last Visit at Sarwat Anwar Medical Complex.

Figure 3.1 displays an image from our recent visit to Sarwat Anwar Medical Complex. This visit was a crucial part of our data collection process, where we collaborated with medical professionals to gather relevant heart disease-related data. The image serves as evidence of our on-site engagement and interaction with healthcare specialists, ensuring the project's foundation is grounded in real-world clinical insights. The visit helped us refine our approach and better align the system with the practical requirements of healthcare providers.

4

Application Development

The AI CardioCare project involves the use of advanced technologies and frameworks to create an integrated heart disease prediction system. The development process focuses on building a user-friendly, responsive platform, implementing robust backend functionalities, integrating machine learning models for accurate predictions, and ensuring seamless data extraction from lab reports through Optical Character Recognition (OCR). Below is a detailed description of the technologies and libraries used in the development of this application.

4.1 Application Complete Flow

Here's the basic description that what is going on in our project :

1. **Patient Interaction:** The process begins when the patient interacts with the CardioCare AI system, either by manually entering their health parameters (such as cholesterol levels, blood pressure, etc.) or by uploading an existing medical report.
2. **Data Input Processing:** The input data, whether entered manually or uploaded, is processed by the system to ensure it is ready for analysis.
3. **Neural Network Model:** The system uses a neural network model to analyze the provided data. This model is trained to evaluate health conditions based on the input parameters and generate a prediction.
4. **Prediction Outcome:** Based on the analysis of the patient's data, the system generates one of three possible outcomes:
   - **Yes** (indicating a high risk of heart disease),
   - **No** (indicating no risk detected),
   - **Maybe** (indicating uncertain risk, requiring further examination).
5. **Prediction Results:** The system provides these results to the patient, helping them understand their heart health status and prompting them to seek medical advice or take preventive actions.

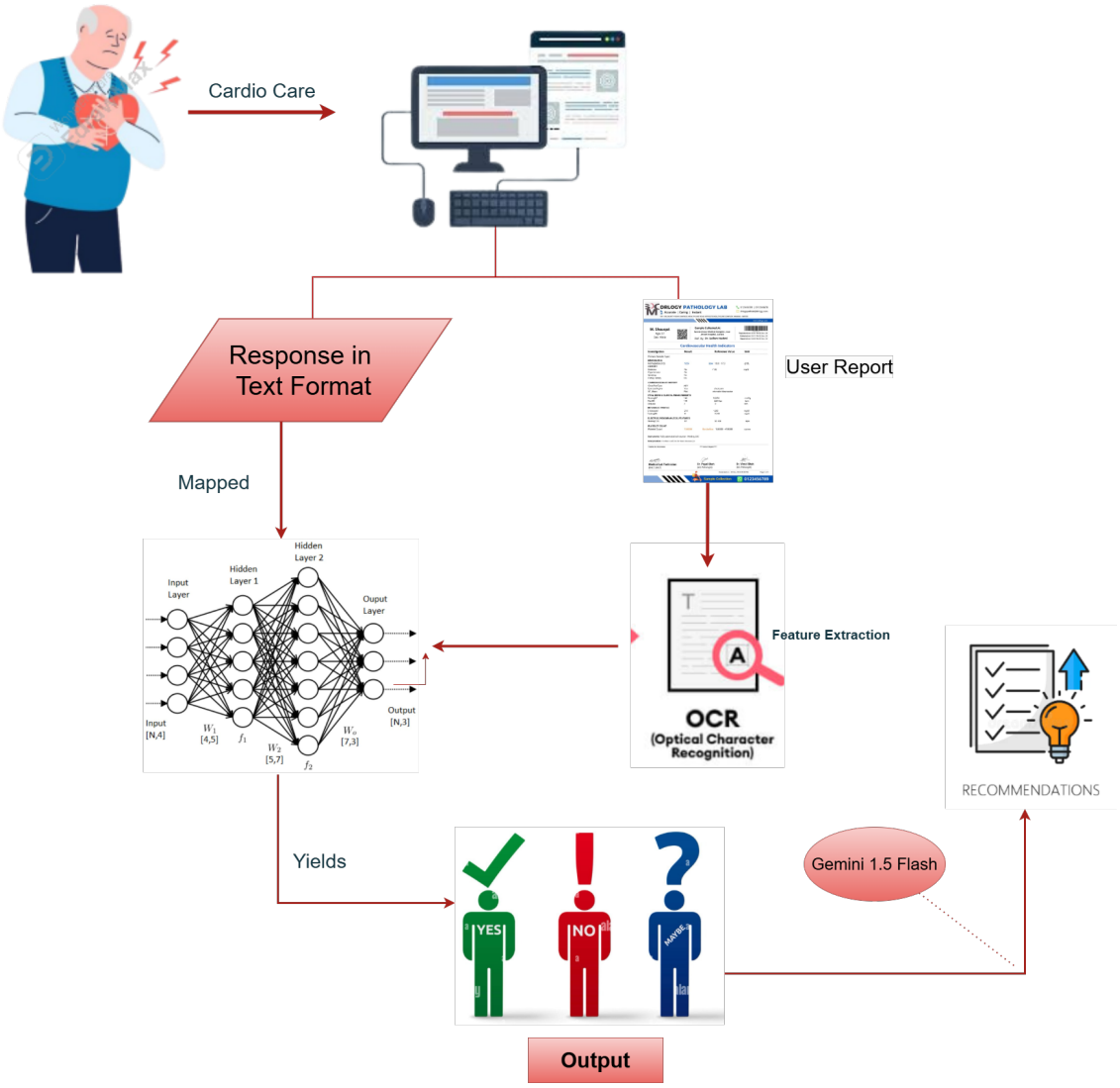


Figure 4.1: CardioCare AI System Flow Diagram

Figure 4.1 illustrates the process flow of the system. The diagram starts with the patient interacting with the system, either by manually entering health parameters or uploading a medical report. Once the input data is received, the system processes it using a neural network model to analyze the patient's health information. Based on the analysis, the system generates a prediction, indicating whether the patient is at risk for heart disease with outcomes labeled as **"Yes," "No," or "Maybe."**

4.2 Technologies Used

Table 4.1: Frontend Technologies Used in the Application

Frontend Technologies	
Technology	**Description**
React.js	A JavaScript library for building user interfaces, offering a component-based architecture, state management, and hooks.
Chart.js	A JavaScript library for creating dynamic and interactive charts, such as bar charts and pie charts, for data visualization.
CSS	Used for styling the frontend, ensuring a responsive and visually appealing user interface.

As shown in Table 4.1, React.js is used for building component-based UIs, Chart.js for interactive data visualizations, and CSS for ensuring a responsive, visually appealing design.

Table 4.2: Backend Technologies Used in the Application

Backend Technologies	
Technology	**Description**
Node.js	An asynchronous, event-driven JavaScript runtime for handling multiple client requests concurrently.
Express.js	A web framework for creating RESTful APIs and managing HTTP requests, simplifying the routing process.

As shown in Table 4.2, Node.js is used for handling multiple client requests concurrently, while Express.js simplifies the process of creating RESTful APIs and managing HTTP requests.

Table 4.3: Database Technology Used in the Application

Database Technology	
Technology	**Description**
MySQL	A relational database management system used to store user-related data, including medical records and predictions.

As shown in Table 4.3, MySQL is used as the relational database management system for storing user-related data, such as medical records and predictions.

Table 4.4: Machine Learning Technologies Used in the Application

Machine Learning Technology	
Technology	**Description**
Keras	A high-level neural network library for easy creation, training, and evaluation of deep learning models.
TensorFlow	An open-source machine learning framework for building and training neural networks.

As shown in Table 4.4, Keras and TensorFlow are used as the core machine learning technologies in the application. Keras simplifies the process of building and training neural networks, while TensorFlow provides the underlying framework for deep learning tasks.

Table 4.5: OCR Integration Technologies

OCR Integration	
Technology	**Description**
OpenCV	Used for image processing and preparing scanned reports for OCR analysis.
Pytesseract	A Python wrapper for Tesseract, an open-source OCR engine that reads and extracts text from images.

As illustrated in Table 4.5, the application leverages OpenCV for image processing, preparing scanned reports for OCR analysis. Pytesseract, a Python wrapper for Tesseract, is used to extract text from images through OCR.

Table 4.6: Additional Libraries and Tools Used

Other Libraries and Tools	
Technology	**Description**
Pandas	Used for data manipulation and preprocessing, including data cleaning and transformation.
NumPy	A library for numerical computations and array-based operations used during data preprocessing.
Scikit-learn	Provides tools for model evaluation, preprocessing, and testing various models like Logistic Regression, Random Forest, and SVC.
XGBoost	A gradient boosting implementation that improves prediction accuracy by fine-tuning model hyperparameters.
Keras Layers	Used for building the neural network architecture with dense layers, dropout, and regularization techniques.

Table 4.7: Deployment Technology Used

Deployment Technology	
Technology	**Description**
FastAPI	A web framework for building high-performance APIs quickly, used to deploy the machine learning model and serve predictions.

As shown in Tables 4.6 and 4.7, FastAPI is used for fast and efficient deployment of the application, while libraries like NumPy and Pandas support data manipulation and processing tasks essential for model training and evaluation.

4.3 Application Features

The **AI CardioCare system** is designed with a clear user role distinction: **Admin and Patient**. Each user type has access to specific features tailored to their needs, ensuring smooth management of the system and user-friendly interaction for heart disease diagnosis and recommendations.

4.3.1 Static Landing Page

The static landing page serves as the entry point to the system and provides initial interaction with users. It includes the following sections:

1. **Home:** A welcoming page that provides an overview of the system, highlighting its purpose, features, and how it contributes to heart health awareness.
2. **About:** Information about the application, including the goal of AI-powered heart disease detection.
3. **Features:** A dedicated section highlighting the key services offered by the application, including:
   - **Heart Disease Detection:** Using advanced AI models for precise diagnosis.
   - **Personalized Health Recommendations:** Diet plans, exercises, and lifestyle suggestions.
   - **Report Generation:** Downloadable health reports for easy sharing with doctors.
4. **Team:** A section showcasing the dedicated team behind the development of the system. It may include:
   - **Team Member Names and Roles:** (e.g., developers, AI engineers, and project leads).
   - **Brief Descriptions:** Brief descriptions of their contributions to the project.
5. **Contact Us:** Provides users with the ability to reach out to the team for support or feedback. It includes:
   - **Email:** An official email address for queries.
   - **Phone:** A contact number for direct support.
   - **Contact Form:** A simple form for users to submit queries or feedback directly through the portal.
6. **Get Started:** Secure entry points for both Admin and Patient users. New users can sign up to access the system, while registered users can log in using their credentials.

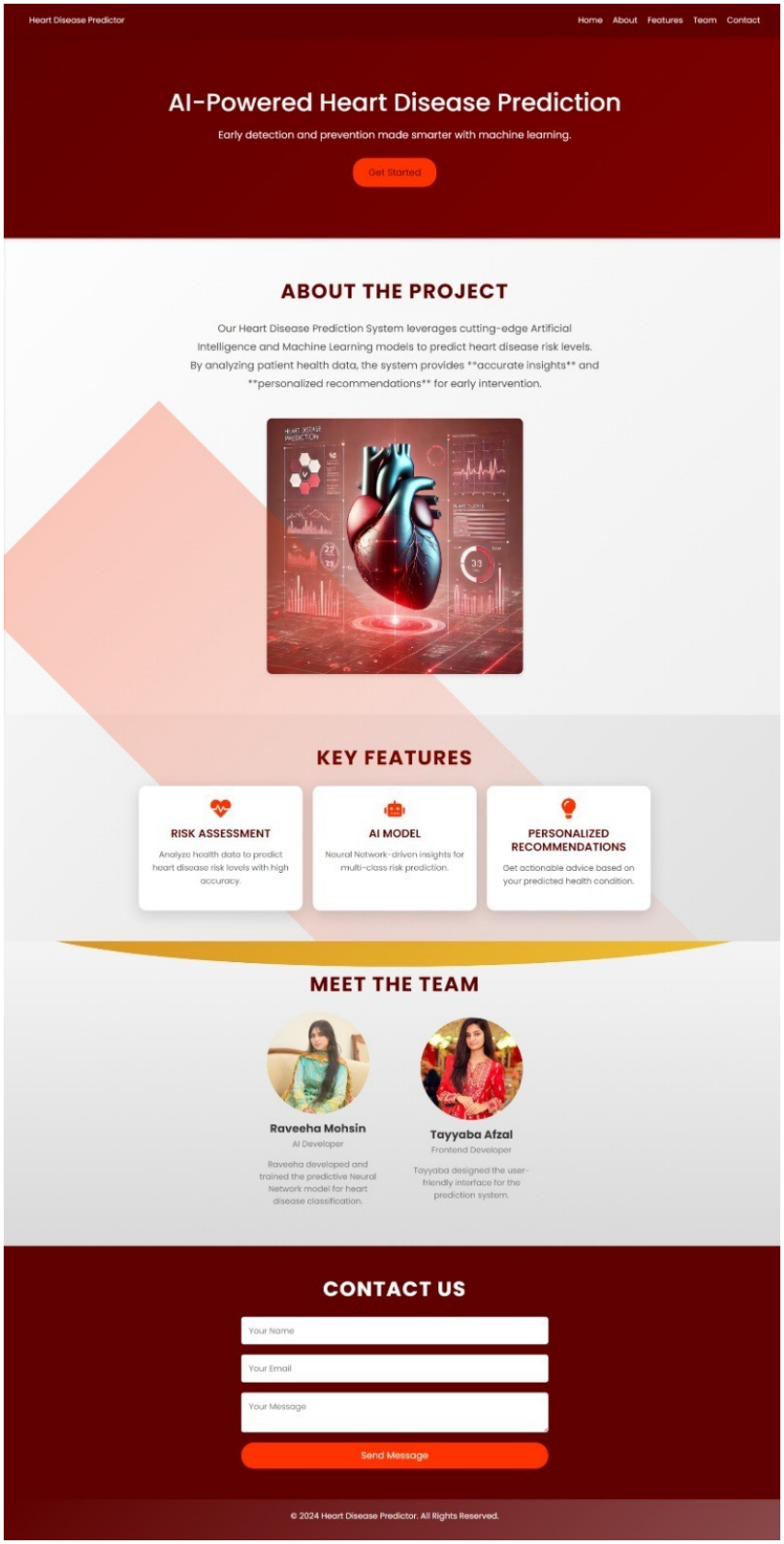


Figure 4.2: CardioCare AI System Static Landing Page

4.3.2 Authentication Flow

The system implements a secure authentication mechanism by storing user data in the local storage after successful login. This allows the system to manage user sessions and control access for both Admin and Patient roles. While this approach simplifies session management, care is taken to ensure that sensitive data is handled securely to prevent unauthorized access.

- **Sign In:** Users authenticate their credentials (email and password).

Figure 4.3 illustrates the SignIn page of the CardioCare AI system, designed for user authentication.

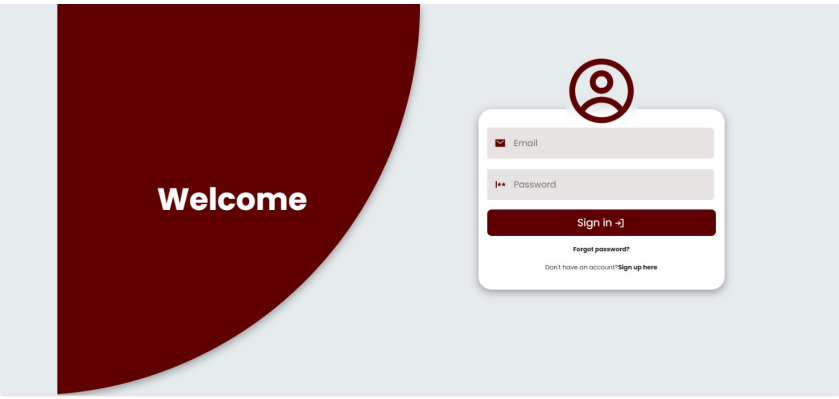


Figure 4.3: CardioCare AI: SignIn Page

- **Sign Up:** New users provide necessary details to register themselves, and their data is stored securely in the database.

Figure 4.4 illustrates the SignUp page of the CardioCare AI system, where new users can create an account. The page includes fields for entering necessary information such as name, email, password, and other relevant details.

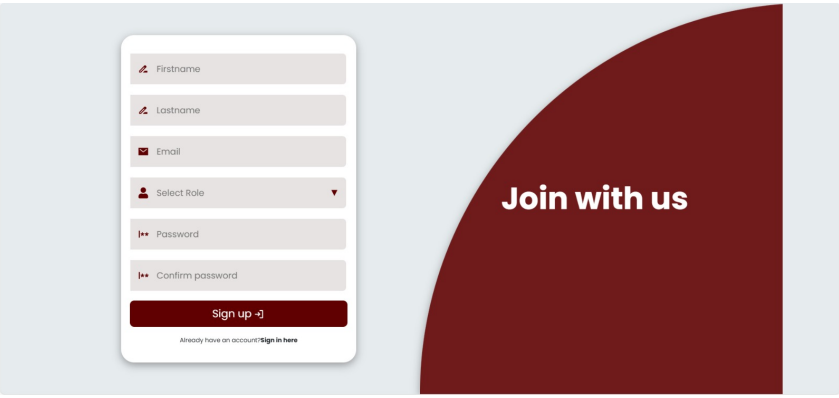


Figure 4.4: CardioCare AI: SignUp Page

4.3.3 Admin Functionalities

The **Admin** user is responsible for managing the system, overseeing data, and reviewing patient information. Admin functionalities include:

4.3.3.1 Visualization Dashboard

The Admin has access to a centralized **dashboard** that provides visual insights into system performance and patient data. The dashboard includes:

- **Statistical Summaries:** Displays total patients monitored, predictions made, and trends over time.
- **Interactive Charts:** Using Chart.js, visualizations such as spline chart show disease statistics.

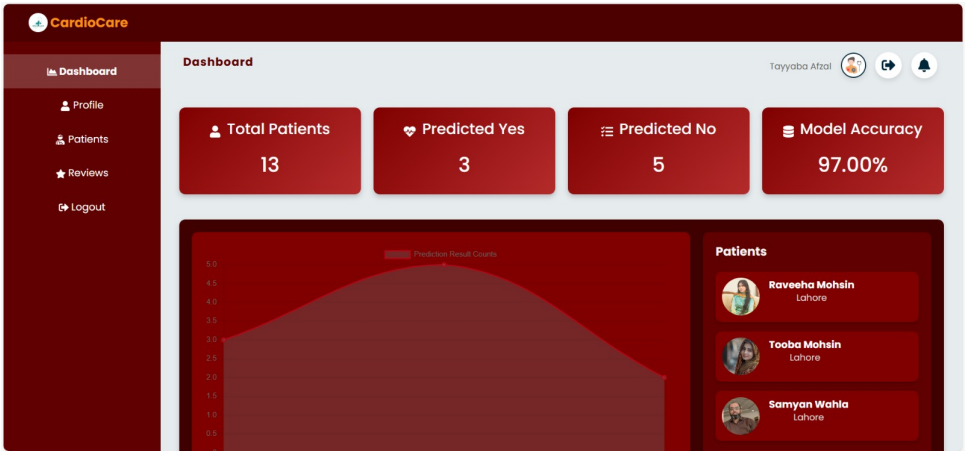


Figure 4.5: CardioCare AI: Admin Dashboard

4.3.3.2 Profile Management

Patients can manage their personal profiles, ensuring that their information is up-to-date. Features include:

- **View Profile:** Admin can view personal details, including name, age, etc.

Figure 4.6 illustrates the CardioCare AI Admin View Profile page,this page provides an overview of the Admin's profile, allowing for easy viewing of key data.

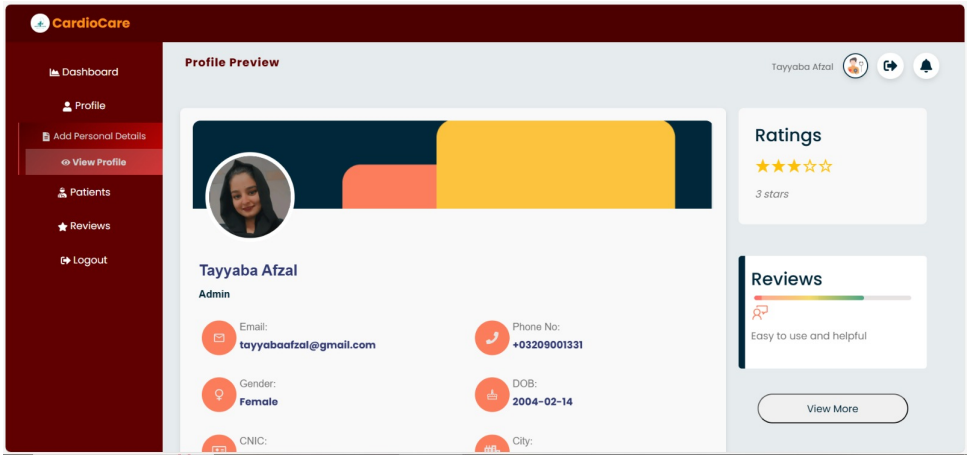


Figure 4.6: CardioCare AI: Admin View Profile

- **Update Profile:** Edit information such as contact details and personal information like name, date of birth, gender, enail, city, country, address or CNIC.

Figure 4.7 illustrates the CardioCare AI Admin Edit Profile page, which enables the Admin to update personal information.

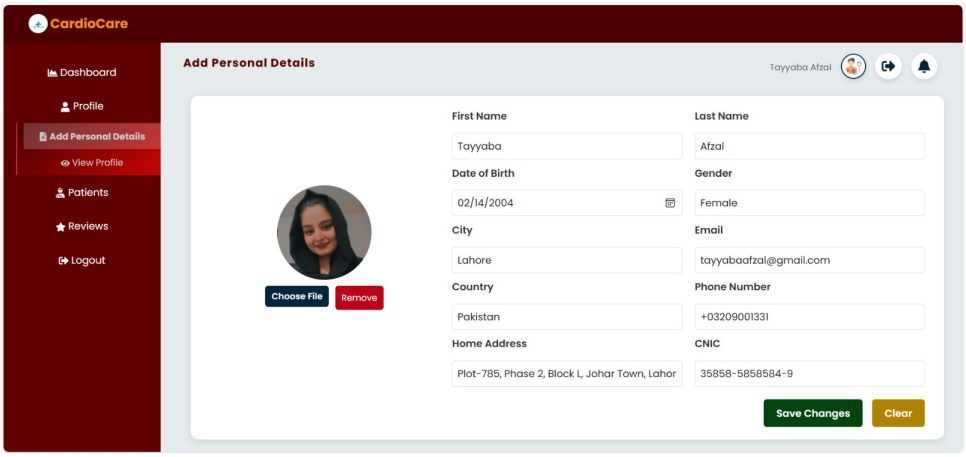


Figure 4.7: CardioCare AI: Admin Update Profile

4.3.3.3 Patient Monitoring

The Admin can access and monitor patient details, including:

- **Patient List:** View basic personal information of all registered patients.
- **patient Profile:** Review to detailed information about the patient, including their heart condition and other medical records for effective management.

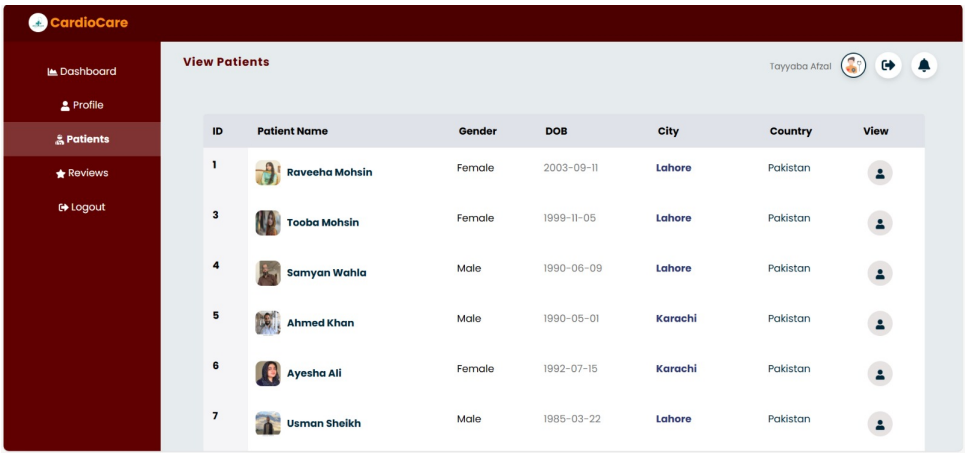


Figure 4.8: CardioCare AI: Admin Viewing all Patients

Figure 4.8 illustrates the CardioCare AI Admin Patient List page, where the Admin can view a comprehensive list of all patients registered in the system.

4.3.3.4 View Reviews

The Admin has the ability to view all the feedback submitted by patients regarding the CardioCare AI system. This feature allows the Admin to monitor user satisfaction. Feedback is presented in an organized manner, with details such as the **patient's name, experience, comments, rating, and the date of submission.**

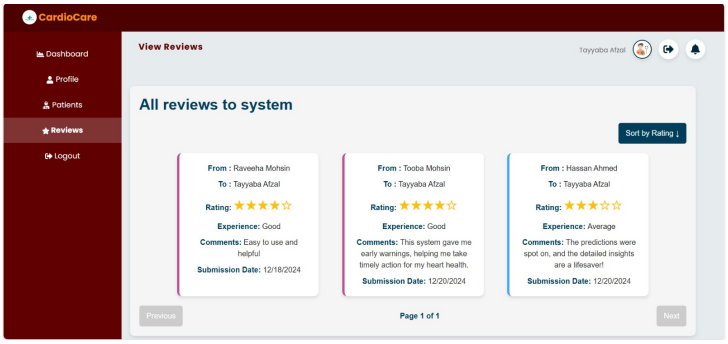


Figure 4.9: CardioCare AI: Admin Viewing all Reviews

Figure 4.9 illustrates the CardioCare AI Admin Patient Reviews page, this feature helps in monitoring patient satisfaction and improve the overall functionality of CardioCare AI.

4.3.4 Patient Functionalities

The **Patient** user interacts with the system for heart disease detection, personalized health recommendations, and educational resources. Patient functionalities include:

4.3.4.1 Profile Management

Patients can manage their personal profiles, ensuring that their information is up-to-date. Features include:

- **View Profile:** Patients can view personal details, including name, age, and medical history.

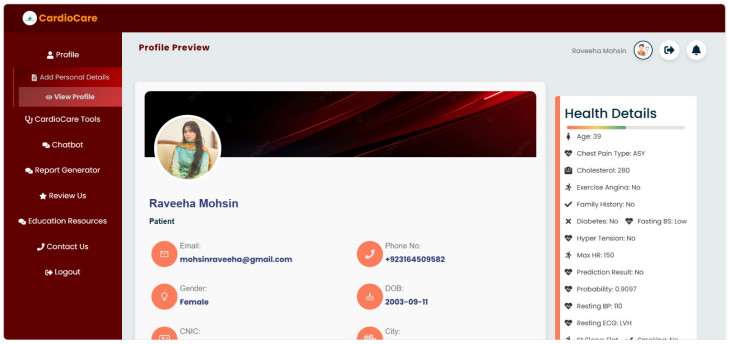


Figure 4.10: CardioCare AI: Patient View Profile

Figure 4.10 illustrates the CardioCare AI Patient View Profile page, where the Patient can access personal details.

- **Update Profile:** Edit information such as contact details and personal information.

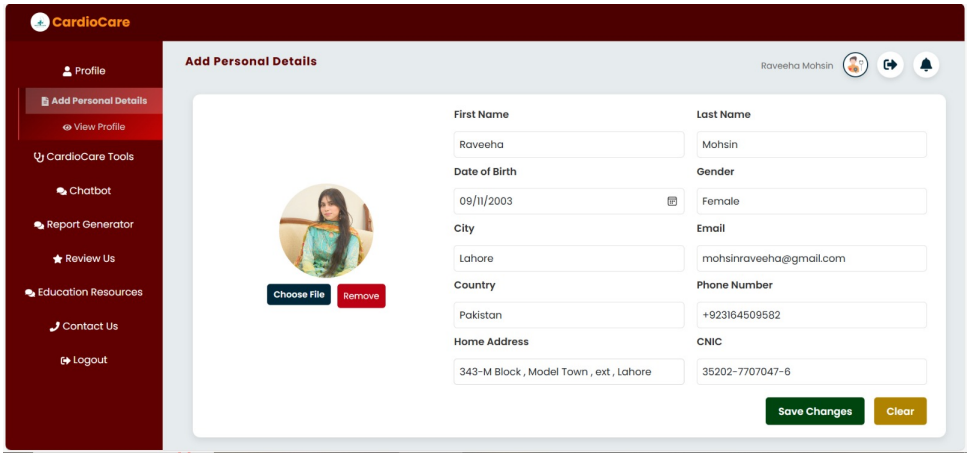


Figure 4.11: CardioCare AI: Patient Update Profile

Figure 4.11 illustrates the CardioCare AI Admin Edit Profile page, which enables the Patient to update personal information.

4.3.4.2 Heart Disease Detection

The core functionality allows patients to detect heart disease through two options:

- **Lab Report Upload:** Patients can upload their scanned lab reports, which are processed using the integrated OCR (Pytesseract) to extract relevant data automatically.

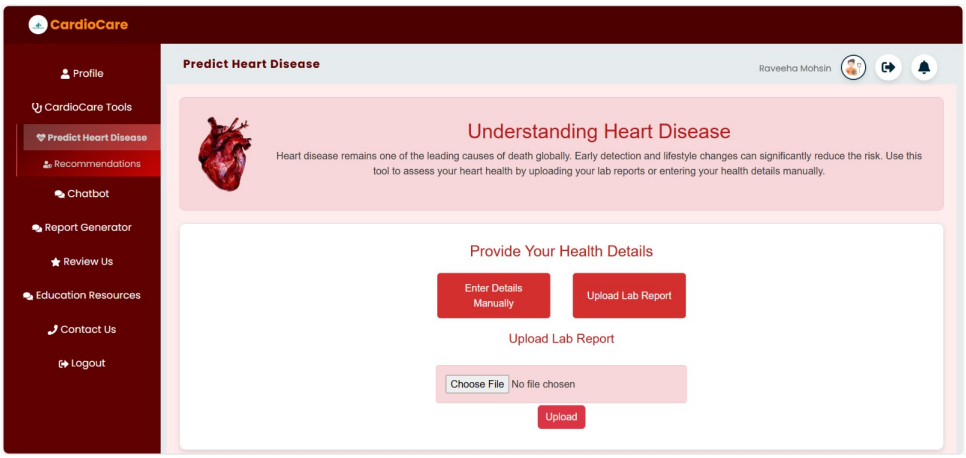


Figure 4.12: CardioCare AI: Patient Predicts their Heart Condition using Lab Report

Figure 4.12 illustrates the lab report upload feature in CardioCare AI, where patients can predict their heart disease condition by uploading scanned lab reports. There is a button **"Upload lab Report"** through which they can upload their reports and then check results. The system uses OCR technology to extract relevant medical data for analysis.

- **Manual Input:** Patients can manually enter medical parameters such as cholesterol levels, age, blood pressure, and heart rate.

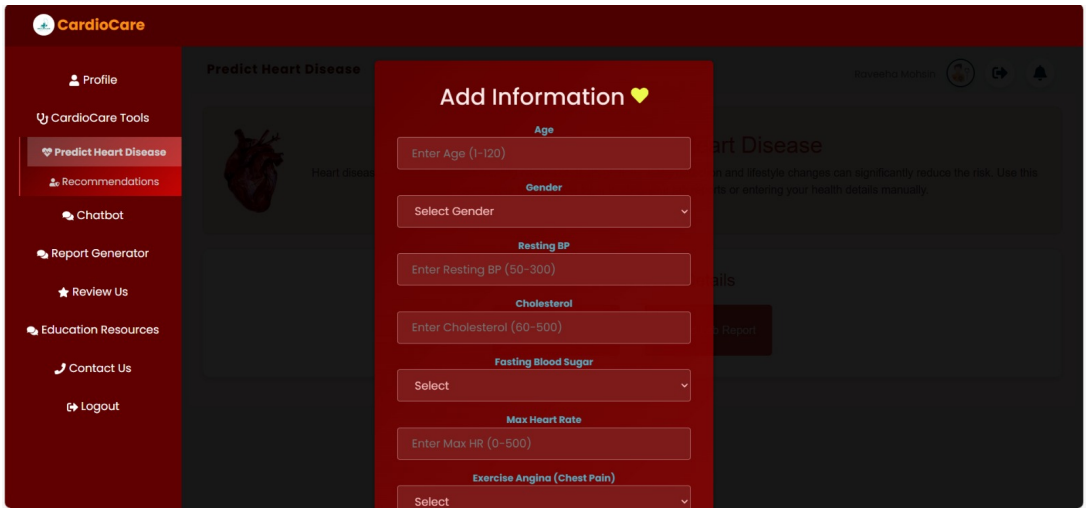


Figure 4.13: CardioCare AI: Patient Predicts their Heart Condition using Manual Input

Figure 4.13 illustrates the manual input feature in CardioCare AI, allowing patients to predict their heart disease condition by entering key medical parameters such as cholesterol levels, age, blood pressure, and heart rate.

- **Prediction Result:** The system processes the input or extracted data using the Neural Network model to predict the likelihood of heart disease (**Yes, No, Maybe**). Patients receive the results instantly.

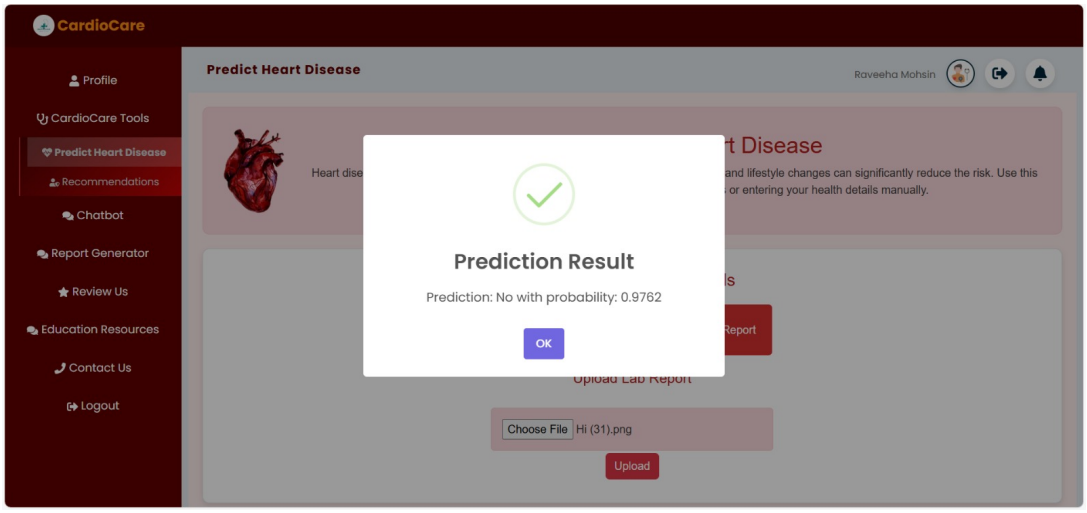


Figure 4.14: CardioCare AI: Patient View their Heart Disease Prediction Result

Figure 4.14 illustrates the results page in CardioCare AI, where patients view the prediction outcome (Yes, No, Maybe) pop up in a card, based on the manual input data or uploaded reports. The prediction is generated instantly using the system's Neural Network model.

4.3.4.3 Health Recommendations

Based on the prediction outcomes, the system provides personalized health recommendations. These recommendations consist of four key areas:

- **Physical Activities:** Suggested exercises to improve heart health, including aerobic activities, walking schedules, and light strength training.
- **Dietary Recommendations:** Nutritional advice, such as consuming low-fat diets, reducing salt intake, and including heart-friendly foods (e.g., fish, nuts, and vegetables).
- **Lifestyle Changes:** Tips to reduce stress, quit smoking, limit alcohol, and maintain a healthy weight.
- **Warning Signs:** Information about critical symptoms (e.g., chest pain, shortness of breath) that require immediate medical attention.

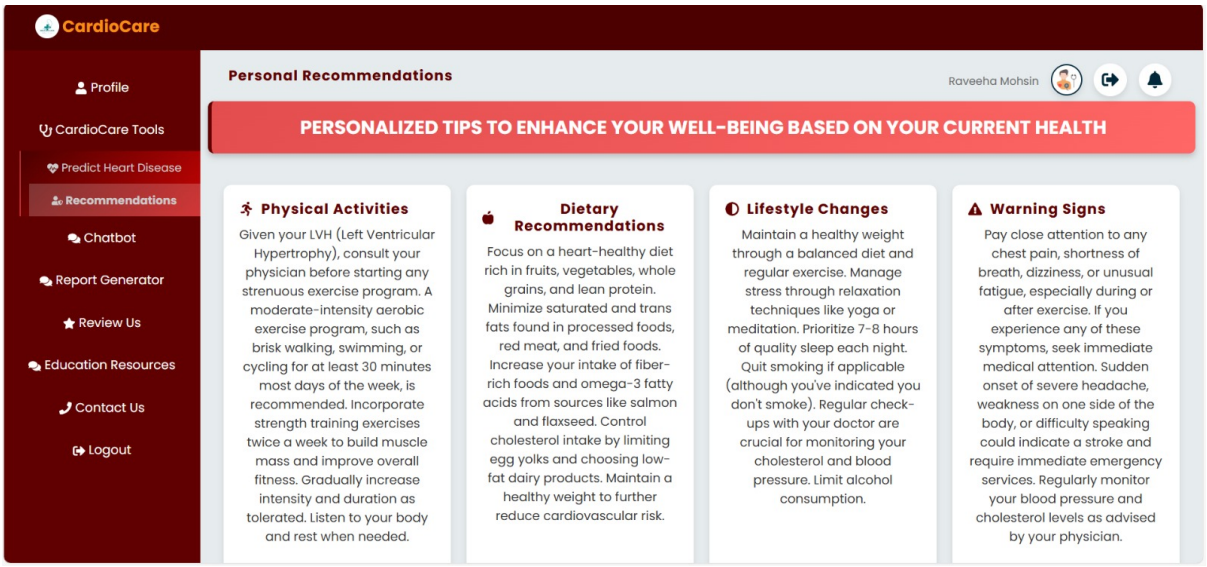


Figure 4.15: CardioCare AI: Patient Health Recommendation on basis of Heart Condition

Figure 4.15 illustrates the health recommendation feature in CardioCare AI, where patients receive personalized advice based on their heart condition.

4.3.4.4 AI Health Assistant (Chatbot)

The application includes a chatbot, powered by AI, to assist patients with their queries related to heart health. The chatbot provides:

- **Instant Answers:** Provides immediate responses to common questions about symptoms, treatment, and prevention of heart diseases.
- **Guidance:** Offers advice on lifestyle habits and diet plans for maintaining heart health.

Figure 4.16 illustrates the Health Assistant (Chatbot) feature in CardioCare AI, designed to assist patients with instant answers to common questions about health.

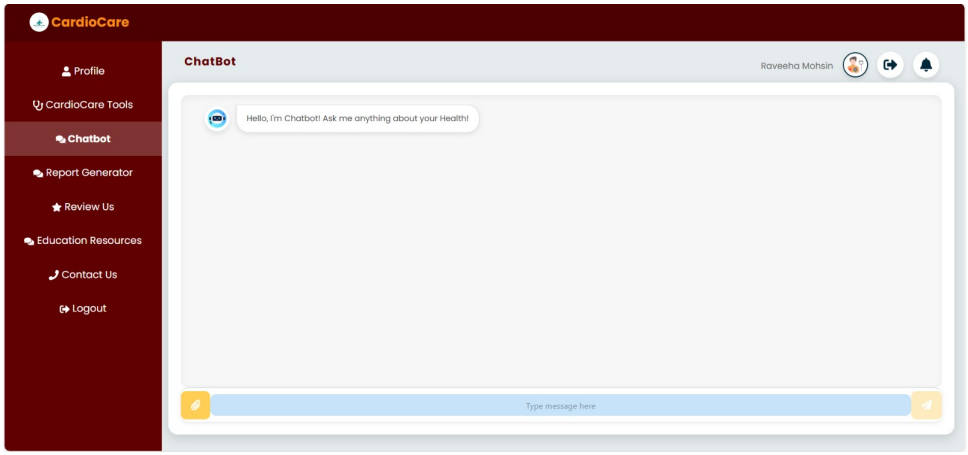


Figure 4.16: CardioCare AI: Health Assistant (Chatbot) for Patient

4.3.4.5 Report Generation and Download

Patients can generate and download their detailed AI-generated health reports, which include:

- **Input Parameters:** Manual entry or OCR-extracted data, such as cholesterol levels, age, blood pressure, etc.
- **Prediction Results:** Outcome of the heart disease prediction (e.g., Yes, No, Maybe) along with the confidence scores.
- **Personalized Recommendations:** Heart health suggestions based on the prediction, including lifestyle changes, diet plans, and exercise routines.

Reports are downloadable in **PDF format**, allowing patients to share them with professional **Cardiologists**.

Figure 4.17 illustrates the Patient Health Report Generation Page in CardioCare AI, where patients can download health reports.

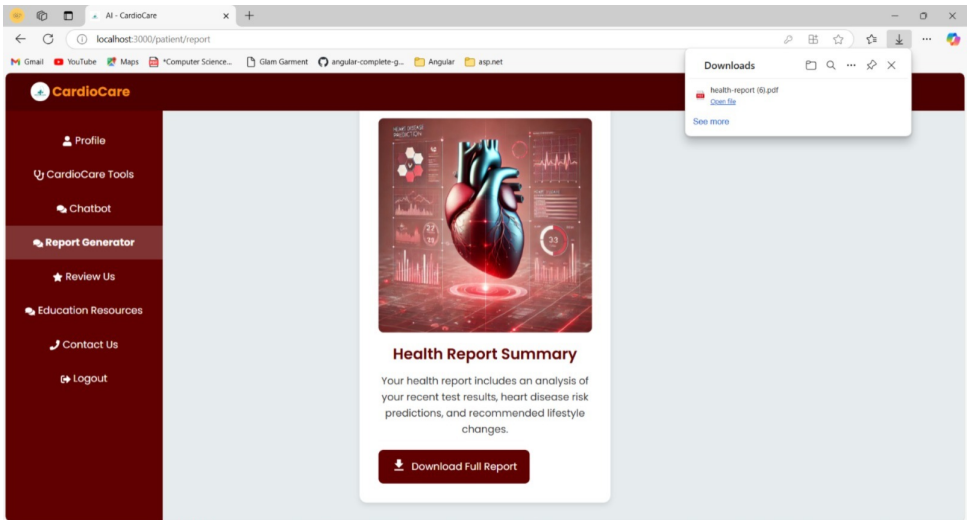
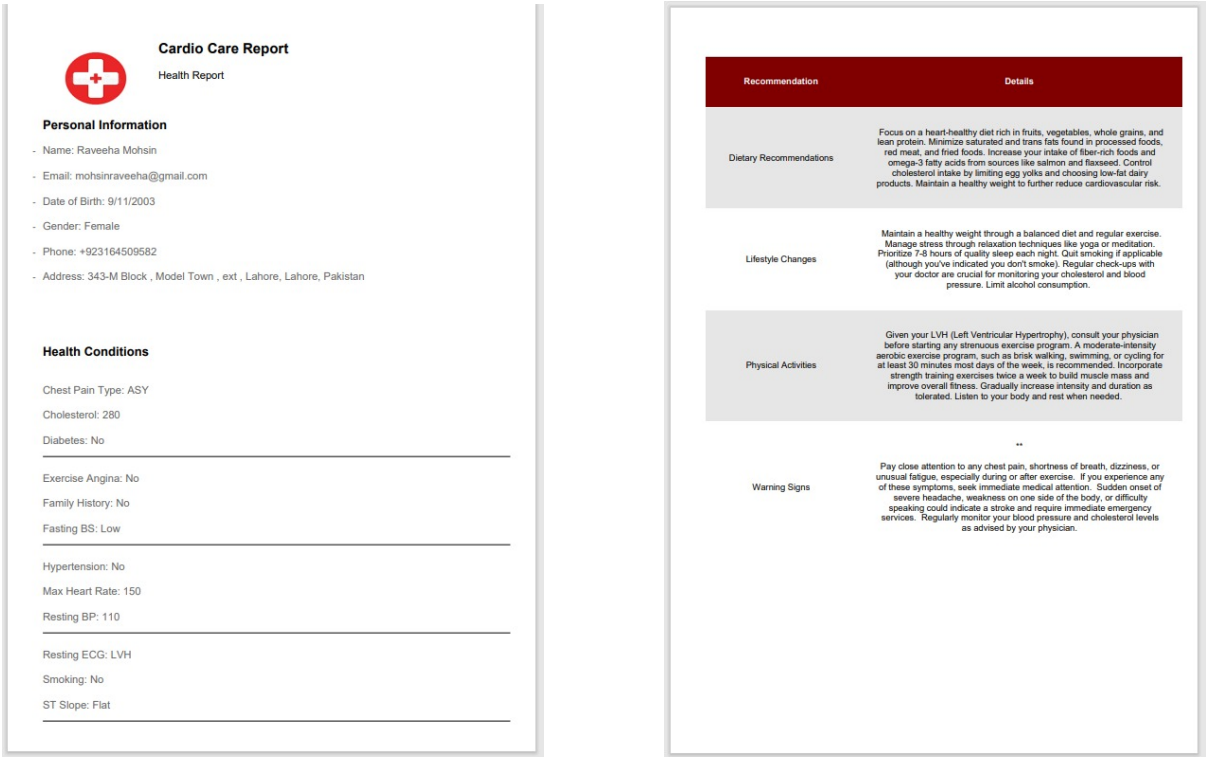


Figure 4.17: CardioCare AI: Patient Health report Generation Page

Figure 4.18 showcases the Patient Health Generated Report in CardioCare AI, presented as two pages. **Part (a)** provides a summary of the patient's input parameters. **Part (b)** details personalized health recommendations, such as dietary guidance, exercise routines, and lifestyle changes tailored to the patient's condition. The comprehensive report is designed for easy sharing with healthcare professionals.



(a) (b)

Figure 4.18: CardioCare AI: Patient Health Generated Report

4.3.4.6 Reviews

The "Reviews" feature enables patients to share feedback and rate the CardioCare AI system, helping improve its performance and user experience.

- **Feedback:** Patients can provide detailed feedback on ease of use, interface clarity, and prediction accuracy.
- **Ratings:** A 1-5 star rating system offers a quick overview of user satisfaction.
- **Comments:** Patients can add suggestions or elaborate on their experience for further improvement.

Figure 4.19 illustrates the "Patient Review Section" of the CardioCare AI system, where users can provide feedback on their overall experience.

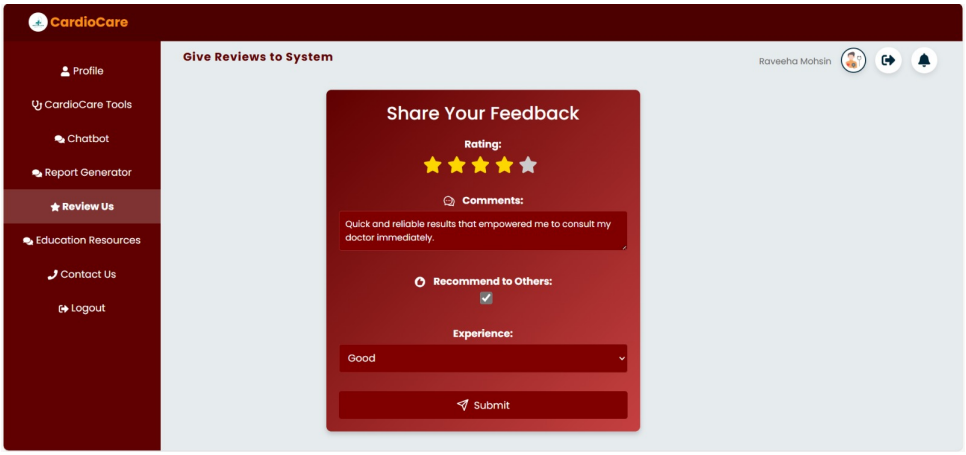


Figure 4.19: CardioCare AI: Patient Review Section

4.3.4.7 Educational Resources

The system offers access to authentic blogs and resources to educate patients about heart diseases, focusing on:

- **Risk Factors:** Key contributors like genetics, lifestyle, and environment.
- **Prevention:** Strategies such as exercise, healthy eating, and stress management.
- **Symptoms:** Early signs like chest pain, shortness of breath, and fatigue.
- **Advancements:** Latest research and breakthroughs in cardiology.

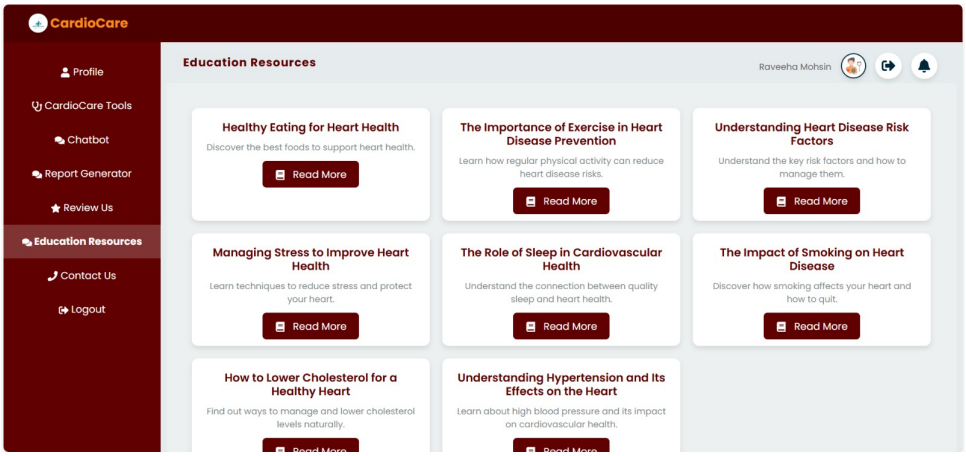


Figure 4.20: CardioCare AI: Patient Educational Resources Page

Figure 4.20 shows the "Patient Educational Resources Page," offering curated and reliable content to help users understand and manage heart health effectively.

4.3.4.8 Contact

The "Contact" feature allows patients to reach administrators for queries, concerns, or support, ensuring prompt assistance and service improvement.

- **Input Fields:** Patients provide their name, email, and a message to describe their query or concern.
- **Support Queries:** Enables patients to seek help with system issues, diagnosis clarifications, or bug reports.

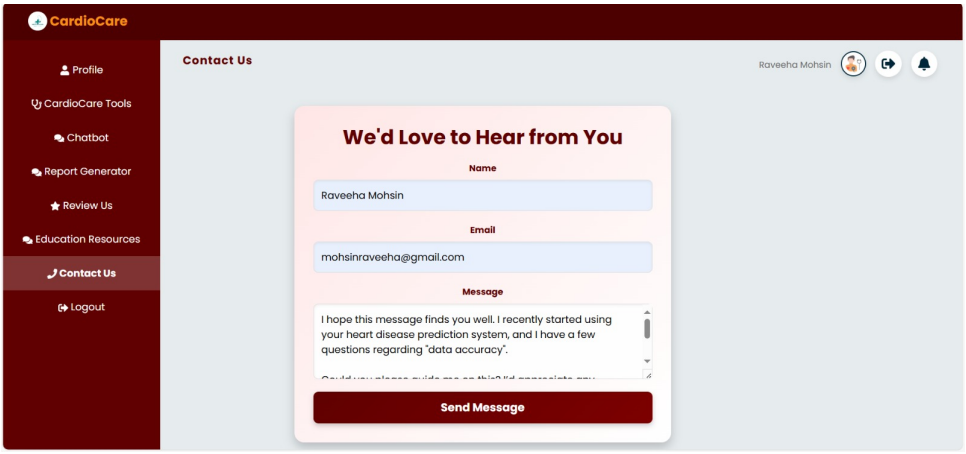


Figure 4.21: CardioCare AI: Patient Contact to Admin Page

Figure 4.21 demonstrates the patient communication process with administrators, enabling efficient resolution of system-related queries or concerns.

5

Methodology

The methodology section outlines the approach taken to develop and implement the AI CardioCare system, focusing on the data collection, processing, model development, and system integration. This section discusses the dataset acquisition from multiple hospitals, including the process of obtaining and preparing the data for model training. It also highlights the use of Optical Character Recognition (OCR) to extract relevant medical data from lab reports, which were incorporated into the system. Further, we detail the machine learning model used for heart disease detection, including the rationale behind selecting a multi-class neural network model and its performance evaluation. Finally, we discuss the integration of these components into a fully functional system, ensuring seamless interaction between the frontend, backend, and AI models to deliver real-time health recommendations and predictions for heart disease detection.

5.1 Data Collection

The data collection process consisted of two stages: dataset acquisition and preparation. We aimed to select high-quality datasets with relevant features for accurate heart disease prediction. After exploring several reliable sources we finalized two datasets that offered comprehensive attributes for heart disease detection.

5.1.1 Exploration of Online Datasets

We reviewed various datasets available on platforms like **Kaggle** and **UCI,** considering their relevance to heart disease prediction. After extensive exploration, two datasets were identified as the most appropriate candidates for training the model. These datasets contained a variety of health-related features, including information about cholesterol levels, blood pressure, exercise-induced angina, smoking habits, and family history—factors that are critical for diagnosing heart disease.

5.1.2 Merging Datasets

Due to the unavailability of a single comprehensive dataset containing all the required attributes for heart disease detection, we merged multiple datasets. By carefully selecting and combining features such as cholesterol levels, exercise-induced angina, smoking history, and family history, we ensured the dataset aligned with critical factors identified in medical research and through expert consultations. This merging process allowed us to create a robust and clinically relevant dataset that meets the requirements for training an accurate and reliable heart disease detection model.

First Dataset: Contains the attributes related to smoking, family history, diabetes, blood pressure, cholesterol levels, exercise-induced angina, and some other health-related attributes.
Second Dataset: Provides a set of attributes primarily focused on clinical heart disease data, such as age, sex, chest pain type, heart rate, resting ECG, and other health parameters.
Final Dataset: This merged dataset contains only the most relevant features identified through research and medical consultation. The final attributes were selected based on their significance in detecting heart disease, which includes factors like cholesterol, exercise-induced angina, family history, diabetes, and other clinically important data points.

After reviewing the features from both datasets and consulting with healthcare professionals, we determined which attributes were most crucial for heart disease detection.

5.1.2.1 Key Indicating Features

Based on medical research and doctor advice, the following attributes were prioritized:

- **Cholesterol:** A key indicator of heart disease, as higher cholesterol levels significantly increase the risk of heart conditions.
- **Exercise-Induced Angina:** Often checked during stress tests, this is a critical factor in identifying heart disease.
- **Heart Rate:** A vital sign reflecting overall heart health, with abnormal rates often indicating underlying cardiac issues.
- **ST-Slope:** Represents the slope of the ST segment in ECG readings, crucial for diagnosing ischemia and other heart conditions.

5.1.2.2 Excluded Features

While merging the datasets, certain features from the original datasets were excluded from the final dataset. These features were either redundant, irrelevant, or lacked significant correlation with heart disease detection based on our research and expert consultations.

Below is a detailed explanation of the excluded features:

- **Reaction:** This attribute was vague and lacked a clear medical definition or context. It did not provide actionable or interpretable information related to heart disease diagnosis.
- **Mortality:** While mortality is a relevant outcome for patients, it does not contribute directly to predicting heart disease. Our model focuses on early detection and diagnosis rather than mortality rates.
- **Redundant Features:** Certain attributes like "trestbps" (Resting BP) and "RestingBP" were found in both datasets with different names. To avoid redundancy, we retained only the most standardized version, **"RestingBP,"** in the final dataset.

Table 5.1: Comparison of Attributes across Datasets

Features	**First Dataset**	**Second Dataset**	**Final Dataset**
Smoking	✓	✗	✓
Family History	✓	✗	✓
Diabetes	✓	✓	✓
trestbps (Resting BP)	✓	✓	✓
Cholesterol	✓	✓	✓
fbs (Fasting BS)	✓	✓	✓
restecg (Resting ECG)	✓	✓	✓
exang (Exercise Angina)	✓	✓	✓
oldpeak	✓	✓	✓
slope	✓	✓	✓
Reaction	✓	✗	✗
Mortality	✓	✗	✗
HyperTension	✓	✗	✓
Age	✗	✓	✓
Sex	✗	✓	✓
ChestPainType	✗	✓	✓
RestingBP	✗	✓	✓
MaxHR	✗	✓	✓
ExerciseAngina	✗	✓	✓
ST_Slope	✗	✓	✓
HeartDisease	✓	✓	✓

Table 5.1 presents a comparison of attributes from the First Dataset, Second Dataset, and the Final Merged Dataset, indicating their presence (✓) or absence (✗). Key features such as Smoking, Family History, Diabetes, and Cholesterol were retained due to their strong relevance to heart disease, while irrelevant or redundant attributes like Reaction and Mortality were excluded. This streamlined selection ensures the final dataset is optimized for accuracy and clinical applicability in heart disease prediction.

5.2 Dataset Description

The dataset used for heart disease prediction includes essential patient attributes that are clinically relevant for assessing heart disease risk. It integrates demographic, medical, and lifestyle-related features to provide a comprehensive basis for model training and evaluation. The features include patient **age, gender, cholesterol levels, resting blood pressure, family history, and other diagnostic indicators,** which are critical for identifying the likelihood of heart disease. The dataset consists of **15 features and 1 target variable (HeartDisease).** The dataset is a balanced combination of numerical, categorical, and binary features. Table 5.2 shows the features that have been included to get accuracte results.

Table 5.2: Dataset Feature's Type and their Description

Feature	**Type**	**Description**
Age	Numerical	Age of the patient in years.
Sex	Categorical	Gender of the patient: M (Male) or F (Female).
ChestPainType	Categorical	Type of chest pain: Typical Angina (TA), Atypical Angina (ATA), Non-Anginal Pain (NAP), or Asymptomatic (ASY).
RestingBP	Numerical	Resting blood pressure measured in mmHg.
Cholesterol	Numerical	Serum cholesterol level in mg/dL.
FastingBS	Binary	Fasting blood sugar: 1 if fasting blood sugar > 120 mg/dL, otherwise 0.
RestingECG	Categorical	Results of resting electrocardiogram: Normal, ST-T wave abnormality (ST), or Left Ventricular Hypertrophy (LVH).
MaxHR	Numerical	Maximum heart rate achieved during physical activity.
ExerciseAngina	Binary	Presence of exercise-induced angina: Y (Yes) or N (No).
Oldpeak	Numerical	ST depression induced by exercise relative to rest.
ST_Slope	Categorical	Slope of the peak exercise ST segment: Up (Upsloping), Flat, or Down (Downsloping).
Smoking	Binary	Smoking status of the patient: YES (Smoker) or NO (Non-smoker).
HyperTension	Binary	Hypertension status: YES if patient has high blood pressure, otherwise NO.
Diabetes	Binary	Presence of diabetes: 1 (Yes) or 0 (No).
FamilyHistory	Binary	Family history of heart disease: YES (Yes) or NO (No).
HeartDisease	Multi Class	Target variable indicating presence of heart disease: 2 (Yes) , 1 (Maybe) , 0 (No).

5.3 Data Preprocessing

Preprocessing is a crucial step to ensure the dataset is clean, structured, and ready for training machine learning models. This section describes the preprocessing techniques applied to the dataset, explaining the methodology and reasons behind each step.

Figure 5.1 illustrates the complete data preprocessing flow, detailing the steps involved in preparing raw data for analysis. The diagram outlines processes such as data cleaning, normalization, and feature extraction, ensuring the dataset is structured and optimized for model training and prediction.

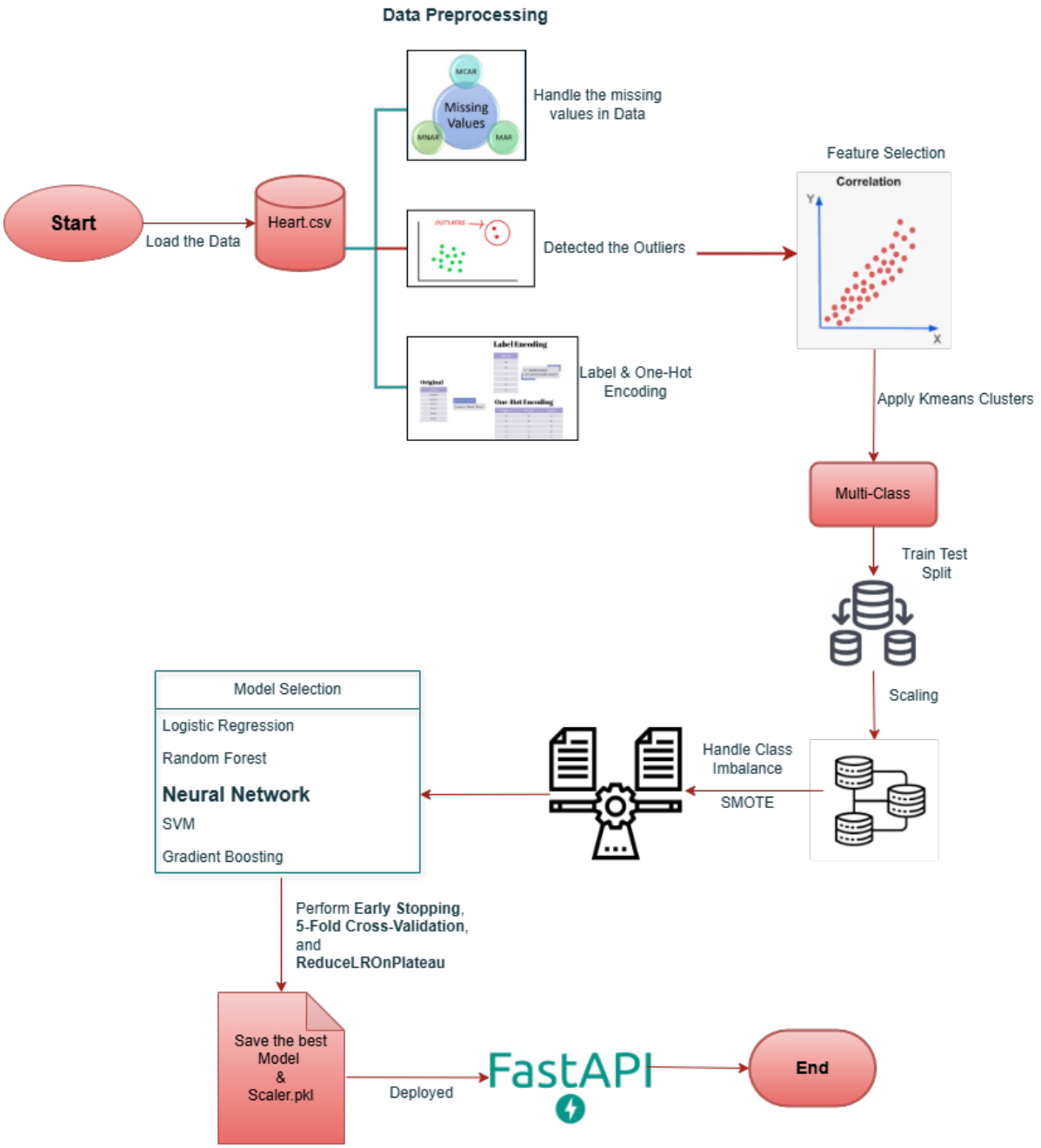


Figure 5.1: Data Preprocessing Complete Flow diagram

5.3.1 Handling Missing Values

5.3.1.1 Description

Missing values in a dataset can cause errors or biases in model training. To address missing values:

- For **numerical columns**, the missing values are replaced with the **mean** of the respective column.

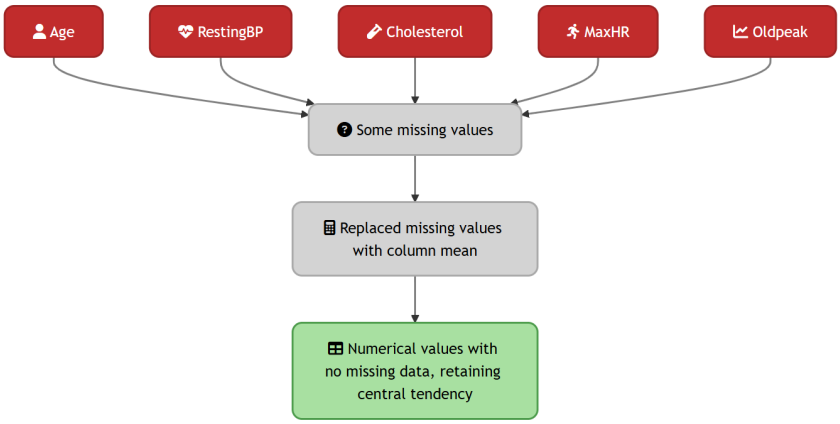


Figure 5.2: Numerical missing values Transformation.

Figure 5.2 illustrates the process of handling missing numerical values in a dataset. Initially, the features **Age, RestingBP, Cholesterol, MaxHR, and Oldpeak** are identified, with some missing values. These missing values are then replaced with the mean of the respective columns, ensuring that the dataset retains its central tendency.

- For **categorical columns**, the missing values are replaced with the **mode** (most frequent value).

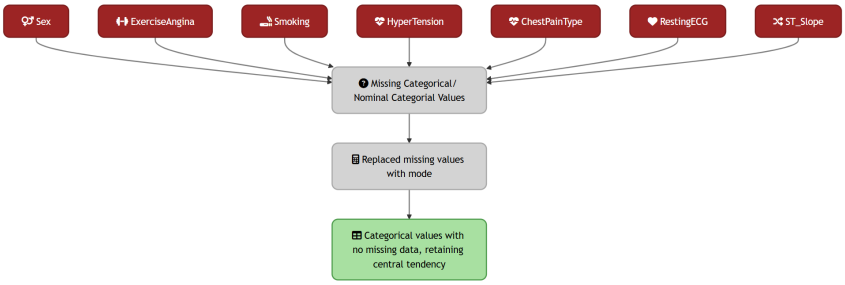


Figure 5.3: Categorial missing values Transformation.

Figure 5.3 illustrates the process of handling missing nominal/categorical values in a dataset. The features **Sex, ExerciseAngina, Smoking, HyperTension, ChestPainType, RestingECG, and ST_Slope** are initially identified, with some missing values. These missing values are then replaced with the mode of each respective column, ensuring that the dataset retains its central tendency.

5.3.1.2 Reasoning

- **Mean for Numerical Data:** It ensures the central tendency of the column remains unaffected without distorting the data distribution. Median or other imputation techniques were not chosen as they may overly simplify the data.
- **Mode for Categorical Data:** Using the most frequent value maintains consistency and does not introduce outliers or artificial categories.

5.3.1.3 Impact

This approach maintains data integrity and avoids information loss caused by dropping rows or columns with missing values.

5.3.2 Encoding Categorical Variables

To prepare the categorical variables for model training, Label Encoding and One-Hot Encoding were applied based on the type of variable.

Table 5.3: Encoding Methods and Their Application

Encoding Method	**Applied Columns**	**Reason**
Label Encoding	Sex, ExerciseAngina, Smoking, FamilyHistory, HyperTension	These are binary (two-category 0 or 1) variables.
One Hot Encoding	ChestPainType, RestingECG, ST_Slope	These are nominal categorical variables with no order.

Table 5.3 summarizes the encoding techniques applied to categorical variables in the dataset. Label Encoding was used for binary features like **Sex, ExerciseAngina, Smoking, FamilyHistory, and HyperTension**, converting them into integer values. One-Hot Encoding was applied to nominal features such as **ChestPainType, RestingECG, and ST_Slope,** creating separate binary columns for each category.

5.3.2.1 Reasoning

- **Label Encoding:** Converts binary variables into integers (0 and 1), making them suitable for numerical models.
- **One-Hot Encoding:** Ensures nominal variables with more than two categories are represented without introducing an artificial order. *drop_first=False* prevents loss of information.

5.3.2.2 Impact

This encoding process allows the categorical variables to be effectively incorporated into the model without creating biases or misleading patterns.

5.3.3 Outlier Detection and Removal

5.3.3.1 Description

Outliers are data points that deviate significantly from other observations and may lead to skewed results in statistical analyses and machine learning models. In this project, outliers in the numerical features of the dataset were detected and removed using the **Interquartile Range (IQR)** method.

The IQR method helps identify outliers by defining the lower and upper bounds of the data distribution:

- **Q1 (1st Quartile)** is the value below which 25% of the data falls.
- **Q3 (3rd Quartile)** is the value below which 75% of the data falls.
- **IQR** is the difference between Q3 and Q1 (i.e., **IQR = Q3 - Q1**).

Outliers are defined as data points that lie beyond the range calculated by:

$$\text{Lower Bound} = Q_1 - 1.5 \times \text{IQR}$$

The upper bound is given by:

$$\text{Upper Bound} = Q_3 + 1.5 \times \text{IQR}$$

Any data point below the lower bound or above the upper bound is considered an outlier and can be removed to improve the model's performance.

5.3.3.2 Visualizing Outliers

Box plots are used to visually represent the distribution of the numerical features and highlight any outliers. These plots show the median, quartiles, and potential outliers for each feature, making it easier to identify data points that fall outside the acceptable range.

The following numerical features were considered for outlier removal:

- **Age:** The age of the patients.
- **RestingBP:** The resting blood pressure of the patients.
- **Cholesterol:** The cholesterol level of the patients.
- **MaxHR:** The maximum heart rate achieved during exercise.
- **Oldpeak:** The depression induced by exercise relative to rest.

Figure 5.4 presents a boxplot used for visualizing outliers in the dataset. The plot highlights the distribution of data, including the median, quartiles, and any potential outliers that fall outside the whiskers, aiding in identifying anomalies for further analysis.

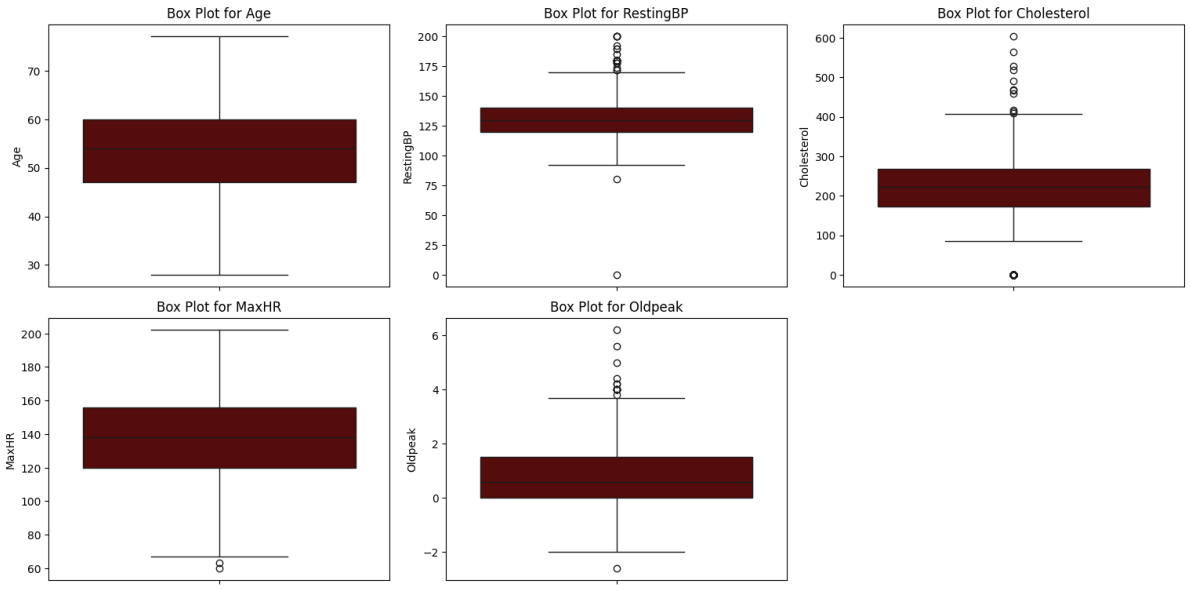


Figure 5.4: Boxplot for visulaizing Outliers

5.3.3.3 Impact of Removing Outliers

Initially, outliers were removed, and the model's accuracy was evaluated. While this approach improved accuracy by reducing noise and ensuring better distribution, it was ultimately not used in the final analysis.
In the medical field, every data point is crucial, as outliers may represent rare but significant cases or anomalies critical for accurate diagnosis and treatment. Therefore, the outlier dataset was retained to ensure the model accounts for all variations in the data, aligning with the importance of preserving medical integrity.

5.3.4 Correlation of Features

5.3.4.1 Description

Correlation measures the statistical relationship between two features in a dataset, quantifying how much one feature changes with respect to another. It is represented as a value between -1 and +1:

- **Positive Correlation** (+1)**:** As one feature increases, the other feature increases proportionally.
- **Negative Correlation** (−1)**:** As one feature increases, the other feature decreases proportionally.
- **No Correlation** (0)**:** There is no linear relationship between the two features.

In the context of our dataset, correlation helps us identify relationships between numerical features such as **Age, RestingBP, Cholesterol, MaxHR, and Oldpeak** allowing us to evaluate how features influence each other.

5.3.4.2 Correlation Matrix Analysis

To understand the relationships between features in our dataset, we computed the correlation matrix for all features.

Figure 5.5 displays the correlation matrix, illustrating the relationships between various features in the dataset. Each cell indicates the strength and direction of correlation, helping to identify highly correlated features for better feature selection and analysis.

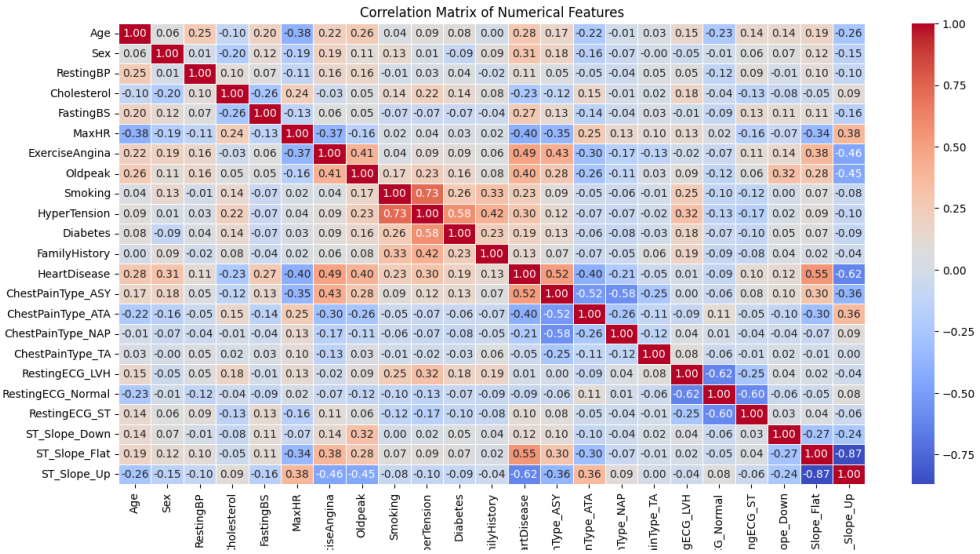


Figure 5.5: Correlation Matrix for CardioCare

5.3.4.3 Impact of Retaining All Features

- **Correlation Analysis:** Through correlation analysis, the features `['Oldpeak', 'MaxHR', 'Cholesterol', 'ExerciseAngina', 'ChestPainType_ASY', 'ST_Slope_Flat']` were identified as positively correlated with heart disease. These features were prioritized for their significant relationship with the target variable.
- **Preparation for K-Means Clustering:** The correlation analysis was a critical step before applying K-Means clustering. Selecting positively correlated features ensures meaningful clusters that accurately group patients based on heart disease risk factors.
- **Improved Data Insight:** By retaining these key features, the analysis focuses on attributes directly influencing heart disease, providing deeper insights for clustering and predictive modeling. .

5.3.5 Clustering and Multi-Class Classification

5.3.5.1 Description

Clustering is an unsupervised machine learning technique used to group similar data points together based on their characteristics. In this context, clustering helps identify patterns or hidden structures in the data without prior knowledge of class labels. The goal is to divide a set of data points into clusters where data points within each cluster are more similar to each other than to those in other clusters.
For heart disease diagnosis, clustering can help us group individuals based on features that might indicate varying levels of risk or stages of heart disease. This grouping can

be used to guide further classification tasks or to gain insights into the structure of heart disease data.

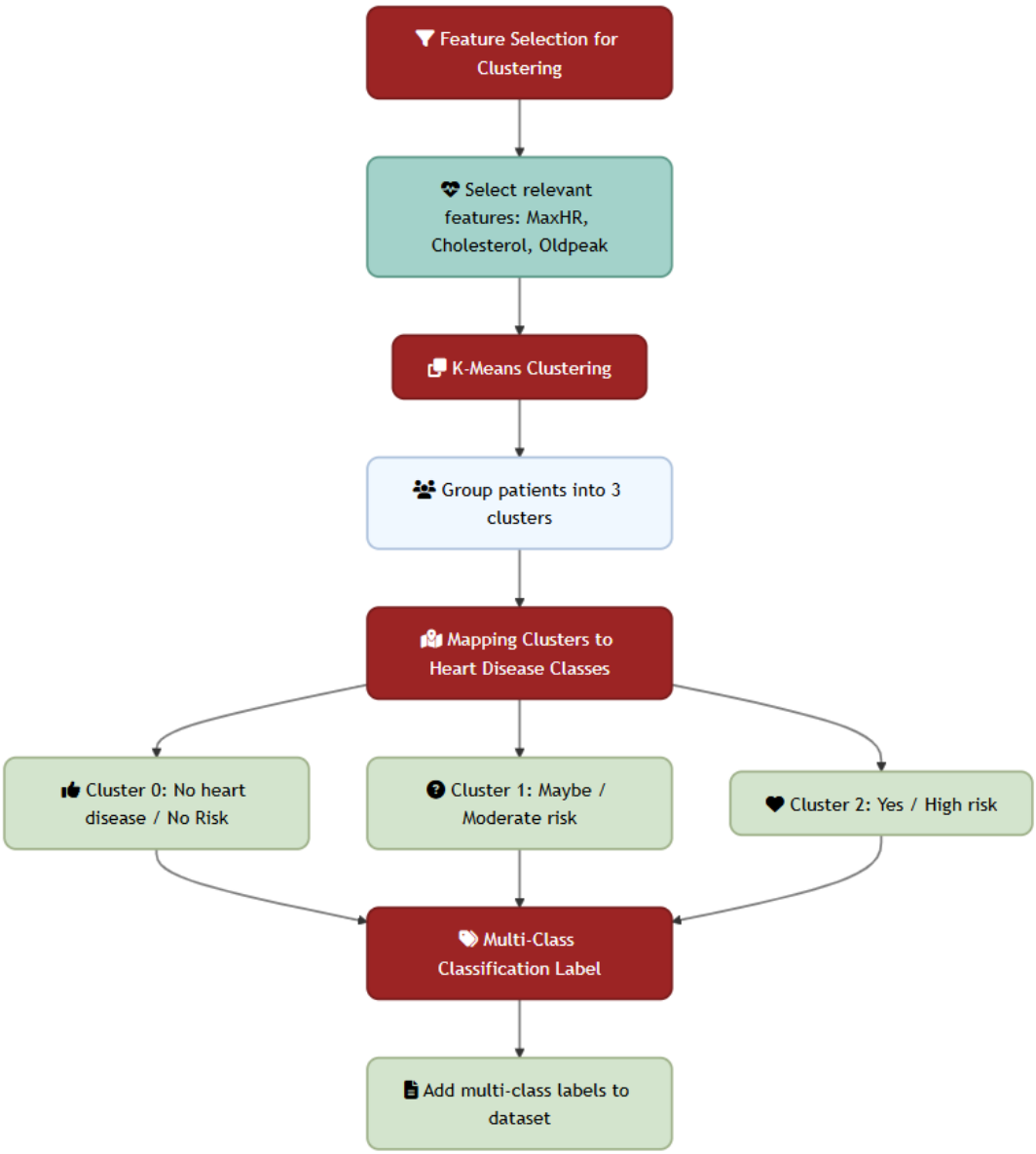


Figure 5.6: Process Flow of Multi-Class Classification.

Figure 5.6 depicts the process of generating multi-class classification labels for heart disease diagnosis. It begins with selecting key features like MaxHR, Cholesterol, and Oldpeak, followed by grouping patients into three clusters using K-Means clustering. The clusters are then mapped to heart disease risk levels: Cluster 0 (No risk), Cluster 1 (Moderate risk), and Cluster 2 (High risk). Finally, the resulting multi-class labels are added to the dataset, providing a structured representation of heart disease likelihood.

5.3.5.2 Techniques Used in Multi-Class Classification

1. **K-Means Clustering**:
   K-Means is an efficient unsupervised machine learning algorithm for clustering data into a specified number of groups (clusters). It minimizes the within-cluster variance by iteratively assigning data points to the nearest cluster center.

2. **Label Mapping**:
   Once the clusters are formed, we use domain knowledge to map the clusters to clinically relevant heart disease classes. This mapping is crucial for interpreting the results of the clustering process and assigning meaningful labels to each data point.

5.3.5.3 Classes Formed in the Multi-Class Classification

The following three classes are formed as a result of the clustering process:

Cluster Number	Class Label	Risk Level	Description
0	No	Low Risk	Patients in this cluster have the lowest likelihood of heart disease. These individuals typically show normal values for key features like cholesterol and exercise-induced pain.
1	Maybe	Moderate Risk	This cluster represents patients who show signs of heart disease risk but are not in an advanced stage. Their features, such as cholesterol and exercise response, suggest moderate concern.
2	Yes	High Risk	Patients in this cluster exhibit high levels of key risk factors, such as elevated cholesterol, abnormal heart rate response, and chest pain during exercise, indicating a high likelihood of heart disease.

5.3.6 Scaling the Dataset

Scaling is a crucial preprocessing step in machine learning and data analysis, particularly when numerical features in the dataset have varying scales or units. Without scaling, features with larger numerical ranges might disproportionately influence the model, leading to biased predictions and slower convergence.

5.3.6.1 Description

Scaling adjusts the range or distribution of feature values so that all numerical features contribute equally to the machine learning model. The goal of scaling is to normalize the dataset while preserving the relationships between features.

5.3.6.2 Features Needing Scaling

The following numerical columns in the dataset require scaling:

- **Age**: Age of the patient (measured in years).
- **RestingBP**: Resting blood pressure (measured in mmHg).

- **Cholesterol**: Cholesterol level (measured in mg/dL).
- **MaxHR**: Maximum heart rate achieved (measured in beats per minute).
- **Oldpeak**: ST depression induced by exercise relative to rest.

These columns have varying units and ranges, and scaling them ensures they contribute uniformly during model training.

5.3.6.3 Impact of Scaling

Scaling has significant benefits on the dataset and model performance:

1. **Faster Convergence**
   - Scaling reduces the range of values, improving the efficiency of optimization algorithms.
   - This leads to faster convergence during training, particularly for neural networks and gradient-boosted models.
2. **Improved Accuracy**
   - Uniformly scaled features prevent any one feature from disproportionately affecting model learning.
   - This leads to a more balanced and accurate model.
3. **Consistent Interpretability**
   - After scaling, coefficients and feature importance scores become comparable.
   - This improves interpretability for linear models.

5.3.7 Data Splitting

5.3.7.1 Description

Data splitting is a crucial step in preparing a dataset for machine learning models. It involves dividing the dataset into separate subsets for training and testing. The purpose is to evaluate the model's performance on unseen data to ensure its ability to generalize beyond the data it was trained on. In our project, we split the dataset into training and testing sets to achieve reliable and unbiased results.

5.3.7.2 How the Data is Divided

1. **Train and Test Sets:**
   - **Training Set:** This subset of the data is used to train the machine learning model. It helps the model learn patterns, relationships, and dependencies between input features (X) and the target labels (y).
   - **Testing Set:** This subset is reserved for evaluating the model's performance on data it has not seen before, helping us assess the model's generalization capability.
2. **Splitting Ratio:**
   In this project, the dataset is divided into:
   - **80% Training Data:** Used to build and train the model.
   - **20% Testing Data:** Used to validate the model's performance on unseen data.

5.3.8 Handling Class Imbalance

5.3.8.1 Description

Class imbalance is a common challenge in machine learning, especially when the distribution of target classes is uneven. In our project, we addressed class imbalance by applying **Synthetic Minority Oversampling Technique (SMOTE).** This technique is designed to balance the dataset by generating synthetic samples for the minority classes, thereby ensuring that the model does not become biased toward the majority class.

5.3.8.2 SMOTE Process

- **SMOTE** works by creating synthetic examples of the minority class by selecting instances that are close in the feature space, drawing a line between the instances, and creating new synthetic instances along that line.
- The result is a more balanced dataset where each class has a similar number of instances, allowing the model to learn better decision boundaries.
- **Before SMOTE:** Class 1 (Maybe) had a significantly smaller number of instances (147) compared to Class 0 (No) and Class 2 (Yes), leading to potential bias in the model toward the majority classes.
- **After SMOTE:** SMOTE generates synthetic samples for Class 1, bringing the count of all classes closer to 325, making the dataset more balanced. This allows the model to train effectively on all classes.

Table 5.5 illustrates the class distribution before and after applying SMOTE. Before SMOTE, Class 0 (No) had 262 instances, Class 1 (Maybe) had 147, and Class 2 (Yes) had 325. After SMOTE, the minority class (Class 1) was oversampled, resulting in a balanced dataset with 325 instances in each class. This ensures the model is not biased towards the majority class and improves prediction accuracy across all classes.

Table 5.5: Class Distribution Before and After SMOTE

Class	Before SMOTE	After SMOTE
Class 0 (No)	262	325
Class 1 (Maybe)	147	325
Class 2 (Yes)	325	325

5.3.8.3 Impact of SMOTE

- **Balanced Model Performance:** By applying SMOTE, we ensure that the model does not show bias toward the majority class, leading to better generalization across all classes. It improves the model's ability to predict all classes more accurately, especially the minority class (Maybe).

- **Improved Metrics:** After balancing the dataset with SMOTE, the model's performance on all classes improves. Metrics like accuracy, precision, recall, and F1 score are more reliable because the model treats all classes with equal importance.

5.4 Model Selection and Evaluation

In this section, we provide a comprehensive explanation of the model selection process, including the models used, the reasoning behind choosing the final model, evaluation criteria, and steps taken to prevent overfitting. This approach ensures that the best performing model for our heart disease diagnosis system is chosen, along with the associated metrics to evaluate its effectiveness.

5.4.1 Model Selection

1. **Choice of Neural Network** Neural networks were chosen as the primary model for this project due to their ability to capture complex, non-linear relationships within data. Here's why we opted for a neural network:
    - **Non-linearity Handling:** Heart disease prediction often involves complex relationships between features (e.g., age, cholesterol levels, ECG signals) that may not be linearly separable. Neural networks, with multiple layers and activation functions like ReLU (Rectified Linear Units), are ideal for learning such non-linear relationships.
    - **Ability to Learn Complex Patterns:** The nature of the dataset, including numerical and categorical features demands a model that can handle multiple types of inputs effectively. Neural networks excel at extracting intricate patterns from large datasets, making them well-suited for this task.
    - **Adaptability:** Neural networks can easily be adjusted in terms of the number of layers, the number of neurons in each layer, and the choice of activation functions. This adaptability allows fine-tuning for optimal performance.

2. **Neural Network Architecture** The chosen neural network model is a feed-forward architecture, consisting of:
    - **Input Layer:** Matching the number of features in the dataset (e.g., age, cholesterol, ECG features).
    - **Hidden Layers:** 2 layers with ReLU activation to introduce non-linearity.
    - **Output Layer:** A softmax activation function to produce probabilities for the multiclass classification problem (e.g., No Risk, mild risk, severe risk).

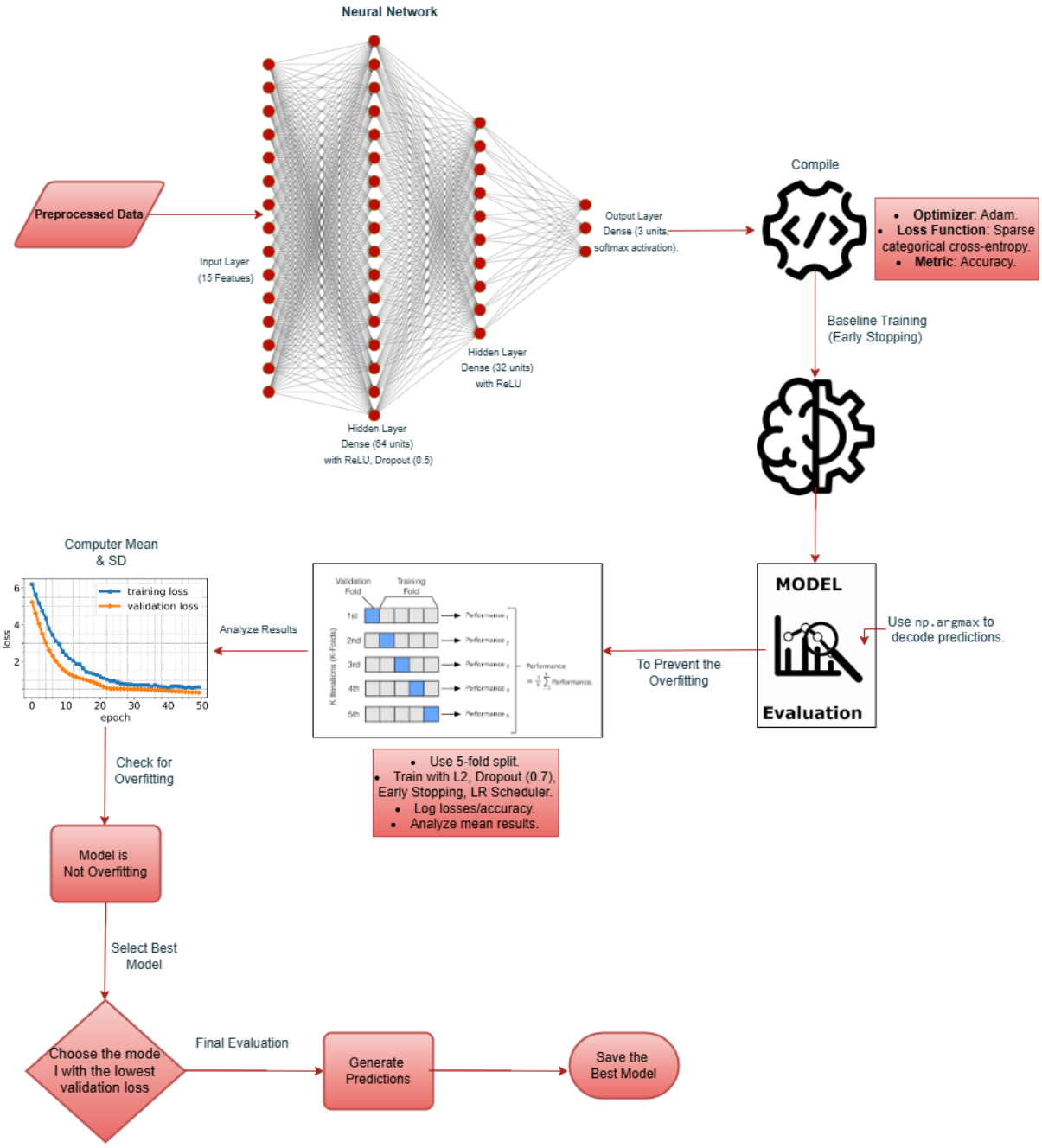


Figure 5.7: Overview of the neural network architecture.

Figure 5.7 provides an overview of the neural network architecture, highlighting its key components and functionality. The input layer incorporates essential features such as Age, Cholesterol, and ECG attributes, serving as the foundation for the model. The hidden layers, equipped with **ReLU** activation functions, introduce non-linearity to enhance the model's capacity for learning complex patterns. Finally, the output layer employs a **Softmax activation function** to produce probabilities for multiclass classification, categorizing patients into **Yes,No or Maybe.** This architecture is designed to effectively process input features and deliver accurate heart disease predictions.

5.4.2 Techniques to Prevent Overfitting

Overfitting occurs when the model learns noise in the training data, resulting in poor generalization to new data.Overfitting was a challenge encountered during model development, especially with the initial high accuracy of 0.94 but poor generalization. To address this, the following techniques were implemented:

- **Early Stopping:** Early stopping was employed to monitor validation loss during training. Training was halted when the validation loss stopped improving for a specified number of epochs, preventing overfitting by stopping before the model started fitting noise in the training data.
- **Dropout Regularization:** Dropout layers were added to randomly deactivate neurons during training, reducing the risk of the model becoming too reliant on specific neurons. Initially, a dropout rate of 0.5 was used and later increased to 0.7 for better regularization.
- **L2 Regularization:** L2 regularization (Ridge) was applied to the model layers to penalize large weights, ensuring that the model does not overfit to specific patterns in the training data. This technique was particularly helpful in controlling the model's complexity.
- **K-Fold Cross-Validation:** A 5-fold cross-validation approach was used to evaluate model performance. The dataset was split into five subsets, with each subset used as a validation set once while the others were used for training. This ensured the model generalized well to unseen data, and overfitting due to random splits was minimized.
- **Learning Rate Scheduler:** A learning rate scheduler (ReduceLROnPlateau) was implemented to dynamically reduce the learning rate when validation loss plateaued. This helped the model converge more effectively during later stages of training.
- **Comparative Training Analysis:** Training and validation losses were analyzed across epochs for all cross-validation folds. Loss curves were plotted to monitor trends, ensuring the model's behavior was consistent across all subsets of data.
- **Best Model Selection:** The model with the lowest validation loss during cross-validation was selected as the final model. This ensured optimal performance on the test set, reducing the risk of overfitting.
- **Performance Evaluation:** The model was tested on unseen data using the test set. A confusion matrix and classification report were generated to evaluate accuracy and ensure balanced predictions across all classes.
- **Final Results:** After implementing these techniques, the average accuracy across all folds improved significantly, and the model demonstrated strong generalization capabilities with reduced overfitting.

Table 5.6: Overview of Regularization and Validation Techniques

Technique	**Description**
L2 Regularization	Penalizes large weights to reduce overfitting.
Early Stopping	Stops training when validation loss stops improving.
K-fold Cross-Validation	Splits data into K subsets to ensure robustness.
Dropout Layers	Randomly deactivates neurons during training to prevent reliance on specific ones.

Table 5.6 summarizes the regularization and validation techniques utilized to enhance model performance and prevent overfitting. These include L2 Regularization, Early Stopping, K-fold Cross-Validation, Dropout Layers. Together, these methods contribute to building a robust and reliable model.

5.4.3 Model Evaluation Metrics

The performance of the model was evaluated using several key metrics, each providing insight into different aspects of the model's predictive power.

Figure 5.8 illustrates a comprehensive overview of the evaluation metrics used to assess the performance of the model. The flowchart outlines key metrics such as **Accuracy, Precision, Recall, F1-Score, and the Confusion Matrix,** along with their respective formulas and relevance to the heart disease prediction task. Each metric is connected to a formula that helps quantify the model's performance, and its relevance is described in the context of the specific challenges posed by imbalanced datasets. This figure highlights the importance of using a combination of these metrics to obtain a thorough understanding of the model's effectiveness, ensuring both accuracy and reliability in predictions.

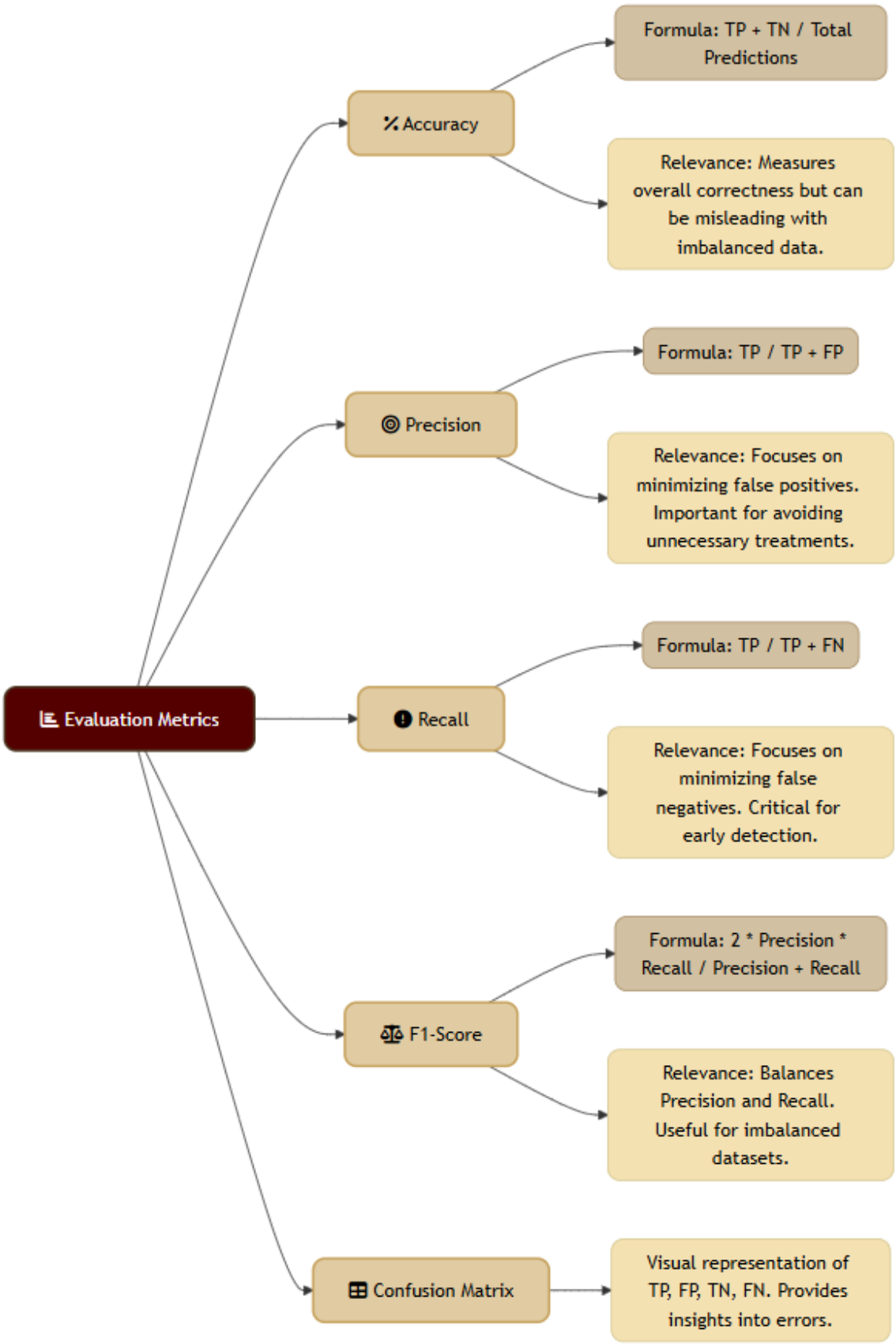


Figure 5.8: Evaluation metrics used to assess the model's performance

1. **Accuracy**
   Accuracy is defined as the ratio of correct predictions to the total number of predictions made:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Predictions}}$$

   Relevance: Accuracy is a common metric, but for imbalanced datasets, it may not provide a full picture. Hence, we also use additional metrics like precision and recall.
2. **Precision**
   Precision indicates how many of the predicted positives were actually positive:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

   where TP = True Positives, FP = False Positives.
   Relevance: In our case, high precision is important because false positives (e.g., incorrectly diagnosing a patient as having heart disease) can lead to unnecessary treatments.
3. **Recall**
   Recall shows how many of the actual positives were correctly identified:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

   where FN = False Negatives.
   Relevance: For heart disease prediction, it is crucial to minimize false negatives (e.g., failing to diagnose a patient who actually has heart disease), as this could lead to missed opportunities for early treatment.
4. **F1-Score**
   The F1-score is the harmonic mean of precision and recall:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

   Relevance: F1-score balances precision and recall, providing a more comprehensive evaluation, especially in imbalanced datasets like ours.
5. **Confusion Matrix**
   A confusion matrix is used to summarize the performance of the classification model by showing the true positives, false positives, true negatives, and false negatives. This matrix gives a clear picture of where the model is making errors.

5.4.4 Findings and Results

The results of the neural network model demonstrated significant improvement after addressing the overfitting issue. Initially, the model exhibited an accuracy of 0.94, but it was overfitting the training data, as evident from a high disparity between training and validation loss. By implementing techniques such as dropout, L2 regularization, early stopping, K-fold cross-validation, and a learning rate scheduler, the model's performance on unseen data improved significantly.

5.4.4.1 Model Training:

During the training process, the model went through several epochs to optimize its performance. Below is a snapshot of the training process in Figure 5.9, which illustrates the model's progression over time. The training loss and validation loss curves reflect the impact of the overfitting prevention techniques, showing improved convergence.

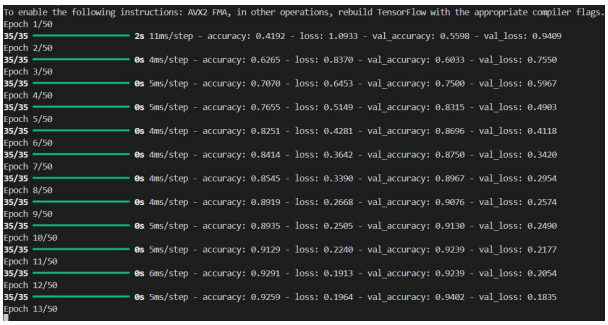


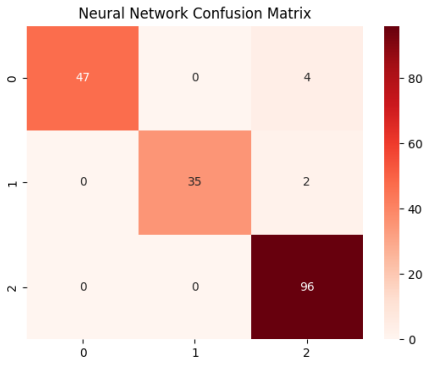
Figure 5.9: Model Training Over Epochs

5.4.4.2 Performance Analysis:

After applying the overfitting prevention techniques, the model achieved balanced performance across all classes. The confusion matrix revealed improved classification accuracy, with most predictions falling into their correct categories. Additionally, the precision, recall, and F1-score for each class indicated that the model generalized well and minimized misclassifications.

- **Confusion Matrix:** The confusion matrix, as in Figure 5.10, showcases the model's ability to correctly classify instances for each category. It demonstrates a significant reduction in misclassifications compared to the initial results.
- **Precision and Recall:** The model's precision and recall for all classes reached competitive levels as shown in Figure 5.10, indicating that it effectively identified true positives while minimizing false positives and false negatives.

The figures below represent the confusion matrix and the precision report obtained after implementing the techniques to combat overfitting:



Neural Network Evaluation on Test Set:

	precision	recall	f1-score	support
0	1.00	0.92	0.96	51
1	1.00	0.95	0.97	37
2	0.94	1.00	0.97	96
accuracy			0.97	184
macro avg	0.98	0.96	0.97	184
weighted avg	0.97	0.97	0.97	184

Figure 5.10: Confusion Matrix and Precision Metrics After Overfitting Prevention

5.4.4.3 Model Evaluation:

The following ROC curve as in Figure 5.11 shows the performance of the model across multiple classes. As seen, the model demonstrates a strong ability to differentiate between classes, with AUC scores providing valuable insights into its effectiveness.

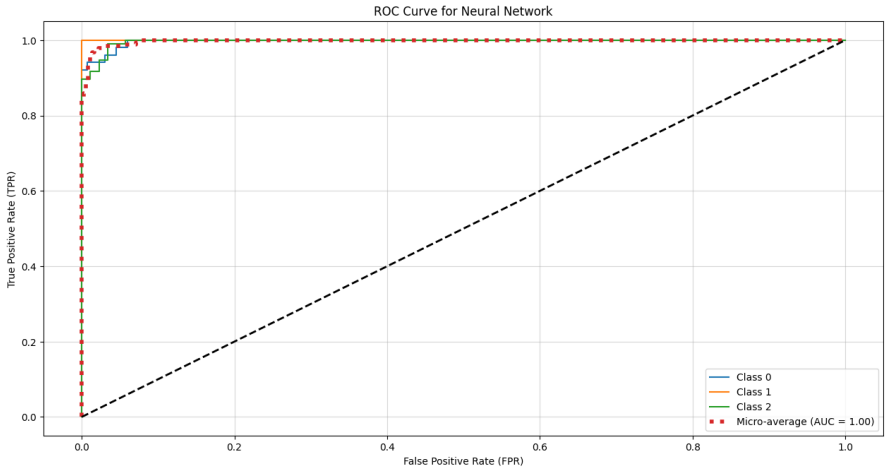


Figure 5.11: ROC Curve for the Neural Network Model

The AUC values, precision-recall curves, and confusion matrix highlight the robustness of the model in making accurate predictions. Various regularization techniques and cross-validation strategies helped prevent overfitting and ensured that the model generalizes well to unseen data.

5.4.5 Comparison with Other Models

Models Evaluated:

- **Logistic Regression:** While logistic regression serves as a useful baseline for multiclass classification tasks, it has limitations in capturing complex relationships within the data. Due to its simplicity, it tends to perform poorly with datasets where intricate patterns are key to accurate predictions, making it unsuitable for this task.
- **Random Forest:** Random Forest is an ensemble method that works well for feature importance and classification tasks, but it struggles with capturing deeper, more complex relationships within the data. Despite its robustness, its decision-making process can be too shallow for problems requiring nuanced pattern recognition, leading to suboptimal performance when compared to more sophisticated models like neural networks.
- **XGBoost:** Although XGBoost is a powerful gradient boosting model known for high performance in structured data, it requires extensive hyperparameter tuning and can be computationally expensive. Given the complexity of this particular task and the need for scalability, it wasn't the best choice for achieving optimal results without additional fine-tuning.
- **Support Vector Machine (SVM):** SVM with a non-linear RBF kernel can be effective for classification tasks but is highly sensitive to feature scaling. Moreover,

SVM models tend to be computationally expensive, especially for larger datasets. Given the nature of the problem, a neural network approach provided a better balance of efficiency and accuracy.

Reasons for Choosing Neural Networks:

- The complexity of the problem, specifically converting the binary classification problem into a multiclass classification problem, made neural networks the ideal choice.
- Neural networks are well-suited for capturing and modeling intricate patterns and relationships in the data.
- They excel in handling high-dimensional data and learning complex, non-linear relationships.
- Neural networks are more flexible in adapting to data variations, which leads to a performance boost compared to simpler models that rely on linear decision boundaries.
- This flexibility and ability to handle complex data led to superior accuracy in the multiclass classification task.

Table 5.7: Comparison among Models

Model	Accuracy (%)	Precision (%)			Recall (%)			F1-Score (%)		
		No	Maybe	Yes	No	Maybe	Yes	No	Maybe	Yes
Logistic Regression	0.97	1.00	1.00	0.95	0.92	0.97	1.00	0.96	0.97	1.00
Random Forest	0.96	0.94	0.97	0.93	0.92	0.87	0.95	0.98	0.93	0.97
XGBoost	0.95	0.94	0.91	0.94	1.00	0.95	0.92	0.89	1.00	0.95
Support Vector Machine	0.93	0.96	1.00	0.90	0.84	0.95	0.98	0.90	0.97	0.94
Neural Network	0.97	1.00	1.00	0.94	0.92	0.95	1.00	0.96	0.97	0.97

5.4.6 Final Model Selection

The chosen model, a Feed-Forward Neural Network, demonstrated superior performance across evaluation metrics. The decision to opt for neural networks was further supported by the following considerations:

- **High Precision and Recall:** Balancing both metrics ensures that we are not only identifying most patients with heart disease (high recall) but also minimizing false alarms (high precision).
- **Non-Linearity:** The dataset includes complex relationships between features that are best captured by the neural network's layered structure.

The model was trained with early stopping, dropout, and regularization to prevent overfitting and ensure the best performance on unseen data.

5.5 OCR Model Integration

The integration of Optical Character Recognition (OCR) into the **CardioCare AI:** Intelligent Heart Disease Prediction and Diagnosis System allows for the extraction of

important medical data from scanned or image-based lab reports, such as cholesterol levels, blood pressure values, and ECG readings. OCR serves as a bridge between unstructured image data and structured information that can be used for further analysis and diagnosis.

5.5.1 Functionality

OCR plays a vital role in the extraction of text data from scanned reports and images, making it useful for the automated analysis of lab reports without manual entry. Below are the key functionalities and how OCR operates within our project.

5.5.1.1 Text Preprocessing and Optimization

- **Image Preprocessing**: Preprocessing techniques, such as resizing (applied one), grayscale conversion, noise reduction, and thresholding, enhance OCR accuracy by improving the clarity of text within images.
- **Text Extraction**: Using libraries like Tesseract (via pytesseract), the processed images are converted into machine-readable text, which can then be parsed for relevant medical data.

5.5.2 OCR Model Integration Steps

The OCR process in the project involves several stages, from uploading the lab report image to extracting structured data for further processing.

Figure 5.12 illustrates the OCR flowchart diagram, which outlines the process of extracting fields from a **lab report**. The system utilizes Optical Character Recognition (OCR) to analyze the report and identify relevant information. The extracted data is then structured into a **JSON format** for further processing, ensuring seamless integration with the CardioCare AI system.

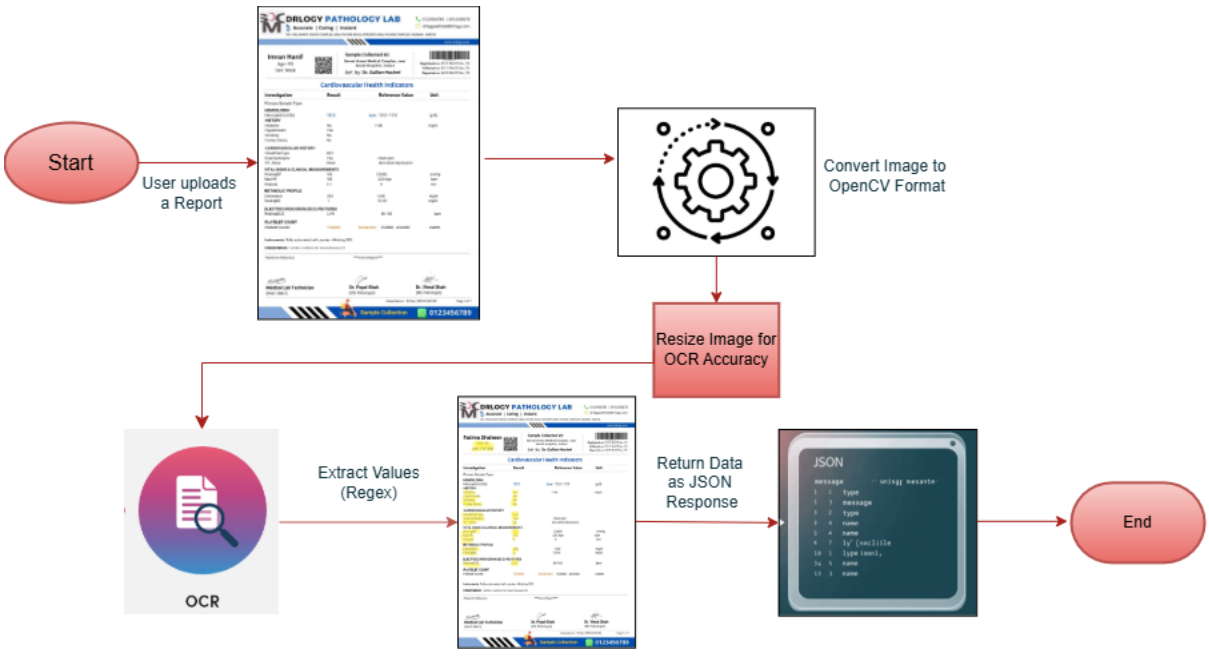


Figure 5.12: OCR Flowchart

5.5.2.1 Upload Lab Report

Users can upload scanned lab reports in image format (e.g., PNG, JPG) for processing. The uploaded image is sent to the backend system for OCR processing.

Figure 5.13 illustrates a sample lab report used as input for the OCR-based text extraction process. The report contains key medical fields, including cholesterol levels, blood pressure, and ECG values, formatted as structured text within the image. The OCR system identifies and extracts these fields by preprocessing the image and parsing the text to retrieve relevant medical data. This processed information is then structured into a machine-readable format for further analysis and integration into predictive models.

DRLOGY PATHOLOGY LAB
Accurate | Caring | Instant
0123456789 | 0912345678
drlogypathlab@drlogy.com
105 -108, SMART VISION COMPLEX, HEALTHCARE ROAD, OPPOSITE HEALTHCARE COMPLEX. MUMBAI - 689578
www.drlogy.com

Sharafat Ali
Age: 28
Sex: Male

Sample Collected At:
Sarwat Anwar Medical Complex , near Jinnah Hospital , Lahore
Ref. By: **Dr. Gulfam Hashmi**

Registered on: 02:31 PM 02 Dec, 2X
Collected on: 03:11 PM 02 Dec, 2X
Reported on: 04:35 PM 02 Dec, 2X

Cardiovascular Health Indicators

Investigation	Result	Reference Value	Unit
Primary Sample Type :			
HEMOGLOBIN			
Hemoglobin (Hb)	12.5	Low 13.0 - 17.0	g/dL
HISTORY			
Diabetes	No	<100	mg/dl
Hypertension	No		
Smoking	No		
Family History	No		
CARDIOVASCULAR HISTORY			
ChestPainType	ATA		
ExerciseAngina	No	chest pain	
ST_Slope	Up	elevvation/depression	
VITAL SIGNS & CLINICAL MEASUREMENTS			
RestingBP	130	120/80	mmHg
MaxHR	185	220-Age	bpm
Oldpeak	0	0	mm
METABOLIC PROFILE			
Cholesterol	132	<200	mg/dl
FastingBS	0	70-99	mg/dl
ELECTROCARDIOGRAM (ECG) FEATURES			
RestingECG	LVH	60-100	bpm
PLATELET COUNT			
Platelet Count	150000	Borderline 150000 - 410000	cumm

Instruments: Fully automated cell counter - Mindray 300

Interpretation: Further confirm for heart disease(2)

Thanks for Reference ****End of Report****

Medical Lab Technician
(DMLT, BMLT)

Dr. Payal Shah
(MD, Pathologist)

Dr. Vimal Shah
(MD, Pathologist)

Generated on : 02 Dec, 202X 05:00 PM Page 1 of 1

Sample Collection 0123456789

Figure 5.13: Report Sample of Patient.

5.5.2.2 Preprocess Image

The image undergoes a preprocessing step to enhance text extraction:

- **Resizing**: The image is resized to enhance the visibility of characters for Optical Character Recognition (OCR). This step improves the accuracy of text extraction, especially when the text is too small to be recognized clearly.

Other preprocessing techniques were initially applied but were ultimately removed due to their negative impact on the image quality:

- **Grayscale Conversion**: Attempted to remove color information, but it reduced the clarity of the text and did not improve OCR accuracy.
- **Denoising**: Tried to reduce noise, but it introduced artifacts that made character recognition less reliable.
- **Dilation**: The text was dilated to enhance faint characters, but this distorted the text and made it harder to read.

5.5.2.3 Extract Text Fields

After preprocessing, OCR is applied to extract text from the image. Tesseract (pytesseract) is used for this purpose. The extracted text is parsed to identify and extract key medical fields such as: **Name, Age, Sex, Diabetes, Smoking, FamilyHistory, ChestPainType, ExerciseAngina, ST_Slope, RestingBP, MaxHR, Oldpeak, RestingECG**

5.5.2.4 Output Structured Data

Once the text is extracted, the next step is to process and structure the data into a machine-readable format (JSON, CSV, etc.). The structured data is then sent to the next stage of the system for use in prediction models, database storage, or analysis.Figure 5.14 presents the OCR output in JSON format, showcasing the structured data extracted from a lab report.

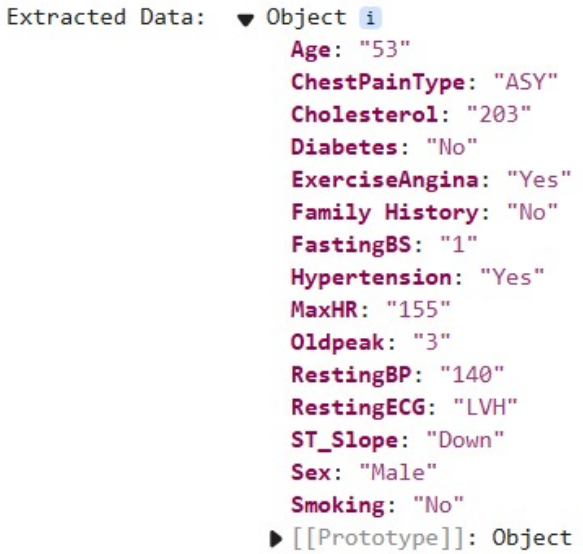


Figure 5.14: OCR Output in JSON format

6

Challenges and Limitations

While the CardioCare AI: Intelligent Heart Disease Prediction and Diagnosis System has demonstrated significant potential in predicting heart disease outcomes, there were several challenges and limitations encountered during its development and evaluation. This section highlights the key obstacles faced in the process of implementing the system, the constraints within the datasets used, and the areas where the system could be further improved. Understanding these challenges is essential for guiding future research and development to enhance the system's performance and applicability in real-world clinical settings.

6.1 Challenges in Data Acquisition and Integration

6.1.1 Data Quality and Completeness

One of the primary challenges encountered during the development of the system was related to the quality and completeness of the datasets. While the datasets from various Pakistani hospitals were invaluable, they were often incomplete, containing missing values or inconsistencies. This made data preprocessing and cleaning a time-consuming task. Missing data points in features such as cholesterol levels, blood pressure, and ECG values had to be handled carefully to avoid introducing bias into the model.

To address this challenge, techniques like mean imputation and interpolation were used for continuous features, while categorical variables were managed by grouping similar classes. However, this process was not perfect, and it is possible that some missing data may still have impacted the accuracy of the model.

6.1.2 Dataset Heterogeneity

The datasets collected from different hospitals exhibited a degree of heterogeneity, meaning that the data recorded by different institutions sometimes followed different standards, formats, or scales. For instance, blood pressure could be recorded as a range or a single value depending on the hospital, and ECG readings were provided in different formats, adding to the complexity of data integration.

To mitigate this, we standardized the features during the data preprocessing phase, ensuring uniformity across all data points. However, despite these efforts, the heterogeneity of the data posed challenges, especially in accurately merging the datasets to form a comprehensive training set.

6.2 Multiclass Classification Complexity

6.2.1 Ambiguity in the "Maybe" Class

In the multiclass classification model used in the CardioCare AI system (Yes, No, Maybe), the "Maybe" category, which represents an ambiguous or uncertain risk of heart disease, proved to be particularly challenging. The "Maybe" class introduces a level of ambiguity, as it represents cases where the system is unsure about the patient's condition based on the available data. While this class adds value by offering a more nuanced approach to prediction, it also presents challenges in terms of model accuracy, especially when the model has to make a decision about an unclear case.

The ambiguity of the "Maybe" class can cause misclassification in cases where the boundary between "Yes" and "Maybe" is not clear-cut. For instance, if a patient shows borderline symptoms or risk factors, the model may struggle to categorize them accurately. This could lead to incorrect diagnoses or delayed intervention for patients who need immediate care.

6.2.2 Data Imbalance in Multiclass Classification

Another challenge in multiclass classification is the imbalance between the classes. Although heart disease prediction is a critical issue, there are generally more "Yes" cases (i.e., patients with heart disease) than "No" or "Maybe" cases, especially in a population with a lower incidence of heart disease. This imbalance can lead to the model becoming biased towards the majority class, resulting in a lower sensitivity for predicting the minority classes, particularly the "No" and "Maybe" categories.

To address this issue, techniques like oversampling the minority class were implemented, but these approaches were not always fully effective in overcoming the imbalance.

6.3 Limitations in Model Evaluation

6.3.1 Lack of External Validation

While the CardioCare AI system was evaluated on the merged dataset from Pakistani hospitals, it still lacks broader external validation. External validation is a crucial step in assessing the generalizability of a model. Given that the dataset is region-specific (i.e., limited to Pakistani patients), it is possible that the model's performance could vary when applied to different populations or healthcare settings.

The model's effectiveness needs to be evaluated on international datasets or at least datasets from diverse regions to assess its true generalizability. Future improvements should include testing the system in different healthcare environments and across a broader demographic.

6.4 Technical and Computational Challenges

6.4.1 Neural Network Training Complexity

Training the neural network on a complex and large dataset introduced significant computational challenges. Despite optimizing the model's architecture, such as using early stopping and dropout layers to prevent overfitting, training times were still lengthy, particularly for models with more complex architectures.
Additionally, fine-tuning the hyperparameters, such as the learning rate, batch size, and number of hidden layers, was a time-consuming process, which may have impacted the model's efficiency during development. Real-time prediction of heart disease in clinical settings could also pose challenges, as model inference times need to be minimized for practical deployment.

6.4.2 Hardware Limitations for Large-Scale Deployment

While the system performs well on a single-instance scale, deploying it in real-world clinical settings with a large number of simultaneous users presents additional challenges. The hardware infrastructure must be capable of handling the high-volume processing required for real-time predictions and continuous data updates from patients. This presents challenges in terms of ensuring adequate scalability and availability, particularly when dealing with large-scale hospital systems or national health databases.

6.5 Ethical and Regulatory Challenges

6.5.1 Privacy and Data Security

Handling sensitive patient data introduces ethical and privacy concerns. Ensuring that the system adheres to relevant data protection laws is crucial to gaining trust from healthcare providers and patients.

6.5.2 Trust and Adoption in Clinical Settings

A major challenge in the healthcare domain is the trust and adoption of AI-powered systems. While the model has been validated by medical professionals, it is still crucial for healthcare providers to trust the recommendations made by the AI system. Overcoming resistance to change and ensuring that doctors feel confident in using the system as a diagnostic tool remains a significant challenge.
Moreover, the system should be seen as a support tool rather than a replacement for human expertise. Educating healthcare providers about the system's functionality, limitations, and potential benefits will be key to successful adoption in clinical environments.

7

Future Work

The CardioCare AI: Intelligent Heart Disease Prediction and Diagnosis System has achieved significant progress in heart disease prediction, utilizing various features and datasets to provide accurate results. However, there are several enhancements and extensions that can be made to the system to further improve its clinical applicability, accuracy, and usability. Due to time constraints, some of the proposed features were not implemented during this phase of the project but are crucial for the system's future development. These enhancements include integrating real-time data from wearable devices, enabling the upload and analysis of ECG images, and incorporating an additional dataset to predict artery narrowing.
This section discusses these planned improvements in detail, exploring how each can contribute to the overall effectiveness of the system and outlines potential future research directions.

7.1 Integration with Wearable Devices for Real-Time Data

7.1.1 Concept and Need

One of the most promising features planned for the future development of the CardioCare AI system is the integration with wearable devices such as smartwatches, fitness bands, and other medical-grade sensors. These devices can provide real-time health data, including heart rate, blood oxygen levels, activity levels, and ECG data. By continuously monitoring patients' vital signs, wearable devices can offer real-time feedback to the system, enabling continuous heart disease risk prediction and early detection of abnormal patterns that could lead to heart conditions.
The addition of wearable device integration will significantly improve the timeliness and accuracy of predictions. Real-time data can be used to adjust risk assessments on the fly, providing dynamic insights into the patient's health status and allowing for proactive interventions. Moreover, it can help create a longitudinal patient profile, allowing the system to track health trends over time and potentially identify early warning signs of heart disease.

7.1.2 Obstacles and Constraints

While the integration of wearable devices holds great potential, it also presents several challenges. These include:

- **Device compatibility:** Different wearable devices use different formats for data collection and transmission. Ensuring seamless integration with multiple devices

will require standardization or the development of custom APIs.

- **Data privacy and security:** Continuous data collection from wearable devices raises concerns about data privacy and security. It will be critical to ensure that the data is anonymized and encrypted to meet legal requirements and protect patients' sensitive health information.
- **Real-time processing:** The system will need to handle continuous streams of data and make real-time predictions. This will require a robust backend infrastructure capable of processing large volumes of incoming data with minimal latency.

7.1.3 Future Steps

The first step in implementing wearable device integration will be to conduct a feasibility study to assess the most suitable devices for the target population and the integration strategies required. Following this, API development will be necessary to allow smooth communication between the wearable devices and the CardioCare AI system. Real-time data handling and processing will be a priority to ensure timely risk assessment and prediction.

7.2 Uploading and Analysis of ECG Images

7.2.1 Concept and Need

Currently, the system focuses on predicting heart disease using structured features such as cholesterol levels, blood pressure, and other clinical measurements. However, ECG images—a critical diagnostic tool in heart disease detection—were not included in the initial implementation due to time constraints. In the future, the system aims to allow users to upload ECG images as part of the diagnostic process, which can provide more in-depth insights into the patient's condition.
ECG images provide essential information about the electrical activity of the heart, and analyzing them can help detect conditions such as arrhythmias, heart attacks, and STEMI (ST-segment elevation myocardial infarction). By incorporating ECG images into the system, the CardioCare AI system will have access to a more comprehensive dataset, improving the overall prediction accuracy and providing more reliable diagnoses.

7.2.2 Obstacles and Constraints

Incorporating ECG image analysis presents several technical challenges:

- **Image preprocessing:** ECG images need to be preprocessed for noise reduction and alignment. This could involve techniques like image normalization, rescaling, and feature extraction from ECG waveforms.
- **Deep learning models:** Analyzing ECG images will require the use of deep learning models, such as Convolutional Neural Networks (CNNs), which are effective in processing and classifying medical images. However, training deep learning models requires large labeled datasets and substantial computational resources.
- **Integration with existing data:** The system will need to integrate ECG images with the existing clinical data, ensuring that predictions made using images align with the predictions based on structured data (e.g., blood pressure, cholesterol).

7.2.3 Future Steps

The first step in implementing ECG image analysis will be to collect a sufficient number of labeled ECG images for training the deep learning models. Collaborating with medical professionals to annotate ECG images and label them with relevant heart disease conditions will be a critical part of the process. Once a substantial dataset is available, the system will be trained using CNNs for classification.
After developing the model, the system will be enhanced to allow users to upload ECG images through the user interface, and the backend will be responsible for processing and integrating the image-based predictions with the overall heart disease risk assessment.

7.3 Additional Dataset for Artery Narrowing Prediction

7.3.1 Concept and Need

A key enhancement for the CardioCare AI system involves integrating an additional dataset focused on predicting the extent of artery narrowing (stenosis). Artery narrowing is a critical factor in diagnosing heart disease, particularly coronary artery disease (CAD). Currently, the system primarily focuses on classifying heart disease risk based on generalized features such as cholesterol levels and blood pressure. However, by incorporating a dedicated dataset on artery narrowing, the system will be able to make more precise predictions about the severity of the disease and provide doctors with valuable information for treatment planning.
The addition of this dataset will enable the system to predict how many arteries are affected, as well as the degree of narrowing in each artery. This level of detail can help doctors decide whether more aggressive treatments, such as angioplasty or stent placement, are required.

7.3.2 Obstacles and Constraints

The main challenges in incorporating artery narrowing prediction include:

- **Availability of data:** Gathering comprehensive and high-quality data on artery narrowing can be difficult, especially from local hospitals. This would require collaboration with medical institutions and access to angiography or CT angiography images or reports.
- **Integration with existing system:** Incorporating this dataset into the existing CardioCare AI system will require modifications to the model architecture to accommodate additional features and ensure that predictions are made using all available data.
- **Model accuracy:** The system will need to be highly accurate in predicting artery narrowing, as false positives or negatives in this prediction could significantly impact patient outcomes.

7.3.3 Future Steps

The first step in implementing artery narrowing prediction will be to obtain the required dataset. Collaborating with hospitals that perform angiograms or CT angiography will be

key in acquiring the data needed. Once the dataset is available, the system will be trained using supervised learning techniques, such as Random Forests or Gradient Boosting, to predict the number of narrowed arteries and their degree of narrowing.
Following successful model development, the system will be enhanced to include this prediction as part of the risk assessment for heart disease, providing doctors with additional information to aid in their clinical decisions.

8

Conclusion

The **CardioCare AI:** Intelligent Heart Disease Prediction and Diagnosis System represents a significant step forward in the application of Artificial Intelligence (AI) for improving healthcare outcomes, specifically in the realm of heart disease diagnosis and prediction. Through the integration of multiple datasets, advanced machine learning models, and a robust system architecture, this project has demonstrated the potential for AI to enhance the accuracy and reliability of heart disease predictions, thus empowering healthcare providers with more precise tools for patient care.

The system's use of **multiclass classification** rather than binary classification is a major innovation, providing a more nuanced and clinically relevant approach to heart disease risk prediction. By offering a risk assessment that includes categories such as **"Yes", "No", and "Maybe",** the system ensures a higher level of certainty in decision-making, particularly crucial in healthcare where a misdiagnosis could have significant consequences.

In addition, the **combination of multiple datasets** containing various medical features has further strengthened the prediction accuracy, enabling a more comprehensive understanding of the factors contributing to heart disease. This integration of diverse medical data provides a holistic view of a patient's health, which is essential for early detection and accurate diagnosis. The validation of the system using real-world datasets from **Pakistani hospitals** further bolsters the system's credibility and relevance in local contexts.

Despite these accomplishments, the project acknowledges several areas for improvement and further development. The **integration of wearable devices**, the **incorporation of ECG image analysis**, and **the addition of a dataset to predict artery narrowing** represent exciting opportunities for future work. These enhancements will serve to refine the system, providing real-time data integration, more accurate diagnoses, and a broader scope of heart disease risk assessment, thereby expanding the system's clinical applicability and impact.

In conclusion, the CardioCare AI system not only meets the current needs of heart disease prediction and diagnosis but also opens the door for continuous improvement and innovation. The work presented here lays the foundation for the next generation of AI-powered healthcare solutions, which have the potential to revolutionize how heart disease is diagnosed and treated globally. With further advancements, the system could become an invaluable tool in the fight against heart disease, contributing to better health outcomes and quality of life for patients worldwide.

Bibliography

[1] Ramalingam, V.V., Dandapath, Ayantan, and Raja, M. Karthik. "Heart disease prediction using machine learning techniques: a survey." *International Journal of Engineering & Technology*, vol. 7, no. 2.8, 2018, pp. 684–687. Science Publishing Corporation.

[2] Gupta, Utkarsh, Paluru, Naveen, Nankani, Deepankar, Kulkarni, Kanchan, and Awasthi, Navchetan. "A comprehensive review on efficient artificial intelligence models for classification of abnormal cardiac rhythms using electrocardiograms." *Heliyon*, 2024. Elsevier.

[3] Saleem, Sehar, Yasin, Seyab, Aslam, Maria, and Ditta, Areeba Allah. "Evaluating Risk Factors for Coronary Heart Disease Among Adults Aged 26-45: A Study from Punjab, Pakistan." *Journal of Statistics*, vol. 28, 2024, pp. 56–65.

[4] Westerlund, Annie M., Hawe, Johann S., Heinig, Matthias, and Schunkert, Heribert. "Risk prediction of cardiovascular events by exploration of molecular data with explainable artificial intelligence." *International Journal of Molecular Sciences*, vol. 22, no. 19, 2021, pp. 10291. MDPI.

[5] Wang, Jinwan, Wang, Shuai, Zhu, Mark Xuefang, Yang, Tao, Yin, Qingfeng, and Hou, Ya. "Risk prediction of major adverse cardiovascular events occurrence within 6 months after coronary revascularization: machine learning study." *JMIR Medical Informatics*, vol. 10, no. 4, 2022, pp. e33395. JMIR Publications Toronto, Canada.

[6] Virani, Salim S., Alonso, Alvaro, Benjamin, Emelia J., Bittencourt, Marcio S., Callaway, Clifton W., Carson, April P., Chamberlain, Alanna M., Chang, Alexander R., Cheng, Susan, Delling, Francesca N., and others. "Heart disease and stroke statistics—2020 update: a report from the American Heart Association." *Circulation*, vol. 141, no. 9, 2020, pp. e139–e596. Am Heart Assoc.

[7] Ahsan, Md Manjurul, Luna, Shahana Akter, and Siddique, Zahed. "Machine-learning-based disease diagnosis: A comprehensive review." *Healthcare*, vol. 10, no. 3, 2022, pp. 541. MDPI.

[8] Rudnicka, Zofia, Proniewska, Klaudia, Perkins, Mark, and Pregowska, Agnieszka. "Cardiac Healthcare Digital Twins Supported by Artificial Intelligence-Based Algorithms and Extended Reality—A Systematic Review." *Electronics*, vol. 13, no. 5, 2024, pp. 866. MDPI.

[9] Umer, Muhammad, Aljrees, Turki, Karamti, Hanen, Ishaq, Abid, Alsubai, Shtwai, Omar, Marwan, Bashir, Ali Kashif, and Ashraf, Imran. "Heart failure patients monitoring using IoT-based remote monitoring system." *Scientific Reports*, vol. 13, no. 1, 2023, pp. 19213. Nature Publishing Group UK London.

[10] Zhang, Daniel, Mishra, Saurabh, Brynjolfsson, Erik, Etchemendy, John, Ganguli, Deep, Grosz, Barbara, Lyons, Terah, Manyika, James, Niebles, Juan Carlos, Sel-

litto, Michael, and others. “The AI index 2021 annual report.” *arXiv preprint arXiv:2103.06312*, 2021.

[11] Gupta, Shashi Kant, Khang, Alex, Somani, Parin, Dixit, Chandra Kumar, and Pathak, Anchal. “Data Mining Processes and Decision-Making Models in the Personnel Management System.” In *Designing Workforce Management Systems for Industry 4.0*, pp. 85–104. CRC Press, 2023.

[12] Azmi, Javed, Arif, Muhammad, Nafis, Md Tabrez, Alam, M Afshar, Tanweer, Safdar, and Wang, Guojun. “A systematic review on machine learning approaches for cardiovascular disease prediction using medical big data.” *Medical Engineering & Physics*, vol. 105, 2022, pp. 103825. Elsevier.

[13] Park, Jeong-Min, Choi, Sung-Kyeong, Kim, Jun-Yeong, Jung, Se-Hoon, and Sim, Chun-Bo. “Implementation of a Drug Information Retrieval System Through OCR API performance comparison.” *The Journal of the Korea Institute of Electronic Communication Sciences*, vol. 18, no. 5, 2023, pp. 989–998. Korea Institute of Electronic Communication Science.

[14] Raihan, Asif. “Incorporating geospatial information into the execution and ongoing evaluation of strategies for attaining sustainable development goals (SDGs).” In *The International Conference on New Quality Productive Forces and Sustainable Development. 21st-22nd September*, 2024.

[15] Kingma, Diederik P. “Adam: A method for stochastic optimization.” *arXiv preprint arXiv:1412.6980*, 2014.

[16] Goodfellow, Ian. *Deep Learning*, 2016. MIT Press.

A

Appendix 1: Project Code and Results

This appendix includes the essential code and results used in the heart disease prediction project.

A.1 Data Preprocessing Code

The following code was used to clean and preprocess the dataset:

```
import pandas as pd
from sklearn.preprocessing import StandardScaler

# Load dataset
data = pd.read_csv('heart_disease_data.csv')

# Handle missing values
if heart_df.isnull().sum().any():
    print("\nMissing values detected.")
    for col in heart_df.columns:
        if heart_df[col].dtype in ['float64', 'int64']:
            heart_df[col].fillna(heart_df[col].mean(), inplace=True)
        else:
            heart_df[col].fillna(heart_df[col].mode()[0], inplace=True)
else:
    print("\nNo missing values detected.")
```

A.2 Model Training and Evaluation

The model was trained using a Random Forest classifier. Here is the code snippet for training and evaluating the model:

```
kf = KFold(n_splits=5, shuffle=True, random_state=42)  # 5-fold cross-
    validation

# Store results
validation_losses = []
train_losses = []
accuracies = []
max_epochs = 0
best_model = None
best_val_loss = np.inf  # Initialize with a large value to track the best
    model
```

```
# Loop through each fold
for train_index, val_index in kf.split(X_train_resampled):
    # Split data
    X_train_fold, X_val_fold = X_train_resampled.iloc[train_index],
        X_train_resampled.iloc[val_index]
    y_train_fold, y_val_fold = y_train_resampled[train_index],
        y_train_resampled[val_index]

    # Define Neural Network Model
    nn_model = Sequential()
    nn_model.add(Dense(32, input_dim=X_train_fold.shape[1], activation='
        relu', kernel_regularizer=l2(0.01))) # L2 regularization
    nn_model.add(Dropout(0.7)) # Increased dropout
    nn_model.add(Dense(16, activation='relu', kernel_regularizer=l2(0.01)))
        # L2 regularization
    nn_model.add(Dense(3, activation='softmax')) # 3 classes

    # Compile the model
    nn_model.compile(optimizer='adam', loss='
        sparse_categorical_crossentropy', metrics=['accuracy'])

    # Early Stopping
    early_stopping = EarlyStopping(monitor='val_loss', patience=10)

    # Learning Rate Scheduler
    reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.1, patience
        =3, min_lr=1e-6)

    # Train the model
    history = nn_model.fit(X_train_fold, y_train_fold, epochs=50,
        batch_size=32,
                           validation_data=(X_val_fold, y_val_fold),
                           callbacks=[early_stopping, reduce_lr], verbose
                               =0)

    # Save the training and validation losses for each fold
    train_losses.append(history.history['loss'])
    validation_losses.append(history.history['val_loss'])
    accuracies.append(history.history['accuracy'][-1])

    # Update max_epochs
    max_epochs = max(max_epochs, len(history.history['loss']))

    # Track the best model based on validation loss
    val_loss = min(history.history['val_loss']) # Get the best validation
        loss for the fold
    if val_loss < best_val_loss:
        best_val_loss = val_loss
        best_model = nn_model
```

A.3 Model Architecture

The model is built using a neural network with the following architecture:

- Input Layer: 15 input features representing various health parameters.
- Hidden Layer 1: 64 neurons with ReLU activation.

- Hidden Layer 2: 32 neurons with ReLU activation.
- Output Layer: 3 neuron with softmax activation function for multiclass classification (Heart Disease: Yes/No/Maybe).
- Optimizer: Adam optimizer.

This appendix contains only the key sections of the code and results relevant to the project's methodology and outcomes.

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ENGINEERING TECHNOLOGY
Lahore, Pakistan
www.uet.edu.pk