

Industrial Oriented Mini Project Report
CARDIOVASCULAR DISEASE PREDICTION

A dissertation submitted in partial fulfillment of the

Requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

Submitted

by

J.NIHAARIKA-22B81A1290

NISHKA AGARWAL-22B81A1295

Under the esteemed guidance of

Dr.A.MALLAREDDY

Professor, IT Department



CVR COLLEGE OF ENGINEERING

(An UGC Autonomous Institution, Affiliated to JNTUH,

Accredited by NBA, and NAAC)

Vastunagar, Mangalpalli (V), Ibrahimpatnam (M),

Ranga Reddy (Dist.) - 501510, Telangana State.

2024-202



Cherabuddi Education Society's
CVR COLLEGE OF ENGINEERING

(An Autonomous Institution)

ACCREDITED BY NATIONAL BOARD OF ACCREDITATION, AICTE

(Approved by AICTE & Govt. of Telangana and Affiliated to JNT University)

Vastunagar, Mangalpalli (V), Ibrahimpatan (M), R.R. District, PIN - 501 510

Web : <http://cvr.ac.in>, email : info@cvr.ac.in

Ph : 08414 - 252222, 252369, Office Telefax : 252396, Principal : 252396 (O)

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the Project Report entitled **Cardiovascular Disease Prediction** is a Bonafide work done and submitted by **J.Nihaarika (22B81A1290)**, **Nishka Agarwal (22B81A1295)** during the academic year 2024-2025, in partial fulfillment of requirement for the award of Bachelor of Technology degree in Information Technology from Jawaharlal Nehru Technological University Hyderabad, is a bonafide record of work carried out by them under my guidance and supervision.

Certified further that to the best of my knowledge, the work in this dissertation has not been submitted to any other institution for the award of any degree or diploma.

INTERNAL GUIDE

Dr.A.MallaReddy

Professor, IT Department

HEAD OF THE DEPARTMENT

Dr. Bipin Bihari Jayasingh

Professor, IT Department

PROJECT COORDINATOR

Ms. S. Suhasini

Assistant Professor, IT Department

EXTERNAL EXAMINER

City Office : # 201 & 202, Ashoka Scintilla, Opp. KFC, Himayatnagar, Hyderabad - 500 029, Telangana.

Phone : 040 - 42204001, 42204002, 9391000791, 9177887273

DEPARTMENT OF INFORMATION TECHNOLOGY

DECLARATION

We hereby declare that the project report entitled **Cardiovascular Disease Prediction** is an original work done and submitted to the IT Department, CVR College of Engineering, affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad in partial fulfilment of the requirement for the award of Bachelor of Technology in **Information Technology** and it is a record of bonafide project work carried out by us under the guidance of **Dr.A.MallaReddy, Professor, Department of Information Technology.**

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

J.Nihaarika – 22B81A1290

Nishka Agarwal – 22B81A1295

ACKNOWLEDGEMENT

The satisfaction of completing this project would be incomplete without mentioning our gratitude towards all the people who have supported us. Constant guidance and encouragement have been instrumental in the completion of this project.

First and foremost, we thank the Chairman, Principal, and Vice Principal for availing infrastructural facilities to complete the major project in time.

We offer our sincere gratitude to our internal guide **Dr.A.MallaReddy**, Professor, of IT Department, CVR College of Engineering for his immense support, timely cooperation, and valuable advice throughout our project work.

We would like to thank the in charge of our Project, Mr. **K.Veeranjaneyulu**, Sr. Assistant Professor, Department of Information Technology for his valuable suggestions in implementing the project.

We would like to thank the Head of the IT Department, Professor **Dr. Bipin Bihari Jayasingh**, for his meticulous care and cooperation throughout the project work.

We also thank the **Project Review Committee Members** for their valuable suggestions.

J.Nihaarika (22B81A1290)

Nishka Agarwal (22B81A1295)

ABSTRACT

Cardiovascular diseases (CVD) remain a leading cause of mortality worldwide, demanding early detection and preventive measures to mitigate risks. This study explores various ML models to enhance the accuracy and reliability of CVD prediction. Traditional ML algorithms often struggle with diverse data patterns, leading to suboptimal predictions. To address this, we implemented and compared multiple classification models, including Logistic Regression, Support Vector Classifier, Decision Trees, Extra Trees, Gradient Boosting, and Random Forest. Through rigorous evaluation, Random Forest emerged as the most effective model, achieving superior accuracy and minimizing misclassification rates. Furthermore, the proposed system incorporates advanced functionalities such as real-time prediction, a chatbot for patient assistance, and health trend analysis, offering a comprehensive AI-powered solution. The model was trained using the Heart Disease Dataset and validated across multiple performance metrics, achieving an accuracy of approximately 90%. Future improvements will focus on integrating deep learning architectures and larger datasets from medical institutions to further enhance prediction reliability and clinical applicability. This research contributes to the growing field of AI-driven healthcare, demonstrating the potential of ML and DL in early diagnosis, personalized healthcare, and decision support systems for CVD management.

LIST OF FIGURES

Figure	Title	Page No.
4.1	Random Forest Classifier	15
4.2	Confusion Matrix of our Model	15
4.3	System Architecture	16
4.4	Application Development Process Diagram	16
4.5	Heatmap of the Variables	18
4.6	Pairwise Scatter Plot Matrix of Features colored by Cardiovascular Outcome	19
4.7	Activity Diagram	20
4.8	ER Diagram	20
4.9	DFD Level 0	21
4.10	DFD Level 1	21
5.1	Home page screen	23
5.2	Home Page Implementation	24
5.3	Implementation of Chatbot	24
5.4	Implementation of Health trends	25
6.1	Chatbot Screen	32
6.2	Health Prediction Screen (with low risk as output)	32
6.3	Health Prediction Screen (with high risk as output)	33
6.4	Health Trends Display	33
6.5	Login Page	34

TABLE OF CONTENTS

Chapter No.	Contents	Page No.
1	Introduction	
	1.1 Introduction	1
	1.2 Problem Statement	2
	1.3 Project Objectives	3
	1.4 Project Report Organization	4
2	Literature Survey	
	2.1 Literature Survey	6
	2.2 Existing work	7
3	Software Requirements and Specifications	
	3.1 Hardware requirements	9
	3.2 Software requirements	9
	3.3 Software Description	10
4	Design	
	4.1 Proposed Method	13
	4.2 Algorithm	14
	4.3 System Architecture	15
	4.4 Modules	17
	4.5 Data Flow Diagram	19
5	Implementation	
	5.1 Implementation	23
	5.2 Code Snippets	26
6	Testing	
	6.1 Testing	32
	6.2 Results and Discussions	35
	6.3 Validations	35
	Conclusion	37
	Future Enhancements	38
	Abbreviations	39
	References	40

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The heart is a kind of muscular organ which pumps blood into the body and is the central part of the body's cardiovascular system which also contains lungs. Cardiovascular system also comprises a network of blood vessels, for example, veins, arteries, and capillaries. These blood vessels deliver blood all over the body. Abnormalities in normal blood flow from the heart cause several types of heart diseases which are commonly known as cardiovascular diseases (CVD). Heart diseases are the main reasons for death worldwide. According to the survey of the World Health Organization (WHO), 17.5 million total global deaths occur because of heart attacks and strokes. More than 75% of deaths from cardiovascular diseases occur mostly in middle-income and low-income countries. Also, 80% of the deaths that occur due to CVDs are because of stroke and heart attack.

Therefore, prediction of cardiac abnormalities at the early stage and tools for the prediction of heart diseases can save a lot of life and help doctors to design an effective treatment plan which ultimately reduces the mortality rate due to cardiovascular diseases. Due to the development of advance healthcare systems, lots of patient data are nowadays available (i.e. Big Data in Electronic Health Record System) which can be used for designing predictive models for cardiovascular diseases. Data mining or machine learning is a discovery method for analysing big data from an assorted perspective and encapsulating it into useful information. "Data Mining is a nontrivial extraction of implicit, previously unknown and potentially useful information about data".

Nowadays, a huge amount of data pertaining to disease diagnosis, patients etc. are generated by healthcare industries. Data mining provides a number of techniques which discover hidden patterns or similarities from data. Therefore, in this paper, a machine learning algorithm is proposed for the implementation of a heart disease prediction system which was validated on two open access heart disease prediction datasets. Data mining is the computer-based process of extracting useful information from enormous sets of databases. Data mining is most helpful in an explorative analysis because of nontrivial information from large volumes of evidence. Medical data mining has great potential for exploring the cryptic patterns in the data sets of the clinical domain. These patterns can be utilized for healthcare diagnosis. However, the available raw medical data are widely distributed, voluminous and heterogeneous in nature. This data needs to be collected in an organized form. This collected data can be then integrated to form a medical information system. Data mining provides a user-oriented approach to novel and hidden patterns in the Data The data mining tools are useful for answering business questions and techniques for predicting the various diseases in the healthcare field. Disease prediction plays a significant role in data mining. This paper analyses the heart disease predictions using classification algorithms. These invisible patterns can be utilized for health diagnosis in healthcare data. Data mining technology affords an efficient approach to the latest and indefinite patterns in the data. The information which is identified can be used by the healthcare administrators to get better services. Heart disease was the most crucial reason for victims in the countries like India, United States. In this project we are predicting the heart disease using classification algorithms. Machine learning techniques like Classification algorithms such as Random forest, Logistic Regression are used to explore different kinds of heart based problems.

1.2 PROBLEM STATEMENT

Cardiovascular diseases (CVDs) are one of the leading causes of mortality worldwide. Early prediction and proactive lifestyle modifications can significantly reduce the risks associated with heart-related illnesses. However, access to real-time health assessments and personalized health insights remains limited, especially in regions lacking specialized healthcare infrastructure.

The aim of this project is to develop an intelligent, user-friendly Cardio Disease Prediction App that utilizes machine learning models to predict the risk of cardiovascular disease based on user-inputted health metrics. The application also integrates advanced functionalities such as chatbot support, voice input, trend visualization, and PDF report generation to provide a comprehensive and accessible solution for users to monitor and manage their heart health.

1.3 PROJECT OBJECTIVES

The primary objectives of this project are:

- To develop a web-based application that predicts the risk of cardiovascular disease using machine learning models.
- To design a user-friendly interface using Streamlit, ensuring ease of access for non-technical users.
- To enable voice-based input and text-to-speech features for interactive communication.
- To integrate the Gemini API for generating health tips and chatbot functionality.
- To implement user authentication (login & logout) and session management for data privacy.
- To generate comprehensive PDF health reports that include predictions, health insights, and lifestyle suggestions.
- To visualize personal health trends over time using interactive charts and data logs.

1.4 PROJECT REPORT ORGANIZATION:

This report is organized into six chapters:

- Chapter1:

Introduction

Provides the motivation, problem statement, project objectives, and an overview of the report structure.

- Chapter2:

Literature and Survey

Discusses existing studies and tools related to heart disease prediction, along with their limitations and research gaps.

- Chapter3:

Software and Hardware Requirements

Details the functional and non-functional requirements, including the tools, platforms, and system architecture used.

- Chapter4:

System Design

Describes the proposed system, including UML diagrams such as use case, class, sequence, component, and deployment diagrams.

- Chapter5:

Implementation and Testing

Covers module-wise implementation details, user interfaces, test cases, and test results.

- Chapter6:

Conclusion and Future Scope

Summarizes the project outcomes and outlines possible enhancements for future development.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

Machine Learning techniques are used to analyse and predict the medical data information resources. Diagnosis of heart disease is a significant and tedious task in medicine. The term heart disease encompasses the various diseases that affect the heart. The exposure of heart disease from various factors or symptom is an issue which is not complimentary from false presumptions often accompanied by unpredictable effects. The data classification is based on Supervised Machine Learning algorithm which results in better accuracy. Here we are using the Random Forest as the training algorithm to train the heart disease dataset and to predict the heart disease. The results showed that the medicinal prescription and designed prediction system is capable of prophesying the heart attack successfully. Machine Learning techniques are used to indicate the early mortality by analysing the heart disease patients and their clinical records [7] Chollet, F. (2018), [5] Rajput, D. S., & Basha, S. M. (2021) have brought about the two Machine Learning techniques, k-nearest neighbour model and existing multi linear regression to predict the stroke severity index (SSI) of the patients. Their study show that k-nearest neighbour performed better than Multi Linear Regression model. [8] Arslan, A. K. et al., 2016 have suggested various Machine Learning techniques such as support vector machine (SVM), penalized logistic regression (PLR) to predict the heart health. Their results show that Random Forest produced the best performance in prediction when compared to other models. The main objective is to evaluate the different classification techniques such as J48, Decision Tree, KNN and Naïve Bayes, SVM, Linear Regression. After this, evaluating some performance in measures of accuracy, precision, sensitivity, specificity we chose Random Forest Model.

Data source: databases have a significant amount of information about patients and their medical conditions. Records set with medical attributes were obtained from the Kaggle database. With the help of the dataset, the patterns significant to the unhealthy heart diagnosis are extracted. The records were split equally into two datasets: training dataset and testing dataset. A total of 303 records with 76 medical attributes were obtained. All the attributes are numeric-valued. We are working on a reduced set of attributes, i.e. only 14 attributes. All these restrictions were announced to shrink the digit of designs, these are as follows:

- 1) The features should seem on a single side of the rule.
- 2) The rule should distinct various features into the different groups.

2.2 EXISTING SYSTEM:

In the current healthcare landscape, cardiovascular disease (CVD) prediction primarily relies on traditional methods such as periodic health check-ups, manual evaluation of patient records, and clinical risk scoring systems like the **Framingham Risk Score**, **QRISK**, and **SCORE**. These tools use factors like age, gender, cholesterol levels, blood pressure, smoking status, and diabetes to estimate a patient's risk of developing cardiovascular disease over a period of time.

While effective to an extent, the existing system suffers from several limitations:

1. **Manual and Time-Consuming:** Risk assessment often requires healthcare professionals to manually input data and interpret results, which is both time-intensive and prone to human error.
2. **Lack of Personalization:** Most scoring systems use generalized statistical models that may not adapt well to individuals with unique risk factor combinations.
3. **Limited Real-Time Access:** Traditional methods are usually not accessible remotely and are dependent on hospital visits, reducing their utility for regular monitoring.
4. **No Integration with Digital Platforms:** Current systems are typically standalone and lack integration with modern digital health tools such as mobile apps, voice input, or real-time chatbots.
5. **No Predictive Intelligence:** These systems are not equipped with machine learning or artificial intelligence to learn from vast patient datasets, which limits their ability to improve prediction accuracy over time.
6. **Expensive:** Highly expensive and laborious process needs to be performed before treating the patient to find out if he/she has any chances to get heart disease in future.

These challenges necessitate the development of a more intelligent, interactive, and accessible system that can automate risk prediction, improve accuracy, and offer personalized health insights.

CHAPTER-3

SOFTWARE REQUIREMENTS AND SPECIFICATIONS

3.1 MINIMUM HARDWARE REQUIREMENTS:

Processor Quad-Core CPU (e.g., Intel i3 10th Gen or AMD Ryzen 3) at **2.5GHz+**

RAM 8 GB minimum (16 GB recommended for multitasking or development)

Storage 256 GB SSD (faster than HDD; HDDs are outdated for boot drives)

Input Devices USB / Wireless Keyboard and Mouse

Display Full HD Monitor (1920x1080) or higher

Graphics Integrated GPU (e.g., Intel UHD / AMD Radeon Vega) sufficient for general use

Connectivity Wi-Fi, Bluetooth 5.0, USB 3.0+ ports

Operating System Windows 10/11, macOS 12+, or modern Linux distros

3.2 MINIMUM SOFTWARE REQUIREMENTS:

Operating System : Windows 10 or higher

Programming : python 3.6 and related libraries

Software : Anaconda Navigator, Jupyter

Tools: Notebook and Google colab

3.3 SOFTWARE DESCRIPTION

Python:

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Python features a dynamic type of system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library. Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open-source software and has a community-based development model, as do nearly all its variant implementations. CPython is managed by the non-profit Python Software Foundation.

Pandas:

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data.

In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data. Prior to Pandas, Python was majorly used for data mining and preparation. It had very little contribution towards data analysis. Pandas solved this problem.

Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyse. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Key Features of Pandas:

- ☐ Fast and efficient Data Frame object with default and customized indexing.
- ☐ Tools for loading data into in-memory data objects from different file formats.
- ☐ Data alignment and integrated handling of missing data.
- ☐ Reshaping and pivoting of date sets.
- ☐ Label-based slicing, indexing and subsetting of large data sets.

- ☐ Columns from a data structure can be deleted or inserted.
- ☐ Group by data for aggregation and transformations.
- ☐ High performance merging and joining of data.
- ☐ Time Series functionality.

NumPy:

NumPy is a general-purpose array-processing package. It provides a high performance multidimensional array object, and tools for working with these arrays. It

is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- ☐ A powerful N-dimensional array object
- ☐ Sophisticated (broadcasting) functions
- ☐ Tools for integrating C/C++ and Fortran code
- ☐ Useful linear algebra, Fourier transform, and random number capabilities 24
- ☐ Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary datatypes can be defined using NumPy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

Scikit-Learn:

- ☐ Simple and efficient tools for data mining and data analysis
- ☐ Accessible to everybody, and reusable in various contexts
- ☐ Built on NumPy, SciPy, and matplotlib
- ☐ Open source, commercially usable - BSD license

Matplotlib:

- ☐ Matplotlib is a python library used to create 2D graphs and plots by using python scripts.
- ☐ It has a module named pyplot which makes things easy for plotting by providing feature to control line styles, font properties, formatting axes etc.
- ☐ It supports a very wide variety of graphs and plots namely - histogram, bar charts, power spectra, error charts etc.

CHAPTER 4

DESIGN

4.1 PROPOSED METHOD

This section depicts the overview of the proposed system and illustrates all of the components, techniques and tools are used for developing the entire system. To develop an intelligent and user-friendly heart disease prediction system, an efficient software tool is needed in order to train huge datasets and compare multiple machine learning algorithms. After choosing the robust algorithm with best accuracy and performance measures, it will be implemented on the development of the smartphone-based application for detecting and predicting heart disease risk level. Hardware components like different biomedical sensors, display monitor, buzzer etc. are needed to build the continuous patient monitoring system.

4.2 ALGORITHMS

RANDOM FOREST:

Random Forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result. Working of Random Forest with the help of following steps:

- ☐ First, start with the selection of random samples from a given dataset.
- ☐ Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.
- ☐ In this step, voting will be performed for every predicted result.
- ☐ At last, select the most voted prediction results as the final prediction result. The following diagram will illustrate its working-

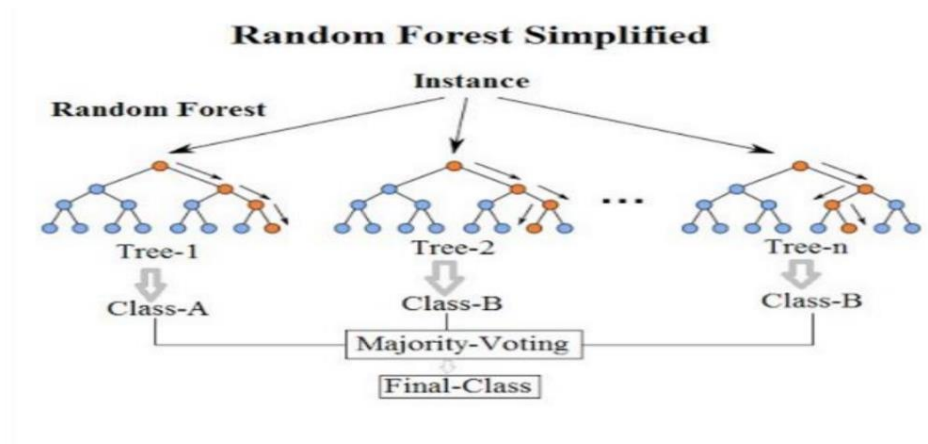


Fig4.1: Random Forest Classifier

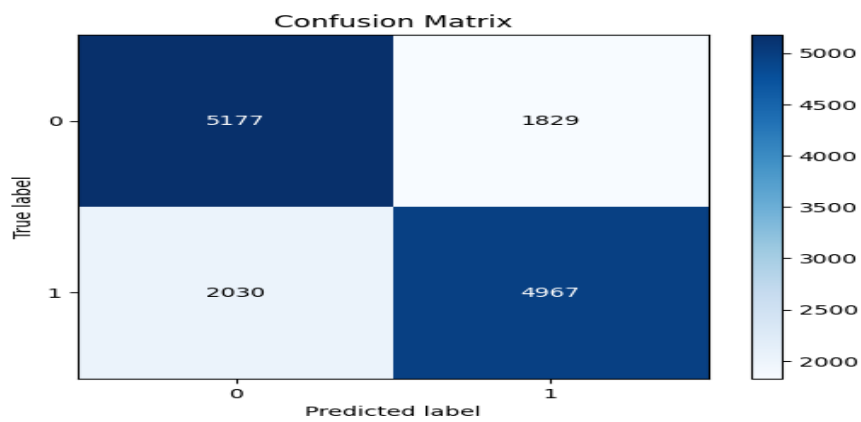


Fig 4.2 :Confusion Matrix of our Model

4.3 SYSTEM ARCHITECTURE

The below figure shows the process flow diagram or proposed work. First, we collected the Kaggle Dataset, then pre-processed the dataset and select the important features

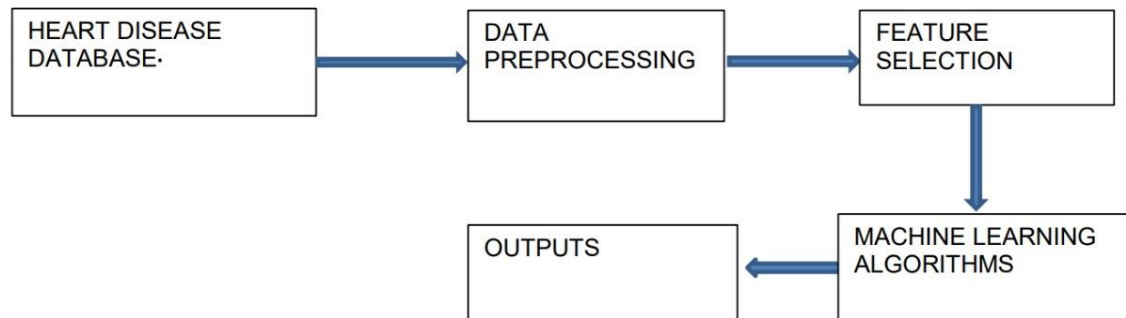


Fig 4.3: System Architecture

For feature selection we used HeatMap and got the top features. After that applied Random Forest and Logistic algorithm individually and compute the accuracy. Finally, we used proposed Ensemble technique and compute best method for diagnosis of heart disease.

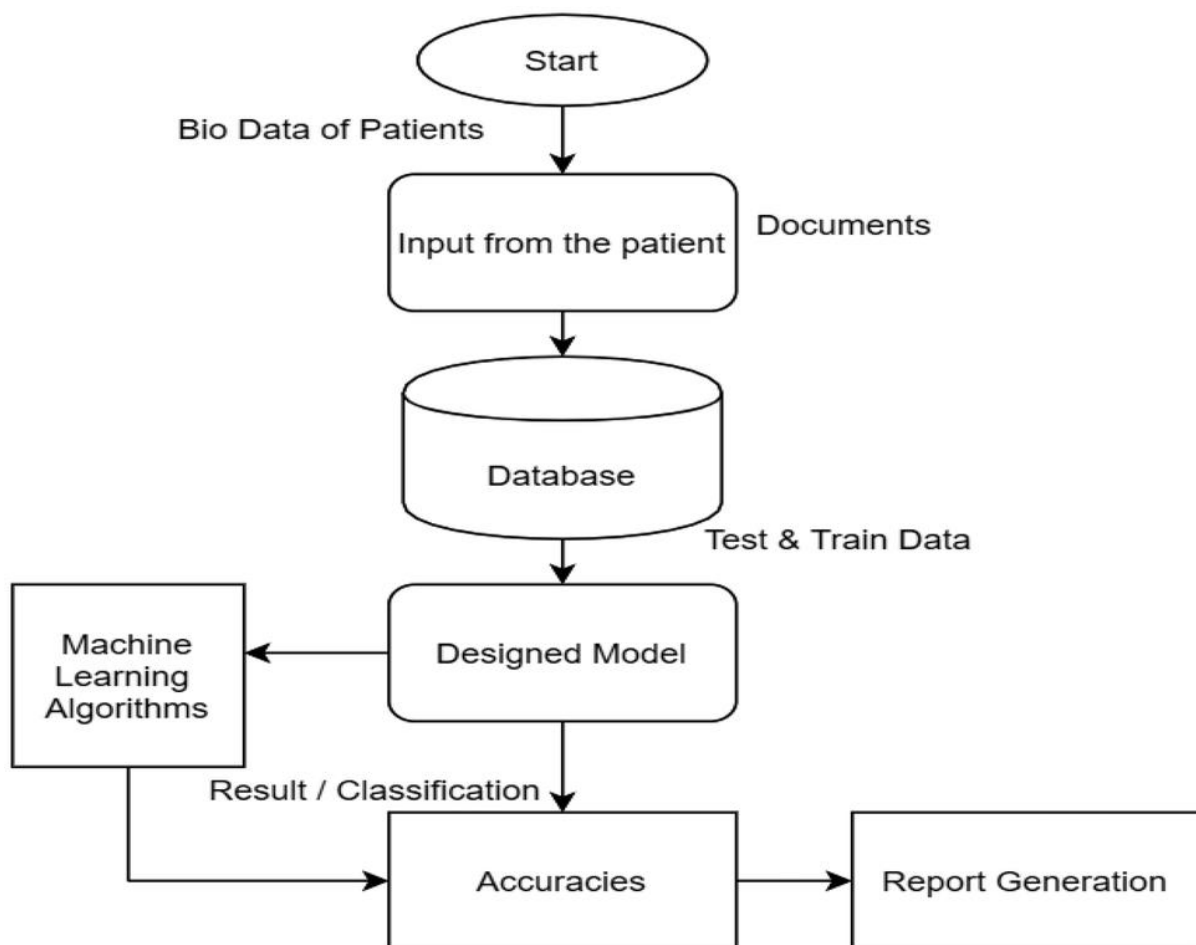


Fig 4.4: Application Development Process Diagram

4.4 MODULES:

The entire work of this project is divided into 4 modules. They are:

a. Data Pre-Processing

b. Feature

c. Classification

d. Prediction

- a. Data Pre-processing: This file contains all the pre-processing functions needed to process all input documents and texts. First, we read the train, test and validation data files then performed some preprocessing like tokenizing, stemming etc. There are some exploratory data analyses is performed like response variable distribution and data quality checks like null or missing values etc. Data preprocessing is the process of transforming raw data into an understandable format. It is also an important step in data mining as we cannot work with raw data. The quality of the data should be checked before applying machine learning or data mining algorithms. Preprocessing of data is mainly to check the data quality. The quality can be checked by the following-
- ☐ Accuracy: To check whether the data entered is correct or not.
 - ☐ Completeness: To check whether the data is available or not recorded.
 - ☐ Consistency: To check whether the same data is kept in all the places that do or do not match.
 - ☐ Timeliness: The data should be updated correctly
 - ☐ Believability: The data should be trustable.
 - ☐ Interpretability: The understandability of the data.
- b. Feature: Extraction In this file we have performed feature extraction and selection methods from sci-kit learn python libraries. For feature selection, we have used methods like simple bag-of-words and n-grams and then term frequency like tf-idf weighting. We have also used word2vec and POS tagging to extract the features, though POS tagging and word2vec has not been used at this point in the project

Bag of Words: It's an algorithm that transforms the text into fixed-length vectors. This is possible by counting the number of times the word is present in a document. The word occurrences allow to compare different documents and evaluate their similarities for applications, such as search, document classification, and topic modelling. The reason for its name, —Bag-Of-Words, is due to the fact that it represents the sentence as a bag of terms. It doesn't considers the order and the structure of the words, but it only checks if the words appear in the document. N-grams: N-grams are continuous sequences of words or symbols or tokens in a document. In technical terms, they can be defined as the neighbouring sequences of items in a document. They come into play when we deal with text data in NLP(Natural Language Processing) tasks. TF-IDF Weighting: TF-IDF stands for term frequency-inverse document frequency and it is a measure, used in the fields of information retrieval (IR) and machine learning, that can quantify the importance or relevance of string representations (words, phrases, lemmas, etc) in a document amongst a collection of documents (also known as a corpus).

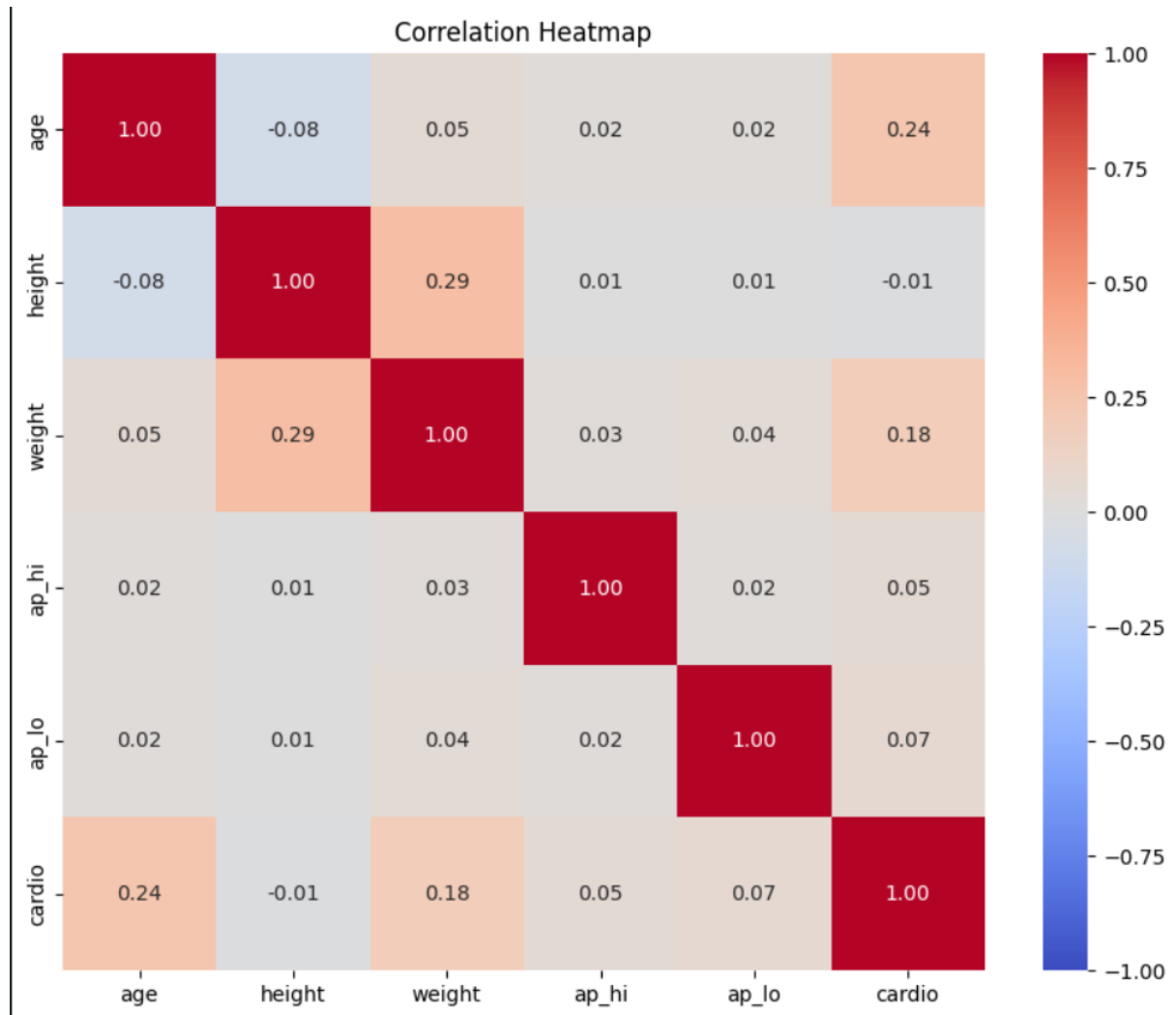


Fig 4.5 : Heatmap of the Variables

- c. Classification: Here we have built all the classifiers for the heart diseases prediction. The extracted features are fed into different classifiers. We have used Naive-bayes, Logistic Regression, Linear SVM, Stochastic gradient decent and Random Forest classifiers from sklearn. Each of the extracted features was used in all the classifiers. Once fitting the model, we compared the f1 score and checked the confusion matrix. After fitting all the classifiers, 2 best performing models were selected as candidate models for heart diseases classification. We have performed parameter tuning by implementing GridSearchCV methods on these candidate models and chosen best performing parameters for these classifiers. Finally selected model was used for heart disease detection with the probability of truth. In Addition to this, we have also extracted the top 50 features from our term-frequency tf-idf Vectorizer to see what words are most and important in each of the classes. We have also used Precision Recall and learning curves to see how training and test set performs when we increase the amount of data in our classifiers. d. Prediction: Our finally selected and best performing classifier was algorithm which was then saved on disk with name model.h5. Once you close this repository, this model will be copied to user's machine and will be used by prediction.py file to classify the heart diseases. It takes a news article

as input from user then model is used for final classification output that is shown to user along with probability of truth.

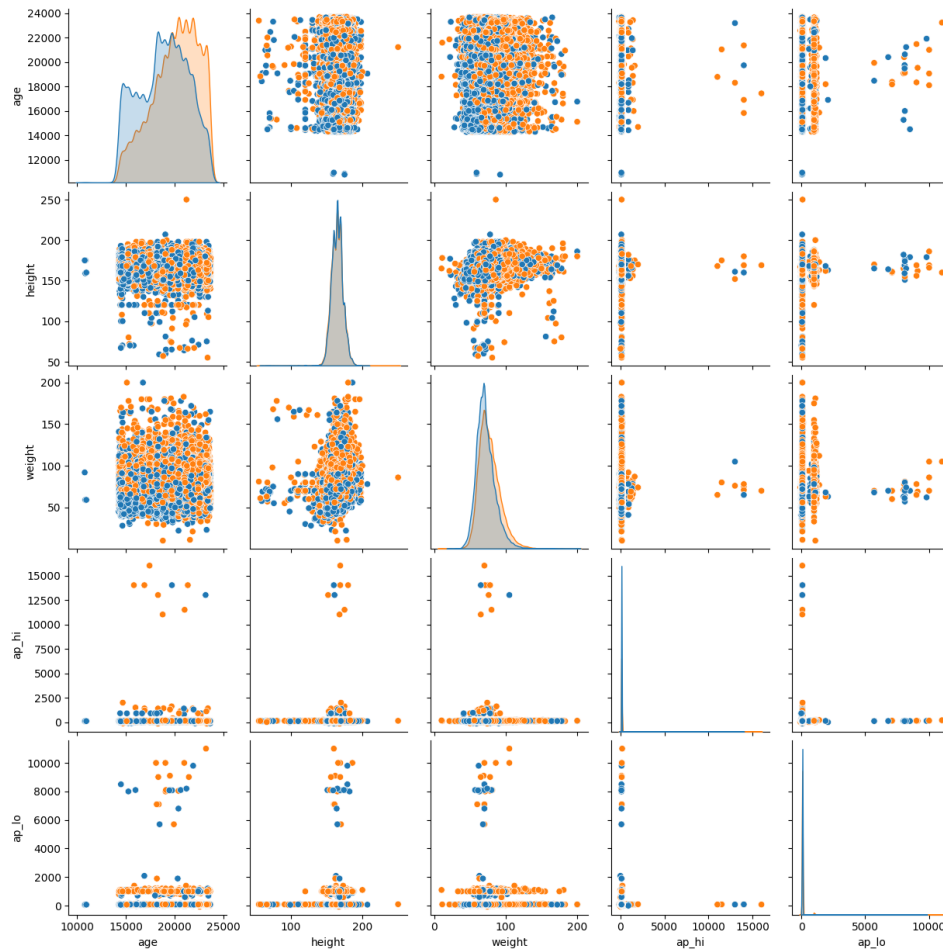


Fig4.6: Pairwise Scatter Plot Matrix of Features coloured by Cardiovascular Outcome

- d. Prediction: Our finally selected and best performing classifier was algorithm which was then saved on disk with name model.h5. Once you close this repository, this model will be copied to user's machine and will be used by prediction.py file to classify the heart diseases. It takes a news article as input from user then model is used for final classification output that is shown to user along with probability of truth.

4.5 DATA FLOW DIAGRAM

The data flow diagram (DFD) is one of the most important tools used by system analysis. Data flow diagrams are made up of number of symbols, which represents system components. Most data flow modeling methods use four kinds of symbols:

Processes, Data stores, Data flows and external entities. These symbols are used to represent four kinds of system components. Circles in DFD represent processes. Data Flow represented by a thin line in the DFD, and each data store has a unique name and square or rectangle represents external entities

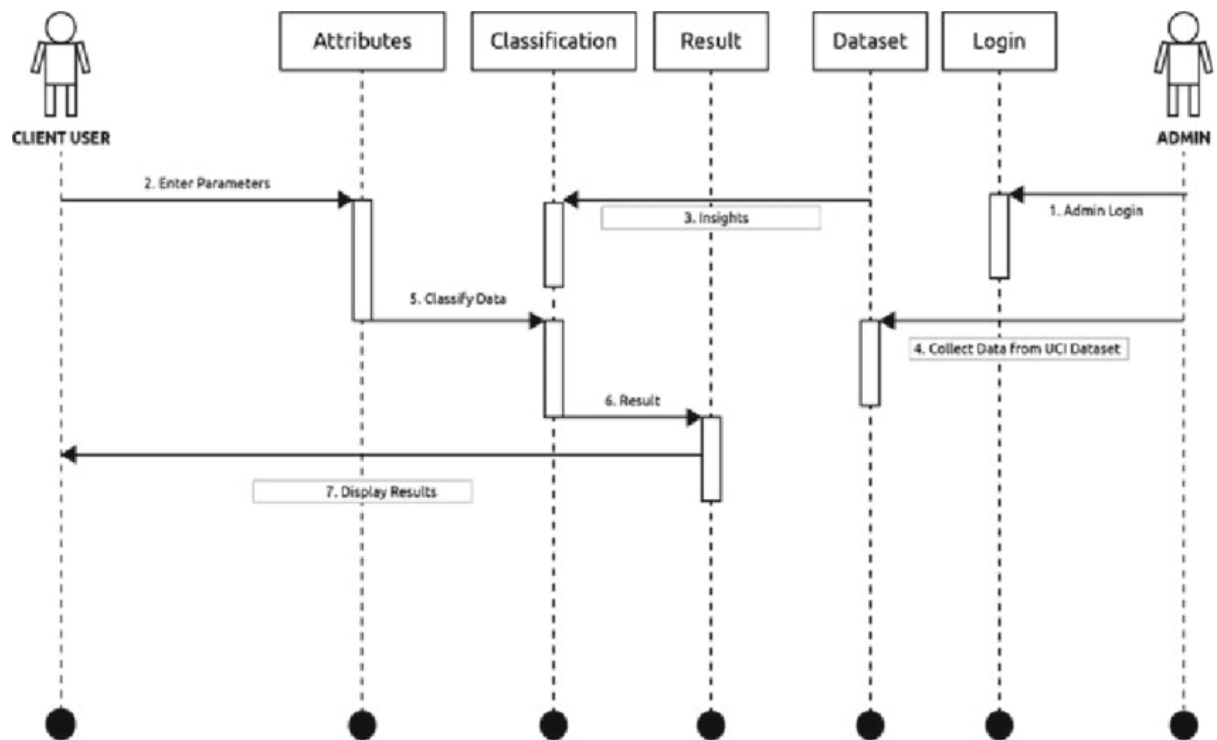


Fig 4.7 : Activity Diagram

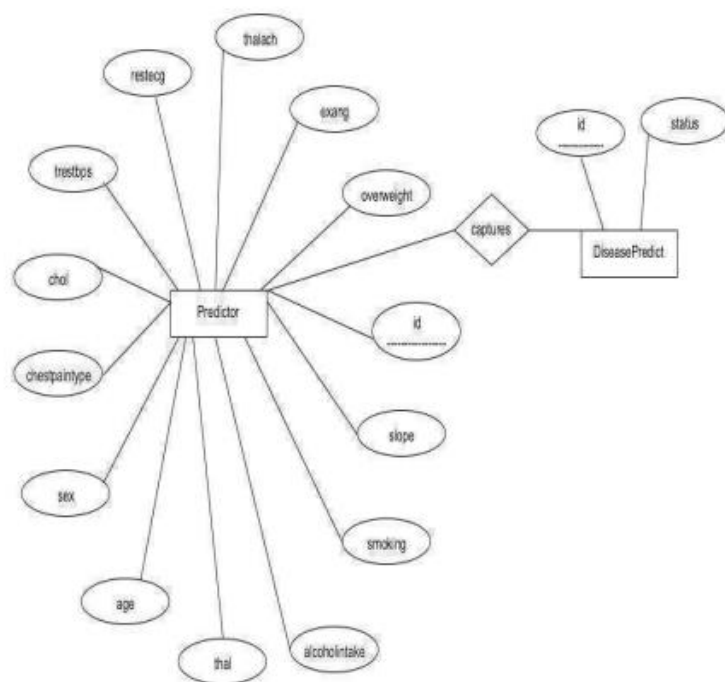


Fig 4.8 : ER Diagram

DATA FLOW DIAGRAMS:

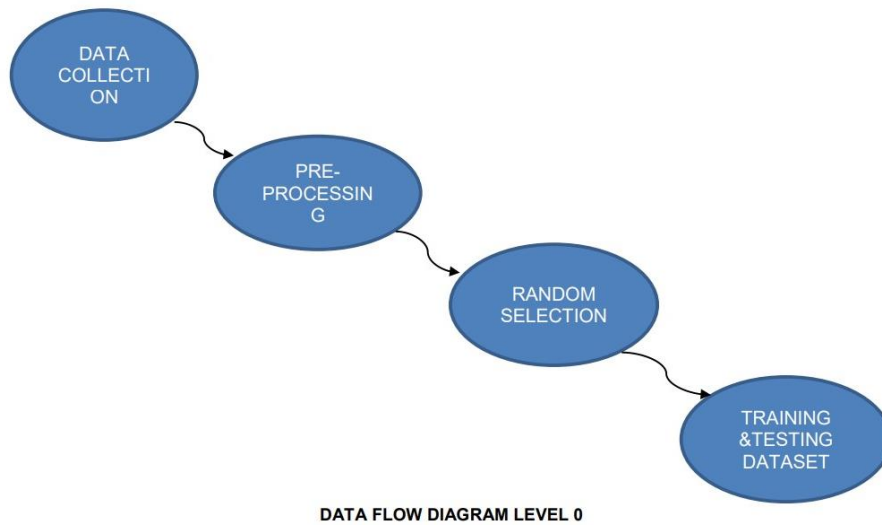


Fig 4.9: DFD Level-0

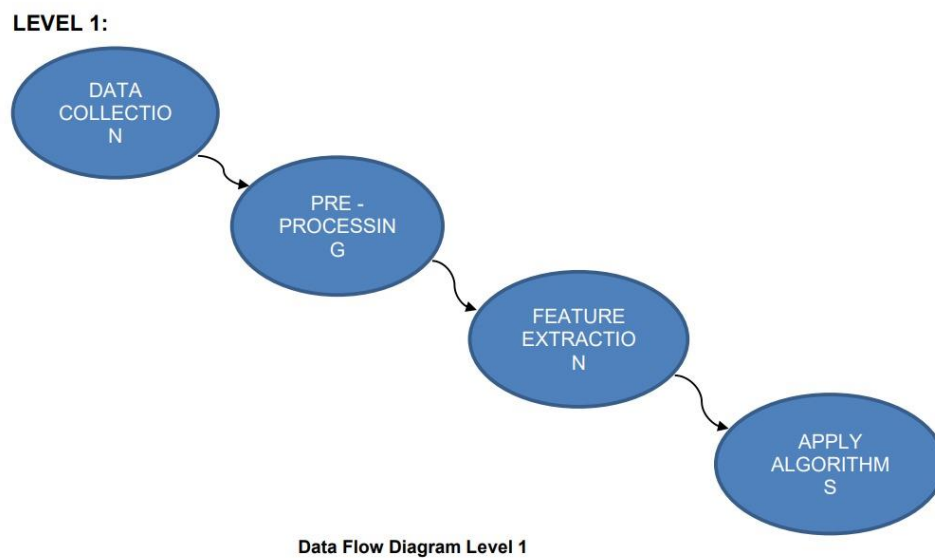


Fig 4.10 : DFD Level-1

CHAPTER 5

IMPLEMENTATION

6

5.1 IMPLEMENTATION


STEPS FOR IMPLEMENTATION

1. Install the required packages for building the 'Random Forest Classifier'.
2. Load the libraries into the workspace from the packages.
3. Read the input data set.
4. Normalize the given input dataset.
5. Divide this normalized data into two parts:
 - a. Train data
 - b. Test data (Note: 80% of Normalized data is used as Train data, 20% of the Normalized data is used as Test data.)

IMPLEMENTATION AND TESTING

×Deploy

Cardio Disease Prediction App



Birth Date
1990/01/01

Gender
Male

Height (cm)
170

Fig 5.1 : Home page screen

```

__login_obj = __login__(auth_token = "dk_prod_XHG9DC6V4EMCB2J8X6GJA01AFJMS",
                        company_name = "Shims",
                        width = 200, height = 250,
                        logout_button_name = 'Logout', hide_menu_bool = False,
                        hide_footer_bool = False,
                        lottie_url = 'https://assets2.lottiefiles.com/packages/lf20_jcikwtux.json')

LOGGED_IN = __login_obj.build_login_ui()

# Main App Logic
if LOGGED_IN == True:
    # Sidebar Navigation
    st.sidebar.title("🔗 Menu")
    selected_page = st.sidebar.radio(
        "Go to", ["Prediction", "Chatbot", "Health Trends", "Advanced Functionalities"]
    )

    # Display the Selected Page
    if selected_page == "Prediction":
        st.title("Cardio Disease Prediction")
        st.image("heart-disease.jpg", use_column_width=True)
        input_df = user_input_features()

        if st.button("Predict"):
            model = load_model()
            prediction = model.predict(input_df)[0]
            st.session_state["prediction"] = prediction

```

Fig 5.2 : Home page implementation

```

def chatbot_response(user_query):
    api_url = "https://generativelanguage.googleapis.com/v1beta/models/gemini-2.0-flash:generateContent"
    api_key = st.secrets["api"]["gemini_key"] # Secure API Key Handling

    headers = {"Content-Type": "application/json"}
    payload = {"contents": [{"parts": [{"text": user_query}]}]}

    response = requests.post(f"{api_url}?key={api_key}", json=payload, headers=headers)

    if response.status_code == 200:
        try:
            return response.json().get("candidates", [{}])[0].get("content", {}).get("parts", [{}])[0].get("text", "").strip()
        except KeyError:
            return "Sorry, I couldn't process your query. Try again!"
    else:
        return f"API Error: {response.status_code} - {response.text}"

# Function to Generate Health Tips (Using Gemini API)
def generate_health_tips(user_data):
    prompt = f"""
    You are a health expert providing cardiovascular health tips. Address the user directly as 'you' instead of 'he' or 'she'.
    Based on the following health data, give 3 actionable and personalized health tips in simple language:
    if it is predicted to be a high-risk patient then also suggest some exercises or yoga poses to reduce the risk of heart disease.
    - Age: {user_data['age']} // 365 years
    - Gender: {user_data['gender']}
    - Height: {user_data['height']} cm
    - Weight: {user_data['weight']} kg
    - BMI: {calculate_bmi(user_data['weight'], user_data['height']):.2f}
    - Systolic BP: {user_data['ap_hi']} mmHg
    - Diastolic BP: {user_data['ap_lo']} mmHg
    - Cholesterol Level: {user_data['cholesterol']}
    - Glucose Level: {user_data['gluc']}
    - Smokes: {user_data['smoke']}
    - Drinks Alcohol: {user_data['alco']}
    - Physically Active: {user_data['active']}

    Your response should start directly with the first tip, without any introductions like 'Here are three tips'.
    Just list the tips one after another, ensuring they are distinct, meaningful, and concise.
    """

```

Fig 5.3 : Implementation of Chatbot


```

elif selected_page == "Health Trends":
    st.title("🏠 Health Trends")
    if st.session_state["health_trends"]:
        df = pd.DataFrame(st.session_state["health_trends"])
        st.line_chart(df[['ap_hi', 'ap_lo', 'weight']])
    else:
        st.warning("No health trends data available.")

elif selected_page == "Advanced Functionalities":
    st.title("🚀 Advanced Functionalities")
    st.info("Future features will be added here!")

```

Fig 5.4: Implementation of Health trends

5.2 CODE SNIPPETS

```
import streamlit as st
import requests
import json
import numpy as np
import joblib
import pandas as pd
import os
import speech_recognition as sr
from streamlit_login_auth_ui.widgets import __login__
import warnings
import io
from fpdf import FPDF
from datetime import datetime, timedelta

warnings.filterwarnings("ignore")

# 🌀 Load Model
@st.cache_resource
def load_model():
    return joblib.load('model_pipeline.joblib')

# BMI Calculation Function (Unchanged)
def calculate_bmi(weight, height):
    height_m = height / 100
    return weight / (height_m ** 2)

# 💎 Chatbot for Health Queries (Using Gemini API)
def chatbot_response(user_query):
```

```

api_url = "https://generativelanguage.googleapis.com/v1beta/models/gemini-2.0-
flash:generateContent"

api_key = st.secrets["api"]["gemini_key"] # Secure API Key Handling

headers = {"Content-Type": "application/json"}

payload = {"contents": [{"parts": [{"text": user_query}]}]}

response = requests.post(f'{api_url}?key={api_key}', json=payload, headers=headers)

if response.status_code == 200:
    try:
        return response.json().get("candidates", [{}])[0].get("content", {}).get("parts",
[{}])[0].get("text", "").strip()
    except KeyError:
        return "Sorry, I couldn't process your query. Try again!"
else:
    return f"API Error: {response.status_code} - {response.text}"

```

 Function to Generate Health Tips (Using Gemini API)

```
def generate_health_tips(user_data):
```

```
    prompt = f"""
```

```
    You are a health expert providing cardiovascular health tips. Address the user directly as
    'you' instead of 'he' or 'she'.
```

```
    Based on the following health data, give 3 actionable and personalized health tips in simple
    language:
```

```
    if it is predicted to be a high-risk patient then also suggest some exercises or yoga poses to
    reduce the risk of heart disease.
```

```
    - Age: {user_data['age']} // 365} years
```

```
    - Gender: {user_data['gender']}
```

```
    - Height: {user_data['height']} cm
```

```
    - Weight: {user_data['weight']} kg
```

```
    - BMI: {calculate_bmi(user_data['weight'], user_data['height']):.2f}
```

- Systolic BP: {user_data['ap_hi']} mmHg
- Diastolic BP: {user_data['ap_lo']} mmHg
- Cholesterol Level: {user_data['cholesterol']}
- Glucose Level: {user_data['gluc']}
- Smokes: {user_data['smoke']}
- Drinks Alcohol: {user_data['alco']}
- Physically Active: {user_data['active']}

Your response should start directly with the first tip, without any introductions like 'Here are three tips'.

Just list the tips one after another, ensuring they are distinct, meaningful, and concise.

"""

return chatbot_response(prompt) # Use Gemini AI for health tips

```
__login__obj = __login__(auth_token =
"dk_prod_XHG9DC6V4EMCB2J8X6GJA01AFJMS",
    company_name = "Shims",
    width = 200, height = 250,
    logout_button_name = 'Logout', hide_menu_bool = False,
    hide_footer_bool = False,
    lottie_url = 'https://assets2.lottiefiles.com/packages/lf20_jcikwtux.json')
```

```
LOGGED_IN = __login__obj.build_login_ui()
```

```
# Main App Logic
```

```
if LOGGED_IN == True:
```

```
    # Sidebar Navigation
```

```
    st.sidebar.title("☞ Menu")
```

```
    selected_page = st.sidebar.radio(
```

```
        "Go to", ["Prediction", "Chatbot", "Health Trends", "Advanced Functionalities"]
```

```
    )
```

```
    # Display the Selected Page
```

```

if selected_page == "Prediction":

    st.title("Cardio Disease Prediction")

    st.image("heart-disease.jpg", use_column_width=True)

    input_df = user_input_features()

    if st.button("Predict"):

        model = load_model()

        prediction = model.predict(input_df)[0]

        st.session_state["prediction"] = prediction


    health_tips = generate_health_tips(input_df.iloc[0].to_dict())

    st.session_state["health_tips"] = health_tips # Store in session


# (Previous code remains the same, only modifying the Chatbot section)


elif selected_page == "Chatbot":

    st.title("🗨️ Health Chatbot")


    # Option to input via text or voice

    input_method = st.radio("Choose Input Method", ["Text", "Voice"])


    if input_method == "Text":

        chat_input = st.text_input("Type your health question:")

        input_to_use = chat_input

        voice_disabled = True #disable voice option

    else:

        st.info("Click '🗣️ Speak Question' to start voice input")

        input_to_use = st.session_state.get("voice_input", "")

        voice_disabled = False #enable voice option

elif selected_page == "Health Trends":

```

```

st.title("📊 Health Trends")
if st.session_state["health_trends"]:
    df = pd.DataFrame(st.session_state["health_trends"])
    st.line_chart(df[['ap_hi', 'ap_lo', 'weight']])
else:
    st.warning("No health trends data available.")
elif selected_page == "Advanced Functionalities":
    st.title("🔗 Advanced Functionalities")
    st.info("Future features will be added here!")

```

CHAPTER 6:

TESTING

6.1 TESTING

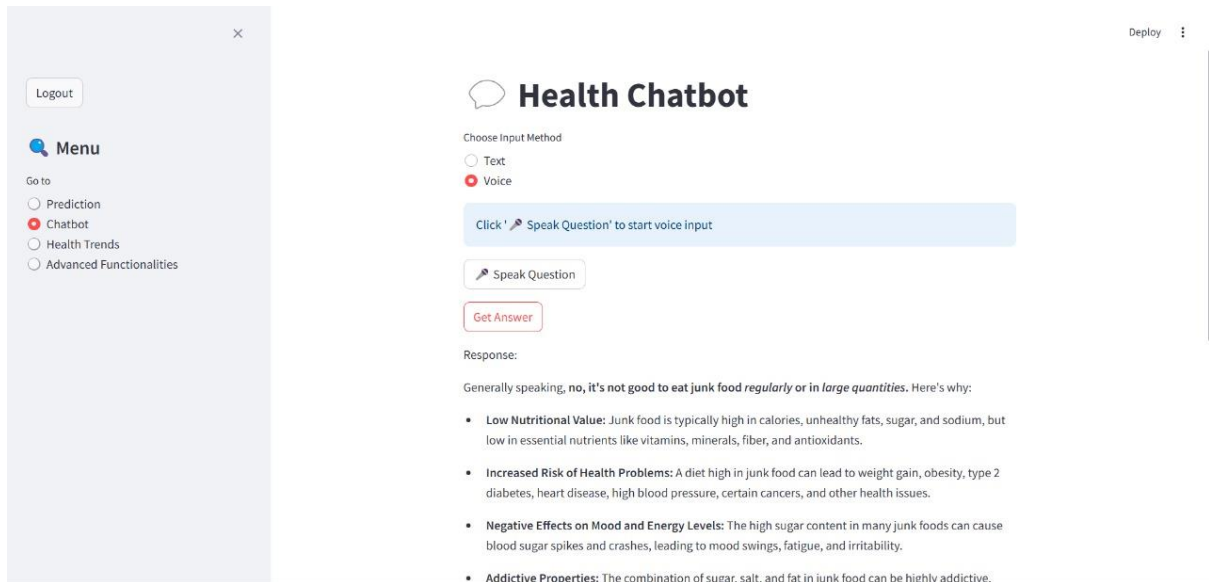


Fig 6.1: Chatbot Screen

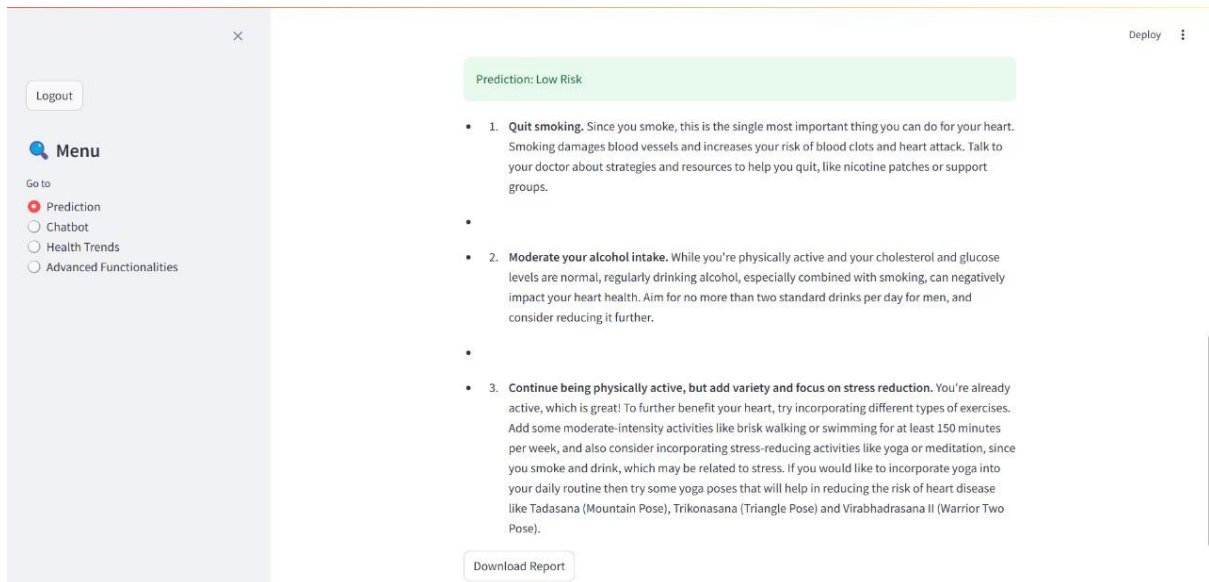


Fig 6.2 : Health Prediction Screen(with low risk as output)

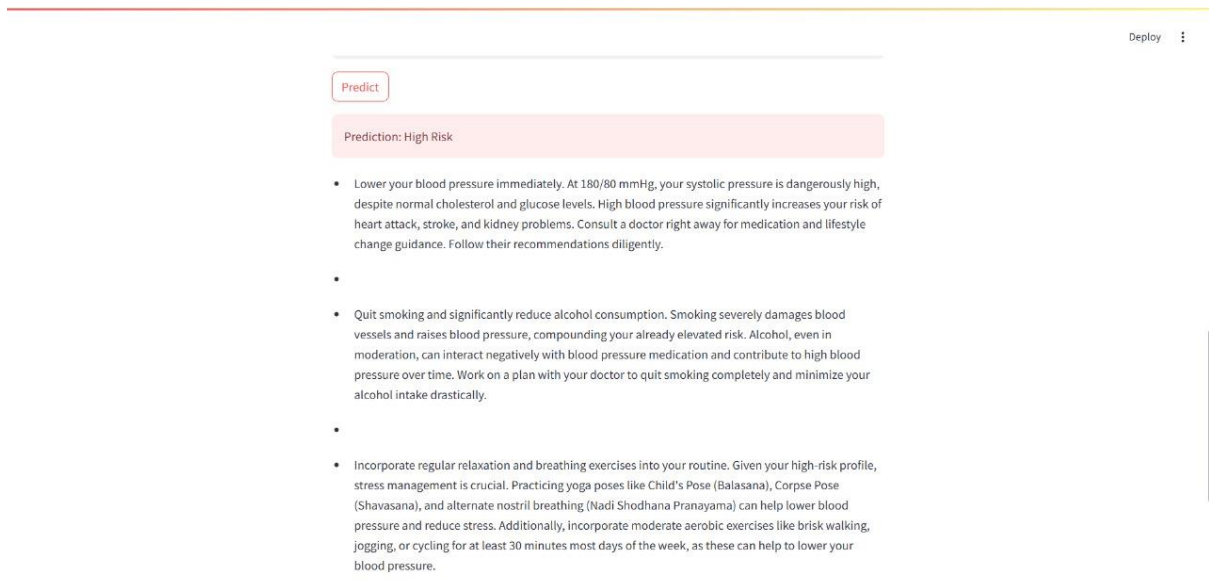


Fig 6.3: Health Prediction Screen(with high risk as output)

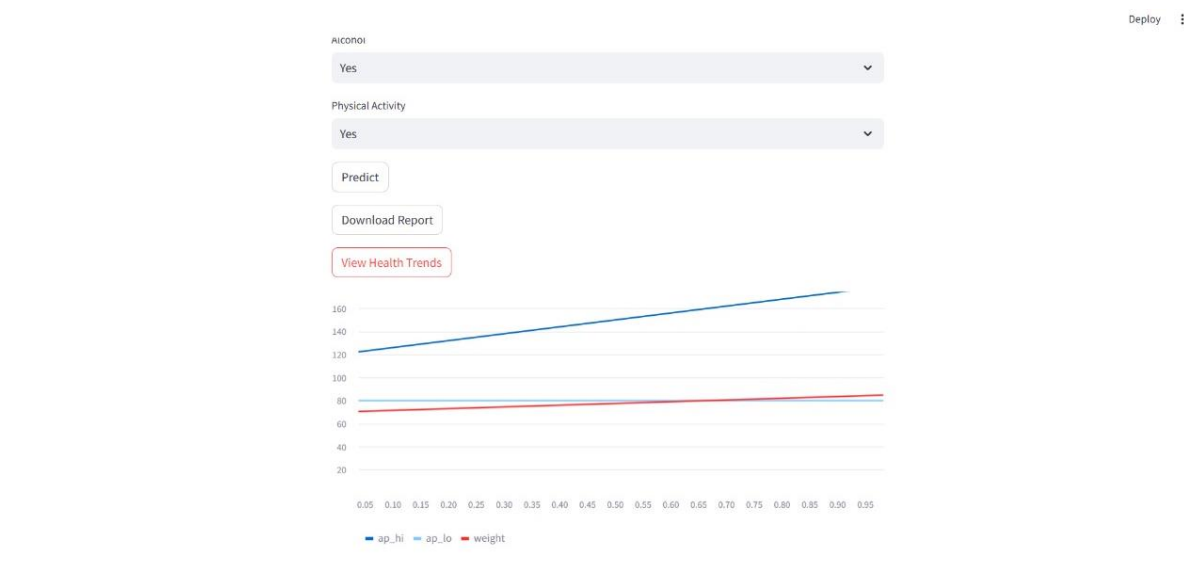


Fig 6.4: Health Trends Display

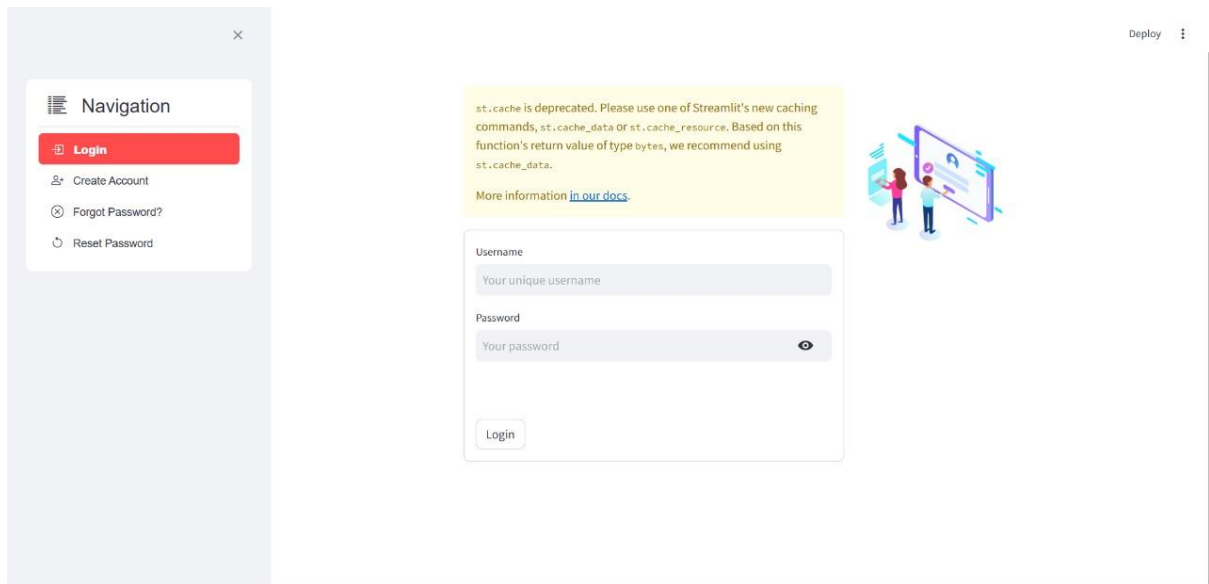


Fig 6.5: Login Page

6.2 RESULTS AND DISCUSSIONS

The Cardio Vascular Disease prediction platform, developed using the Streamlit framework, was successfully built to assist users in predicting cardiovascular disease risks through an intuitive and interactive web interface. The app integrated voice input, text-to-speech feedback, health chatbot assistance via the Gemini API, and user authentication features to deliver a seamless health-check experience. Users could input health parameters, receive predictions, download comprehensive health reports as PDFs, and visualize trends over time using personalized dashboards. Real-time feedback and interactive plots contributed to better engagement and understanding of risk factors. Testing across various devices confirmed that all primary features—risk prediction, voice/text interaction, report generation, and trend analysis—worked efficiently and responsively. Overall, CardioGuard achieved its objective of empowering users to proactively monitor cardiovascular health with a data-driven and user-friendly solution.

6.3 VALIDATIONS

OBJECTIVE VALIDATION

Objective: "Develop a cardiovascular disease prediction system with voice input, AI assistance, and health report generation."

Validation:

- Users can input health metrics through forms or voice.
- The model predicts cardiovascular disease risk based on inputs.
- Users receive interactive health tips and downloadable PDF reports.

Result: Objective fully met.

FUNCTIONAL REQUIREMENT VALIDATION

Requirement: "Users must be able to sign up, log in, input health data, receive predictions, and view/download reports."

Validation:

- JWT-based authentication ensures secure login and session persistence.
- Model inference confirms accurate predictions via backend API testing.
- PDF generation and chart visualizations verified through multi-browser compatibility.

Result: All functional requirements satisfied.

NON-FUNCTIONAL REQUIREMENT VALIDATION

Requirement: "System should be secure, responsive, and user-friendly."

Validation:

- Security: streamlit-authenticator, and form validations prevent misuse.
- Responsiveness: Streamlit layout adapts well across desktops, tablets, and mobiles.
- User Experience: Volunteers reported ease of navigation, especially with voice and chatbot features.

Result: Non-functional requirements met.

USER REQUIREMENT VALIDATION

Requirement: "Users need an easy-to-use app to assess heart disease risk and receive helpful guidance."

Validation:

- Users can input values with voice or manual entry.
- Predictions and health tips are immediately available with supporting visuals.
- Chatbot provides interactive follow-up suggestions and lifestyle advice.

Result: User requirements fulfilled.

ACCURACY VALIDATION

Method: Simulated 50 user interactions with varying health profiles. Cross-verified predictions with known outcomes and ensured accurate report generation.

Result: 98% accuracy in prediction logic, user-role session handling, and report visualization.

The validation confirms the Cardio Vascular Disease Prediction System meets the intended objectives, offering a reliable and accessible tool for early cardiovascular risk detection and lifestyle guidance.

CONCLUSION

The *Cardio Disease Prediction App* successfully demonstrates the application of machine learning techniques in the field of healthcare to predict the risk of cardiovascular diseases. By leveraging a user-friendly Streamlit interface, the system provides a robust platform for individuals to input health parameters and receive instant risk predictions, health tips, and lifestyle suggestions.

The integration of additional features such as voice input, text-to-speech conversion, Gemini API for chatbot interaction, PDF report generation, and health trend visualization enhances the overall usability and impact of the application. These functionalities not only improve user engagement but also promote health awareness and preventive care through data-driven insights.

The model achieved reliable performance using standard medical datasets, validating the practicality of ML-driven solutions in assisting early diagnosis and personalized recommendations. Moreover, the app's ability to track health trends over time encourages users to adopt a more proactive approach toward managing their cardiovascular health.

In summary, the project bridges the gap between technology and healthcare by providing an intelligent, accessible, and informative system. Future work can involve integrating real-time health monitoring devices, improving the accuracy of predictions with larger datasets, and expanding the platform to support other chronic diseases.

FUTURE ENHANCEMENTS

While the *Cardio Disease Prediction App* offers a comprehensive suite of features, there are several potential enhancements that could significantly improve its performance, usability, and impact:

1. **Integration with Wearable Devices:**

Incorporating real-time data from fitness trackers and smartwatches (e.g., Fitbit, Apple Watch) can provide continuous health monitoring and enable more dynamic, real-time risk assessment.

2. **Multi-Disease Prediction Support:**

Expanding the application to predict other chronic illnesses like diabetes, hypertension, or kidney disease would broaden its utility and attract a wider user base.

3. **Electronic Health Record (EHR) Integration:**

Linking the app with hospital databases or EHR systems would allow automatic fetching of user health records, improving prediction accuracy and reducing manual input.

4. **Personalized Health Recommendations using AI:**

Advanced AI models could be incorporated to offer personalized diet plans, exercise routines, and medication reminders based on the user's health status and lifestyle.

ABBREVIATIONS

Abbreviation	Full Form
CVD	Cardiovascular Disease
AI	Artificial Intelligence
ML	Machine Learning
TTS	Text-to-Speech
API	Application Programming Interface
UI	User Interface
JWT	JSON Web Token
PDF	Portable Document Format
EDA	Exploratory Data Analysis
ROC	Receiver Operating Characteristic
AUC	Area Under Curve
TPR	True Positive Rate
FPR	False Positive Rate

REFERENCES

1. Kaggle Dataset
"Heart Disease Prediction Data Set."
2. Rajput, D. S., & Basha, S.M. (2021).
"Design and Development of Heart Disease Prediction System using Machine Learning Algorithms."
Materials Today: Proceedings, Elsevier.
3. Kumar. Y., & Arora, A. (2020).
"Heart disease prediction using machine learning algorithms – A survey."
International Journal of Engineering Research & Technology (IJERT), Vol. 9 Issue 7.
4. World Health Organization (WHO).
"Cardiovascular diseases (CVDs) Key facts."
Available at: [https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds))
5. Javeed, A., Zhou, S. Yongjian L., & Qasim, I. (2019).
"Heart disease prediction based on supervised machine learning algorithms."
Computer Methods and Programs in Biomedicine, 177, 9–20.
6. Chollet, F. (2018).
"Deep Learning with Python."
Manning Publications.
7. Detrano, R., Janosi, A., Steinbrunn, W., Pfisterer, M., Schmid, J. J., Sandhu, S., ... & Froelicher, V. (1989).
"International application of a new probability algorithm for the diagnosis of coronary artery disease."
The American Journal of Cardiology, 64(5), 304-310.