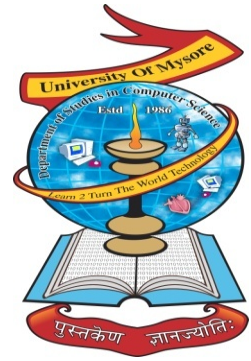




UNIVERSITY OF MYSORE



DEPARTMENT OF STUDIES IN COMPUTER SCIENCE

A project report entitled

**“Developing an Efficient Prediction Model of Heart Disease
Using Machine Learning Techniques”**

Submitted in partial fulfillment of the requirement for the award of the
Master of Science in Computer Science

Submitted by

Mr. BOUBACAR BOUREIMA MOHAMED

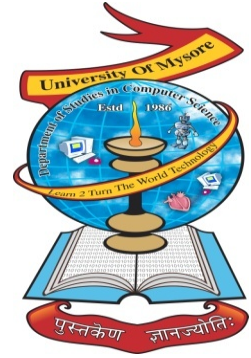
Reg. No. 20MSC10

**Department of Studies in Computer Science
University of Mysore
Mysuru - 570006**

August 2022



UNIVERSITY OF MYSORE



DEPARTMENT OF STUDIES IN COMPUTER SCIENCE

CERTIFICATE

This is to certify that the project entitled **“Developing an Efficient Prediction Model of Heart Disease Using Machine Learning Techniques”** is a bonafide work carried out by **Mr. Boubacar Boureima Mohamed**, with register no.20MSC10, a student of **Department of Studies in Computer Science, Manasagangotri, University of Mysore, Mysuru** in partial fulfillment for the award of the degree of **Master of Science in Computer Science (M. Sc. (CS))** by the **University of Mysore during the academic year 2021 – 22.**

The project work is approved as it satisfies the academic requirements in respect of project work prescribed for the aforesaid degree. This project report has not been submitted previously by anybody for the award of any degree or diploma to any other university.

Internal Guide

Smt. L Hamsaveni
Associate Professor,
DoS in Computer Science
University of Mysore,
Manasagangotri, Mysuru

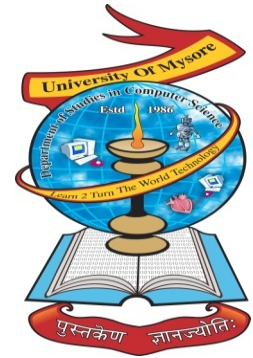
Chairman

Dr. D S Guru
Professor & Chairman,
DoS in Computer Science
University of Mysore,
Manasagangotri, Mysuru

External Examiners:



UNIVERSITY OF MYSORE



DEPARTMENT OF STUDIES IN COMPUTER SCIENCE

Declaration

I, **Mr. Boubacar Boureima Mohamed**, a student of **IV Semester M.Sc., Department of Studies in Computer Science, Manasagangotri, University of Mysore, Mysuru**, do hereby declare that the project entitled **“Developing an Efficient Prediction Model of Heart Disease Using Machine Learning Techniques”** has been carried out by me at Department of Computer Science, Manasagangotri, University of Mysore, Mysuru during the period, March – July 2022. This project report is submitted in partial requirement for the award of the degree **Master of Computer Science (M.Sc)** by the **University of Mysore**.

This is a bonafide work and the matter embodied in the report has not been submitted previously by anybody for the award of any degree or diploma to any other university.

Place: Mysuru

Date: 24 / 08 / 2022

Mr. Boubacar Boureima Mohamed

ACKNOWLEDGEMENT

The satisfaction that implies the successful completion of any work would be incomplete without the mention of people who made it possible.

I am indebted to my internal guide **Smt. L. Hamsaveni, Associate Professor, Department of Studies in Computer Science, Manasagangotri, University of Mysore, Mysuru**, who has motivated and guided me throughout this project work. She has made the entire task simple with her valuable suggestions.

I wish to place on record my deepest gratitude to our beloved Chairman **Dr. D.S. Guru, Professor, Department of Studies in Computer Science, Manasagangotri, University of Mysore, Mysuru**. I am thankful for his support and guidance throughout the completion of my project.

Also, my special thanks to our Course Coordinator **Smt. L. Hamsaveni, Associate Professor, Department of Studies in Computer Science, Manasagangotri, University of Mysore, Mysuru**, for her kind support in completing this project.

Finally, I would like to thank my parents, my friends and all those who have contributed directly or indirectly in my effort to complete this project.

Mr. Boubacar Boureima Mohamed

TABLE OF CONTENTS

1	Introduction	1 – 3
	1.1 Introduction	2
	1.2 Existing System	3
	1.3 Objectives of the project	3
2	Literature Survey	4 - 4
	2. 1 Related Work	
3	Proposed Methodology	5 - 8
	3.1 Motivation for the Proposed System	5
	3.2 Proposed System	5
	3.3 Features of the System	6
	3.4 Advantage of Proposed System	6
	3.5 System Architecture	7
	3.5.1 Architecture of the System	8
4	System Analysis	9 – 10
	4.1 Introduction	9
	4.2 System Analysis	9
	4.3 Feasibility Study	9
	4.3.1 Economic Feasibility	10
	4.3.2 Technical Feasibility	10
	4.3.3 Operational Feasibility	10
5	System Requirements	11 – 13
	5.1 Introduction	11
	5.2 System Specification	11
	5.2.1 Hardware Requirements	11
	5.2.2 Software Requirements	12
	5.3 Functional Requirements	12
	5.4 Non Functional Requirements	13

6	Software Specification	14 – 18
	6.1 Python Programming Language	14
	6.1.1 Pip	14
	6.1.2 Numpy	15
	6.1.3 Scikit-Learn	15
	6.1.4 Matplotlib	15
	6.2 Flask Python Web Framework	16
	6.2.1 Flask SQLAlchemy	17
	6.3 Localhost	18
7	System Design	19 – 25
	7.1 Introduction	19
	7.2 Use Case Diagrams	19
	7.3 Sequence Diagrams	214
	7.4 Data Flow Diagrams	23
8	Implementation	26-38
	8.1 Introduction	26
	8.2 Machine Learning Techniques	26
	8.2.1 Supervised Learning	26
	8.2.2 Unsupervised Learning	27
	8.3 Machine Learning Algorithms	27
	8.3.1 Support Vector Machine (SVM)	27
	8.3.2 Decision Tree	28
	8.3.3 Random Forest	29
	8.3.4 Logistic Regression	30
	8.4 Collection of Dataset	31
	8.5 Selection of Attributes	32
	8.6 Pre-Processing of Data	33
	8.7 Prediction of Heart Disease	33
	8.8 Dataset Details	33
	8.9 Screenshots	34

9	Software Testing	39– 40
	9.1 Testing	39
	9.2 Testing Levels	39
	9.3 System Test Cases	39
	9.4 Test Objectives	39
	9.4.1 Navigation from Index page to the Login or Register page	40
	9.4.2 Navigation from Login page to the Home page	40
	9.4.3 Navigation from Home page to the Prediction page	40
10	Conclusion and Future Enhancements	41-43
	10.1 Conclusion	41
	10.2 Future Enhancements	41
	References	42
	Bibliography	43

LIST OF FIGURES

Figure 3.1 Architecture of the System	8
Figure 6.1 Flask Python Web Framework	16
Figure 6.2 Flask SQLAlchemy	17
Figure 7.1 Use Case Diagram	20
Figure 7.2 Sequence diagram	22
Figure 7.3 Data Flow Diagram-level 0	23
Figure 7.4 Data Flow Diagram- level 1	24
Figure 8.1 Support Vector Machine	28
Figure 8.2 Decision Tree	29
Figure 8.3 Random Forest	30
Figure 8.4 Logistic Regression	31
Figure 8.5 Dataset Classification	31
Figure 8.6 Correlation Matrix	32
Figure 8.7 Dataset Attributes	33
Figure 8.8 Index and About Pages	34
Figure 8.9 Login and Registration Forms	35
Figure 8.10 Home Page	36
Figure 8.11 Import Dataset Form	36
Figure 8.12 User profile	37
Figure 8.13 Prediction Form	37
Figure 8.14 Heart Disease Prediction	38

LIST OF TABLES

Table: 7.1 Test case from the index page to login page	39
Table: 7.2 Test case from login page to home page	39
Table: 7.3 Test case form home page to predication page	40

Abstract

A great diversity comes in the field of medical sciences because of computing capabilities and improvements in techniques, especially in the identification of human heart diseases. Nowadays, heart disease is one of the most critical human diseases in the world and has very serious effects the human life. Accurate and on time diagnosis of heart disease is important for heart failure prevention and treatment. The diagnosis of heart disease through traditional medical history has been considered as not reliable in many aspects. To classify the healthy people and people with heart disease, noninvasive-based methods such as machine learning are reliable and efficient. This project presents a Smart Heart Disease Prediction (SHDP) system that is quicker and more proficient than the usual system of diagnosis. SHDP is based on machine learning techniques for predicting the heart disease and knowing present heart status. In order to demonstrate the suitability of the proposed SHDP application, accurate and timely identification of human heart disease can be very helpful in preventing heart failure in its early stages and will improve the patient's survival. This project is an attempt to solve a healthcare problem currently society is facing. The main objective of the project was to design a remote healthcare system. It's comprised of three main parts. Detection of patient's heart disease, Knowing the stage of heart disease properly, Perceiving about heart disease prediction result and the cause of heart related problems.

CHAPTER 1

Introduction

In today's world, cardiovascular disease is the leading cause of death. Heart disease prediction is a critical challenge in the medical data processing. The emergence of machine learning techniques has demonstrated their effectiveness in disease prediction from massive amounts of healthcare data. Heart disease is difficult to recognize due to a variety of risk factors such as high blood pressure, cholesterol, and abnormal pulse rate. Because of the disease's complexity, it must be handled with care.

Otherwise, the effects of heart or death may occur. With computer-aided, decision-support/prediction systems, technological advancements have aided the field of medicine. In the healthcare industry, machine learning techniques have demonstrated accurate disease prediction in less time. In the case of heart disease, early detection is critical in saving patients' lives. It is also necessary to protect patients from such diseases. To address cardiac risk, accurate decision-making and optimal treatment are required. This project analyses the algorithms and methods used to implement prediction of heart diseases. It is directed towards machine learning and also to provide a detailed analysis of heart disease risk based on several factors that may have a direct impact on cardiovascular health.

The aim of the project was to come up with the implementation of an efficient low cost predicting system with high reliability for the protection of valuable life of affected patients by heart disease. Hence the proposed architecture of Smart Heart Disease Prediction system receives the health record data through the web interface where it is processed and analyzed for further using.

Existing System

There are hundreds of algorithms implemented to classify, cluster, and find hidden patterns in data. Also, there has been a lot of research done in the field of automated disease detection. However, domain specific issues of healthcare are still to be resolved. To get better accuracy of heart disease prediction, precisely work shall be done. In the world, we can commonly see cardiovascular disease is the leading cause of death in middle-aged people.

The type of disease and its manifestations are also evolving over time due to a variety of factors. Heart disease identification is also difficult due to the difficulty of obtaining data since changing of symptoms.

Objectives of the project

The main objective is to implement an efficient low cost predicting system with high reliability for the protection of valuable life of affected patients by heart disease. This project will make it possible to propose a solution to a concrete problem, starting from a definition of needs.

Overall, the system will help in:

- Reducing the cost to detect heart disease.
- Knowing the stage of heart disease properly.
- Finding out heart disease prediction with best accuracy.
- Increasing aware about heart disease after prediction to the patients.
- Developing an effective prediction model in distributed environment.
- Perceiving about heart disease prediction result and the cause of heart related problems.

CHAPTER 2

Related Work

A quiet significant amount of work related to the diagnosis of Cardiovascular Heart disease using Machine Learning techniques has motivated this study. An efficient Cardiovascular heart disease prediction has been made by using many techniques some of them include Logistic Regression, KNN, Random Forest Classifier etc. It can be seen in Results that each technique has its strength to register the defined objectives.

The study for the Prediction of Cardiovascular Disease by comparing the accuracies of applying rules to the individual results of Support Vector Machine, Random forest, Naive Bayes classifier and logistic regression on the dataset taken in a region to present an accurate model of predicting cardiovascular disease. The machine learning algorithms used in that study were able to predict cardiovascular disease in patients with accuracy between 58.71% and 77.06%. Logistic Regression achieved the highest Accuracy (77.06 %) when compared to different Machine-learning Algorithms. In this study, we will use some classifiers of machine learning such as Support Vector Machine, Random forest, Decision Tree, Logistic Regression and KNN. Furthermore, will check their accuracy on the standard heart disease dataset by performing certain preprocessing, standardization of dataset, and hyper parameter tuning. Additionally, to train and validate the machine learning algorithms. Finally, the experimental result should indicate that the accuracy of the prediction classifiers with hyper parameter tuning improved and achieved notable results with data standardization and the hyper parameter tuning of the machine learning classifiers. After reviewing the previous studies, it seems that the dataset used is the one in UCI Machine Learning Heart Disease dataset which is different from the dataset which is in the kaggle depository. We will use similar machine learning algorithms for training, validating and testing these algorithms.

CHAPTER 3

Motivation for the proposed system

In the medical field area, there is no easy way to identify heart disease. It's a little bit costly specially, lower middle-class people. People are suffering from cardiovascular disease surreptitiously of their mind. These affected people do not have the ability to diagnose these types of heart related disease for its huge amount of cost. The cost of diagnosis of the heart disease is very high because it is one of the most difficult tasks in the medical field and it's difficult to identify cardiovascular disease. Some people are infected congenital heart disease. To identify heart disease, patients have to follow various tests concurrently. It's a little bit hard and costly to catch up the syndrome. We know, world of technology is moving towards artificial intelligence and machine learning techniques, huge amount of raw data is being produced in real time. This data gives us with the opportunity to analyze this huge amount of data using data mining and machine learning techniques.

That's why the primary motivation of this work is to find out the collection of cardiovascular disease related syndrome data to predict heart disease perfectly at low cost based machine learning techniques. So, those patients can easily get treatment

Proposed System

According to previous problems as mentioned above, there have been a modest number of studies in this field. The studies have been fairly successful in their own way. From studying different algorithms to make re-optimization to the existing algorithm to find better results, researchers have gone through many different ways. The observable factor is that although the accuracy has been quite good, yet we have not seen any real implementation of these processes. Hence, developing an efficient prediction model of heart disease based machine learning techniques at low cost to predict cardiovascular disease.

The proposed system not only performs the task of disease prediction but also identifies and classifies heart disease patient and non-heart disease patient automatically along with the reducing of the cost to early detect heart disease using machine learning techniques.

This system will also take data about whether heart disease patients smoke too much, drink a lot, have high diabetes level, and so on and the data will be categorized by male and female. After analysis the various categories data, the system will predict the risk of heart disease with higher accuracy.

Proposed Methodology

Features of the system

- A health data classification mechanism to enable good patient care.
- Performance analysis of the SHDP application to show its effectiveness.
- A flexible, energy-efficient, and scalable remote patient's heart disease status monitoring system.
- A case study where the capabilities of the SHDP application are exploited for patients with heart disease.

Advantage of the proposed system

- It is a multipurpose so that overall conditions are easily measured.
- It helps in faster detection of heart disease :
 - Easy to operate
 - Compare with compact it gives better performance
 - Predicting save the life and protect the health

Proposed Methodology

System Architecture:

- User/Admin Interface: It is the Front End of the system through which the user can interact with the system. The Web Interface contains the following parts:
 - Login Interface
 - Registration Interface
 - User Data Form
- Prediction System: It is the part of the system which contains the actual raw data provided by the user, dataset, and machine learning algorithms and helps process that data for further use. It performs the following tasks:
 - Processing User Input Data
 - Handling Database
 - Machine Learning algorithms
 - Predict cardiovascular diseases
- The application is designed in such way we can easily view patient's data and analyzed it.

Proposed Methodology

Architecture of the system:

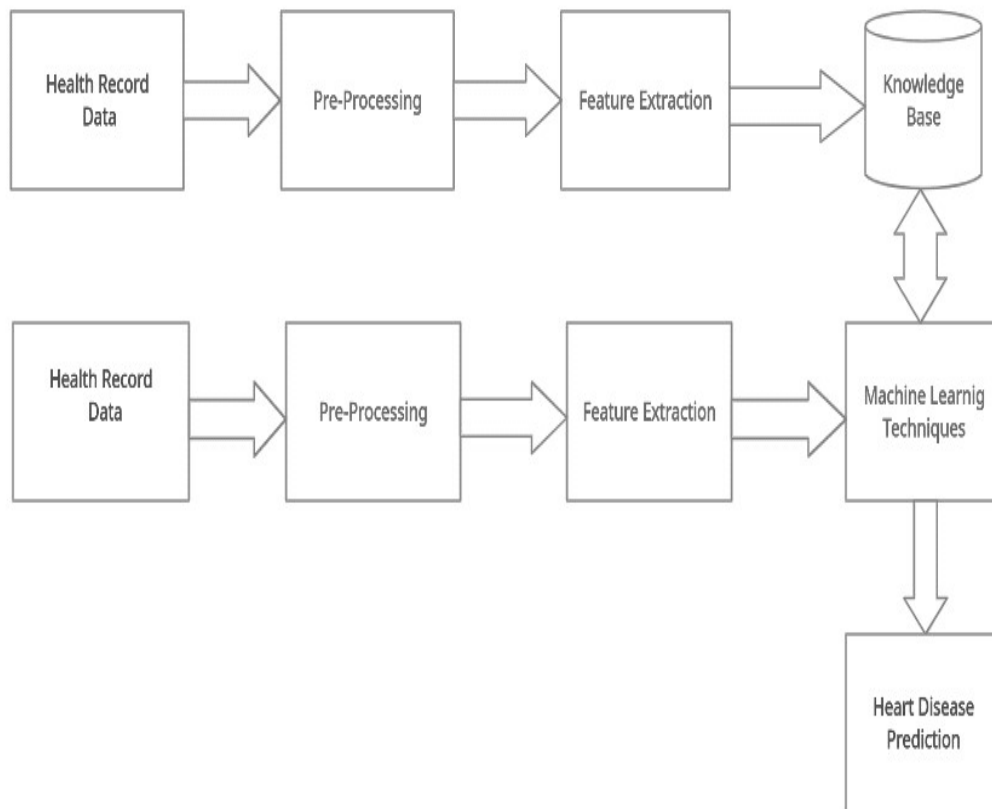


Figure: 3.1 Architecture of the system.

CHAPTER 4

Introduction

Systems analysis is a process of collecting factual data, understand the processes involved, identifying problems and recommending feasible suggestions for improving the system functioning. This involves studying the business processes, gathering operational data, understand the information flow, finding out bottlenecks and evolving solutions for overcoming the weaknesses of the system so as to achieve the organizational goals. System Analysis also includes subdividing of the complex process involving the entire system, identification of data store and manual processes.

System Analysis

The major objectives of systems analysis are to find answers for each business process: What is being done, How is it being done, Who is doing it, When is he doing it, Why is it being done and How can it be improved? It is more of a thinking process and involves the creative skills of the System Analyst. It attempts to give birth to a new efficient system that satisfies the current needs of the user and has scope for future growth within the organizational constraints. The result of this process is a logical system design. Systems analysis is an iterative process that continues until a preferred and acceptable solution emerges.

Feasibility Study

The feasibility of the project is analyzed in this phase and the business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis, the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are :

Economic Feasibility

In the economic feasibility study, we have asked the following question:

What is the project cost and is it reasonable?

What are the financial benefits of the project?

We have found that the project cost is reasonable and will help the healthcare organization to know the status of patient's heart, saving time, money, and effort. The system provides the heart disease detection's service to the patients. We conclude that the system cost is reasonable and it has beneficial benefits for both organization and patients.

Technical Feasibility

In the Technical feasibility study, we have asked the following question:

- What are the technical requirements (hardware and software) for the project?
- Are they available?
- What is their cost is it reasonable for organization and patient?
- What are the required techniques?

Answering the above questions, we have found that the project technical requirement hardware and software, other techniques and tools are available with the reasonable cost that makes it easy to implement the system with less cost, effort and time. The system will help the healthcare organization to know the status of patient's heart, saving time, money, and effort. The system provides the heart disease detection's service to the patients. We conclude that the availability of technical requirement of the project with cost will benefit the organization and patients.

Operational Feasibility

The operational feasibility of this project is high since it is user-friendly because the environment of patient or company appropriate to apply system, nowadays the internet networks are available in everywhere, those factors will help in project implementation and apply the system in all environments operational Feasibility.

The aspect of the study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

CHAPTER 5

Introduction

Software requirements specification (SRS) is a description of a software system to be developed, laying out functional and non-functional requirements, and may include a set of use cases that describe interactions the users will have with the software. Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers (in market-driven projects, these roles may be played by the marketing and development divisions) on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules.

The software requirements specification document enlists enough and necessary requirements that are required for the project development. To derive the requirements we need to have the clear and thorough understanding of the products to be developed or being developed. This is achieved and refined with detailed and continuous communications with the project team and customer till the completion of the software.

System Specification

Hardware Requirements:

- Processor Speed: 1.6 GHz
- RAM: 4 GB or higher
- Disk Space: 80 GB or higher

System Requirements

Software Requirements:

- Operating System: Windows 7 or higher.
- IDE: Visual Studio Code or Pycharm
- Technologies used: Flask, Python, Jupyter Notebook, MySQL

Functional Requirements:

A functional requirement document defines the functionality of a system or one of its subsystems. It also depends upon the type of software, expected users and the type of system where the software is used.

Functional user requirements may be high-level statements of what the system should do but functional system requirements should also describe clearly about the system services in detail.

- Log on
- Register
- User data from
- Processing user input data
- Handling database
- Predict cardiovascular diseases.

Register:

The first step for the patient after he visits the hospital and receives the tools is to be registered by filling in some information about him and he will get id as the primary key for him.

User data from:

After patient registration, he must get the health records data of the patient and start processing the user input data.

System Requirements

Handling database

It is the part of the system which contains the actual raw data provided by the user, dataset, and helps process that data for further use.

Predict cardiovascular diseases

Using our prediction module the doctor can do the analysis to data using machine learning algorithms to predicate whether the patient has heart disease or not.

Non-Functional Requirements:

Non-functional requirements are constraints that must be adhered to during development. They limit what resources can be used and set bounds on aspects of the software's quality. One of the most important things about non-functional requirements is to make them verifiable. The verification is normally done by measuring various aspects of the system and seeing if the measurements conform to the requirements. Non-functional requirements are divided into several groups:

The first group of categories reflects the five qualities attributes:

- **Usability:** The application can be used by the doctor and other users easily. It is user-friendly.
- **Efficiency:** Our application takes less time to accomplish a particular task such as predicting which also reduces time complexity.
- **Reliability:** The application designed to provide a set of services as expected by the user. The application provides many modules and each module is developed to satisfy the non-functional requirements.
- **Maintainability:** By using some storage backups for the database which is available, so, maintainability is very easy.
- **Portability:** The Software is a web-based application, built using Python and MySQL, so the application is independent of operating system.

CHAPTER 6

Python Programming

Language

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. In July 2018, Van Rossum stepped down as the leader in the language community after 30 years. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms are also supported and this can be good, including object oriented, imperative, functional and procedural, and has a large and comprehensive standard library. Python is used in web, game, desktop applications development, AI, data science and so on. There are the following why python is popular now:

- Presence of third-party module
- Open source and community development
- User-friendly data structures
- High-level language.

PIP

Pip is the package installer for python. It is used to install packages from the Python Packages Index and other indexes. This is what was used to install all the back-end modules that were used in this system. First introduced as pyinstall in 2008 by Ian Bicking (the creator of the virtualenv package) as an alternative to easy install, pip was chosen as the new name from one of several suggestions that the creator received on his blog post. According to Bicking himself, the name is a recursive acronym for "Pip Installs Packages". In 2011, the Python Packaging Authority (PyPA) was created to take over the maintenance of pip and virtualenv from Bicking, led by Carl Meyer, Brian Rosner, and Jannis Leidel.

NumPy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors.

Scikit-Learn

Scikit-learn, which has a lot of usages, is a well-known and free of charge software machine learning library in the Python programming language.

It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k -means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. The scikit-learn project started as scikits.learn, a Google Summer of Code project by David Cournapeau. Its name stems from the notion that it is a "SciKit" (SciPy Toolkit), a separately-developed and distributed third-party extension to SciPy. The original codebase was later rewritten by other developers.

Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK. There is also a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely resemble that of MATLAB, though its use is discouraged. SciPy makes use of Matplotlib.

Matplotlib was originally written by John D. Hunter. Since then it has an active development community and is distributed under a BSD-style license. Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in August 2012 and was further joined by Thomas Caswell.

Flask Python Web Framework

Flask is a web application framework written in Python. It was developed by Armin Ronacher, who led a team of international Python enthusiasts called Poocco. Flask is based on the Werkzeug WSGI toolkit and the Jinja2 template engine. A Web Framework represents a collection of libraries and modules that enable web application developers to write applications without worrying about low-level details such as protocol, thread management, and so on.

The Web Server Gateway Interface (Web Server Gateway Interface, WSGI) has been used as a standard for Python web application development. WSGI is the specification of a common interface between web servers and web applications.

Flask is considered more Pythonic than the Django web framework because in common situations the equivalent Flask web application is more explicit.



Figure 6.1 Flask Python Web Framework

Flask SQLAlchemy

Flask-SQLAlchemy is an extension for [Flask](#) that adds support for [SQLAlchemy](#) to your application. It aims to simplify using SQLAlchemy with Flask by providing useful defaults and extra helpers that make it easier to accomplish common tasks.

SQLAlchemy is the Python SQL toolkit and the Object Relational Mapper that gives application developers the full power and flexibility of SQL. It provides a full suite of well known enterprise-level persistence patterns, designed for efficient and high-performing database access, adapted into a simple and Pythonic domain language.

SQL databases behave less like object collections the more size and performance start to matter; object collections behave less like tables and rows the more abstraction starts to matter. SQLAlchemy aims to accommodate both of these principles. SQLAlchemy is most famous for its object-relational mapper (ORM), an optional component that provides the **data mapper pattern**, where classes can be mapped to the database in open ended, multiple ways - allowing the object model and database schema to develop in a cleanly decoupled way from the beginning.

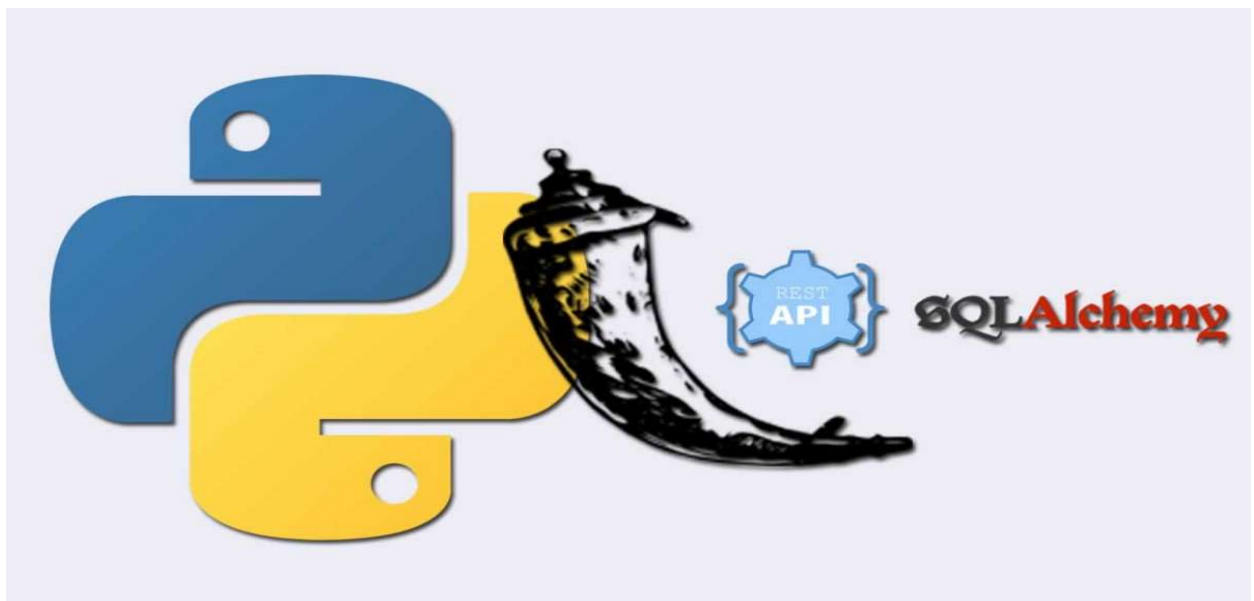


Figure 6.2 Flask SQLAlchemy

Local host

The local loopback mechanism may be used to run a network service on a host without requiring a physical network interface, or without making the service accessible from the networks the computer may be connected to. For example, a locally installed website may be accessed from a Web browser by the URL *http://localhost* to display its home page.

The name *localhost* is reserved for loopback purposes by RFC 6761 (*Special-Use Domain Names*), which achieved the Proposed Standard maturity level in February 2013. The standard sets forth a number of special considerations governing the use of the name in the Domain Name System:

CHAPTER 7

Introduction

System design is the process, which involves conceiving planning and carrying out the plan by generating the necessary reports and inputs. In other words, design phase acts as a bridge between the software requirement specification and implementation phase, which satisfies those requirements. System Design is the transformation of the analysis model into a system design model. The design of a system is correct if a system built precisely according to the requirements of that system. Design should be clearly verifiable, complete and traceable. The goal is to divide the problem into manageably small modules that can be solved separately, the different modules have to cooperate and communicate together to solve the problem. The complete project is broken down into different identifiable modules. Each module can be understood separately. All the modules at last are combined to get the solution of the complete system.

Use Case Diagrams

A Use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

Interaction among actors is not shown in the use case diagram. If this interaction is essential to a coherent description of the desired behavior, perhaps the system or use case boundaries should be re-examined. Alternatively, interaction among actors can be part of the assumptions used in the use case. Terms used in use case diagram:

➤ Use cases:

A use case describes a sequence of actions that provide something of measurable value to an actor and is drawn as a horizontal ellipse.

➤ Actors:

An actor is a person, organization, or external system that plays a role in one or more interactions with the system.

➤ System boundary boxes:

A rectangle is drawn around the use cases, called the system boundary box, to indicate the scope of the system. Anything within the box represents functionality that is in scope and anything outside the box is not.

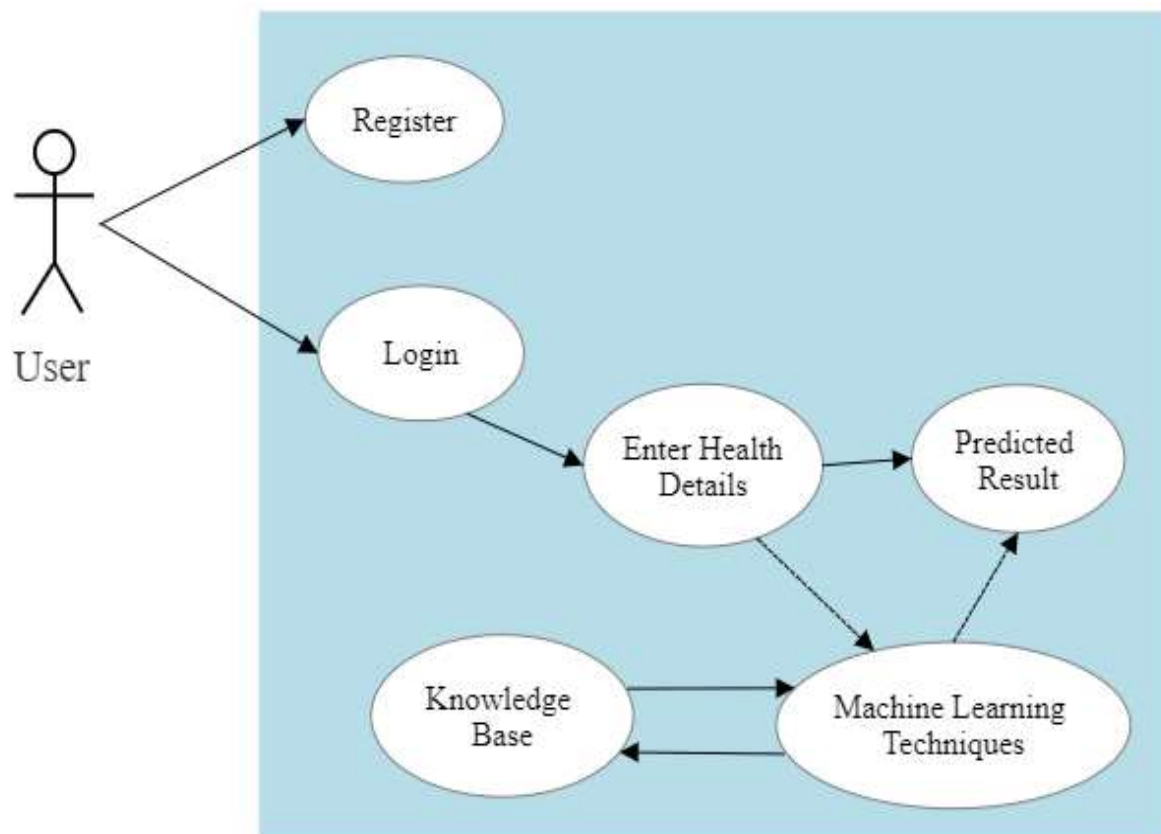


Figure: 7.1 Use Case Diagram

Sequence Diagrams

A sequence diagram shows interaction among objects as a two-dimensional chart. The chart is read from top to bottom. The objects participating in the interaction are shown at the top of the chart as boxes attached to a vertical dashed line. Inside the box, the name of the object is written with a colon separating it from the name of the class and both the name of the object and the classes are underlined.

The objects appearing at the top signify that the object already existed when the use case execution was initiated. However, if some object is created during the execution of the use case and participates in the interaction (e.g. a method call), then the object should be shown at the appropriate place on the diagram where it is created. The vertical dashed line is called the object's lifeline.

The lifeline indicates the existence of the object at any particular point in time. The rectangle drawn on the lifetime is called the activation symbol and indicates that the object is active as long as the rectangle exists. Each message is indicated as an arrow between the lifelines of two objects.

The messages are shown in chronological order from the top to the bottom. That is, reading the diagram from the top to the bottom would show the sequence in which the messages occur. Each message is labeled with the message name. Some control information can also be included. Two types of control information are particularly valuable.

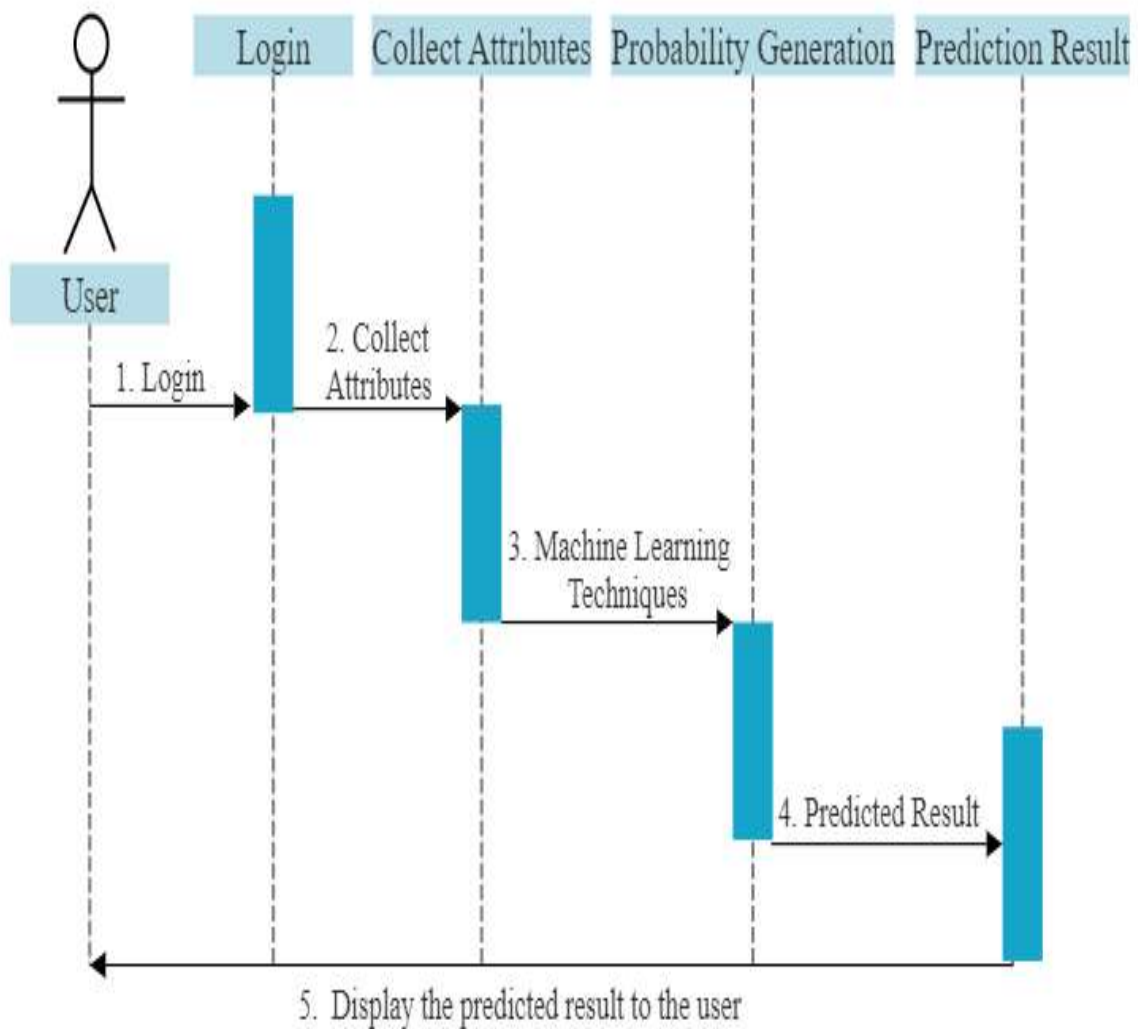


Figure: 7.2 Sequence diagram

Data Flow Diagrams

The DFD (also known as a bubble chart) is a hierarchical graphical model of a system that shows the different processing activities or functions that the system performs and the data interchange among these functions.

Data Flow Diagram – Level 0

The level 0 DFD shows the entire system as a single process. Interactions with users and other external entities are shown as the data flow. The level 0 DFD clarifies the scope of the proposed system, the kinds of users the system will have, and the data coming out from and going into the system. It motivates and establishes a framework for the more complicated next level.

It provides an abstract view of how the end users of our application interact with the system and also illustrates the input the output of the system.

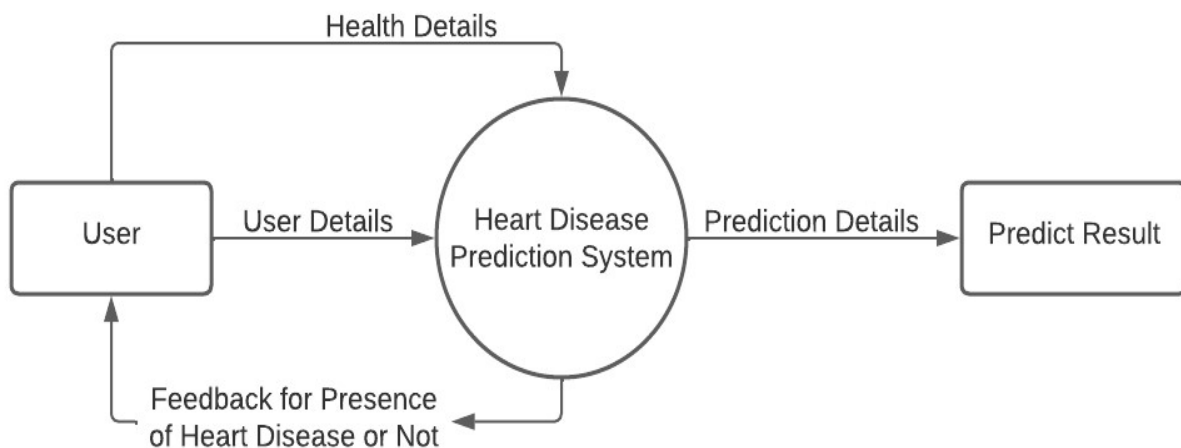


Figure: 7.3 Data Flow Diagram-level 0

Data Flow Diagram- level 1

Level 1 Data Flow Diagram gives more information than level 0 Data Flow Diagram. It Shows additional functionality that provided by the application to the user. The flow diagram illustrates the activities in more depth compared to data flow diagram level 0. The functions the flow of the system are depicted clearly.

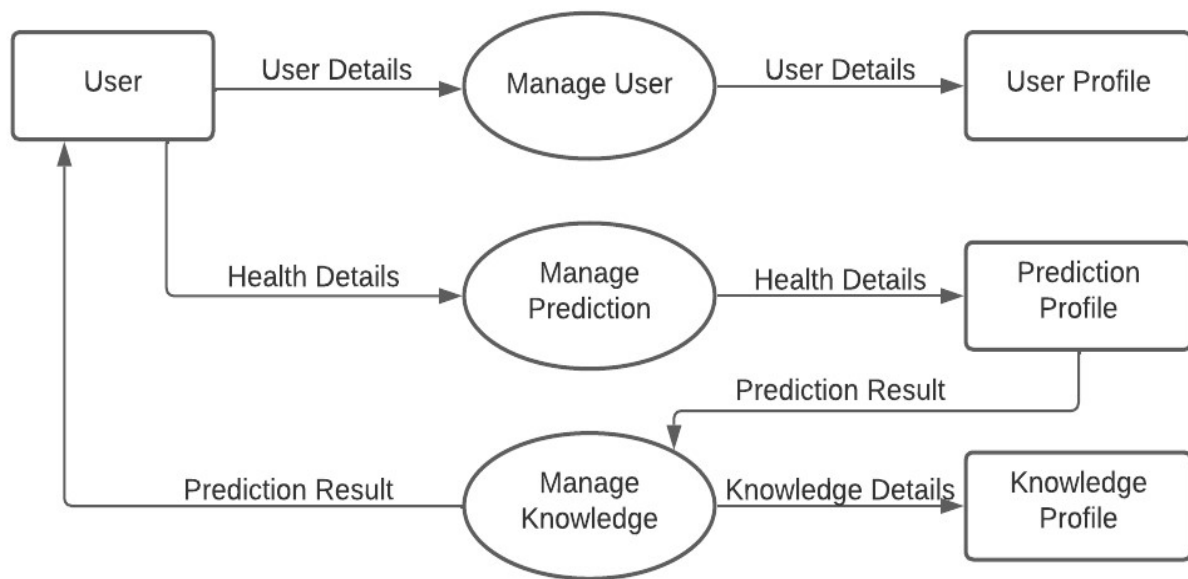


Figure: 7.4 Data Flow Diagram- level 1

Data Flow Diagram- level 1 (Manage User)

The Data Flow Diagram Level 1 (Manage User) is specifically designed for user management. Users need to register to the application and fill in the registration details.

Then, they can log in to the application by submitting the email and password created by them in register process previously.

Users can update their profile and all this process will be store in user profile data store.

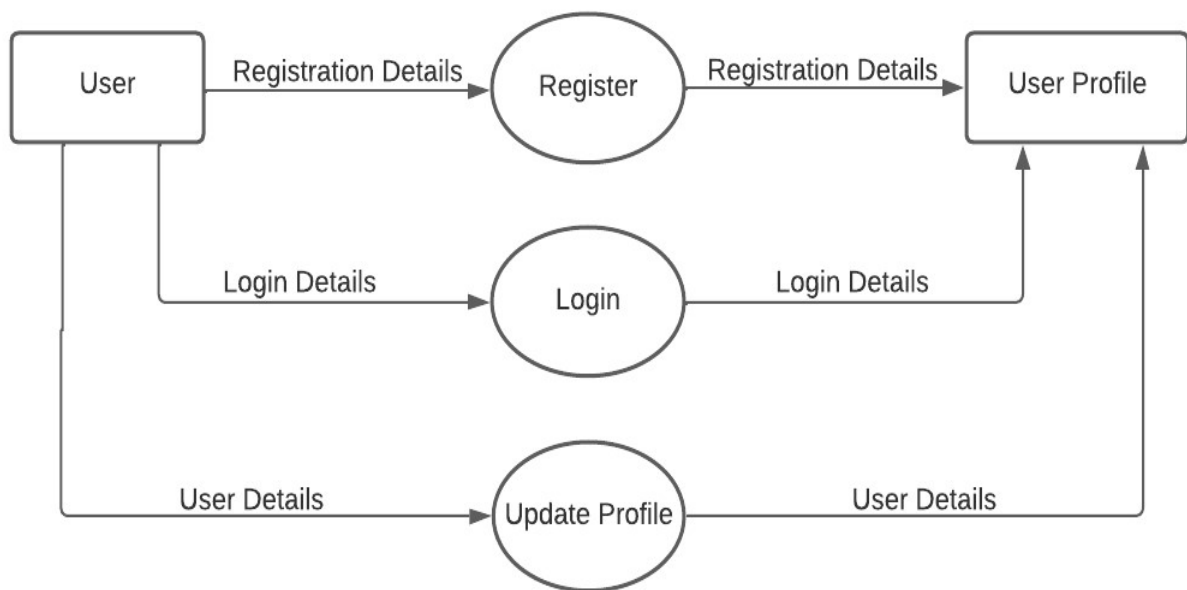


Figure: 7.4 Data Flow Diagram- level 1

Data Flow Diagram- level 1 (Manage Prediction)

The Data Flow Diagram Level 1 (Manage Prediction) is specifically designed for prediction management.

Users can check their prediction result after completing the health details and the prediction result will be displayed with better accuracy.



Figure: 7.4 Data Flow Diagram- level 1

CHAPTER 8

Introduction

The goals of implementation phase are to translate the design of the system procedure during the phase into code from in a given programming language which can then be executed by the computer performing the computation specified by the design, the coding phase affects both testing and maintenance profoundly. Well written code can reduce the testing and maintains cost. A crucial phase in the system lifestyle is the successful implementation of the system design. Implementation simply means converting the system designs into operation.

This stage is considered to be most crucial stage in the development of a successful system since a new system is developed and the users get information in effective manner. Implementation is a stage in which the design is converted into working system that is it the stage of the project where theoretical design is turned into a working system. The implementation involves careful planning, investing of the current system and its constraint on implementation, design of methods to achieve the changeover.

Machine Learning Techniques

In machine learning, classification refers to a predictive modeling problem where a class label is predicted for a given example of input data.

Supervised Learning

Supervised learning is the type of machine learning in which machines are trained using well "labelled" training data, and on the basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.

In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher.

Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to find a mapping function to map the input variable(x) with the output variable(y).

Unsupervised Learning

Unsupervised learning cannot be directly applied to a regression or classification problem because unlike supervised learning, we have the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of dataset, group that data according to similarities, and represent that dataset in a compressed format.

- Unsupervised learning is helpful for finding useful insights from the data.
- Unsupervised learning is much similar to how a human learns to think by their own experiences, which makes it closer to the real AI.
- Unsupervised learning works on unlabeled and uncategorized data which make unsupervised learning more important.
- In real-world, we do not always have input data with the corresponding output so to solve such cases, we need unsupervised learning.

Machine Learning Algorithms

Support Vector Machine (SVM)

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence the algorithm is termed as Support Vector Machine.

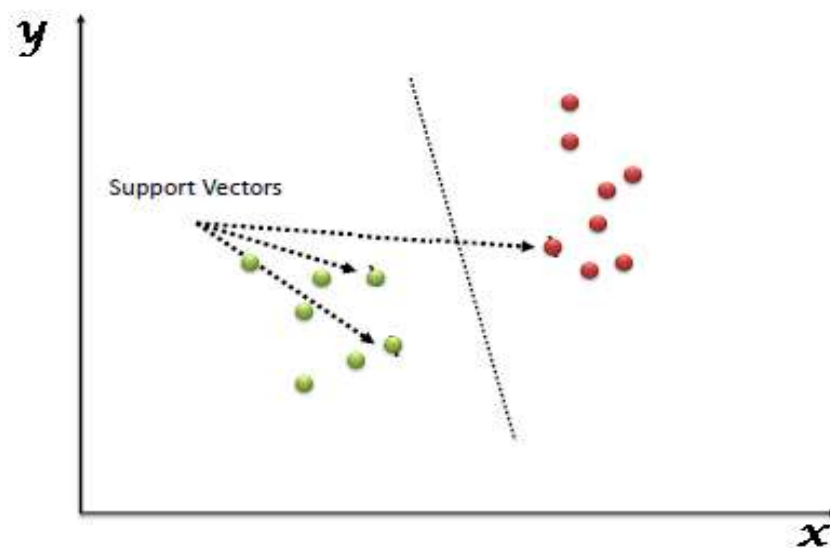


Figure: 8.1 Support Vector Machine

Decision Tree

Decision Tree is a supervised learning technique that can be used for both classification and regression problems, but mostly it is preferred for solving classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a Decision Tree, there are two nodes, which are the Decision Node and Leaf Node.

Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions. It is called a Decision Tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure. In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm. A Decision Tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

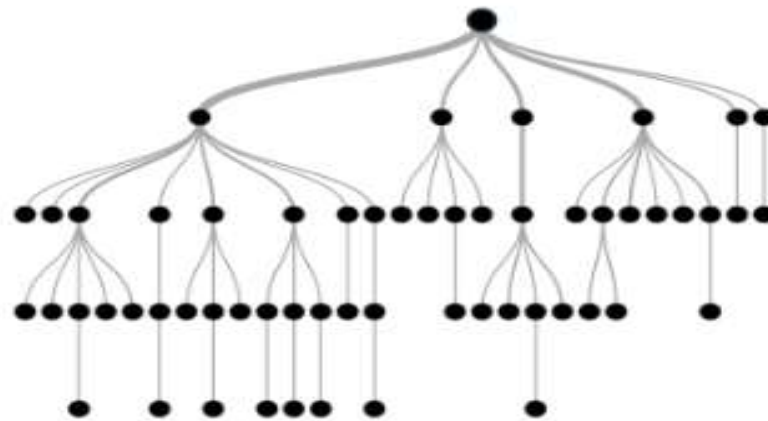


Figure: 8.2 Decision Tree

Random Forest

Random Forest is a supervised learning algorithm. It is an extension of machine learning classifiers which include the bagging to improve the performance of Decision Tree. It combines tree predictors, and trees are dependent on a random vector which is independently sampled. The distribution of all trees is the same. Random Forests splits nodes using the best among of a predictor subset that are randomly chosen from the node itself, instead of splitting nodes based on the variables. The time complexity of the worst case of learning with Random Forests is $O(M(dn \log n))$, where M is the number of growing trees, n is the number of instances, and d is the data dimension.

It can be used both for classification and regression. It is also the most flexible and easy to use algorithm. A forest consists of trees. It is said that the more trees it has, the more robust a forest is. Random Forests create Decision Trees on randomly selected data samples, get predictions from each tree and select the best solution by means of voting. It also provides a pretty good indicator of the feature importance.

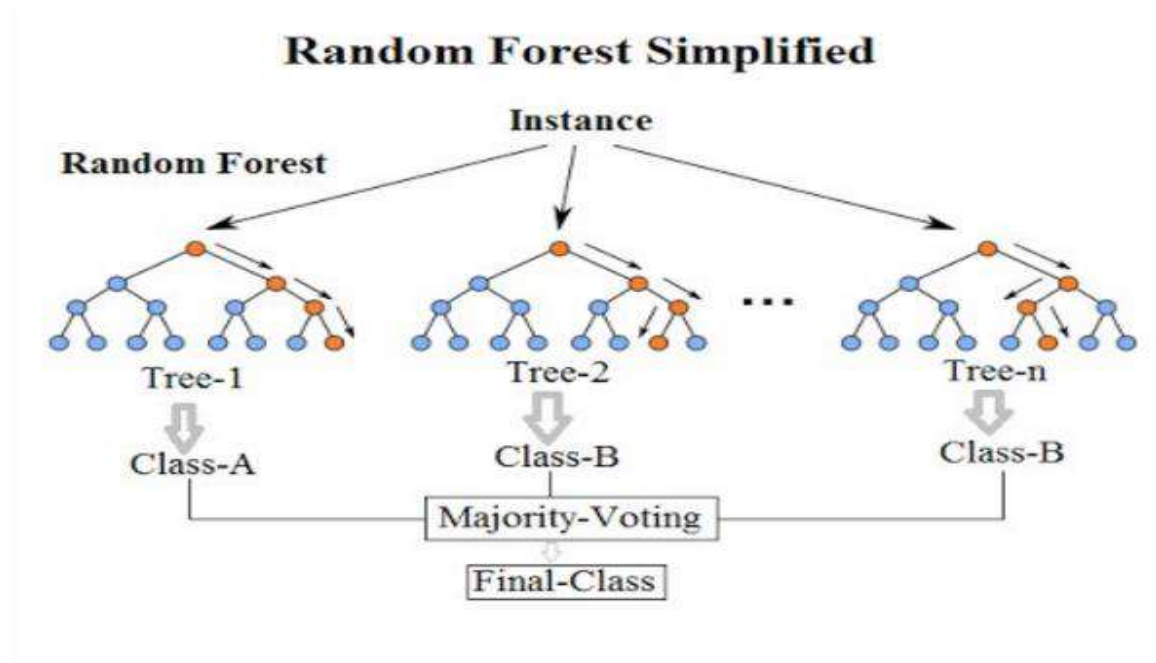


Figure: 8.3 Random Forest

Logistic Regression

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas logistic regression is used for solving the classification problems. In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1). The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.

Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.

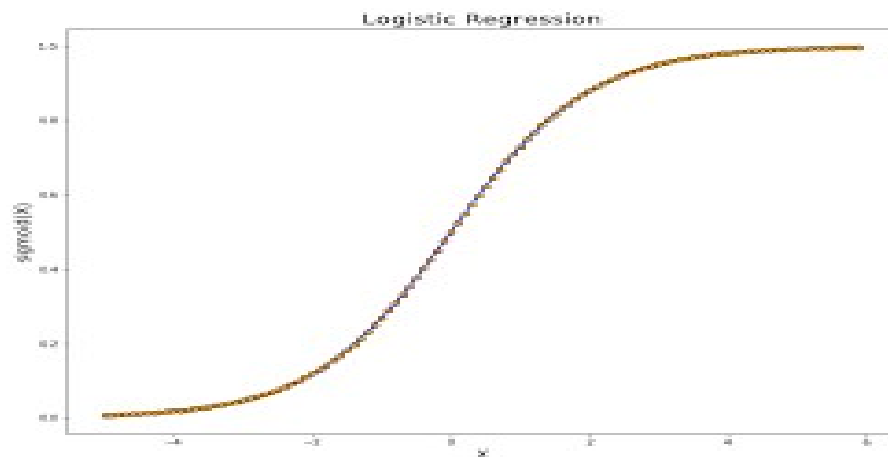


Figure: 8.4 Logistic Regression

Collection of Dataset

Initially, we collect a dataset for our heart disease prediction system. After the collection of the dataset, we split the dataset into training data and testing data. The training dataset is used for prediction model learning and testing data is used for evaluating the prediction model. For this project, 75% of training data is used and 25% of data is used for testing. The dataset used for this project is Heart Disease Cleveland UCI. The dataset consists of 76 attributes; out of which, 14 attributes are used for the system.

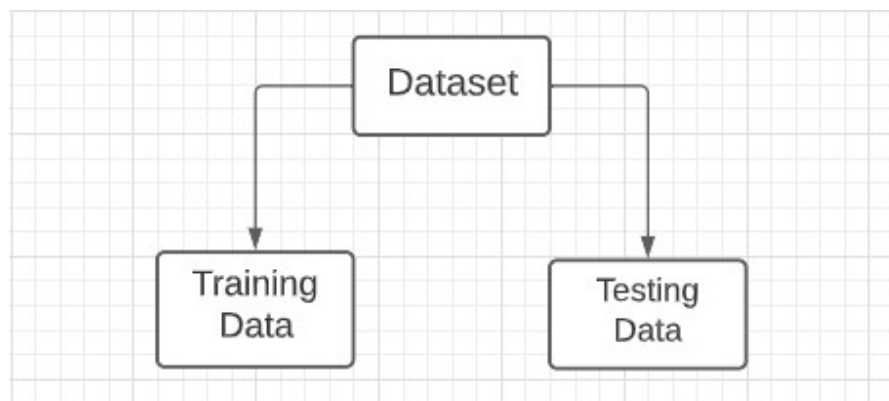


Figure: 8.5 Dataset Classification

Selection of Attributes

Attribute or Feature selection includes the selection of appropriate attributes for the prediction system. This is used to increase the efficiency of the system. Various attributes of the patient like gender, chest pain type, fasting blood pressure, serum cholesterol, exang, etc are selected for the prediction. The Correlation matrix is used for attribute selection for this model.

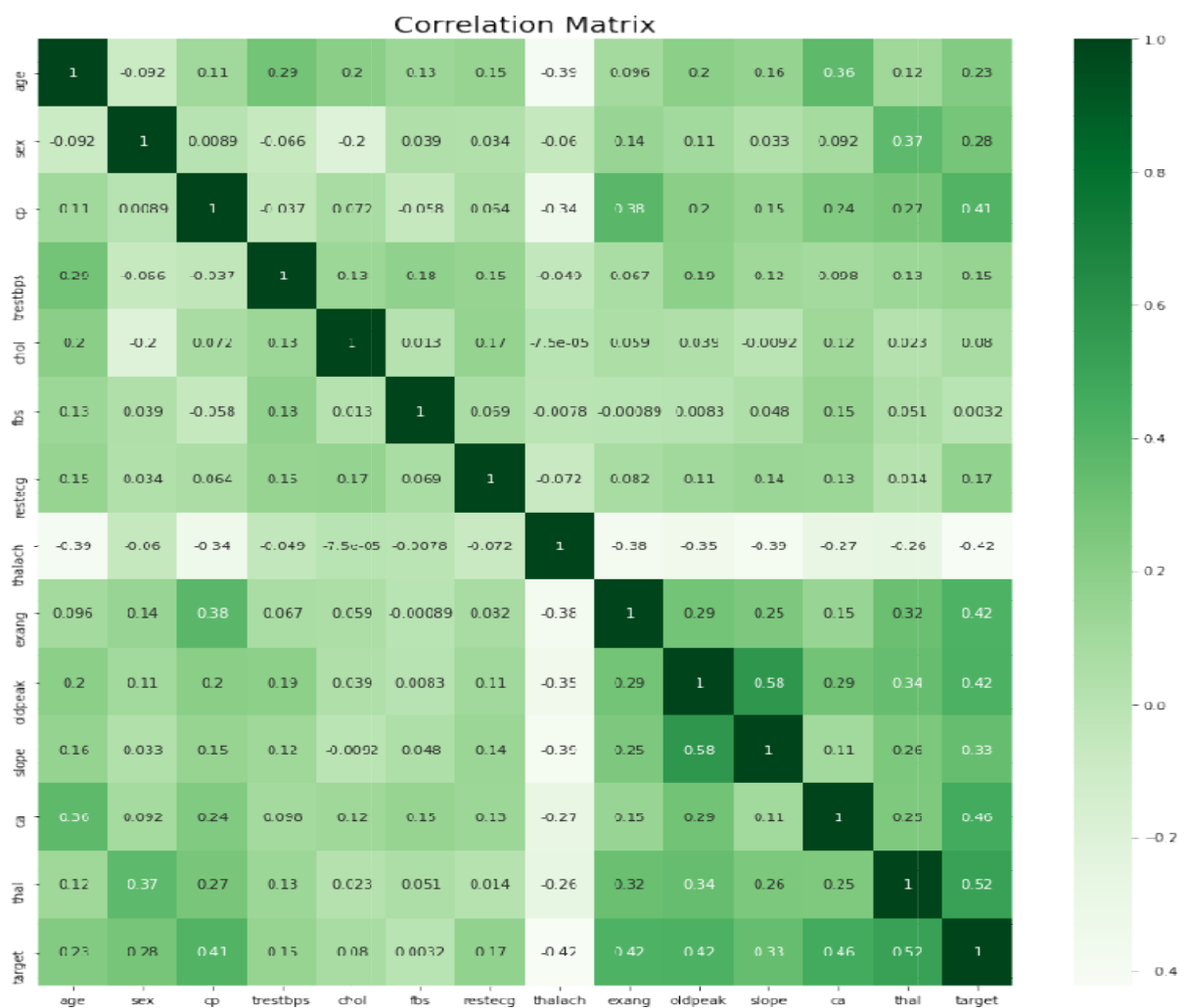


Figure: 8.6 Correlation Matrix

Pre-processing of data

Data pre-processing is an important step for the creation of a machine learning model. Initially, data may not be clean or in the required format for the model which can cause misleading outcomes. In pre-processing of data, we transform data into our required format. It is used to deal with noises, duplicates, and missing values of the dataset. Data pre-processing has the activities like importing datasets, splitting datasets, attribute scaling, etc. Preprocessing of data is required for improving the accuracy of the model.

Prediction of Heart Disease

Once you have gone through collecting data, preparing the data, selecting the model, and training and evaluating the model and tuning the attributes, it is time to use various machine learning algorithms like Support Vector Machine, Logistic Regression, Decision Tree, and Random Forest for classification. Comparative analysis is performed among algorithms and the algorithm that gives the highest accuracy is used for heart disease prediction.

Dataset Details

Of the 76 attributes available in the dataset, 14 attributes are considered for the prediction of the output.

- Heart Disease Cleveland UCI: <https://www.kaggle.com/chemngs/heart-disease-cleveland-uci>

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	condition
0	69	1	0	160	234	1	2	131	0	0.1	1	1	0	0
1	69	0	0	140	239	0	0	151	0	1.8	0	2	0	0
2	66	0	0	150	226	0	0	114	0	2.6	2	0	0	0
3	65	1	0	138	282	1	2	174	0	1.4	1	1	0	1
4	64	1	0	110	211	0	2	144	1	1.8	1	0	0	0
5	64	1	0	170	227	0	2	155	0	0.6	1	0	2	0
6	63	1	0	145	233	1	2	150	0	2.3	2	0	1	0
7	61	1	0	134	234	0	0	145	0	2.6	1	2	0	1
8	60	0	0	150	240	0	0	171	0	0.9	0	0	0	0
9	59	1	0	178	270	0	2	145	0	4.2	2	0	2	0

Figure: 8.7 Dataset Attributes

Screenshots

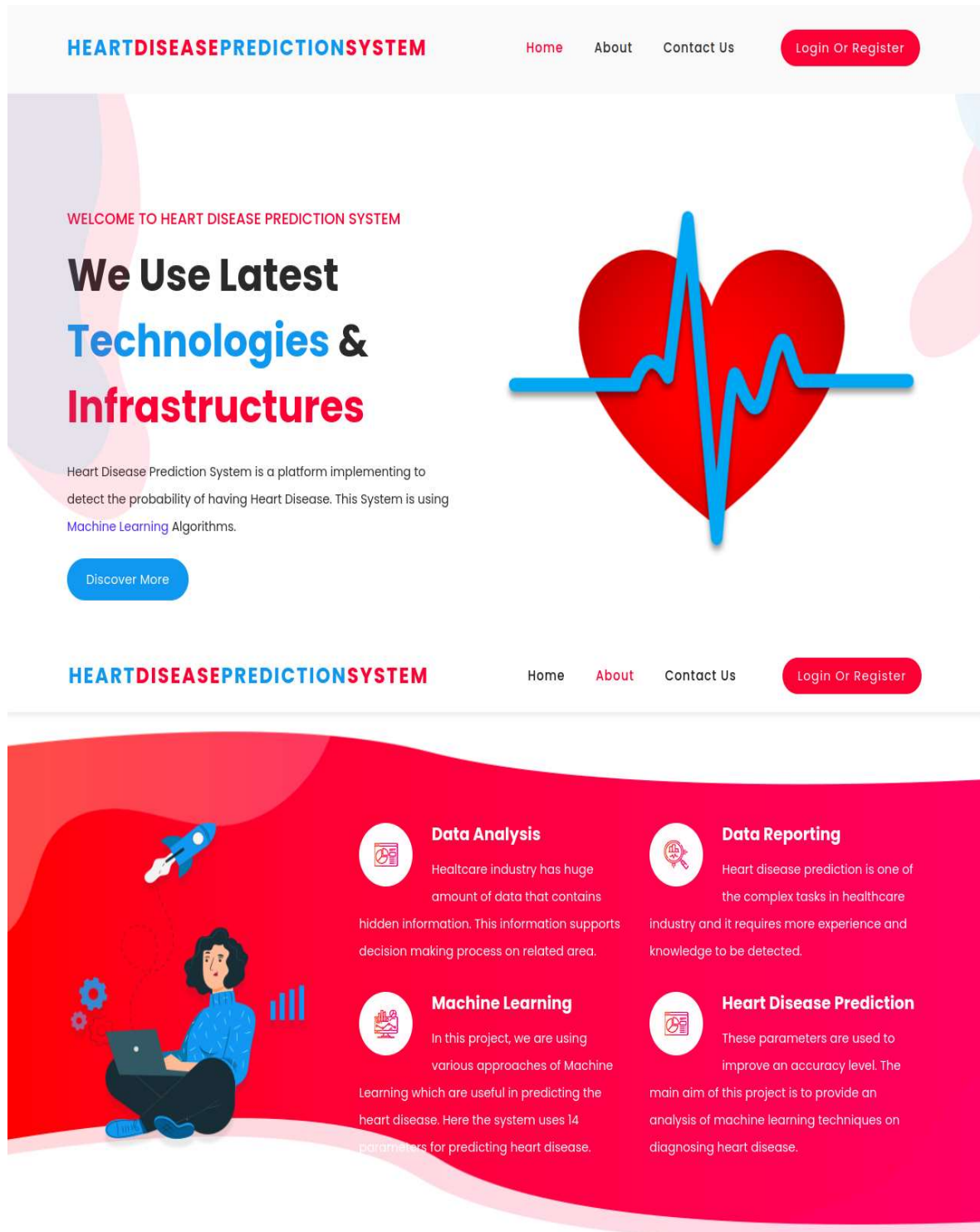


Figure: 8.8 Index and About Pages

The figure displays two screenshots of a web application interface for a heart disease prediction system. Both screenshots feature a header with the text "HEARTDISEASEPREDICTIONSYSYSTEM" and a "Home" button. The background is a vibrant red-to-pink gradient.

The top screenshot shows the "Authentication" form, which includes fields for "Email" and "Password", a "Log In" button, and a link to "Sign Up" for users who don't have an account.

The bottom screenshot shows the "Registration" form, which includes fields for "Username", "Email", "Password", and "Confirm Password", a "Sign Up" button, and a link to "Log In" for users who already have an account.

Figure: 8.9 Login and Registration Forms

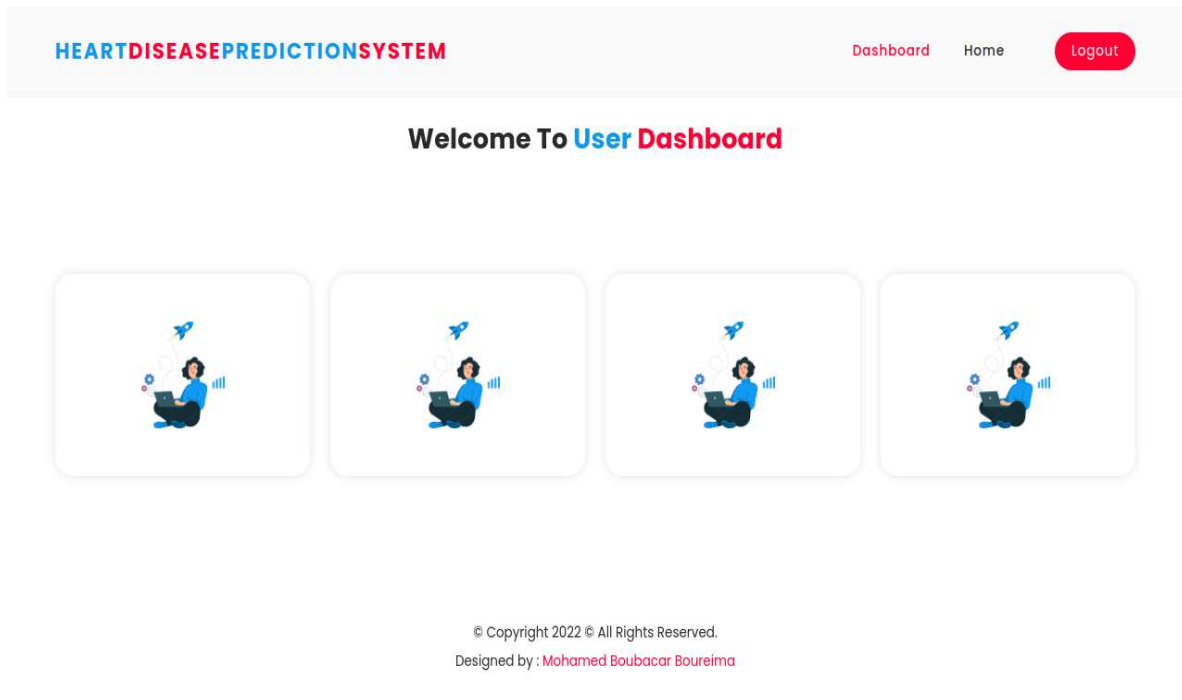



Figure: 8.10 Home Page

The screenshot displays the 'Import a new dataset' form within the same system interface. The form includes input fields for 'File Name' and 'Description', an 'Upload Dataset File' section with a 'Choose file' button and 'No file chosen' text, and 'Cancel' and 'Submit' buttons at the bottom. The footer contains the same copyright and design credit information as the previous figure.

Figure: 8.11 Import Dataset Form

HEARTDISEASEPREDICTIONSYSYSTEM
Dashboard
Home
Logout



Dear Admin, you can update your account information here!

Username
Admin

Email
admin@gmail.com

Upload Profile Picture No file chosen

© Copyright 2022 © All Rights Reserved.
Designed by : Mohamed Boubacar Boureima

Figure: 8.12 User profile

HEARTDISEASEPREDICTIONSYSYSTEM
Dashboard
Home
Logout

See What Our Heart Disease System Predicts

Age :
Please enter the age

Gender :
--- Please select an option ---

Chest Pain :
--- Please select an option ---

Resting Blood Pressure :
Please enter a number in range [94-200] mmHg

Serum Cholesterol :
A number in range [120-504] mg/dL

Fasting Blood Sugar :
--- Please select an option ---

Resting ECG :
--- Please select an option ---

Heart Rate :
Please enter a number in range [71-202] bpm

ST Depression :
Please enter a number typically in range [0-0.2]

Exercise-Induced Angina :
--- Please select an option ---

Slope of the peak exercise ST segment :
--- Please select an option ---

Number of Major Vessels :
Please enter a number typically in range [0-4]

Thalassemia :
--- Please select an option ---

© Copyright 2022 © All Rights Reserved.
Designed by : Mohamed Boubacar Boureima

Figure: 8.13 Prediction Form

HEARTDISEASEPREDICTIONSYSYSTEM

DashboardHomeLogout

See What Our *Heart Disease System* Predicts

Age :
54

Chest Pain :
Atypical Angina

Serum Cholesterol :
264

Resting ECG :
Having ST-T wave abnormality

ST Depression :
2

Slope of the peak exercise ST segment :
Upsloping

Thalassemia :
Fixed Defect

Gender :
Male

Resting Blood Pressure :
150

Fasting Blood Sugar :
Greater than 120 mg/dl

Heart Rate :
100

Exercise-Induced Angina :
No

Number of Major Vessels :
1

Click Here To Check The Prediction.

© Copyright 2022 © All Rights Reserved.
Designed by : Mohamed Boubacar Boureima

HEARTDISEASEPREDICTIONSYSYSTEM

DashboardHomeLogout

Prediction : Oops ! You Have Chances Of Heart Disease.

(Note : This model is 82.67% accurate)

© Copyright 2022 © All Rights Reserved.
Designed by : Mohamed Boubacar Boureima

Figure: 8.14 Heart Disease Prediction

CHAPTER 9

Testing:

Software Testing is a process of executing the application with intent to find any software bugs. It is used to check whether the application met its expectations and all the functionalities of the application are working. The final goal of testing is to check whether the application is behaving in the way it is supposed to under specified conditions. All aspects of the code are examined to check the quality of the application. The primary purpose of testing is to detect software failures so that defects may be uncovered and corrected. The test cases are designed in such way that scope of finding the bugs is maximum.

Testing Levels

There are various testing levels based on the specificity of the test.

- **Unit testing:** Unit testing refers to tests conducted on a section of code in order to verify the functionality of that piece of code. This is done at the function level.
- **Integration Testing:** Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Its primary purpose is to expose the defects associated with the interfacing of modules.
- **System Testing:** System testing tests a completely integrated system to verify that the system meets its requirements.
- **Acceptance testing:** Acceptance testing tests the readiness of application, satisfying all requirements.
- **Performance testing:** Performance testing is the process of determining the speed or effectiveness of a computer, network, software program or devices such as response time or millions of instructions per second etc.

System Test Cases

A test case is a set of test data, preconditions, expected results and post conditions, developed for a test scenario to verify compliance against a specific requirement. I have designed and executed a few test cases to check if the project meets the functional requirements.

Test Objectives

Test Objectives: Navigation from Index page to Login or Register page

TEST CONDITION	INPUT SPECIFICATION	OUTPUT SPECIFICATION	PASS/FAIL
The user is currently on the index page	The user clicks on the login or register button	Directs to the login or register page	PASS

Table 1: The test case from the Index page to login page

Test Objectives: Navigation from Login page to Home page

TEST CONDITION	INPUT SPECIFICATION	OUTPUT SPECIFICATION	PASS/FAIL
The user is currently on the login page	The user enters credentials and clicks the login button	Directs to the home page of the particular user	PASS

Table 2: The test case from login page to home page

Test Objectives: Navigation from the Home page to Prediction page

TEST CONDITION	INPUT SPECIFICATION	OUTPUT SPECIFICATION	PASS/FAIL
The user is currently on the home page	The user enters values and clicks on Click Here and Check The Prediction button	Directs to Heart Disease Prediction page	PASS

Table 3: The test case form home page to prediction page

CHAPTER 10

Conclusion

In the conventional hospital-centric healthcare system, patients are often tethered to several various test concurrently. The heart disease prediction system would assist the hospitals to get to know who has chances to have heart disease and who do not. In this project, we develop an inexpensive but flexible and scalable heart disease prediction system that integrates the capabilities of AI and Machine Learning Techniques for cardiovascular disease prediction of a patient's health status.

Through experimental analysis, we have shown that the proposed framework is scalable and reliable with high classification accuracy. We believe that the proposed work can address the healthcare spending challenges by substantially reducing inefficiency and saving valuable lives. We are currently implementing the proposed algorithm and testing it in a real-life environment.

Future Enhancements

For future work, we plan to increase the size of the dataset and analyze the dataset by referring to various respected medical sources. Also, we can add some new features of deep learning with various other optimizations that can be used and more promising results can be achieved.

Combination of multiple machine learning algorithms with a large number of various datasets and various optimization techniques can also be used so that the evaluation results can again be increased. More different ways of normalizing the data can be used and the result can be compared and in future studies, more ways could be found where we could integrate heart-disease-trained machine learning and deep learning models with certain multimedia for the ease of patients and doctors.

References

- [1] Abbasi, J. (2022). The COVID heart—One year after SARS-CoV-2 infection: Patients have an array of increased cardiovascular risks. *JAMA*, 327(12), 1113-1114.
- [2] A. Chauhan, A. Jain, P. Sharma and V. Deep, "Heart Disease Prediction using Evolutionary Rule Learning," 2018 4th International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, 2018, pp. 1-4. doi: 10.1109/CICT.2018.8480271
- [3] A. Dewan and M. Sharma, "Prediction of heart disease using a hybrid technique in data mining classification," 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, 2015, pp. 704-706.
- [4] Atlanta, GA: Centers for Disease Control and Prevention, US Dept of Health and Human Services. Hahad, O., Kröller-Schön, S., Daiber, A., & Münzel, T. (2019). The cardiovascular effects of noise. . *Dtsch Arztebl Int*, 116, 245–50. DOI: 10.3238/arztebl.2019.0245. WHO (n.d). Cardiovascular diseases.
- [5] Best practices for cardiovascular disease prevention programs: A guide to effective health care system interventions and community programs linked to clinical services.
- [6] Cardiovascular Diseases (Cvds), "World health organization,"
- [7] C. B. C. Latha, S. C. Jeeva, and S. Carolin Jeeva, "Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques," *Informatics in Medicine Unlocked*, vol. 16, Article ID 100203, 2019.
- [8] Purushottam, K. Saxena and R. Sharma, "Efficient heart disease prediction system using decision tree," *International Conference on Computing, Communication & Automation*, Noida, 2015, pp. 72-77. doi: 10.1109/CCAA.2015.7148346

Bibliography

- <https://jamanetwork.com/>.
- https://www.who.int/health-topics/cardiovascular-diseases#tab=tab_1
- <https://medicalxpress.com/news/2019-05-machine-humans-death-heart.html>
- [https://www.who.int/news-room/fact%20sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/news-room/fact%20sheets/detail/cardiovascular-diseases-(cvds)).
- <https://towardsdatascience.com/preventing-deaths-from-heart-disease-using-machinelearning-c4f8dba250c6>
- <https://towardsdatascience.com/preventing-deaths-from-heart-disease-using-machine-learning-c4f8dba250c6?gi=a147b81ce98f>