Heart Disease Detection Model by using Machine Learning and flask API

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

Bachelor of Technology

in

Electronics and Communication Engineering

by

M. Aneesh Reddy (160118735023)
K. Anurag (160118735024)

Under the Guidance of

Sri. G.V. Pradeep Kumar
Assistant Professor, Dept. of ECE, CBIT

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY
Affiliated to O.U, Accredited by NBA NAAC, ISO 9001:2015 Certified
Institution ChaitanyaBharathi P.O. Gandipet, Hyderabad-500075

Heart Disease Detection Model by using Machine Learningand flask API

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

Bachelor of Technology

in

Electronics and Communication Engineering

by

M. Aneesh Reddy (160118735023)
K. Anurag (160118735024)

CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY
Affiliated to O.U, Accredited by NBA NAAC, ISO 9001:2015 Certified Institution ChaitanyaBharathi P.O. Gandipet, Hyderabad-500075

Department of Electronics and Communication Engineering

ChaitanyaBharathi Institute of Technology (A),

Hyderabad-500075

CERTIFICATE

This is to certify that the project titled **“Heart Disease Detection Using Machine Learning”** is the bonafide work carried out by Aneesh Reddy Mekala (160118735023), Anurag Kalapala (160118735024) students of B.E.(ECE) of Chaitanya Bharathi Institute of Technology(A), Hyderabad, affiliated to Osmania University, Hyderabad, Telangana(India) during the academic year 2022-2023, submitted in partial fulfilment of the requirements for the award of the degree in Bachelor of Engineering (Electronics and Communication Engineering) and that the project has not formed the basis for the award previously of any other degree, diploma, fellowship or any other similar title..

G.V. Pradeep Kumar
Supervisor
Assistant Professor , Department of ECE
Chaitanya Bharathi Institute of Technology
Hyderabad-500075

Dr. D. Krishna Reddy
Head of the Department
Electronics and Communication Engineering
CBIT (A)
Hyderabad-500075

DECLARATION

This is to certify that the work reported in this project titled “Heart disease detection model by using machine learning and flask api ” is a record work done by us in the Chaitanya Bharathi Institute of Technology, Hyderabad in fulfilment of degree for the award of Bachelor of Engineering is a bonafide work done by us, which was carried under the supervision of G.V.Pradeep Kumar.

Also, we declare that the matter embedded in this report has not been submitted by us in full or partial thereof for the award of any degree/diploma of any other institution or University previously.

M. Aneesh Reddy	:	160118735023
K. Anurag	:	160118735024

Date:

Place: Hyderabad

Acknowledgements

At the very outset of this report, we would like to thank with utmost gratitude and sincerity to all the people who have helped us in this effort. Without their guidance, support and encouragement we wouldn’t have made it through this project.

We were gratified and beholden to our gratitude to our guide Sri. G . V. Pradeep Kumar, Assistant Professor, Dept of ECE CBIT, for her constant guidance and support through our work, attending to our needs and helping us in finding solutions to various problems encountered.

We sincerely thank our project coordinator Dr. K. Suman, Dr.Vinod Kumar Assistant professor, Dept. of ECE CBIT, for various suggestions which have benefited our work in various ways.

We express our professional gratitude to Dr. D. Krishna Reddy, Head of the Dept. ECE CBIT, for his variable guidance, enormous support and providing required facilities in carrying out our project as well as preparing this document.

We also thank our parents who have been supporting us from the very beginning. It is their support and cooperation without which this work wouldn’t have been possible. We also thank our seniors for their devices and guidance whenever we require it.

Finally, we express our immense gratitude with pleasure to all our friends and classmates who have extended their time, energy and valuable knowledge for successfully completing our project work.

ABSTRACT

Machine Learning has been effectively used in real-world situations; nonetheless, its application in real-world problems is relatively new, with the goal of identifying and extracting new and potentially valuable knowledge from data. The goal of this Machine Learning research is to create a model that can predict a person's cardiovascular illness.

One person dies every 36 seconds from cardiovascular disease. About 18.2 million of adults age 20 and older have CAD (about 6.7%). By 2030, almost 23.6 million people will die from CVDs. About 2 in 10 deaths from CAD happen in adult less than 65 years. Maximum of deaths are occurred before the diagnosis due to high cost for tests. This prediction model is to identify the initial stage whether the patient need to proceed for the further diagnosis.

The patients' medical histories are gathered, the data is combined, and predictions are made. The performance of the learning methods is evaluated based on their accuracy, ease of learning, and user-friendly qualities. Supervised algorithms are used to predict whether the patient has a high likelihood of getting CAD. A user interface has been designed for regular people to enter data using the FLASK API.

Keywords: CAD, Machine Learning, Supervised Learning, Flask

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Statement

Heart disease is the leading cause of death for people of most racial and ethnic groups Deaths are occurred before the diagnosis for CAD. Machine Learning can be used to predict the cardiovascular disease using Random Forest algorithm. The predicted value is used to estimate the occurrence of the CAD and can be treated accordingly.

1.2 Existing System

Many existing machine learning models for health care analysis focus on a single disease. For example one analysis for diabetes another for cancer, and so on. There is no universal approach that can forecast multiple diseases with a single analysis. This model is used to examine diabetes, diabetic retinopathy, heart disease, and cancer. They used machine learning algorithms and the Flask API to create this model. Python pickling is used to save model behaviour, while Python unpicking is used to load the pickle file**.**

Disadvantages:

1.The collected Data set is very less in number around one thousand, hence it cannot produce more accurate results compared with the proposed system.

2.The attributes(data) which they have used such as cp, trestbps, fbs, restecg, exang, etc. are not available with all the users and It is not user friendly.

1.3 Proposed System

This model was created with all of the general characteristics in mind, such as age, weight, gender, cholesterol, glucose level, smoking habit, and so on. 70,000 rows of data were collected and trained with five Supervised Machine Learning Algorithms to generate this model. The Random Forest Algorithm, which produced the maximum accuracy of 73 percent, was used to train the machine.The model's front-end was constructed using the Flask API, and it informs users about their current health state.

Advantages:

1.This produces more accurate results with the proposed system.

2.This system has user-friendly interface.

1.4 System Requirements

1.4.1 Hardware Requirements:

1. System	: Intel i5 octa core
2.Processor	: 1.5 GHZ to 2.6 GHZ
3.RAM	: 4 GB Minimum
4.Hard disk	: 256 GB

1.4.2 Software Requirements:

1. Operating System	: Windows/Mac/Linux
2.Platforms	: Anaconda prompt,Jupyter Notebook
3. Software	: Python 3.8.5

1.5 Aim

The goal of the heart disease detection system is to use a machine learning algorithm on a medical data set to help forecast heart disorders.

1.6 Technical Approach

The automated personality method analyzes the user's personality to the outcomes of conventional personality tests. Personality prediction is primarily based on a person's behaviour. Various assessments will be conducted by asking a series of questions, and personality will be assessed based on the responses provided by the user. The primary function is to provide an outline for the construction of personality prediction based on the answers to the many questions.

Python:

Python is a scripting language that is high-level, interpreted, and object-oriented. Python is intended to be a very understandable language. Python has been used to build the classifiers.

Pandas:

Pandas library provide easy to use data structures and it has been used for data analysis.**Tkinter:**

Tkinter is Python's standard GUI library. Python and Tkinter make it simple and quick to design graphical user interfaces. Tkinter is an object-oriented foundation for the Tk GUI Frameworks. The GUI was built using this interface.

Data Base:

One of the most common formats for datasets used in machine learning and data science is CSV (comma separated values). MS Excel may be used to manipulate data in CSV format in a simple way. However, we must convert CSV files to data tables before we can run complicated SQL queries on them. Create a script that reads the CSV data and populates the appropriate data table. Although this method is faster than copy-pasting, it still necessitates the use of a manual script. With a few mouse clicks, we can import and transform CSV files into data tables in SQL Server using SSMS (SQL Server Management Studio). However, a manual script is still required.

Chapter 2

Literature Review

Machine literacy algorithms have a huge potential for prognosticating cardiovascular or heartrelated disorders, according to [2]. In certain cases, each algorithm has performed beautifully, while in others it has failed spectacularly. Interspersing decision trees have performed exceptionally well when used with PCA, but they have fared poorly in other situations, probably owing to over-fitting. Because they utilise numerous algorithms to solve the problem of over-fitting, Random Forest and Ensemble models have performed well (multiple Decision Trees in case of Random Forest). The Naive Bayes classifier-based models were computationally quick and performed well. SVM performed brilliantly in the great majority of situations.

In [5] ,To prognosticate the state of a heart disease, many types of data mining and machine literacy have been applied. Determine the vaticination performance of each algorithm and put the suggested system in place in the needed region. Use more suitable point selection styles to increase algorithm accuracy. There are numerous treatment choices available if they have been diagnosed with a certain type of cardiac condition.

From similar data sets, data mining can generate verified information. As a consequence of the literature study, creating predictive models for heart complaint patients has only achieved a minimal success, and that there is a need for combinational and more sophisticated models to enhance the delicacy of prognosticating the early beginning of heart illness.

The project in [9] purpose is to employ data mining techniques to better forecast the circumstances around a cardiac disease. In this work, researchers used the UCI data bank to compare three algorithms that are comparable to Random Forest, Decision Trees, and Naive Bayes. Random Forest has been scientifically proved to give optimum outcomes when compared to Decision Tree and Naive Bayes.

In study of [3], decision trees frequently lead to incorrect results in small datasets, but Nave Bayes findings produce more accurate results with chances of all other possibilities, yet decision trees may overlook leads since they only guide to one conclusion. Finally, this experiment reveals that if the input data is gutted and well kept, the Nave Bayes is more accurate. While ID3 can clean up the tone, it can't always deliver correct findings, and Nave Bayes is no exception. This implies that the outcomes of many algorithms must be taken into account, and any vaticination performed will be precise. However, Nave

Bayes treats variables as individuals and may combine algorithms such as Nave Bayes and K-means.

In article [7] looked at data sets for diabetes analysis, diabetic retinopathy, heart disease, and cancer detection. Many additional disorders, such as skin diseases, fever-related diseases, and others, may be included in the future. This study is adaptable, and it has now been expanded to encompass a variety of disorders. When adding a new illness analysis to this API, the developer must also include the model file for the new disease's analysis. It is feasible to forecast more than one illness at a time using a multi disease model. As a result, there is no need for the user to go among many models in order to anticipate illnesses. It will save time, and it has the potential to lower death rates by predicting numerous illnesses at once.

Chapter 3
Technical Background

3.1 Python :

Python is a high-level, general-purpose programming language that is interpreted. Python's design philosophy prioritises code readability, as evidenced by the use of indentation. Its language components and object-oriented philosophy are intended to assist programmers in writing clear, concise code for both small and big projects. Python is garbage-collected and dynamically structured. Objectoriented programming, structured (especially procedural) programming, and functional programming are all supported. because of its vast standard library collection, Python is sometimes referred to as a "batteries included" language. Guido van Rossum and others developed Python as a replacement for the ABC programming language in the late 1980s, and Python 0.9.0 was released in 1991. In the year 2000, Python 2.0 was released, with significant enhancements such as list comprehensions and a garbage collection system based on reference counting. In 2008, Python 3.0 was published, and it was a major upgrade to the language. However, it wasn't completely backwards compatible, and many Python 2 scripts don't work as well with Python 3. Python is one of the most widely used programming languages in the world today.Python is a high-level, general-purpose programming language that is interpreted. Python's design philosophy prioritises code readability, as evidenced by the use of indentation. Its language components and object-oriented philosophy are intended to assist programmers in writing clear, concise code for both small and big projects. Python is garbage-collected and dynamically structured. Object-oriented programming, structured (especially procedural) programming, and functional programming are all supported. because of its vast standard library collection, Python is sometimes referred to as a "batteries included" language. Guido van Rossum and others developed Python as a replacement for the ABC programming language in the late 1980s, and Python 0.9.0 was released in 1991. In the year 2000, Python 2.0 was released, with significant enhancements such as list comprehensions and a garbage collection system based on reference counting. In 2008, Python 3.0 was published, and it was a major upgrade to the language. However, it wasn't completely backwards

compatible, and many Python 2 scripts don't work as well with Python 3. Python is one of the most widely used programming languages in the world today.

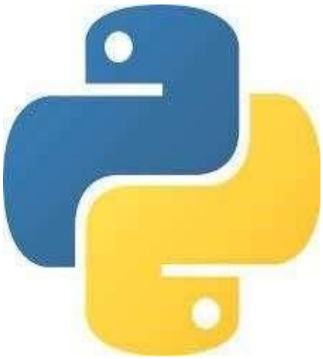


Figure 3.1 python logo

The above figure is the interface of python logo.The Python logo consists of an iconic emblem on the right and a wordmark on the left. The symbol is based on ancient Mayan designs and displays a dual coloured serpent figure. A python with a shortened tail and a large head was generally depicted by them.

3.1.1 Features in Python

•Python is a high-level computer language that is simple to code.

•Open Source and Free Software.

•Object-Oriented Programming.

•Tools for GUI development

•Sophisticated Language.

Python is a versatile & flexible programming language.

3.1.2 Syntax and Semantics

Python is designed to be a user-friendly platform. It has clean structure and frequently employs English terms where other languages employ punctuation. It does not employ curly brackets to delimit blocks, unlike many other languages rather it uses indentation , and semicolons following statements are permitted but seldom, if ever, used. In comparison to C or Pascal, it features minimal syntactic exclusions and special circumstances.

3.1.3 Python Development Environment:

A read–eval–print loop (REPL) is contained in numerous Python executions (including C, Python), allowing these to serve as a command line interpreter wherein the user logs instructions sequentially and receives responses instantly. Other environments, like as IDLE and IPython, include additional features such as auto-execution, session state preservation, and syntax underscoring. There are Webclient IDEs, Sage Math (for executing science and math-related Python applications), Python. Anywhere, a browser-based IDE and hosting platform, and Canopy IDE, a commercial Python IDE highlighting scientific processing, in addition to regular desktop integrated development environment.

3.1.4 Python Libraries:

One of Python's major features is its vast standard library , which contains tools for a wide range of activities. Many common types and protocols, such as MIME and HTTP, are accessible for Web projects. It has packages for constructing graphical user interfaces, connecting to relational database systems, producing pseudorandom values, doing arithmetic using random decimals, processing regular expressions, and functional testing.

Some elements of the standard library are addressed by specific requirements example, the execution of the Web Server Gateway Interface (WSGI) follows PEP 333—but the majority are described by their script, intrinsic literature, and test scripts. Hardly just few packages need to be changed or rebuilt for implementations because the majority of the standard library is cross-platform Python functionality.

3.1.5 Uses:

Python can be used to create web-based applications using Apache's mod wsgi. The Web Server Gateway Protocol has built a standard API for these services. Web frameworks such as Django, Pylons, Pyramid, TurboGears, web2py, Tornado, Flask, Bottle, and Zope allow developers to create and manage complex systems. Pyjs and IronPython can be used to create the client side of Ajax-based apps. SQLAlchemy can be used to export material from relational databases. Dropbox, for example, employs the Twisted architecture to manage machine connections.

3.2 Jupyter Notebook:

Project Jupyter is an open-source, member-supported project that grew from the IPython Project in 2014 to facilitate effective data research and systematic computing between all computer languages. Jupyter will always be 100% opensource software, distributed under the provisions of the altered BSD licence.

Jupyter is open-source and built on GitHub with the assistance of the Jupyter society. The Jupyter Code of Practice governs all digital and in-person exchanges and interactions directly linked to the project. This Code of Practice establishes guidelines for a heterogeneous group of users and collaborators to engage in the initiative with dignity and safety.

Distinguished Project Jupyter Creators are credited for making significant volume and quality improvements to Jupyter over a time of at least two years.Code, code review, infrastructure upgrades, mailing list maintenance, and other contributions are all possible.Engagement in chats, group assistance/building, public outreach, and financing ,promotion, advertising, diversity and inclusion, user experience design and research, and so on. Up to 10 , Each year, the aggregate group of Outstanding Creators selects new Exceptional Contributors.

Figure 3.2 Jupyter Logo

The above figure represents the jupyter notebook logo.

3.3 Pandas:

Pandas is an open-source Python library that uses strong data architectures to provide high- data processing and evaluation. Pandas gets its name from the term Panel Data, which is an Econometrics term for multivariate data.

When developer Wes McKinney required a high-efficient, versatile tool for data evaluation, he started designing pandas in 2008.

Python was primarily used for data mining and pre processing before Pandas. It made just a minor contribution to data analysis. This issue was fixed by pandas.

We can import, arrange, modify, classify, and analyze the data with Pandas regardless of its origin.

Python with Pandas is utilised in a variety of sectors, including finance, economics, statistics, and analytics, in both academic and commercial settings. Pandas is mostly used to analyse data. Pandas supports loading data from commaseparated data, JSON, SQL, and Microsoft Excel files. Pandas supports a number of data manipulation techniques, including combining, reshaping, and selecting, as well as data cleaning and wrangling. There are several tools in pandas that allow us to interface with databases and alter them in various ways.

3.4 Numpy:

NUMPY is a Python library that adds substructure for massive, multivariate arrays and matrices, as well as a wide number of high-level numerical methods to work on these arrays. Numeric, NumPy's forerunner, was built by Jim Hugunin with help from a number of other people. Travis Oliphant built NumPy in 2005 by heavily modifying Numeric and combining features from the competitor Numarray. NumPy is an open source project with numerous developers.

3.5 Tkinter:

Tkinter is a popular Python framework for generating graphical user interfaces for desktop programs. It is not difficult to create desktop apps with Python Tkinter. The techniques below can be used to generate an empty Tkinter top-level panel:

- Tkinter should be imported.
- Create the application's primary panel/window.
- Add components to the panel, such as labels, buttons, frames, and so on.
- To make activities to take place on the user's computer screen, main event loop need to be called. Widgets such as button, canvas, checkbutton, entry, and others are used to create python GUI Programs.

Table 3.1

S. No	Widget	Description
1	Button	The Button class is used to add several types of controls to a Python programme.
2	Canvas	The canvas component is utilised to draw on the window's canvas.
3	Check Button	The CheckButton is shown on the panel using the Checkbutton.
4	Entry widget	The entry widget is availed to show the user a quick text field. It's frequently used to take user input.
5	Frame	It may be characterised as a variable to which another applet can be added and arranged.
6	Label	A label is a piece of text that is utilised to show a particular message about other widgets.
7	List Box	The ListBox widget is availed to show a user a range of suggestions.
8	Menu Button	The Menubutton is needed to show the viewer the dropdown menus.
9	Radio Button	A radiobutton is not the same as a checkbutton. The user is presented with several alternatives, from which he or she may choose just one.
10	Top Level	It's required to make a different window frame.

The above table gives the information about total weight and their description.

3.6 Scikit-Learn

Scikit-learn is without a doubt Python's most useful machine learning library. Among the many useful tools in the sklearn package for machine learning and statistical modelling are classification, regression, clustering, and dimensionality reduction.

It is important to remember that sklearn is a tool for creating machine learning models.

It should not be used for data reading, manipulation, or summarization.

There are more appropriate libraries for this (e.g. NumPy, Pandas etc.) Scikit-learn components include:

Scikit-learn has numerous features.

1) Supervised machine learning

Any supervised machine learning algorithm might be found in Scikit-learn. From generalised linear models (like linear regression) to Support Vector Machines (SVM), Decision Trees, and Bayesian approaches, the scikit-learn toolbox has it all. One of the main reasons for scikit-popularity learn's success is the extensive use of machine learning technologies..

2) Cross-validation:

There are numerous approaches to check the validity of supervised models on unknown data using sklearn..

3) Unsupervised learning techniques:

Clustering, factor analysis, principal component analysis, and unsupervised neural networks are just a few of the machine learning algorithms included in the offering.

several toy datasets:

This was quite useful while studying scikit-learn. We Having these on hand while studying a new library was quite beneficial.

Extraction of features: Scikit-learn is a Python library that extracts attributes from photos and text (e.g. Bag of words).

3.7 Matplotlib

Matplotlib is a fantastic Python visualizing library for 2D array charts. Matplotlib is a multi-platform data visualisation package based on NumPy arrays and intended to operate with the SciPy stack as a whole. It was first presented in 2002 by John Hunter.

One of the most significant advantages of visualisation is that it provides us with visual access to massive volumes of data in simply understandable graphics. Line, bar, scatter, histogram, and more graphs are available in Matplotlib.

Matplotlib and several of its components are available as wheel packages in Windows, Linux, and macOS distributions.

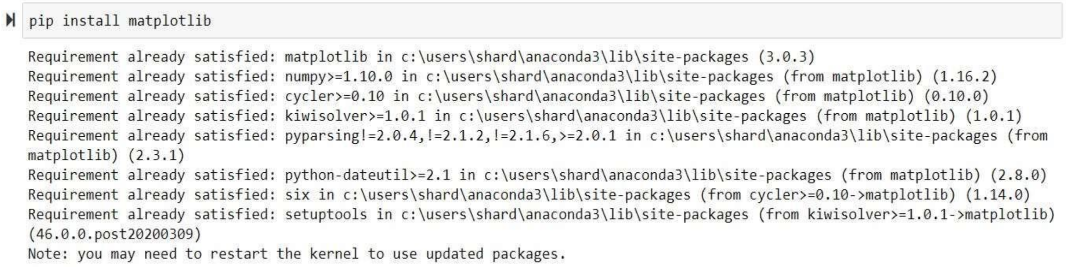
Matplotlib is a Python library that uses NumPy, Python's mathematical extension. It has various plots, including:

- Line
- Bar
- Scatter
- Histrogram
- And many more

Installation:

- Using pip, install Matplotlib.

The Python package controller, pip, may also be used to install Matplotlib. To use pip to install Matplotlib, open a terminal window and typepip install matplotlib.



```
pip install matplotlib

Requirement already satisfied: matplotlib in c:\users\shard\anaconda3\lib\site-packages (3.0.3)
Requirement already satisfied: numpy>=1.10.0 in c:\users\shard\anaconda3\lib\site-packages (from matplotlib) (1.16.2)
Requirement already satisfied: cycler>=0.10 in c:\users\shard\anaconda3\lib\site-packages (from matplotlib) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\shard\anaconda3\lib\site-packages (from matplotlib) (1.0.1)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in c:\users\shard\anaconda3\lib\site-packages (from
matplotlib) (2.3.1)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\shard\anaconda3\lib\site-packages (from matplotlib) (2.8.0)
Requirement already satisfied: six in c:\users\shard\anaconda3\lib\site-packages (from cycler>=0.10->matplotlib) (1.14.0)
Requirement already satisfied: setuptools in c:\users\shard\anaconda3\lib\site-packages (from kiwisolver>=1.0.1->matplotlib)
(46.0.0.post20200309)
Note: you may need to restart the kernel to use updated packages.
```

Figure 3.3

The above figure explains the installation of Matplotlib and how it is done sequentially.

If this command causes Matplotlib to be generated from source and you encounter problems, use —prefer-binary to pick the most recent version of Matplotlib for which a precompiled wheel exists for your OS and Python.

3.8 FLASK API

Flask is a popular micro web platform to build Python APIs. It's a basic but robust development tool that's meant to get you up and running quickly while also allowing you to grow up to more advanced systems.

"Micro" does not imply that your whole webservice must fit inside a sole Python script (though it may), nor to imply as Flask is limited in its capabilities. The "micro" in microframework refers to Flask's goal of keeping the foundation minimal while still being extendable. Flask isn't going to make many decisions for you, like which database to use.

It may easily adjust the judgments it does make, such as the scripting technology to employ. All else is up to you, so Flask can be everything you need while leaving nothing out.

By standard, Flask doesn't provide a network abstract model, form authenticate, or anything else that may be handled by other frameworks. Rather, Flask allows you to use extensions to add functionality to your application that would otherwise be built in Flask. Database integration, form validation, upload handling, multiple open validation protocols, and more are all available through plugins. Flask could be "micro," but it really is capable of handling a wide range of tasks.

When we first start using Flask, you'll see that there are a lot of configuration options with reasonable settings and just few rules to follow. Templates and dynamic documents are saved in folders with the labels templates and basic inside the Python root directory of the program. While one can adjust this, we normally don't need to, notably when first starting off.

Once we got Flask up and operating, you may use the ecosystem to identify plugins to help you connect your projects for operation. Modifications are reviewed by the Flask team members to guarantee that they do not clash with later Flask versions.

We are allowed to make design choices that are acceptable for our application as our codebase increases. Flask will remain to ensure a very basic glue shell for Python's finest features. You may use SQL, Alchemy or another database tool to create sophisticated features, as well as non-relational record preservation and conceptual model utilities designed for WSGI, the Python web functionality.

The resource class, defined by Flask restful, has methods for each HTTP method. The name of the function should be the same as the HTTP method it corresponds to, and it should be typed in lowercase. This may be seen in the script ahead. Moreover, because these approaches lack a channel decorator, they rely on resource routes. We define the route to any class we construct using the add resource function, as well as the route we must call it on.

Decorators are used in conjunction with APIs to track IP addresses, cookies, and other data. So we'll learn using the flask API with decorators within this category. A decorator is a procedure that accepts an argument and returns another function. It may also be thought of as a method that adds new functionality to an existing function without altering or altering the original function.

We must security requirements while designing the API as it will be used by a large number of individuals. What if you only need authorised persons to access the API because it may include sensitive information shared between clients. Basic authentication can be implemented with Flask. This flask package must be installed with the pip instruction.

Safeguarding Python APIs with Auth0 is simple and comes with a lot of useful functions. We merely need to write a few lines of code with Auth0 to get:

- Single trace is part of a good identity management platform.
- User administration
- Support for developers of social identities (like Facebook, GitHub, Twitter, etc.)
- Identity management services for businesses (Active Directory, LDAP, SAML, etc.)
- Users from our own database

3.9 WEB SOCKET

The WebSocket API is a cutting-edge technology which enables a two-way interactive communication exchange to be established between a user's server and a browser. You may use this API to send messages to a server and obtain event-driven answers instead of polling the service.

WebSocket is a reversible, full-duplex protocol that is used in the same context as HTTP, however it starts with ws:/ or wss:/ instead of HTTP. It is a stateful protocol, which implies that the communication between the client and the server will remain open until any side terminates it (client or server). The link is ended from both ends once either the client or the host closes the communication.

When can a web socket be used:

Real-time web application:

A real-time web application employs a web socket to display data that is continually delivered by the webserver at the user end. WebSocket is quicker and enhances application availability since data is continually pushed/transmitted into the same port that is already available.

For example, in a trading website or bitcoin transaction, the database server continually pushes price movements and dynamic contact to the user end over a WebSocket connection.

Chat application:

Chat apps employ WebSockets to create a link just once for the purpose of exchanging, distributing, and transmitting messages among members. For exchanging messages, as well as one-to-one message transmission, it leverages this very same WebSocket session.

When not to use WebSocket:

If we want any real-time updated or continuous streams of data to be communicated over the network, we may utilise WebSocket. If we want to get old data or get it only once to process it with a program, we may utilise HTTP protocol. Old data that isn't needed very often or retrieved just once may be requested with a simple HTTP request, thus we shouldn't use WebSocket in this case.

Chapter 4

Design Methodology

4.1 Block Diagram

A diagram depicting the general layout of the pieces or components of a complex system or process in schematic form.

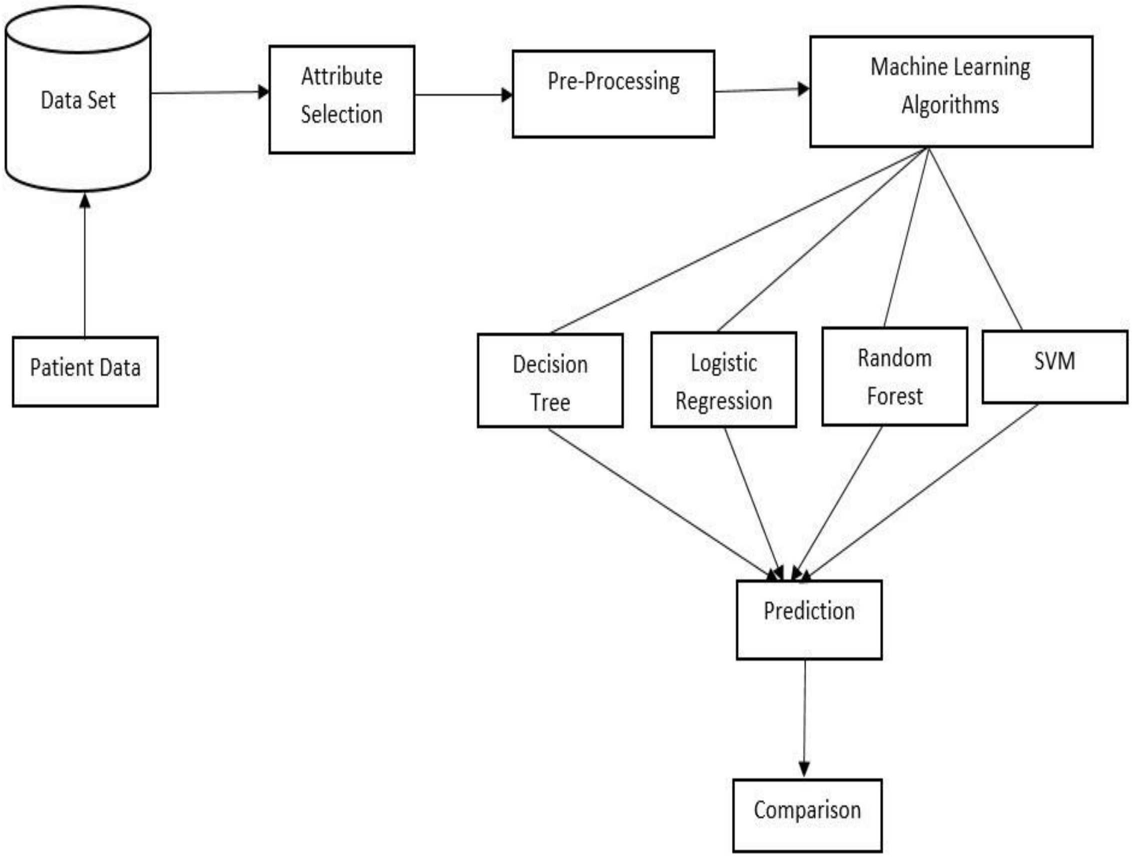


Figure 4.1: Block Diagram

In above figure, The block diagram for cardiovascular disease prediction is explained. Patient dataset is been taken with the following attributes Age ,Gender ,Height ,Weight ,Systolic blood pressure ,Diastolic blood pressure ,Glucose ,Smoke ,Active ,Cardio. From this the required featured attributes is been selected and been undergone pre-processing to remove the outliers for optimization. Machine learning algorithms like decision tree , logistic regression, random forest ,svm are been chosen and trained until the best accuracy acquired . The results for the model are analyzed. Best model from them with the high accuracy is chosen.

4.2 UML Diagrams:

A common language for writing software designs is the Unified Modeling Language. It can be used to visualise, specify, build, and document software-intensive system artefacts. Modelling is a well-known and widely used engineering approach.

4.2.1 Data Flow Diagram:

A Data Flow Diagram is a graphical representation of data flow in an information system that represents the system's process characteristics. A DFD is typically used as a preliminary step to provide a high-level overview of a system without going into great detail, which can then be elaborated on later. A DFD can also be used to visualise data processing and emphasise the sorts of data that will be input to and output from the system, as well as where the data will originate from, go to, and be stored. It doesn't show how long processes will take or if they'll execute in sequence or in parallel. Unlike a traditional structured flowchart, which emphasises control flow, this one emphasises data flow.

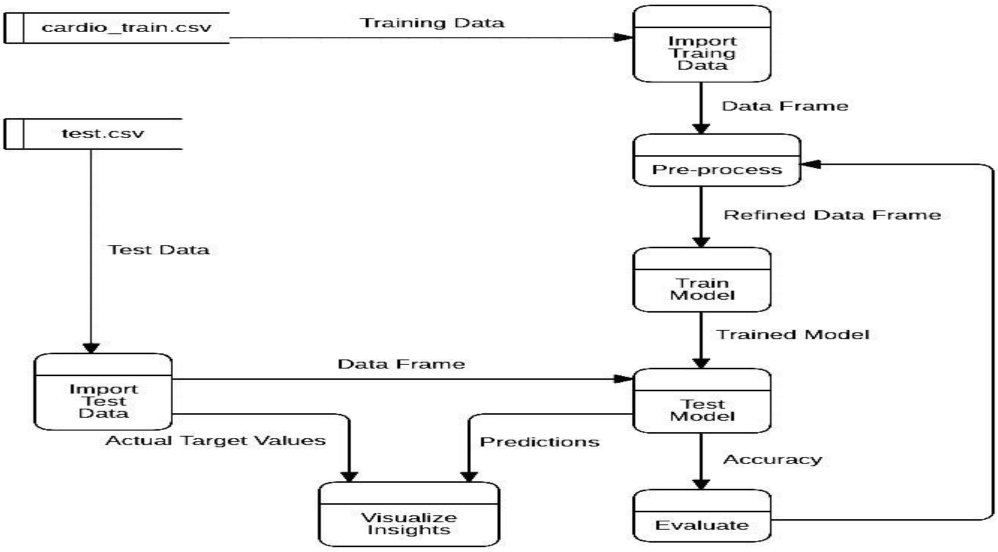


Figure 4.2

The above figure shows the data flow diagram for training and testing the model.

The data collection is pre-processed before being imported as a.csv file. Pre-processing is essential because real-world data contains noise. To make the data more precise, it is cleansed. The model is trained by dividing the data set into 2/3 and 1/3 training and testing datasets, respectively. After successful training, the model is validated across the actual data set by measuring the accuracy of the related model. After that, the predicted values are compared to the actual desired values, with the results displayed.

4.2.2 Sequence Diagram

The time ordering of messages is highlighted in a sequence diagram. A sequence diagram depicts a group of objects and the communications they send and receive. Typically, the objects are named or anonymous instances of classes. Sequence diagrams are used to model a system's dynamic features.

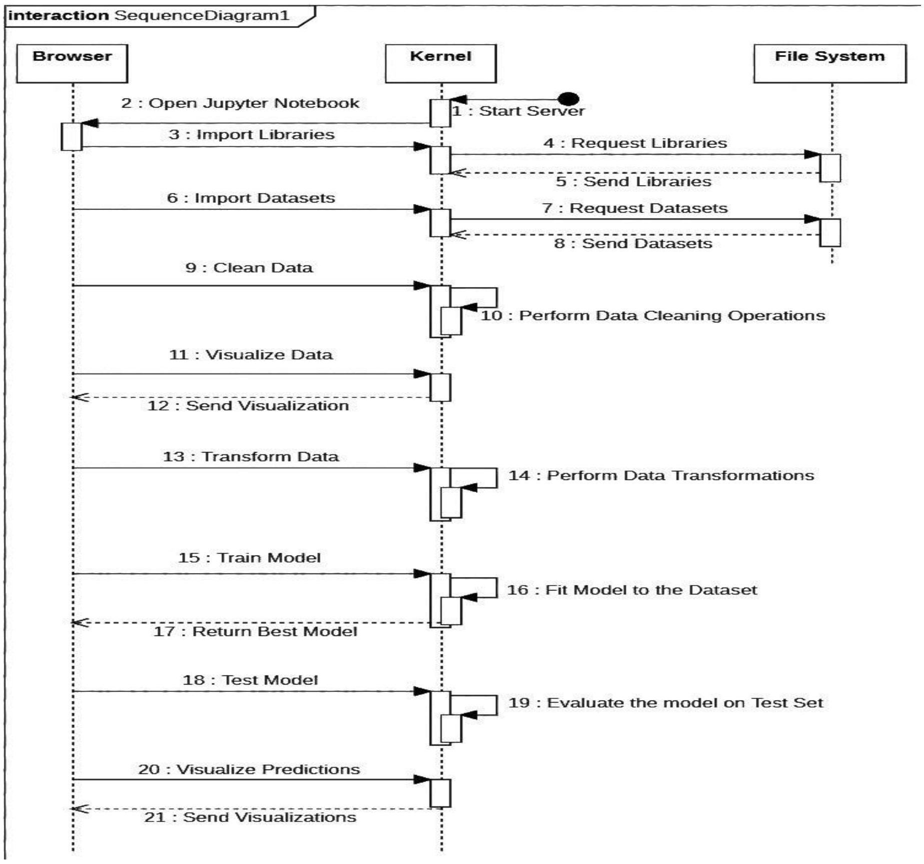


Figure 4.3

The above figure is the sequence Diagram for cardiovascular disease prediction model The steps illustrated in the Figure 4.3 are as follows:

1. The kernel began by launching the server.

2. A command to launch a Jupyter notebook is given to the browser.

3. The browser uses the kernel to request libraries from the file system (Ex: Pandas, Numpy, etc.,)

4. The kernel communicates directly with the file system, requesting the necessary libraries.

5. The file system replies to the kernel's request by retrieving the needed libraries.

6. The browser then asks for the data set that corresponds to the problem statement.

7. The kernel receives the browser's request and forwards it to the file system.

8. The file system fulfils the request by retrieving the needed data.

9. The browser requests that cleaning actions be performed on the data-set.

10. To improve accuracy, several cleaning processes are applied to the data frame.

11. To comprehend the behaviour of the target data, it is visualised.

12. The kernel responds with the types of visualisations that the browser requested.

13. To standardise the attribute values in the data collection, data transformation techniques are used.

14. The Kernel applies many transformation techniques to the data frame.

15. The data frame is used to train a model.

16. To produce a highly accurate model, a tailored model is fitted across the dataset.

17. The test data's associated predictions are shown to assess their accuracy in relation to actual property prices.

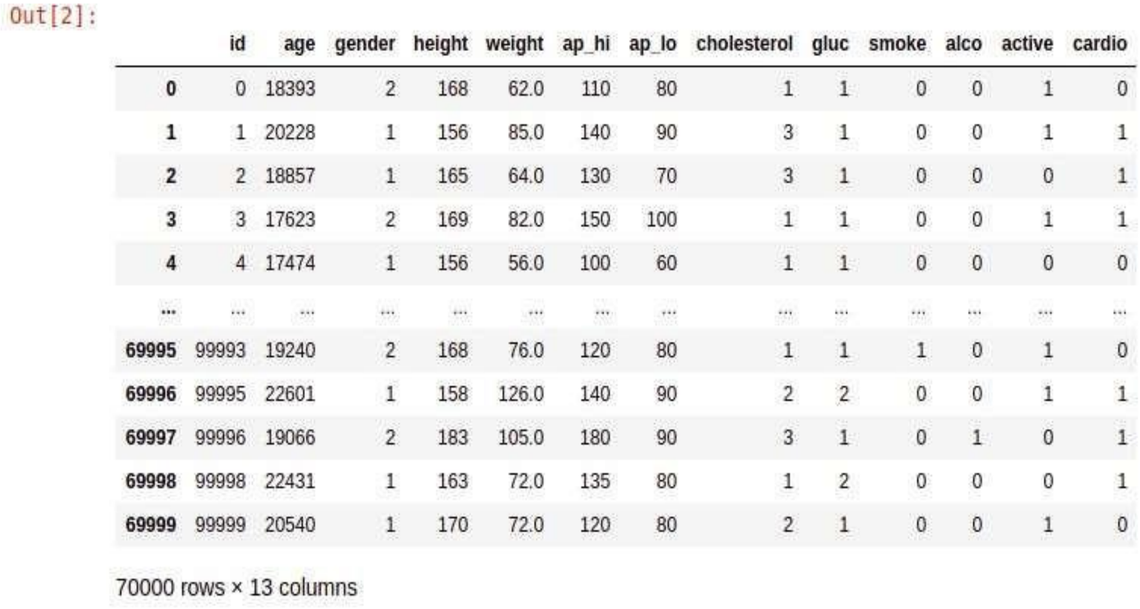
18. Requested visualisations are provided in the browser to help you better comprehend the trained model's accuracy.

4.3 Modules

4.3.1 Data Acquisition

Data acquisition is the process of bringing raw data sets into an analytical platform. It can come from a variety of places, including traditional databases (SQL and query browsers), remote data (web services), text files (scripting languages), No SQL storage (web services, programming interfaces), and other sources. Locating data sets, retrieving data, and querying data from the data set are all part of data acquisition..

The Cardiovascular data-set is extracted from the kaggle platform. It consists of total 45000 samples out of which 50000 are training samples and remaining 20000 are testing samples.



Out[2]:

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
0	0	18393	2	168	62.0	110	80	1	1	0	0	1	0
1	1	20228	1	156	85.0	140	90	3	1	0	0	1	1
2	2	18857	1	165	64.0	130	70	3	1	0	0	0	1
3	3	17623	2	169	82.0	150	100	1	1	0	0	1	1
4	4	17474	1	156	56.0	100	60	1	1	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
69995	99993	19240	2	168	76.0	120	80	1	1	1	0	1	0
69996	99995	22601	1	158	126.0	140	90	2	2	0	0	1	1
69997	99996	19066	2	183	105.0	180	90	3	1	0	1	0	1
69998	99998	22431	1	163	72.0	135	80	1	2	0	0	0	1
69999	99999	20540	1	170	72.0	120	80	2	1	0	0	1	0

70000 rows × 13 columns

Figure 4.4

The above figure defines the Cardiovascular patients Data set.The following are the attributes present in the data set:

Age ,Gender ,Height ,Weight ,Systolic blood pressure ,Diastolic blood pressure ,Glucose ,Smoke Active, Cardio.

4.3.2 Data Exploration

The following exploration tactics are used in the project: finding correlation between variables in the data set to decrease feature attribute redundancy, which reduces training time and improves efficiency.

Outliers are plotted to discover uncommon events and fix errors. The data set contained big outliers, which reduced the predictor's accuracy by 8%. As a result, recognising and eliminating outliers from the sample was crucial for accurate prediction.

4.3.3 Pre-Process Data

Pre-processing of data involves 2 criteria:

Inconsistent values, duplicate entries, missing values, erroneous data, and outliers are all removed during data cleaning.

Scaling, transformation, feature selection, dimensionality reduction, and data manipulation are all strategies used in data wrangling. To prevent some features with big values from dominating the results, scaling is applied across the dataset. The data-noise set's and variability are reduced using the transformation technique. Multiple features were hand-selected to eliminate redundant/irrelevant features and simplify analysis.

4.3.4 Data Analysis Module

The data analysis scenario includes feature selection, model selection, insight production, and result analysis.

When forecasting and predicting future values, a variety of strategies can be applied. When constructing a forecasting model, a variety of different methodologies are typically used and then compared. Forecasting methodologies can be separated into two types, and which one is best depends on the type of data provided. Qualitative forecasting, which is used when historical data is limited, and quantitative forecasting, which is used when numerical information about the past is accessible, are the two approaches. Because all predictions are dependent on past data, the second strategy will be applied in this thesis.This chapter is broken down into sections. The first section will cover regression analysis. The methods utilised in this thesis were designed with the philosophy of starting with a simple method termed multiple linear regression and using it as a baseline for more advanced methods in the field of machine learning. The second section will discuss machine learning approaches with the purpose of better understanding their benefits and how they work..

1) Regression Analysis:

Regression analysis is a well known and normally used statistical approach for analysing quantitative information. Regression analysis is a set of statistical strategies for calculating relationships among variables. Of route, due to the fact it is based on training and trying out information, this technique can be characterized as a device studying method. but, the primary purpose of using those strategies was to look at the statistics and the correlations between variables. when there are a couple of variables in a model, regression is regularly employed to construct a mathematical link among the established and unbiased variables. it can help us realise how the price of the established variable varies whilst one or greater of the impartial variables is modified. The variables are referred to by using a variety of names. The reaction variable or target is commonly called the dependent variable, even as the impartial elements are referred to as predictors or explanatory variables. To maintain matters clean, this thesis will use dependent and independent variables any further. to produce predictions, many regression strategies are used. whilst choosing a strategy, the wide variety of impartial variables, the shape of the regression line, and the kind of goal variables are the maximum vital elements to do not forget.

2) Simple Linear Regression:

simple linear regression is the maximum fundamental form of regression, and it is often the primary approach people examine when constructing a predictive model. A single unbiased variable is used to are expecting the cost of a dependent variable in simple linear regression. there is an assumption that the variables have a linear connection. The motive of the usage of those techniques is to both forecast a destiny fee for the based variable or quantify the strength of the hyperlink among the structured and impartial variables. This strategy become utilised to recognize the connection among the variables on the begin of the thesis.

Mathematically the linear relationship may be defined with the following equation:

($Y = ß\ 0 + ß1\ X + €$) simple linear regression is the maximum fundamental form of regression, and it is often the primary approach people examine when constructing a predictive model. A single unbiased variable is usedto are expecting the cost of a dependent variable in simple linear regression. there is an assumption that the variables have a linear connection. The motive of the usage of those techniques is to both forecast a destiny fee for the based variable or quantify the strength of the hyperlink among the structured and impartial variables. This strategy become utilised to recognize the connection among the variables on the begin of the thesis.

Mathematically the linear relationship may be defined with the following equation:

$$\hat{Y} = \hat{}ß0 + \hat{}ß1\ x$$

In the above equation ^ indicates the predicted value of Y and the estimated value of the coefficient parameters after training the data.

Multiple Linear Regression:

The least squares approach, which is an extension of simple linear regression, is also known as multiple linear regression. Multiple independent variables are typically related with the dependent variable in practise. Because the ß coefficient parameters in each of these models can be written as linear combinations, they are known as linear regression models. Multiple linear regression aims to model the relationship between two or more independent variables (Xi) and a dependent variable by fitting a linear equation to observed data (Yi). Assume there are p independent variables in the model (X1, X2,...., X p), and the equation is:

$$\hat{Y} = ß0 + ß1\ X1 + ß2\ X2 + ... + ßpXp + €$$

The expected value of the dependent variable is in Equation 4.3. X through X are the predictors or independent variables. In both basic and multiple linear regression, the least square technique is employed to estimate the coefficients. The least squares approach reduces the overall vertical distance between the observed and expected values by a factor of two.

Machine Learning Methods:

Arthur Lee Samuel created the time period gadget learning (ML) in 1959. He turned into a pioneer within the fields of computer games and synthetic intelligence within the united states, and he advanced the world's first self-learning programme. He claimed in his look at [1] that programmed computers can play checkers higher than the individual that designed the software. The word "device getting to know" refers back to the automated detection of styles in facts [5], that's strongly reliant on statistics availability and processing capability. gadget learning algorithms fluctuate from previous statistical strategies in that they do not require earlier assumptions, and they will analyze from facts without being explicitly undertaking programmed to do so. Splitting facts into schooling and checking out sets is one of the concepts of gadget mastering methods. The schooling set is used to excellent-music the version's parameters, whereas the trying out set is used to estimate model performance. There are varieties of system mastering algorithms: supervised and unsupervised. all of the records in supervised learning is labelled, and the output prediction is learned from the input records. The output label does no longer exist in unsupervised learning, which means that the process is only dependent on input variables with out a corresponding output values. due to the fact the output is to be had on this assignment, all applied system getting to know algorithms are supervised. In latest years, the application of device learning algorithms inside the discipline of hydrology has accelerated, as it's far a beneficial device for forecasting and prediction. Random forest, neural community, and guide vector device were selected because the 3 ML mastering algorithms for use on this thesis.

1) Random Forest:

Random forests are one of the most often utilised supervised machine learning approaches (RF). One of the algorithm's main benefits is that it can do classification and regression tasks as well as make predictions. The algorithm is an ensemble approach that creates combinations by combining many decision trees. The goal of the ensemble learning strategy is to mix several methods in order to improve model performance. Tin Kam Ho was the first to propose the method, which used the random subspace method to pick a sample of features at random from the entire collection of data. The algorithm can be simply described in the four following steps:

Assume that the training set contains N instances. Then, with replacement, a random sample of N instances is selected. If there are M input variables, a number m M is provided to choose m variables at random at each node. The best split on these factors is used to split the node. The value of m remains constant as the forest expands. .All trees are allowed to reach their full potential without being pruned.Combine the n trees' predictions to forecast fresh data.

The "Random Forest" R package was used to implement the approach.

The "Random Forest" R package was used to implement the approach.

On the other hand, single tree models might be unstable and overly reliant on specific training data. To overcome this problem, ensemble techniques create a collection of models and combine their predictions to get a data point's class designation. A random forest model is made up of a collection of classification (or regression) trees. Individual trees in random forests derive variance from two sources: first, each tree is created on independent bootstrapped samples of the training data, and second, only a randomly selected subset of data characteristics is examined at each node in generating the individual trees.

2) Logistic Regression:

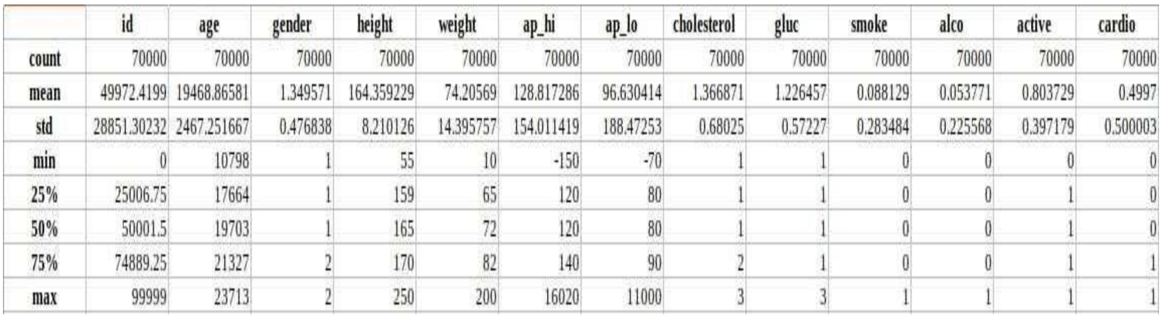
The logistic model is used to forecast the possibility of a specific class or event occurring, such as pass/fail, win/lose, alive/dead, or healthy/sick. In its most basic form, logistic regression employs a logistic function to model a binary dependent variable, while there are many more advanced variations. In regression analysis, logistic regression (or logit regression) is a technique for estimating the parameters of a logistic model (a form of binary regression). In a binary logistic model, a dependent variable has two alternative values, such as pass/fail, which is represented by an indicator variable, and the two values are labelled "0" and "1" in mathematics.

The logistic model's log-chances (logarithm of the odds) for the value labelled "1" are a linear combination of one or more independent variables ("predictors"); the independent variables can be binary variables (two classes, each coded by an indicator variable) or continuous variables (any real value). The related likelihood of the value labelled "1" may vary between 0 (certainly the value "O") and 1 (definitely the value "1"), therefore the labelling; the logistic function transforms logodds to probability, hence the name.

In logistic regression, sigmoid function is used . The equation for sigmoid function is as follows S(x) =1/(1+e-(ß0+ß1x)).In equation above here, 0+1x is comparable to the linear model y = axe + b where x is the input predictor variables. S(x) is a sigmoid function and e is the Euler Number.

4.3.5 Data Visualisation

The data in the data-set is as follows:



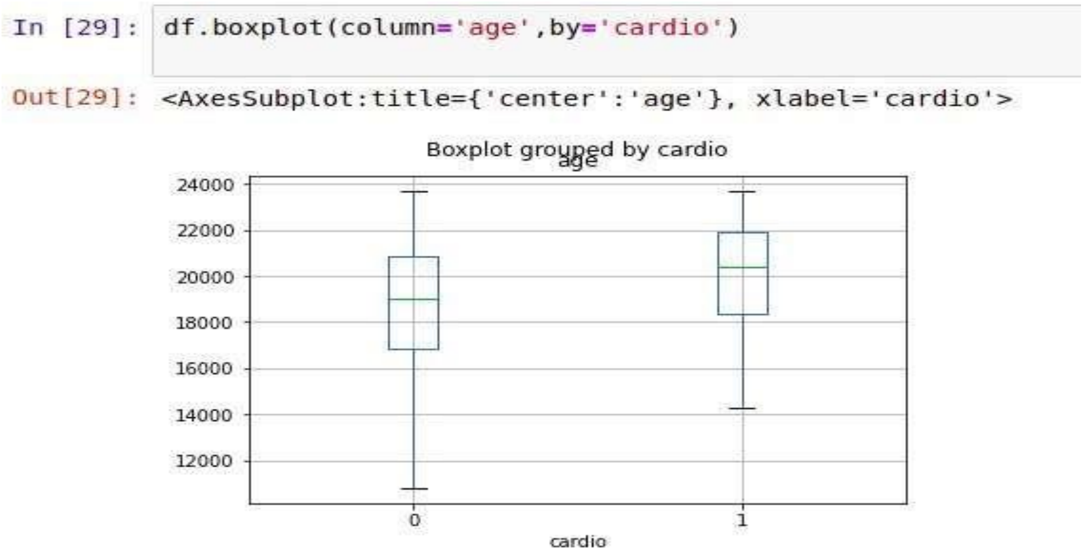
	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
count	70000	70000	70000	70000	70000	70000	70000	70000	70000	70000	70000	70000	70000
mean	49972.4199	19468.86581	1.349571	164.359229	74.20569	128.817286	96.630414	1.366871	1.226457	0.088129	0.053771	0.803729	0.4997
std	28851.30232	2467.251667	0.476838	8.210126	14.395757	154.011419	188.47253	0.68025	0.57227	0.283484	0.225568	0.397179	0.500003
min	0	10798	1	55	10	-150	-70	1	1	0	0	0	0
25%	25006.75	17664	1	159	65	120	80	1	1	0	0	1	0
50%	50001.5	19703	1	165	72	120	80	1	1	0	0	1	0
75%	74889.25	21327	2	170	82	140	90	2	1	0	0	1	1
max	99999	23713	2	250	200	16020	11000	3	3	1	1	1	1

Figure 4.5

The above figure, the description of the data is shown which shows the aggregate values for the dataset.

Box Plot:

A box plot is a way for graphically showing groups of numerical data through their quartiles in descriptive statistics. The terms box-and-whisker plot and box-and-whisker diagram refer to box.plots with vertical lines (whiskers) indicating variability beyond the top and lower quartiles .Individual points can be plotted to represent outliers.

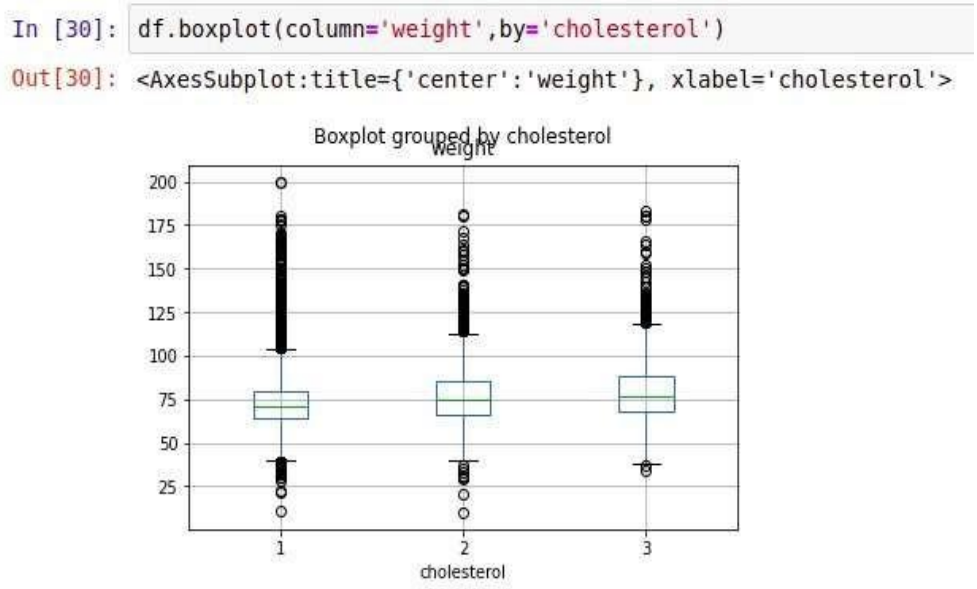


```
In [29]: df.boxplot(column='age',by='cardio')

Out[29]: <AxesSubplot:title={'center':'age'}, xlabel='cardio'>
```

Figure 4.6

In the above figure ,the box plot against age and cardio , weight and cholesterol . The outliers can be visualized.



```
In [30]: df.boxplot(column='weight',by='cholesterol')
Out[30]: <AxesSubplot:title={'center':'weight'}, xlabel='cholesterol'>
```

Figure 4.7

In the above figure, the box plot against age and cardio , weight and cholesterol . The outliers can be visualized.

Chapter 5

Testing and Results

Training is performed for the chosen part of the data using regression methods and the accuracy is observed.

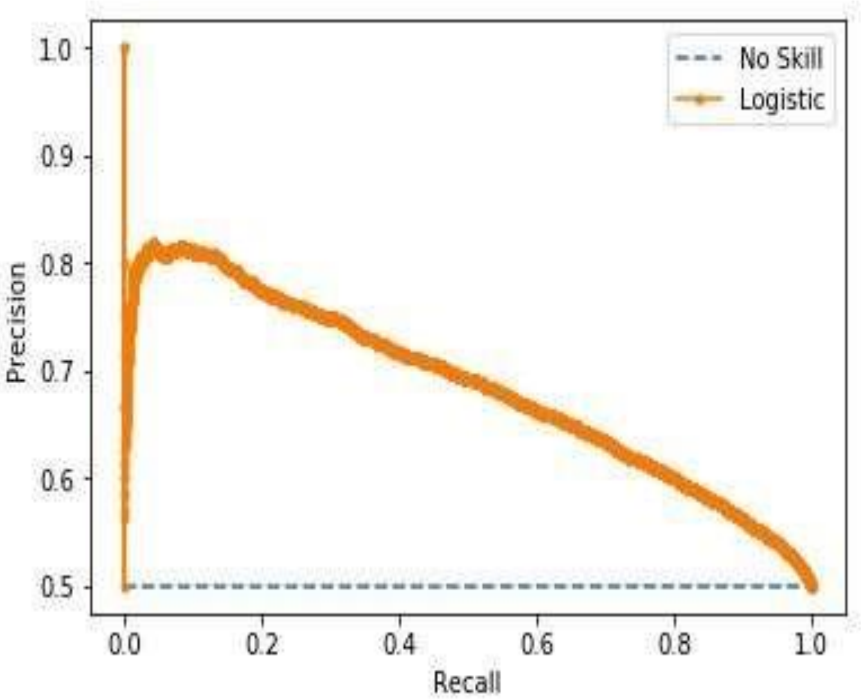


Figure 5.1

The above figure is the precision-recall curve and f1 of Logistic Regression for prediction model

In figure above figure, Training is done using logistic regression, it gives an accuracy of around 64.69% Linear regression is also used to train the data set ,it gives an accuracy of around 11.57%.

Training is performed for chosen part of the data using SVM , Decision Tree and the accuracy is observed.

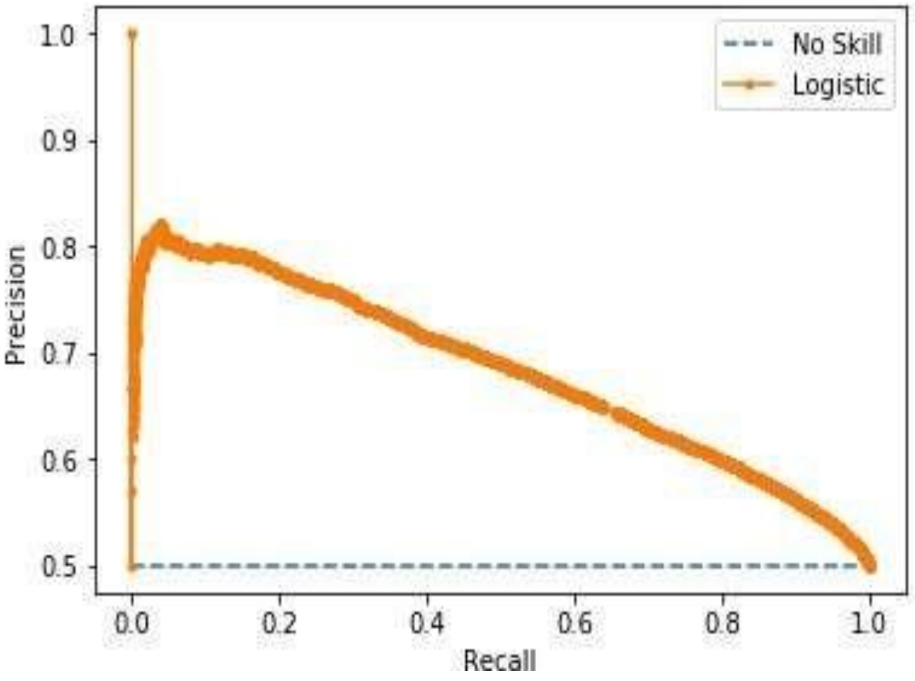


Figure 5.2

The above figure is the Precision-recall curve and f1 of SVM for cardiovascular disease prediction

In figure 5.2, Training is done using SVM , it produced an accuracy of 64.63%.

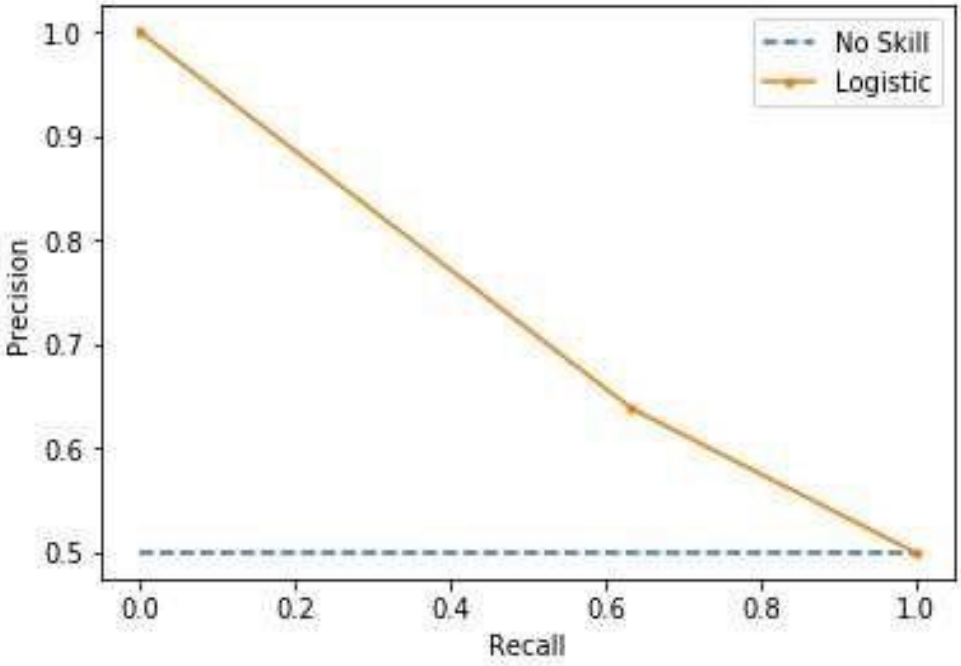
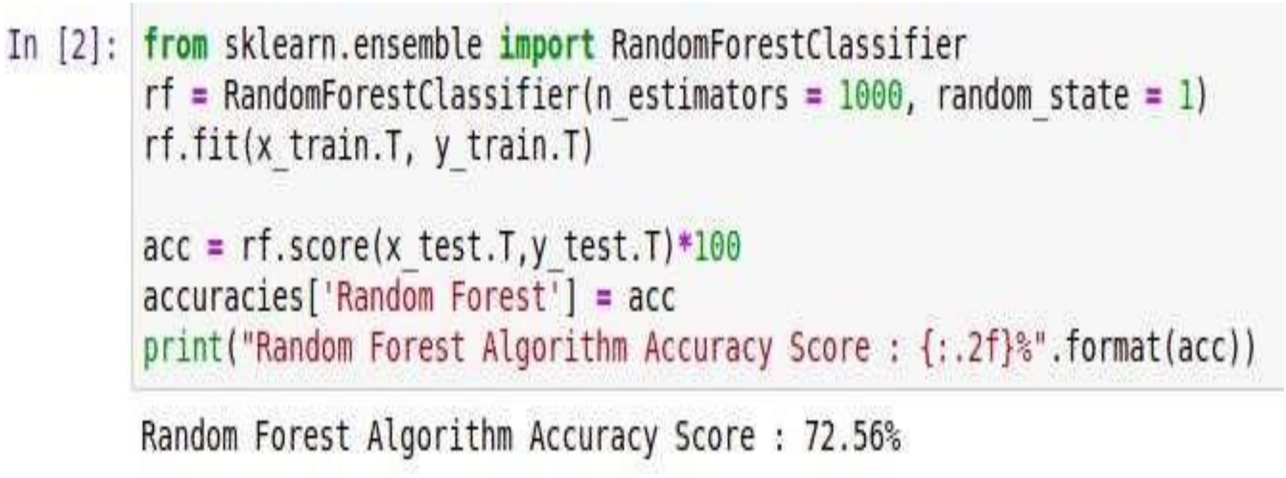


Figure 5.3

The above figure is Precision-recall curve and f1 for Decision tree of cardiovascular disease predictionIn figure 5.3, training is done using decision tree, it gives an accuracy of 63.62%.

A 72.56% accuracy is obtained on the dataset with the feature attributes.



```
In [2]: from sklearn.ensemble import RandomForestClassifier
        rf = RandomForestClassifier(n_estimators = 1000, random_state = 1)
        rf.fit(x_train.T, y_train.T)

        acc = rf.score(x_test.T,y_test.T)*100
        accuracies['Random Forest'] = acc
        print("Random Forest Algorithm Accuracy Score : {:.2f}%".format(acc))

        Random Forest Algorithm Accuracy Score : 72.56%
```

Figure 5.4

The above figure is the Accuracy score for random forest .In figure 5.4, The accuracy score for Random Forest Model is displayed.

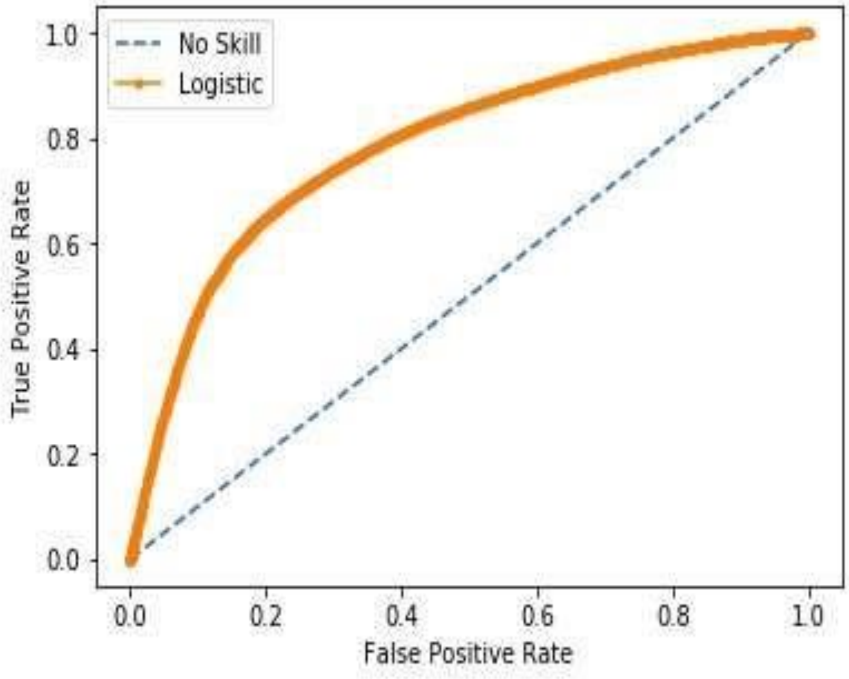


Figure 5.5

Figure 5.5 is the Precision-recall curve and f1 for Random Forest Model of Cardiovascular Disease. In figure 5.5, The Precision-recall curve and f1 for random forest model is shown.

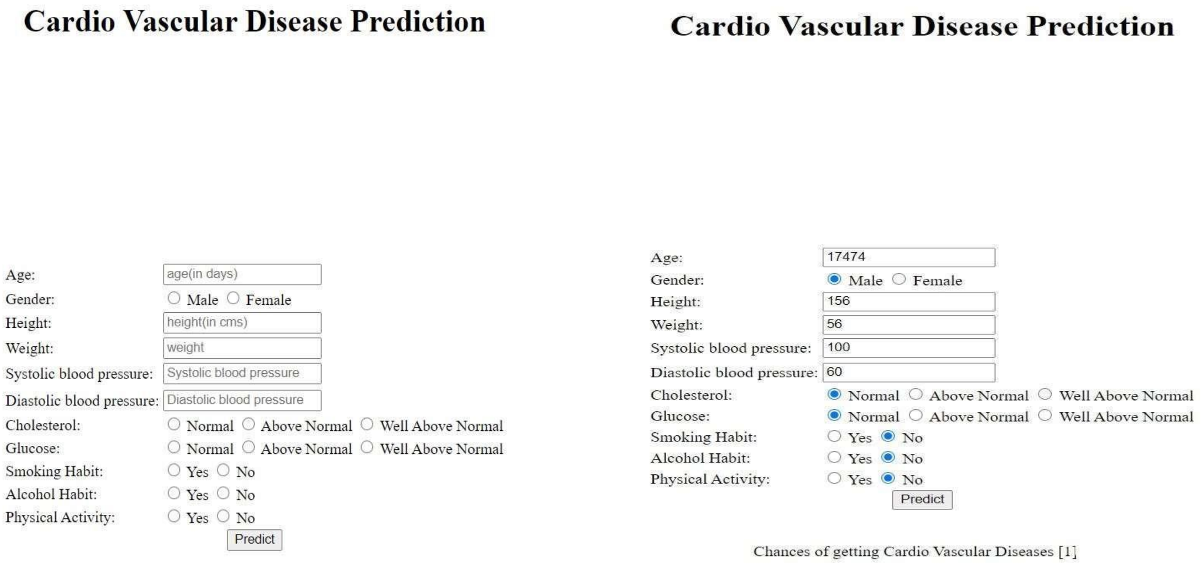
The accuracy for the models are given below :

Linear Regression Technique	11.57%
Logistic Regression Technique	64.78%
SVM Technique	64.34%
Decision Regression Technique	63.62%
Random Forest Technique	72.56%

Observation:

The accuracy of linear regression is substantially lower than that of the Random Forest Regressor, although the accuracy of logistic regression, decision trees, and svm is extremely close but still lower than that of the Random Forest Model.

The Prediction Result is as follows:



Cardio Vascular Disease Prediction

Age: age(in days)
Gender: ○ Male ○ Female
Height: height(in cms)
Weight: weight
Systolic blood pressure: Systolic blood pressure
Diastolic blood pressure: Diastolic blood pressure
Cholesterol: ○ Normal ○ Above Normal ○ Well Above Normal
Glucose: ○ Normal ○ Above Normal ○ Well Above Normal
Smoking Habit: ○ Yes ○ No
Alcohol Habit: ○ Yes ○ No
Physical Activity: ○ Yes ○ No
Predict

Cardio Vascular Disease Prediction

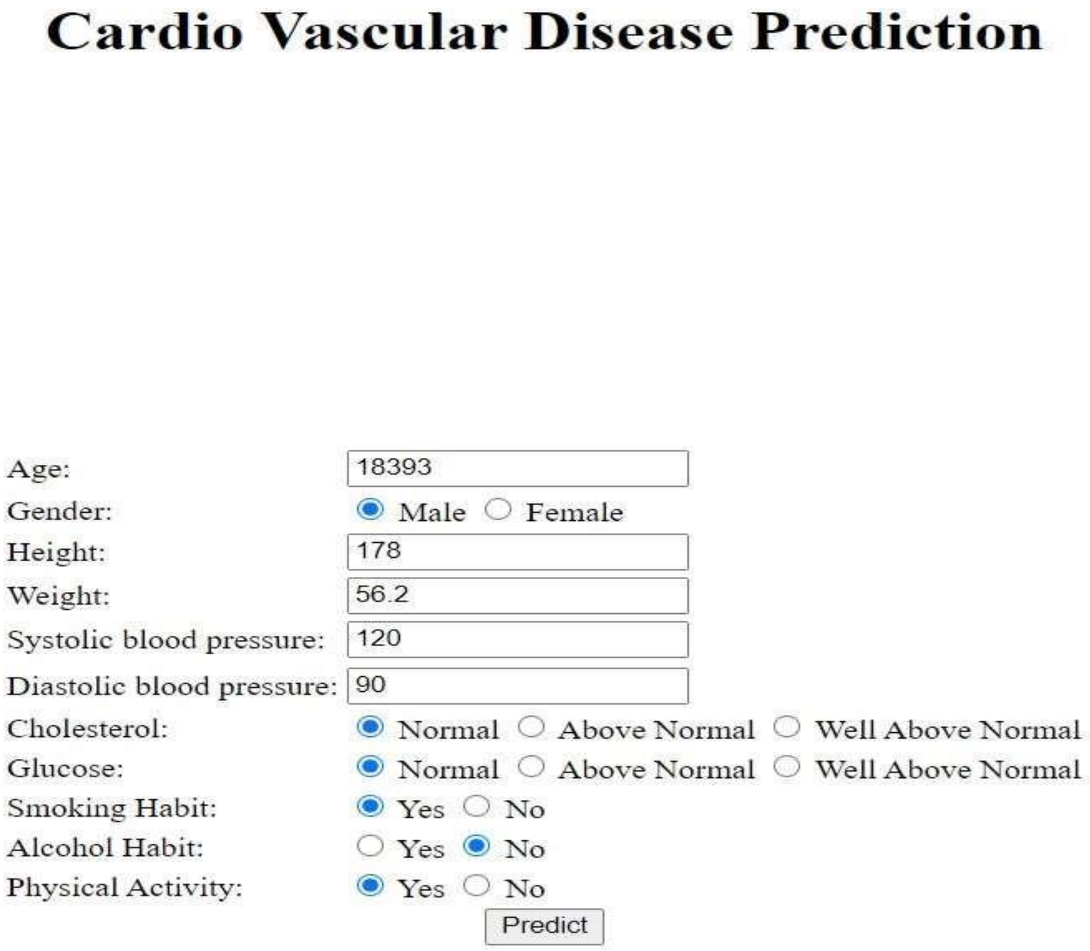
Age: 17474
Gender: ● Male ○ Female
Height: 156
Weight: 56
Systolic blood pressure: 100
Diastolic blood pressure: 60
Cholesterol: ● Normal ○ Above Normal ○ Well Above Normal
Glucose: ● Normal ○ Above Normal ○ Well Above Normal
Smoking Habit: ○ Yes ● No
Alcohol Habit: ○ Yes ● No
Physical Activity: ○ Yes ● No
Predict

Chances of getting Cardio Vascular Diseases [1]

Figure 5.6

The above figure is the Prediction Interface for the test sample.In the Figure 5.6, The prediction interface the test sample is shown , where the the data is to be entered by the user and the output is shown as the chance of getting cardio vascular disease is (1/0).

5.1 Test Case 1

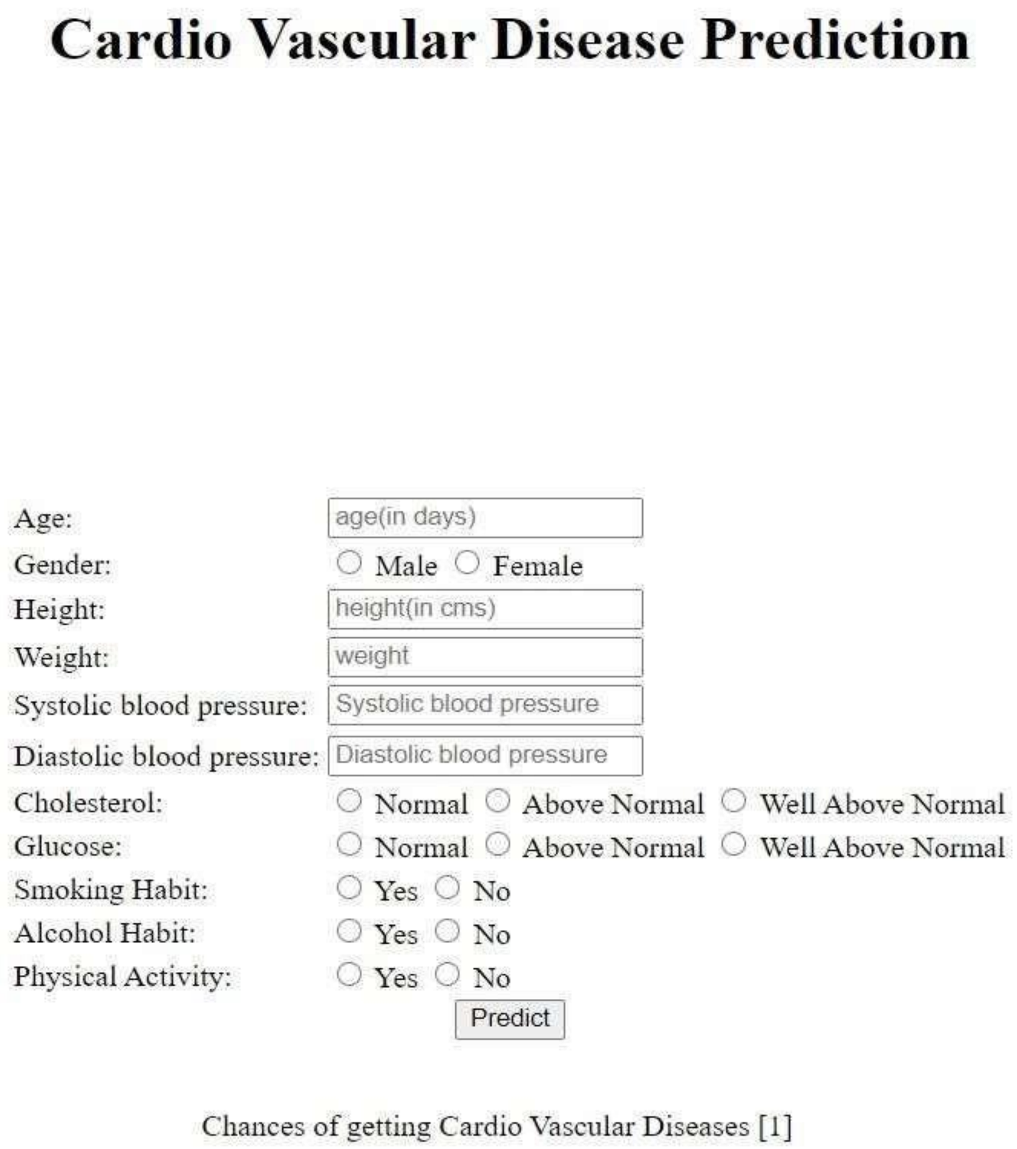


Cardio Vascular Disease Prediction

Age:	18393
Gender:	● Male ○ Female
Height:	178
Weight:	56.2
Systolic blood pressure:	120
Diastolic blood pressure:	90
Cholesterol:	● Normal ○ Above Normal ○ Well Above Normal
Glucose:	● Normal ○ Above Normal ○ Well Above Normal
Smoking Habit:	● Yes ○ No
Alcohol Habit:	○ Yes ● No
Physical Activity:	● Yes ○ No

Figure 5.7

The above figure is the front end of the web page which is designed using flask api.In the figure 5.7 required fields such as age in days need to be entered as shown above.Height should be given in centimetres and weight should be given in kilo grams and so on.



Cardio Vascular Disease Prediction

Age:	age(in days)
Gender:	○ Male ○ Female
Height:	height(in cms)
Weight:	weight
Systolic blood pressure:	Systolic blood pressure
Diastolic blood pressure:	Diastolic blood pressure
Cholesterol:	○ Normal ○ Above Normal ○ Well Above Normal
Glucose:	○ Normal ○ Above Normal ○ Well Above Normal
Smoking Habit:	○ Yes ○ No
Alcohol Habit:	○ Yes ○ No
Physical Activity:	○ Yes ○ No

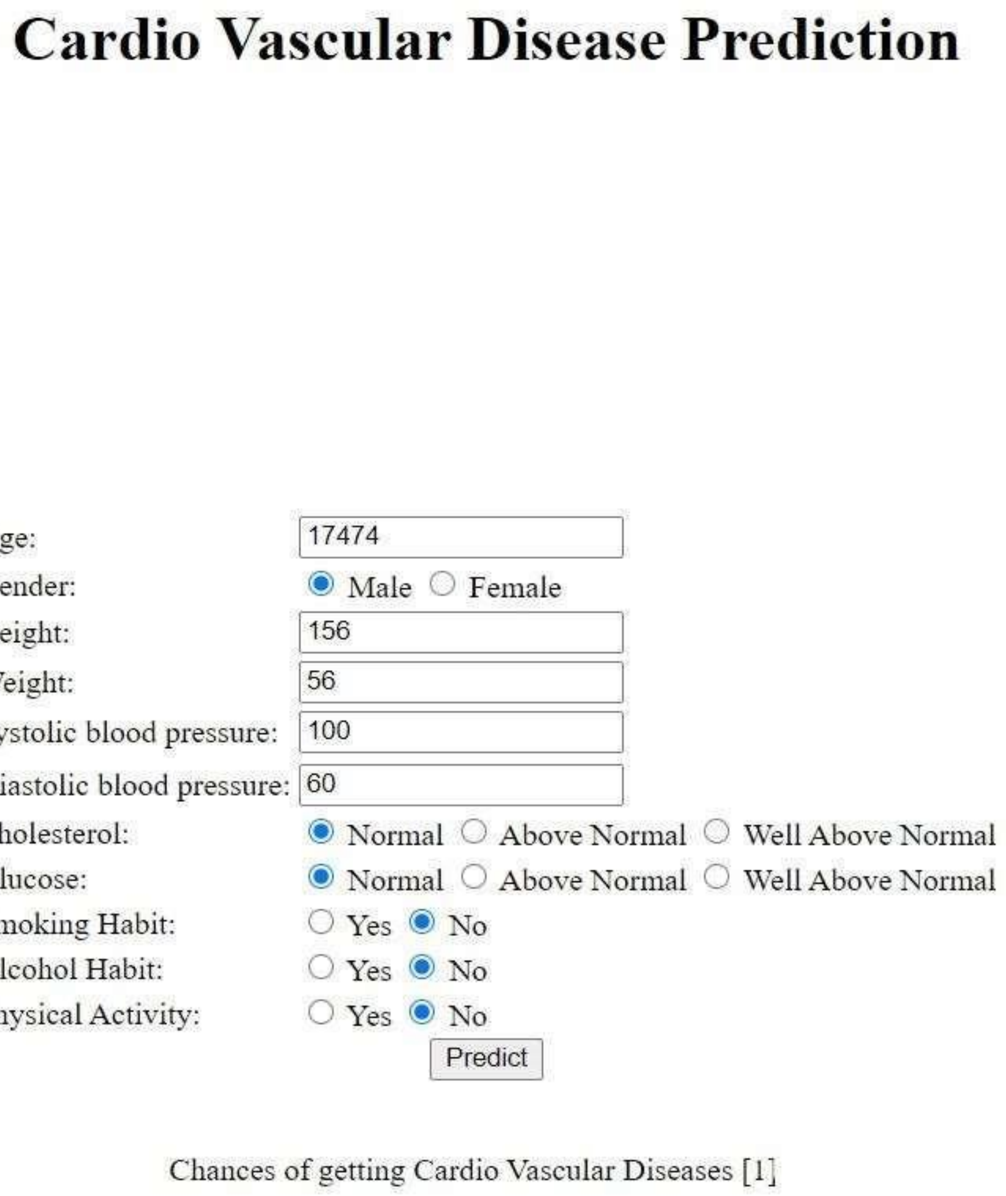
Chances of getting Cardio Vascular Diseases [1]

Figure 5.8

The above figure is the front end of the web page with output for test case 1.

Result of the test case 1 is given as 1. If the test case results give 1 then the person who filled the details is likely to have a heart disease. When the result is shown in the interface ,values of the attributes are cleared as shown above.

5.2 Test Case 2



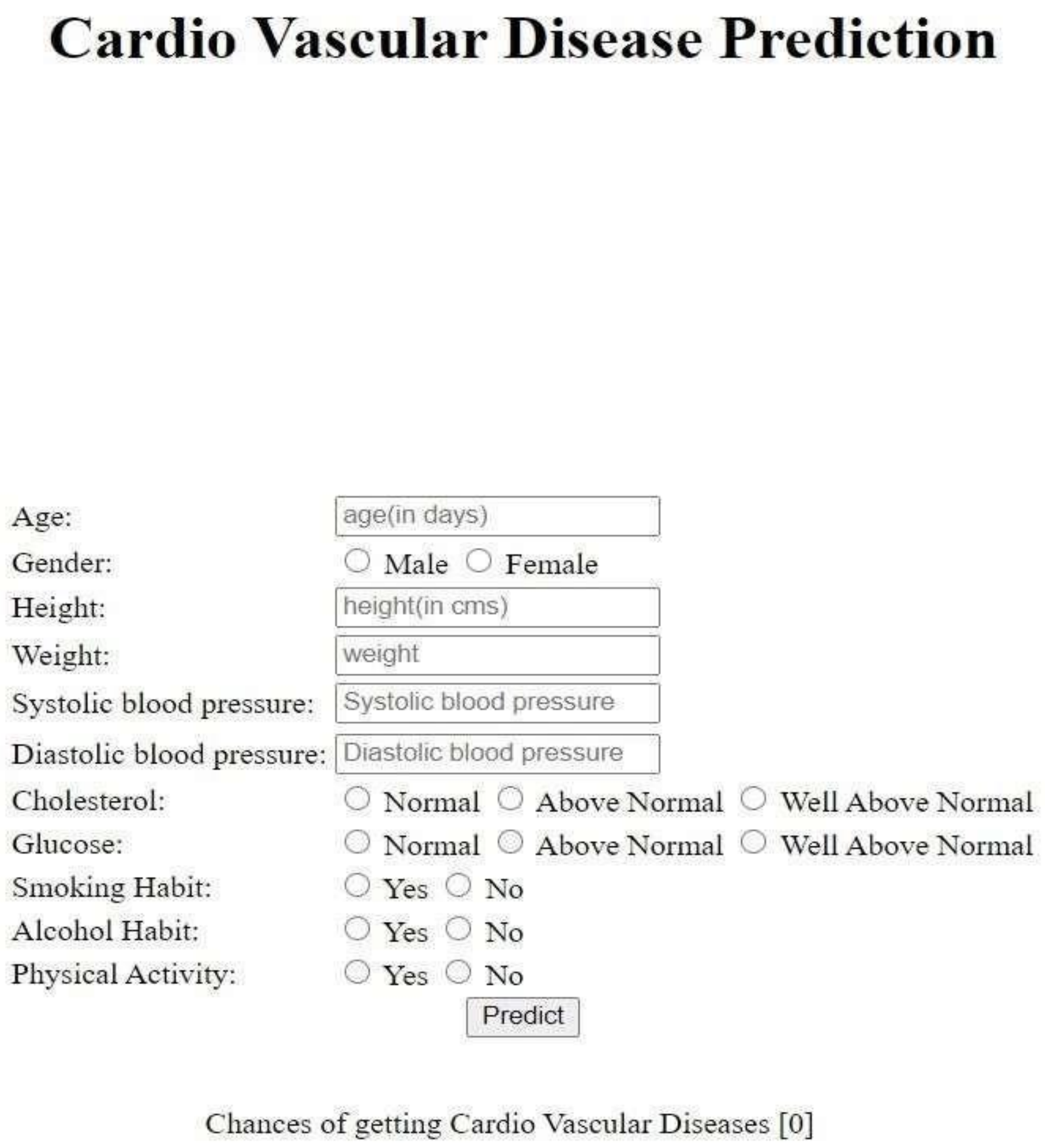
Cardio Vascular Disease Prediction

Age:	17474
Gender:	● Male ○ Female
Height:	156
Weight:	56
Systolic blood pressure:	100
Diastolic blood pressure:	60
Cholesterol:	● Normal ○ Above Normal ○ Well Above Normal
Glucose:	● Normal ○ Above Normal ○ Well Above Normal
Smoking Habit:	○ Yes ● No
Alcohol Habit:	○ Yes ● No
Physical Activity:	○ Yes ● No

Chances of getting Cardio Vascular Diseases [1]

Figure 5.9

The above figure is the front end of the web page which is designed using flask api.In the figure 5.9, required fields such as age in days need to be entered as shown above. Height should be given in centimetres and weight should be given in kilo grams.



Cardio Vascular Disease Prediction

Age:	age(in days)
Gender:	○ Male ○ Female
Height:	height(in cms)
Weight:	weight
Systolic blood pressure:	Systolic blood pressure
Diastolic blood pressure:	Diastolic blood pressure
Cholesterol:	○ Normal ○ Above Normal ○ Well Above Normal
Glucose:	○ Normal ○ Above Normal ○ Well Above Normal
Smoking Habit:	○ Yes ○ No
Alcohol Habit:	○ Yes ○ No
Physical Activity:	○ Yes ○ No

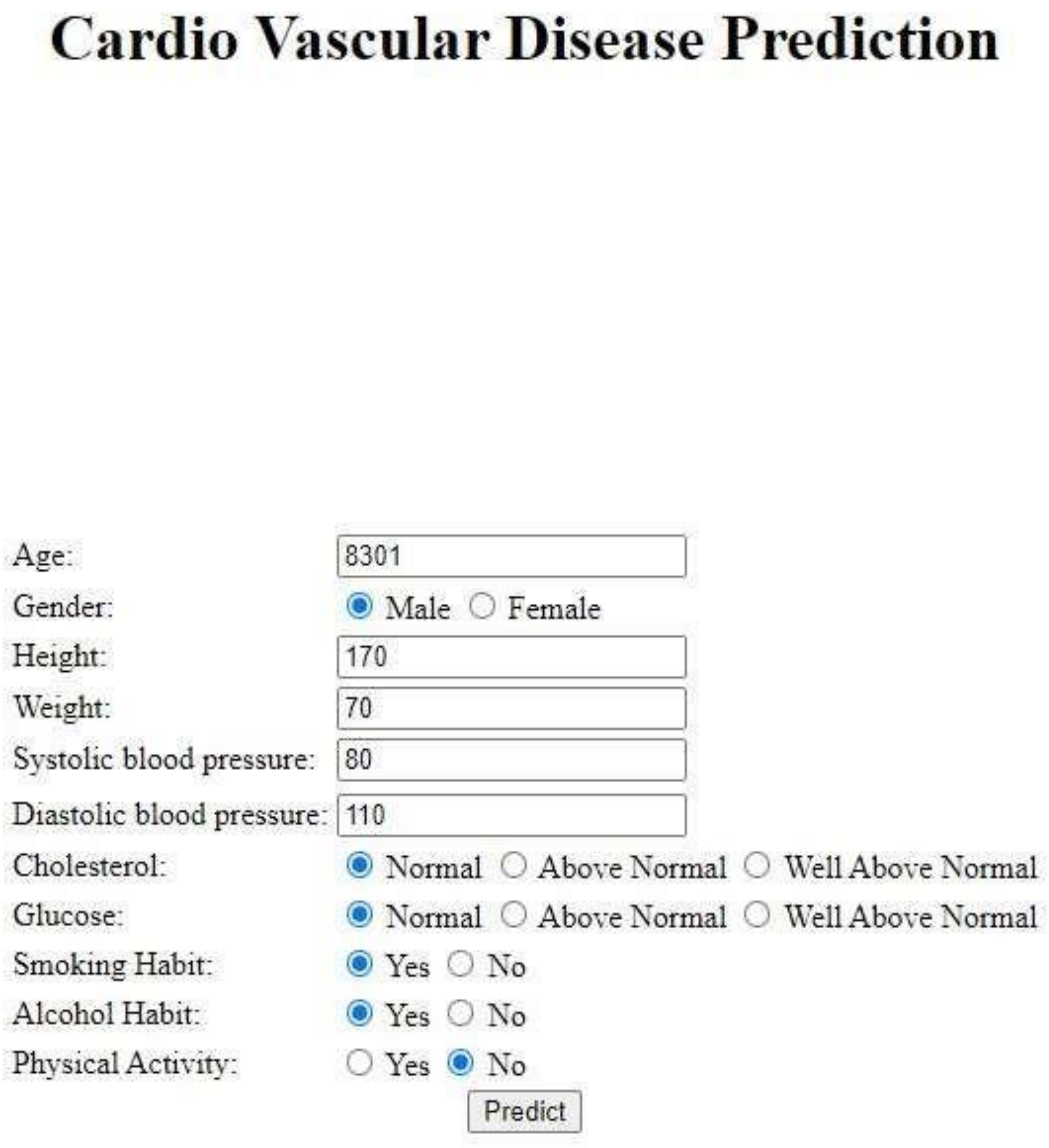
Chances of getting Cardio Vascular Diseases [0]

Figure 5.10

The above figure is the front end of the web page with output for test case 2.

Result of the test case 2 is given as 0. If the test case results give 0 then the person who filled the details is not likely to have a heart disease. When the result is shown in the interface ,values of the attributes are cleared as shown above.

5.3 Test Case 3

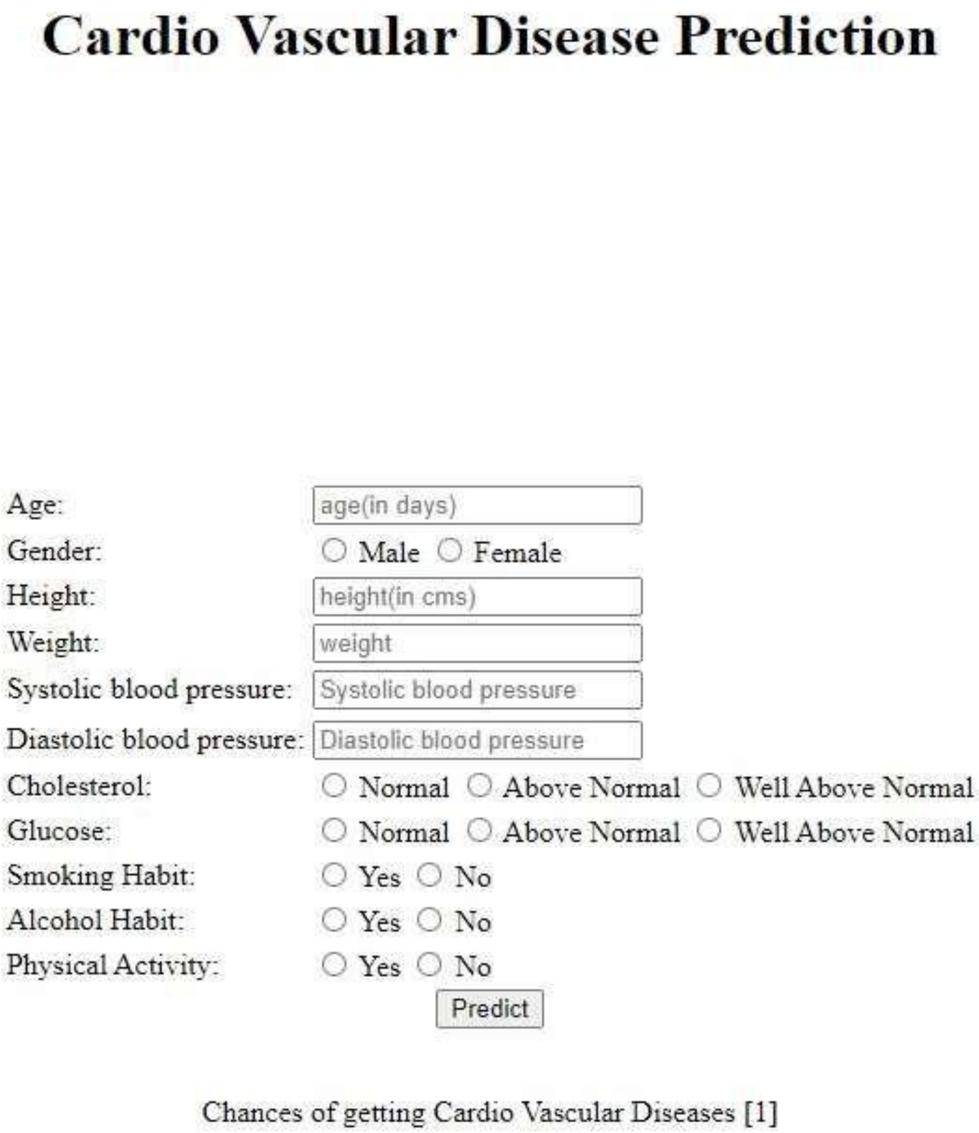


Cardio Vascular Disease Prediction

Age:	8301
Gender:	● Male ○ Female
Height:	170
Weight:	70
Systolic blood pressure:	80
Diastolic blood pressure:	110
Cholesterol:	● Normal ○ Above Normal ○ Well Above Normal
Glucose:	● Normal ○ Above Normal ○ Well Above Normal
Smoking Habit:	● Yes ○ No
Alcohol Habit:	● Yes ○ No
Physical Activity:	○ Yes ● No

Figure 5.11

The above figure is the front end of the web page which is designed using flask api.In the above figure, required fields such as age in days need to be entered as shown above. Height should be given in centimetres and weight should be given in kilo grams.



Cardio Vascular Disease Prediction

Age:	age(in days)
Gender:	○ Male ○ Female
Height:	height(in cms)
Weight:	weight
Systolic blood pressure:	Systolic blood pressure
Diastolic blood pressure:	Diastolic blood pressure
Cholesterol:	○ Normal ○ Above Normal ○ Well Above Normal
Glucose:	○ Normal ○ Above Normal ○ Well Above Normal
Smoking Habit:	○ Yes ○ No
Alcohol Habit:	○ Yes ○ No
Physical Activity:	○ Yes ○ No

Chances of getting Cardio Vascular Diseases [1]

Figure 5.12

The above figure is the front end of the web page with output for test case 3.

Result of the test case 3 is given as 1. If the test case results give 1 then the person who filled the details is likely to have a heart disease. When the result is shown in the interface ,values of the attributes are cleared as shown above

Chapter 6

Conclusion and Future Scope

Conclusion:

The prediction of cardiovascular illnesses has been performed successfully. The project's aim has been met, and the issues have been resolved. The Random forest model was successfully created. By eliminating outliers from the data collection, the accuracy is enhanced. In some circumstances, logistic regression, SVM, and Decision Tree are also highly quick and perform well. With an accuracy of 72.56 percent, the trained model can now predict cardiovascular illness.

The accuracy of the model is affected by changes in the data set. As a result, the model must be updated at regular intervals with fresh records. The model will get more accurate as more data is entered into the system.

Future Scope:

1). Future work of this model can be done to improve the accuracy. There is need of more complex techniques to obtain better accuracy. Adding of other feature attribute might improve this model.

2). This could be used in the application for android and ios with user-friendly interface for early diagnosis of cardiovascular disease**.**

References

[1]. Chen, A. H., Huang, S. Y., Hong, P. S., Cheng, C. H., & Lin, E. J. (2011, September). HDPS: “Heart disease prediction system”. In 2011 Computing in Cardiology (pp. 557-560). IEEE.

[2]. Shetty, Deeraj, Kishor Rit, Sohail Shaikh, and Nikita Patil. "Diabetes disease prediction using data mining."In 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), pp. 1-5. IEEE, 2017.

[3]. Rajesh , T Maneesha, Shaik Hafeez, Hari Krishna“Prediction of Heart Disease Using Machine Learning Algorithms“May 2018International Journal of Engineering & Technology 7(2):363-366DOI: 10.14419/ijet. v7i2.32.15714 North-Holland/American Elsevier) p517

[4]. Krishnan Santana; S. Geetha “Prediction of Heart Disease Using Machine Learning Algorithms”. 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)Publisher: IEEE

[5]. RajdhanApurb, Agarwal Avi, Sai Milan, Ravi Dundigalla, Ghuli Poonam.” Heart Disease Prediction using Machine Learning” INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY

[6]. Singh, A., & Kumar, R. (2020). “Heart Disease Prediction Using Machine Learning Algorithms”. 2020 International Conference on Electrical and Electronics Engineering (ICE3).

[7]. ‘Multi Disease Prediction Model by using Machine Learning and Flask API'‘AkkemYaganteeswaruduInfoshare Systems', Pyramid Softsol Pvt Limited.

[8]. Rairikar, A., Kulkarni, V., Sabale, V., Kale, H., &Lamgunde, A. (2017, June). “Heart disease prediction using data mining techniques”. In 2017 International Conference on Intelligent Computing and Control(I2C2) (pp. 1-8). IEEE

[9]. Benjamin Fredrick David and S. Antony Belcy(October 2018) . "Data Miningand Warehousing." American Society for EngineeringEducation (ASEE Zone 1) Journal (2014).

[10]. National Health Council, ‘Heart HealthScreenings’,2017. [Online]Available: http://www.heart.org/HEARTORG/Conditions/HeartHealthScreenings_UCM_428687_Article.jsp.

[11]. Jaymin Patel, Prof.Tejal Upadhyay, Dr. Samir Patel (2016). Heart Disease Prediction Using Machine Learning and Data Mining Technique. In International Journal of Computer Science & Communication (IJCSC), Volume-7, Number 1 Sept 2015-March 2016 Page No.129 – 137. Shi, D.-P.,Wu, C. “The influence of infrared temperature measurementbased on reflection temperature compensation and incidenttemperature compensation” Electron. Measure. Technol. 08, 2321–2326 (2015)

[12]. Shamsher Bahadur Patel, Pramod Kumar Yadav, Dr. D. P.Shukla (2013). Predict the Diagnosis of Heart Disease Patients Using Classification Mining Techniques. In IOSR Journal of Agriculture and Veterinary Science (IOSR-JAVS) E - ISSN: 2319-2380, p-ISSN: 2319-2372. Volume 4, Page No. 61-64.

[13]. SanchayitaDhar, Krishna Roy, Tanusree Dey, Pritha Datta, Ankur Biswas(2018).A Hybrid Machine European Journal of Molecular & Clinical Medicine ISSN 2515-8260 Volume 7, Issue 4, 2020 2459 Learning Approach for Prediction of Heart Diseases. In 2018 4th International Conference on Computing Communication and Automation (ICCCA).

[14]. Anjan Nikhil Repaka, Sai Deepak Ravikanti, Ramya G Franklin(2019). Design And Implementing Heart Disease Prediction Using Naives Bayesian. In Proceedings of the Third International Conference on Trends in Electronics and Informatics (ICOEI 2019) IEEE Xplore Part Number: CFP19J32-ART; ISBN: 978-1-5386-9439-8.

[15]. Shanmugasundaram G, Malar Selvam V, R. Saravanan(2018). An Investigation of Heart Disease Prediction Techniques. In Conference at SMVEC.

[16]. Aditi Gavhane, GouthamiKokkula, Isha Pandya, Kailas Devadkar(2018). Prediction of Heart Disease Using Machine Learning. In Proceedings of the 2nd International conference on Electronics, Communication and Aerospace Technology (ICECA 2018).

[17]. RahmaAtallah, Amjed Al-Mousa(2019). Heart Disease Detection Using Machine Learning Majority Voting Ensemble Method.InConferenece at Princess Sumaya University for Technology.

[18]. Nidhi Bhatla , Kiran Jyoti(2012). An Analysis of Heart Disease Prediction using Different Data Mining Techniques. In International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 8, October - 2012 ISSN: 2278-0181.

[19]. AyantanDandapath, M Karthik Raja(2018).Heart disease prediction using machine learning techniques a survey. In International Journal of Engineering & Technology IJET.

[20]. AkshayJayraj Suvarna, Arvind Kumar M, Ajay Billav, Muthamma K M, GadugSudhamsu(2019). Predicting The Presence of Heart Disease Using Machine Learning.In International Journal of Computer Science and Mobile Computing, IJCSMC, Vol. 8, Issue. 5, May 2019, pg.119 – 125.

[21]. Abhijeet Jagtap, Priya Malewadkar, Omkar Baswat3,Harshali Rambade4(2019). Heart Disease Prediction using Machine Learning.In International Journal of Research in Engineering, Science and Management Volume-2, Issue-2, February-2019).

[22]. Gnaneswar B, EbenezarJebarani M.R.(2017). A Review on Prediction and Diagnosis of Heart Failure. In 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS).

[23]. Md. Jamil-Ur Rahman1, Rafi Ibn Sultan2, Firoz Mahmud3, Ashadullah Shawon4 and Afsana Khan5(2018). Ensemble of Multiple Models For Robust Intelligent Heart Disease Prediction System. In 4th International Conference on Electrical Engineering and Information & Communication Technology.Sudharsan, RR, J. Deny, E. Muthukumaran, and R. Varatharajan. “FPGA based peripheral myopathy monitoring using MFCV at dynamic contractions.” Journal of Ambient Intelligence and Humanized Computing, 1-9 (2020).

[24]. Purushottam , Prof. (Dr.) Kanak Saxena, Richa Sharma(2016). Efficient Heart Disease Prediction System. In Procedia Computer Science 85 (2016) 962 – 969.

[25]. AvinashGolande, Pavan Kumar T(2019). Heart Disease Prediction Using Effective Machine Learning Techniques.In International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8, Issue-1S4, June 2019.

PROJECT OUTCOMES CBIT(A)

Dept. of ECE Common guidelines for uniform evaluation of BE Projects, AY 2021-22

Course: Project (18ECC34)

Course Outcomes:

Upon completion of this course, students will be able to:

1 Recall the details of the approach for the selected problem.

2 Interpret the approach to the problem relating to the assigned topic.

3 Determine the action plan to conduct investigation.

4 Analyze and present the model / simulation /design as needed.

5 Evaluate, present and report the results of the analysis and justify the same.

S.No	Project Outcome	CO	PO	Blooms Level
1	Studied and summarized the Literature Review to identify and formulate the problems faced in heart disease detection.	CO1	PO1, PO2, PO4	Understand, analyze
2	Selected and gained adequate knowledge on Machine Learning to solve the inadequacies in the current detection system.	CO2	PO5	Analyze, Understand
3	Selected and developed a Random Forest Algorithm based Machine Learning model to detect heart disease.	CO3	PO3, PO5	Analyze, Apply
4	Designed an End-user Web application through which patient can give his information	CO3	PO6,PO5,PO9, PO10,PO11	Analyze, Apply
5	Written a concise and articulate report by emphasizing on the key points and gained presentation skills	CO4	PO8,PO9,PO10	Create, Evaluate, Understand

report group 120 after extra added.docx

ORIGINALITY REPORT

26%
SIMILARITY INDEX

PRIMARY SOURCES

1	V. Sireesha, Nagaratna P. Hegde, Palve Nanditha, B. Ram Naresh. "Chapter 18 Cardiovascular Disease Prediction using Machine Learning with Flask API", Springer Science and Business Media LLC, 2022 Crossref	429 words — 7%
2	ebin.pub Internet	364 words — 6%
3	en.wikipedia.org Internet	199 words — 3%
4	www.ijemr.net Internet	75 words — 1%
5	"ICCCE 2020", Springer Science and Business Media LLC, 2021 Crossref	72 words — 1%
6	Akkem Yaganteeswarudu. "Multi Disease Prediction Model by using Machine Learning and Flask API", 2020 5th International Conference on Communication and Electronics Systems (ICCES), 2020 Crossref	62 words — 1%
7	www.analyticsvidhya.com Internet	59 words — 1%

8	Bhattacharyya, S.. "Data mining for credit card fraud: A comparative study", Decision Support Systems, 201102 Crossref	51 words — 1%
9	www.ijrte.org Internet	36 words — 1%
10	www.hindawi.com Internet	25 words — < 1%
11	V V. Ramalingam, Ayantan Dandapath, M Karthik Raja. "Heart disease prediction using machine learning techniques : a survey", International Journal of Engineering & Technology, 2018 Crossref	23 words — < 1%
12	thereaderwiki.com Internet	21 words — < 1%
13	ijircce.com Internet	18 words — < 1%
14	www.coursehero.com Internet	18 words — < 1%
15	www.cdc.gov Internet	16 words — < 1%
16	www.slideshare.net Internet	16 words — < 1%
17	"Applications of Networks, Sensors and Autonomous Systems Analytics", Springer Science and Business Media LLC, 2022 Crossref	15 words — < 1%

18	norma.ncirl.ie Internet	13 words — < 1%
19	www.mygreatlearning.com Internet	13 words — < 1%
20	M S Sruthi, S Sarath, R Sathish, S Shanthosh. "A Fast and Accurate Face Recognition Security System", Journal of Physics: Conference Series, 2021 Crossref	12 words — < 1%
21	ijisrt.com Internet	12 words — < 1%
22	sersc.org Internet	11 words — < 1%
23	Abu-Jamous, Basel, Rui Fa, and Asoke K. Nandi. "Visualisations and Presentations", Integrative Cluster Analysis in Bioinformatics, 2015. Crossref	10 words — < 1%
24	Sherin J Rouhani, Jonathan A Trujillo, Athalia R Pyzer, Jovian Yu et al. "Severe COVID-19 infection is associated with aberrant cytokine production by infected lung epithelial cells rather than by systemic immune dysfunction", Cold Spring Harbor Laboratory, 2021 Crossref Posted Content	10 words — < 1%
25	filibeto.org Internet	10 words — < 1%
26	www.edureka.co Internet	10 words — < 1%
27	www.ijcaonline.org Internet	10 words — < 1%

28 www.semanticscholar.org
Internet
10 words — < 1%

29 Krishna Battula, Raghupatruni Durgadinesh, Killi Suryapratap, Gedela Vinaykumar. "Use of Machine Learning Techniques in the Prediction of Heart Disease", 2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), 2021
Crossref
9 words — < 1%

30 digitalcommons.du.edu
Internet
9 words — < 1%

31 docplayer.net
Internet
9 words — < 1%

32 Md. Riyazudin, Mousmi Ajay Chaurasia, Syed Ibrahim Ibaad, Alisha Lalani, Salva Fathima. "Chapter 11 Estimation of Quantity and Nutritional Information of Food Using Image Processing", Springer Science and Business Media LLC, 2022
Crossref
8 words — < 1%

33 Rini Puji Astutik, Eliyani, Hendra Ari Winarno, Risma Wahyuni. "Wireless Nurse Call System Using IoT Implementation", 2021 3rd International Conference on Research and Academic Community Services (ICRACOS), 2021
Crossref
8 words — < 1%

34 pdfs.semanticscholar.org
Internet
8 words — < 1%

35 www.ijeat.org
Internet
8 words — < 1%

Appendix

```
# Importing modules
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import pickle
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
df = pd.read_csv('cardio_train.csv',sep=";")
#Display the data
df
series.describe()
pf.columns
split_point=len(series)-10
dataset,validation=series[0:split_point],series(split_point:]
print('Dataset %d, Validation %d' %(len(dataset),len(validation)))
dataset.to_csv('dataset.csv')
validation.to_csv('validation.csv')
#Prepare data
df = pd.read_csv('cardio_train.csv',sep=";")
x = df.iloc[:, 1:12]
y = df.iloc[:, -1]
y = df.cardio.values
x_data = df.drop(['cardio'], axis = 1)
# Normalize
x = (x_data - np.min(x_data)) / (np.max(x_data) - np.min(x_data)).values
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.2,random_state=0)
#import Sklearn
from sklearn.model_selection import train_test_split
```

```
import sklearn.metrics as metrics
#transpose matrices
x_train = x_train.T
y_train = y_train.T
x_test = x_test.T
y_test = y_test.T
# precision-recall curve and f1
from sklearn.datasets import make_classification
from sklearn.model_selection import train_test_split
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import f1_score
from sklearn.metrics import auc
from matplotlib import pyplot
# split into train/test sets
trainX, testX, trainy, testy = train_test_split(x, y, test_size=0.5, random_state=2
# fit a model
model = RandomForestClassifier(n_estimators = 1000, random_state = 1)
model.fit(trainX, trainy)
# predict class values
yhat = model.predict(testX)
lr_precision, lr_recall, _ = precision_recall_curve(testy, lr_probs)
lr_f1, lr_auc = f1_score(testy, yhat), auc(lr_recall, lr_precision)
#Boxplot
df.boxplot(column= ‘age’,by = ‘cardio’)
df.boxplot(column= ‘weight’, by= ‘cholesterol’)
#Heatmap
import seaborn as sns
corradf.corr()
sns.heatmap(corr,xticklabels-corr.columns,yticklabels-corr.columns)
# plot the precision-recall curves for logistic regression
```

```
no_skill = len(testy[testy==1]) / len(testy)
pyplot.plot([0, 1], [no_skill, no_skill], linestyle='--', label='No Skill')
pyplot.plot(lr_recall, lr_precision, marker='.', label='Logistic')
pyplot.xlabel('Recall')
pyplot.ylabel('Precision')
pyplot.legend()
pyplot.show()
# plot the precision-recall curves for SVM
no_skill = len(testy[testy==1]) / len(testy)
pyplot.plot([0, 1], [no_skill, no_skill], linestyle='--', label='No Skill')
pyplot.plot(lr_recall, lr_precision, marker='.', label='Logistic')
pyplot.xlabel('Recall')
pyplot.ylabel('Precision')
pyplot.legend()
pyplot.show()
# plot the precision-recall curves for Decision tree
no_skill = len(testy[testy==1]) / len(testy)
pyplot.plot([0, 1], [no_skill, no_skill], linestyle='--', label='No Skill')
pyplot.plot(lr_recall, lr_precision, marker='.', label='Logistic')
pyplot.xlabel('Recall')
pyplot.ylabel('Precision')
pyplot.legend()
pyplot.show()
#Random Forest
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import pickle
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
import sklearn.metrics as metrics
df = pd.read_csv('cardio_train.csv',sep=";")
x = df.iloc[:, 1:12]
y = df.iloc[:, -1]
y = df.cardio.values
x_data = df.drop(['cardio'], axis = 1)
x = (x_data - np.min(x_data)) / (np.max(x_data) - np.min(x_data)).values
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.2,random_state=0)
accuracies = {}
x_train = x_train.T
y_train = y_train.T
x_test = x_test.T
y_test = y_test.T
#Accuracy of Random Forest
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier(n_estimators = 1000, random_state = 1)
rf.fit(x_train.T, y_train.T)
acc = rf.score(x_test.T,y_test.T)*100
accuracies['Random Forest'] = acc
print("Random Forest Algorithm Accuracy Score : {:.2f}%".format(acc))
#Model3.py
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import pickle
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
df = pd.read_csv('cardio_train.csv',sep=";")
x_data = df.iloc[:, 1:12]
```

```
y = df.iloc[:, -1]
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()
rf.fit(x_data,y)
pickle.dump(rf, open('model3.pkl','wb'))
#App3.py
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle
app = Flask(__name__) #Initialize the flask App
model = pickle.load(open('model3.pkl', 'rb'))
@app.route('/')
def home():
 return render_template('index2.html')
@app.route('/predict',methods=['POST'])
def predict():
 '''
 For rendering results on HTML GUI
 '''
 feature1=int(request.form['age'])*365
 feature2=int(request.form['gender'])
 feature3=int(request.form['height'])
 feature4=float(request.form['weight'])
 feature5=int(request.form['ap_hi'])
 feature6=int(request.form['ap_lo'])
 feature7=int(request.form['cholesterol'])
 feature8=int(request.form['gluc'])
 feature9=int(request.form['smoke'])
 feature10=int(request.form['alco'])
 feature11=int(request.form['active'])
```

```
int_features=[feature1,feature2,feature3,feature4,feature5,feature6,feature7,feature8,feature9,
feature10,feature11]
 #int_features = [int(x) for x in int_features]
 final_features = [np.array(int_features)]
 prediction = model.predict(final_features)
 output = prediction
```