CardioPredict

Submitted by

Muhammad Kashir Khan (BSSE-Sp20-BSSE-008-A)

Muhammad Daniyal Khan (BSSE-Sp20-BSSE-005-A)

Session

Spring 2020

Supervised by

Ms. Ayesha Kiran

Department of Software Engineering

Lahore Garrison University

Lahore

CardioPredict

A project submitted to the

Department of Software Engineering

In

Partial Fulfillment of the Requirements for the

Bachelor's Degree in Software Engineering

By

Muhammad Kashir Khan

Muhammad Daniyal Khan

Supervisor

Ms. Ayesha Kiran

_____

Lecturer

Department of Software Engineering

Chairperson

Dr. Waqar Azeem

_____

Head of Department

Department of Software Engineering

COPYRIGHTS

This is to certify that the project titled "**CardioPredict"** is the genuine work carried out by **Muhammad Daniyal Khan** & **Muhammad Kashir Khan**, students of Software Engineering Department, Lahore Garrison University, Lahore. During the academic year 2023-2024, in partial fulfilment of the requirements for the award of the degree of Bachelor of Science in Software 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.

Muhammad Kashir Khan _____

Muhammad Daniyal Khan _____

DECLARATION

This is to declare that the project entitled "**CardioPredict**" is an original work done by undersigned, in partial fulfilment of the requirements for the degree "Bachelor of Science in Software Engineering" at Software Engineering Department, Lahore Garrison University, Lahore.

All the analysis, design and system development have been accomplished by the undersigned. Moreover, this project has not been submitted to any other college or university.

Muhammad Kashir Khan _____

Muhammad Daniyal Khan _____

ACKNOWLEDGEMENTS

We extend our deepest appreciation towards our exceptional supervisor, **Ms. Ayesha Kiran**, whose unwavering assistance, and mentorship have been instrumental in the successful completion of this project. Her wealth of knowledge and expertise provided invaluable insights that considerably enhanced the depth and quality of our work. We are truly grateful for her dedication in guiding us through the complexities of the research process, offering constructive feedback, and encouraging critical thinking. Furthermore, her commitment to foster a positive and promising learning environment has played a significant role in the development of the project. Her encouragement to explore innovative ideas and her confidence in our capabilities have boosted our poise and motivation. We want to express our gratitude for her patience, understanding, and the sincere interest she has shown in this voyage. We feel honored to have had the opportunity to work under the supervision of such a keen and inspiring supervisor. The knowledge and skills we have extended during this cooperation will definitely leave an enduring impact on our academic and professional pursuits. We have highest regards and esteemed honor for **Ms. Ayesha Kiran**, for her requisite involvement, mentorship, and unwavering support throughout this journey.

DEDICATION

This research project is dedicated to our dearest parents and family who always support us as the foundation of our academic activities. Their enduring trust in our abilities, steadfast encouragement during difficult times, and their generous resourcing have been instrumental in our journey. We express our deepest gratitude for your unwavering commitment to our success. Your unwavering commitment to our success created the foundation needed to complete this research project. This achievement is tribute to the values instilled in us and the endless support that continues to shape our aspirations. With the deepest gratitude, we dedicate this work to our parents and beloved family, whose ever-present presence is our greatest inspiration.

Table of Contents

List of Tables

List of Figures

List of Abbreviation

Abbreviation	Description
CVD	Cardiovascular Disease, related to the disorders of heart and blood vessels.
UCI	University of California, Irvine. A machine learning repository for useful resources and datasets.
ReLU-NNR	An activation function used in the layers of the neural networks.
KNN	K-Nearest Neighbor (A supervised learning technique).
GNB	Gaussian Naïve-Bayes (A supervised learning technique).
SVM	Support Vector Machine (A supervised learning technique).
BMI	Body-Mass Index is measure derived from the mass and the height of a person.
IDE	An integrated development environment which facilitates a comprehensive development.
MVC	Model-View Controller is an architectural approach separating system into three logical components.

Abstract

Cardiovascular diseases (CVD) pose a significant global health challenge, necessitating accurate and efficient prediction models for timely intervention. This study addresses the pressing need for an effective predictive system by employing three machine learning algorithms: Logistic Regression, Random Forest Classifier and Decision Tree Classifier. The primary objective is to develop a robust CVD risk prediction system, with a particular focus on assessing the feasibility of deploying a web application using the superior accuracy of the random forest algorithm. The challenges in CVD risk prediction include handling diverse and high-dimensional clinical data by identifying relevant features. To overcome these challenges, logistic regression, random forest, and decision tree classifier models are trained and evaluated using a comprehensive dataset. Each algorithm's performance is analyzed based on metrics such as accuracy, recall, precision, and f1-score. The proposed solution involves implementing a web application for CVD risk prediction, with the random forest model selected as the core algorithm due to its superior accuracy in comparison to logistic regression and decision tree classifier. The web application provides a user-friendly interface for individuals to input their health parameters and obtain a personalized risk prediction.

In conclusion, the random forest-based web application demonstrates promising results in CVD risk prediction. Future directions include refining the model with additional features and incorporating real-time data for continuous model improvement. The success of this approach lays the foundation for a practical and accessible tool that can aid individuals and healthcare professionals in proactive CVD risk management.

Chapter 1

Introduction

1.1 Overview

Cardiovascular disease primarily occurs due to the blockage of arteries and is known by various names such as heart disease and arterial hypertension. Globally, approximately 26 million people are affected by heart disease. The concerning aspect is that this number is expected to rise significantly in the coming years unless efficient precautions are taken. In addition to adopting a healthy lifestyle and controlling diet, timely diagnosis and comprehensive analysis are crucial factors that can ultimately contribute to saving lives. Therefore, this paper takes a modest step toward enhancing the lives of cardiovascular risk patients by proposing a method to improve the risk diagnosis of patients based on their medical parameters [1].

Patients frequently undergo numerous examinations, resulting in heightened physical exertion, time investment, and additional financial costs. Previous studies have identified prevalent factors associated with cardiovascular disease, such as age, tobacco usage, excessive glucose, high cholesterol and being overweight. Common indications of the condition may include discomfort in the arms and chest. Therefore, this system attempts to improve the performance of the classifiers by doing experiments using multiple machine-learning (Logistic Regression, Random Forest, and Decision Tree Classifier) models and different optimizations to make better use of the dataset [2].

The project aims to develop an accurate and accessible cardiovascular disease (CVD) prediction system using machine learning algorithms, specifically logistic regression, random forest, and decision tree classifier. The primary goal is to address the unmet need for a reliable predictive tool that can assist healthcare professionals in assessing CVD risk [3]. The proposed solution involves the creation of a web application centered on the random forest algorithm due to its superior accuracy.

1.2 Related Work

Cardiovascular Disease Risk Prediction is an area where the implementation of the machine learning models can be found to accurately predict the CVD risk based on the patients' clinical parameters. cardiovascular failure occurs when the heart becomes too weak to adequately pump

blood throughout the body. For many people, this is their fate. About 2% of people in affluent countries suffer from cardio failure, and that number rises to 6% - 10% among those aged 60 and beyond.

Standard clinical risk factors, such as hypertension and diabetes are used for most applications. High-Risk areas for deaths due to cardiovascular disease have also been the subject of research in Brazil. Moreover, half of the people analyzed in this study had multiple difficult-to-treat diseases that increased their risk of dying.

Several research took place in different countries, and they concluded high accuracy of the disease being rightly predicted by the models. They concluded that machine learning can be used to anticipate cardiac emergencies. The subfield of artificial intelligence known as "machine learning" focuses on the process of teaching a computer to learn new things on its own. They are under constant pressure because of the prevalence of heart attacks among their patients. It is crucial to find ways to reduce the number of deaths caused by heart attacks. Machine learning plays an important role in this study. Scientists have developed a way to reliably predict cardiovascular disease in patients [4].

Researchers at UCI use the Cleveland Heartland Registry to screen for and validate heart disease in patients. This dataset contains the following types of information. The optimum algorithm for large-scale classification will then be suggested. Data mining can be used to find correlations between patient data and heart disease Risk factors to get more accurate diagnoses for patients [5] [6].

In a study in University of Pennsylvania, they used logistic regression, random forest, and artificial neural network with the RELU-activated neural network (NNR), K-nearest neighbors (KNN), and (Naive Bayes) GNB selection methods to predict the probability of cardiac sickness [7]. The model was developed in Jupyter Notebook with Flask and Python using the Kaggle dataset. To evaluate the model's effectiveness in each of these contexts, we run tests on a wide range of parameters. Another investigation revealed that the test was accurate 90% of the time, precise 91% of the time, and accurate and precise 91% of the time. Since ensemble modelling is more precise than utilizing individual models, these findings demonstrate that it contributes to saving lives [8].

In another study conducted in a German university, The University of Paderborn, it is explained, how a more accurate model for predicting survival in people with heart failure can be developed. Two hundred nine patients with cardio failure are used to evaluate a survival

prediction model. To determine the best ensemble tree method and feature selection approach, a model optimization is utilized. Accuracy on five of the twelve variables increased from 79.5 to 85.1 using the extra tree classifier thanks to the use of cross-validation.

Table 1. Related Studies

Authors	Novel Approach	Best Accuracy	Dataset
Maiga et al., 2019	- Random Forest - Naive Bayes - Logistic Regression - KNN	70%	Kaggle cardiovascular disease dataset (70,000 patients, 12 attributes)
Shorewall, 2021	Stack of KNN, Random Forest, & SVM with Logistic Regression as metaclassifier	75.1%	Kaggle cardiovascular disease dataset (70,000 patients, 12 attributes)
Khan & Mondal, 2020	Cross-Validation with neural network	71.82%	Kaggle cardiovascular disease dataset (70,000 patients, 12 attributes)
	Cross-Validation with Logistic Regression (solver:lgfgs), (k = 30)	72.72%	Kaggle cardiovascular disease dataset 1 (462 patients, 12 attributes)
	Cross-Validation with linear SVM (k = 10)	72.72%	Kaggle cardiovascular disease dataset (70,000 patients, 12 attributes)
Waigi et al., 2020	Decision Tree	72.77%	Kaggle cardiovascular disease dataset (70,000 patients, 12 attributes)
Our & ElSeddawy, 2021	Random Forest	89.01%	UCI cardiovascular dataset (303 patients, 14 attributes)

1.3 Data Overview

The dataset used for this system was collected from Kaggle platform, the dataset is also known as Heart dataset. It is an open dataset, having number of attributes, suggested by different scholars that selected 18 attributes are most useful to predict the cardiovascular disease risk in a patient. In addition, the file contains the record of 319,795 patients. The complete description of each attribute and the number of values for each attribute is shown in the table below:

Table 2. Dataset Details

Sr. No.	Attribute Description	Distinct Values
1.	HeartDisease	Yes or No
2.	BMI	Continuous Values
3.	Smoking	Yes or No
4.	AlcoholDrinking	Yes or No
5.	Stroke	Yes or No
6.	PhysicalHealth	0 - 30
7.	MentalHealth	0 - 30
8.	DiffWalking	Yes or No
9.	Sex	Male or Female
10.	AgeCategory	24 – 80+
11.	Race	White, Hispanic, Asian, Black
12.	Diabetic	Yes, No, Borderline, High

13.	PhysicalActivity	Yes or No
14.	GenHealth	Good, V. Good, Fair, Excellent and Poor
15.	SleepTime	Continuous Values
16.	Asthma	Yes or No
17.	KidneyDisease	Yes or No
18.	SkinCancer	Yes or No

This above given table demonstrates the columns of the dataset that is to be used in the research as follows. On this raw data, feature selection was applied, and the relevant features were selected to be tried as features in the models.

The feature engineering was applied and resulted in the extraction of some of these columns. The following are the visual description of the columns of the dataset [9].

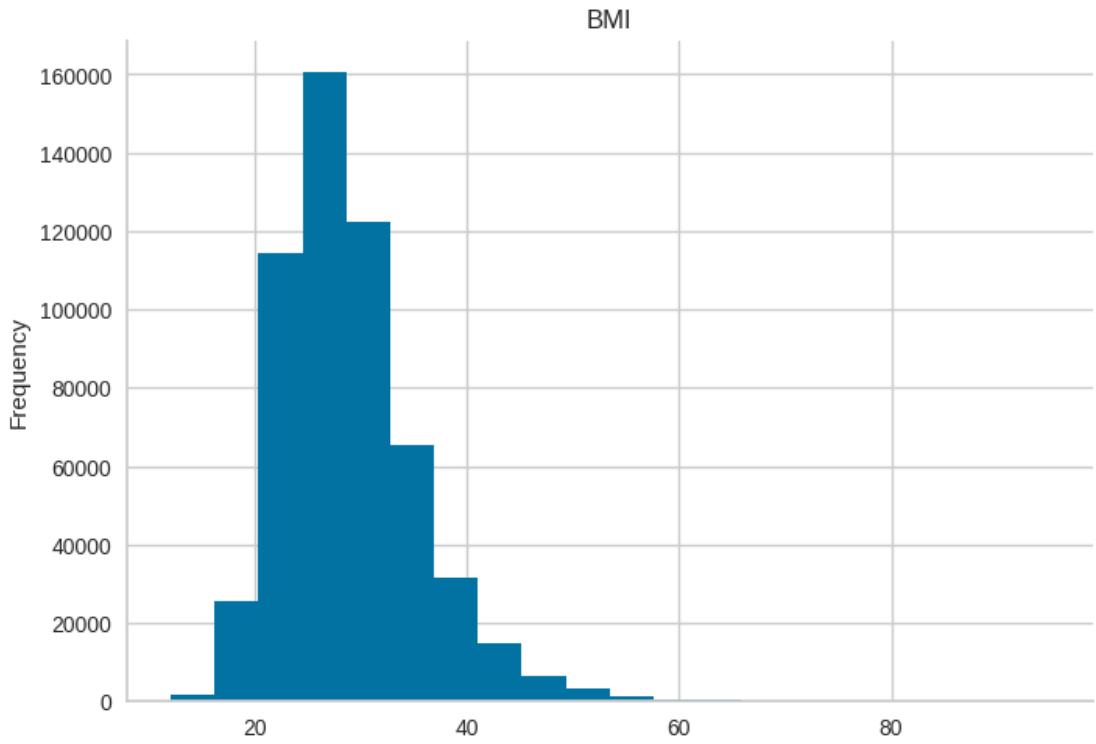


Figure 1. Body Mass Index Data

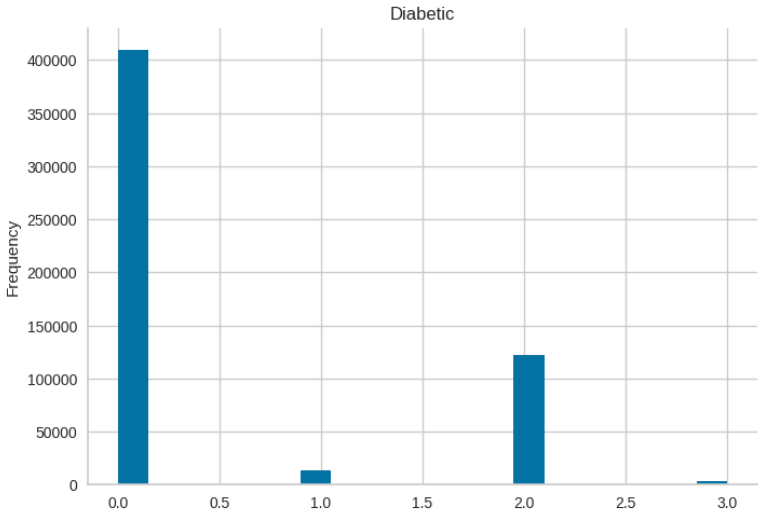


Figure 2. Diabetes Data

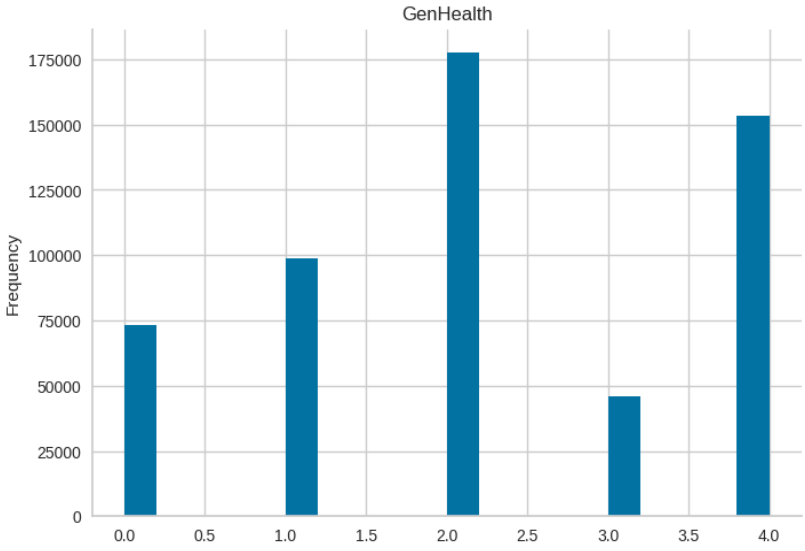


Figure 3. General Health Data

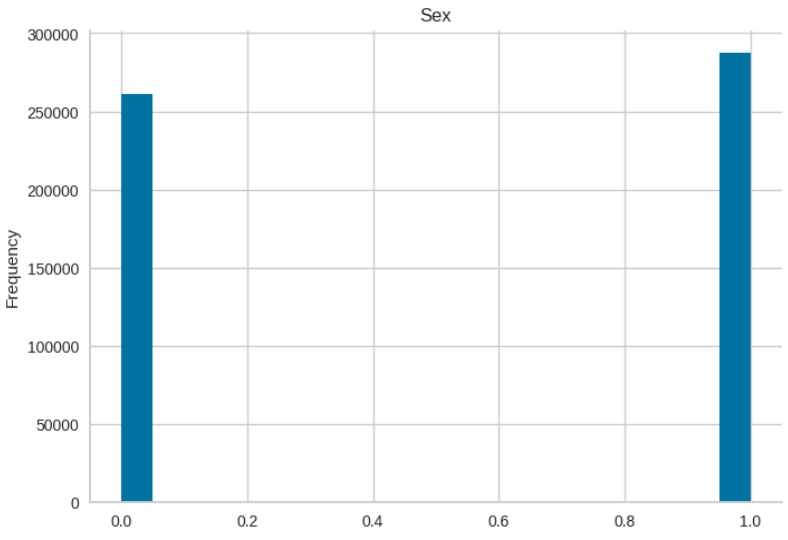


Figure 4. Gender Data

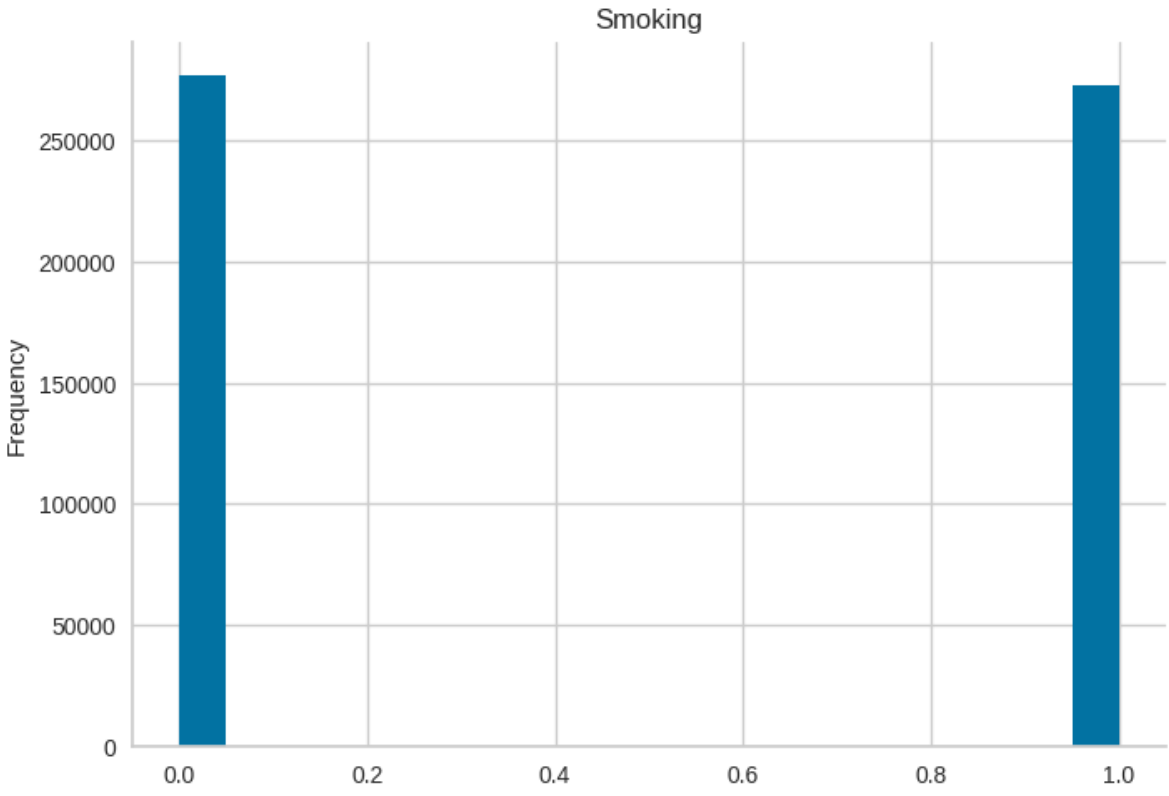


Figure 5. Smoking Data

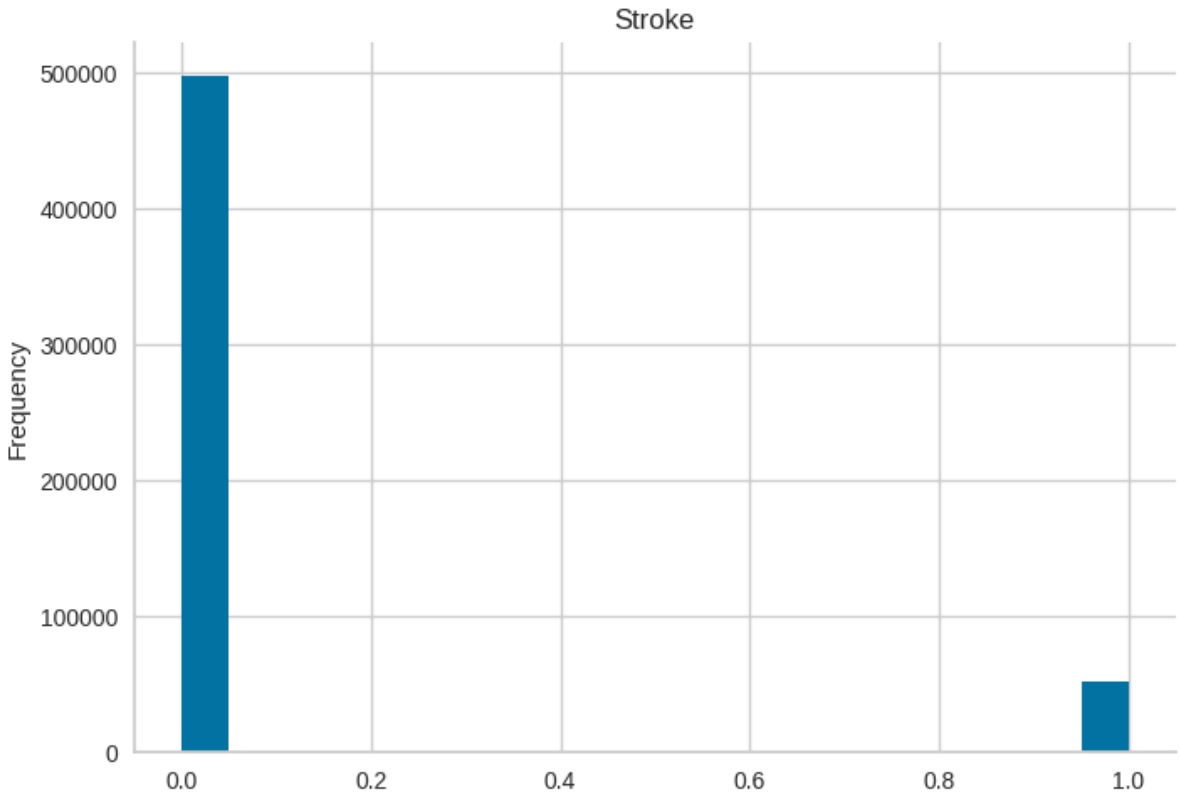


Figure 6. Stroke History Data

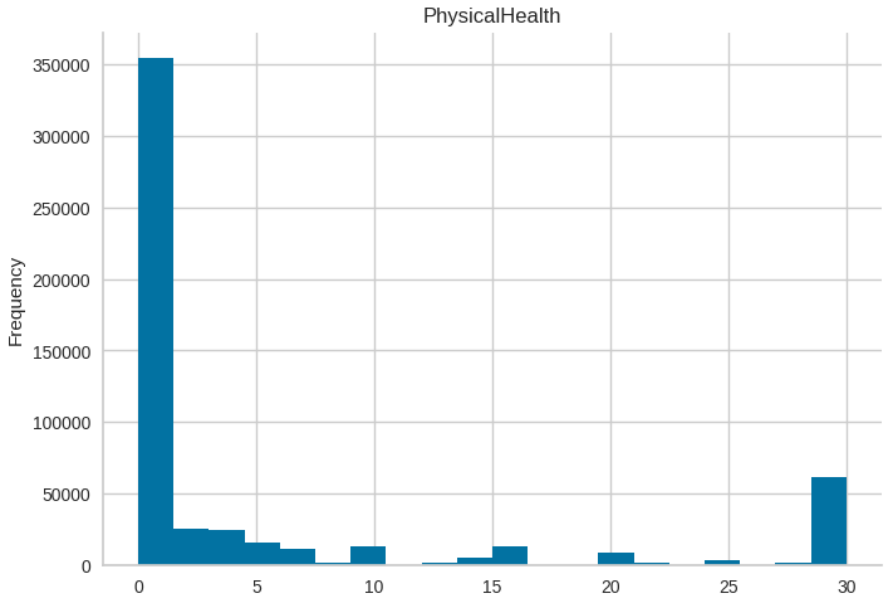


Figure 7. Physical Health Data

1.4 Structure of the System

Algorithmic Comparison: The project involves training and evaluating three machine learning algorithms; logistic regression, random forest, and decision tree classifier using a comprehensive dataset that includes diverse clinical parameters.

Web Application Development: A user-friendly web application is designed to facilitate CVD risk prediction. The random forest model is integrated into the application, allowing users to input their health parameters and receive personalized risk predictions.

Performance Metrics: The solution incorporates rigorous evaluation metrics, including accuracy, recall, precision, and f1-score to assess the predictive performance of each algorithm.

1.5 Unmet Need or Problem Addressed

The existing gap in CVD risk prediction tools lies in the need for a solution that is not only accurate but also user-friendly and accessible. Traditional risk assessment methods may lack the precision required for personalized predictions. The proposed system caters to this unmet need by offering a comprehensive approach that combines machine learning algorithms and a web application interface. This system is designed to benefit both individuals seeking personalized risk assessments and healthcare professionals aiming for an efficient and reliable CVD risk management tool.

1.6 Target Audience

The system caters to a broad audience, including individuals concerned about their cardiovascular health and healthcare professionals involved in preventive care. Individuals can use the web application for a quick and personalized risk assessment, while healthcare professionals can incorporate the tool into their practice to enhance the accuracy and efficiency of CVD risk management. By addressing the unmet need for a reliable and accessible CVD risk prediction system, the project aims to empower individuals and healthcare professionals in making informed decisions for better cardiovascular health outcomes.

1.7 Structure of the Report

After the first chapter, the remaining chapters are structured as explained below.

Chapter 2: Latter to the former, this chapter encompasses the problem definition, its description and the challenges that could occur during the process.

Chapter 3: In this chapter, the functional and non-functional requirements are outlined. Its ought to serve as an explicit and comprehensive section guiding the development process.

Chapter 4: In this chapter, the methodological design processes of the research that has been conducted, are explained.

Chapter 5: This chapter is an outline of the overall structure and organization of the proposed system, the components, their interactions, and dependencies through visual elements. Predominantly, it is the blueprint of the basic layout and functionality.

Chapter 6: In this phase, the system is rendered into actual substance lead from the outline. This phase includes the evaluation and testing of performance and guaranteeing that it is in agreement with specified requirements.

Chapter 7: This chapter presents and showcases the outcomes of the implementation through metrics and observations. The analysis is done, leading to insights.

Chapter 8: In this chapter, the findings and outcomes are summarized, highlighting achievements, and addressing limitations. The potential enhancements are suggested, areas of further research and advancements are drawn.

Chapter 2

Problem Definition

2.1 Problem Description

Cardiovascular diseases continue to be a leading cause of morbidity and mortality globally. Despite advancements in medical science, predicting and preventing CVD risks remains a complex challenge. Traditional risk assessment methods often rely on simplistic models that may not capture the complexity of various clinical parameters. This poses a significant problem as it can lead to inaccurate risk predictions, resulting in delayed interventions or unnecessary treatments [10].

Moreover, the existing tools may lack the adaptability required for diverse patient populations and may not provide the granularity needed for personalized risk assessments. There is a pressing need for a more sophisticated and accurate CVD risk prediction system that considers the complexity of individual health profiles.

The problem statement encompasses several key challenges:

2.1.1 Data Complexity:

Clinical data related to cardiovascular health is often multifaceted and high-dimensional. Traditional methods may struggle to extract relevant patterns from such complexity.

2.1.2 Feature Relevance:

Identifying the most relevant features for CVD risk prediction is a challenge. Not all clinical parameters contribute equally to the risk [10], and the system must recognize which factors are most indicative. The following correlation matrix (Figure. 8) shows the relationship of the features that were selected during the feature engineering process [11].

2.1.3 User Accessibility:

Existing tools may lack a user-friendly interface, hindering widespread adoption. There is a need for a system that is accessible to individuals with varying levels of health literacy.

2.1.4 Accuracy and Generalization:

Achieving a balance between high accuracy and generalization across diverse patient populations is critical. The system must perform well across different datasets to ensure its applicability in real-world scenarios.

The project aims to address these challenges by leveraging machine learning algorithms, specifically logistic regression, random forest, and decision tree classifier, to develop a more accurate and accessible CVD risk prediction system. The selection of the random forest algorithm for the web application is grounded in its superior accuracy, providing a solution that not only caters to the complexity of clinical data but also meets the practical needs of users seeking reliable and personalized risk assessments.

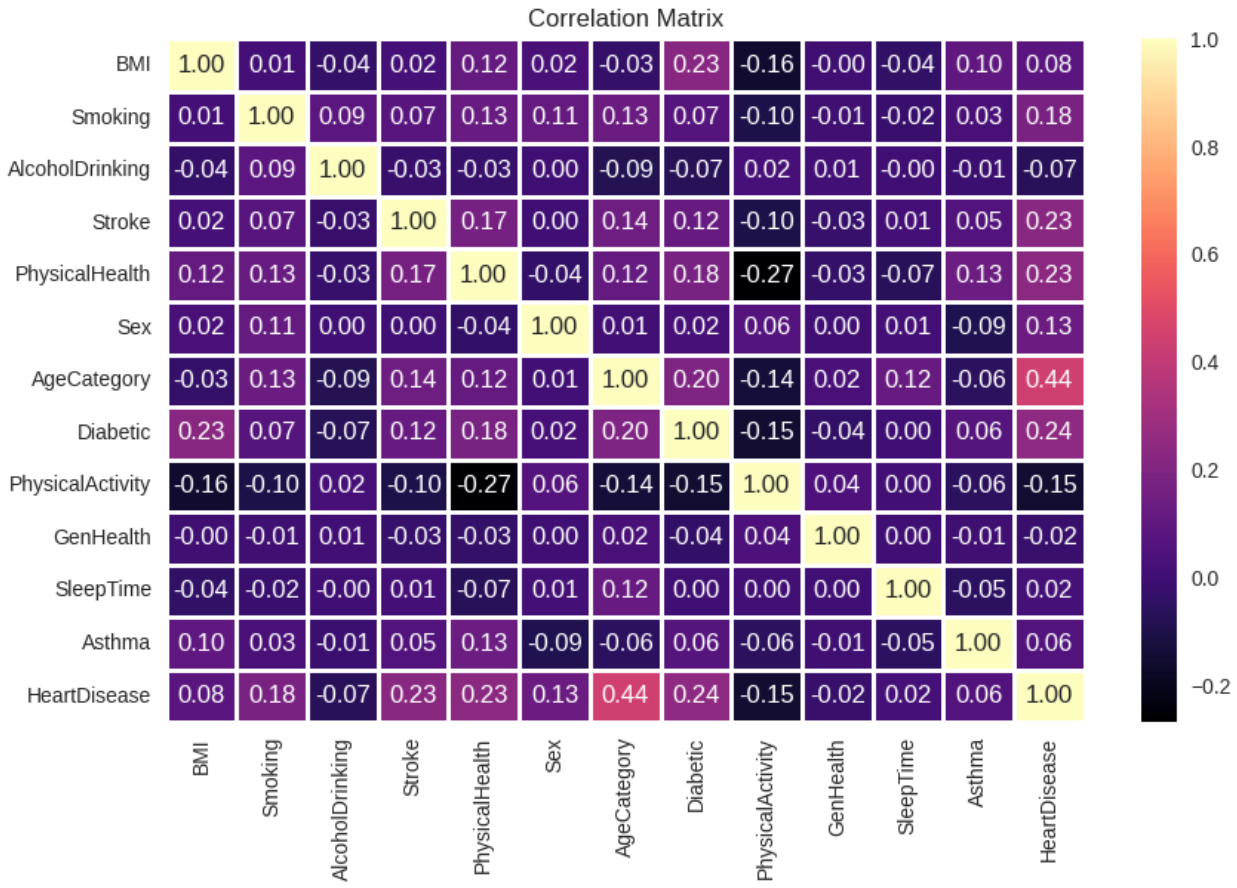


Figure 8. Correlation Matrix

Chapter 3

Software Requirement Specification

3.1 Purpose

The purpose of the cardiovascular disease risk prediction system is to provide a reliable and accessible tool for predicting individualized risk of cardiovascular diseases. The system aims to leverage machine learning algorithms, including logistic regression, random forest, and decision tree classifier, to analyze diverse clinical parameters and generate personalized risk assessments. The primary goals are:

3.1.1 Accuracy:

Develop a predictive model that accurately assesses an individual's risk of developing cardiovascular diseases based on their health parameters.

3.1.2 Accessibility:

Create a user-friendly web application interface that allows individuals to easily input their health data and receive personalized risk predictions.

3.1.3 Usability:

Provide healthcare professionals with a practical tool for preventive care, enhancing the accuracy and efficiency of cardiovascular risk management.

3.1.4 Innovation:

Integrate state-of-the-art machine learning algorithms, with a focus on the random forest model, to ensure the system's predictive capabilities align with the complexity of clinical data.

3.2 Product Scope

The Cardiovascular Disease Risk Prediction System encompasses the following features and functionalities:

3.2.1 Web Application:

A secure and user-friendly web interface allowing individuals to input their health parameters. User authentication for personalized access to risk assessments.

3.2.2 Machine Learning Models:

Implementation of logistic regression, random forest, and decision tree classifier models for CVD risk prediction. The random forest model chosen as the primary algorithm for the web application due to its superior accuracy.

3.2.3 Risk Assessment:

Generation of personalized CVD risk assessments based on input health parameters. Presentation of risk scores to users.

3.2.4 Scalability:

The system designed to handle a diverse range of health data and user inputs for scalability and adaptability.

3.2.5 Performance:

Efficient response times to user inputs for a seamless and responsive user experience. Capability to handle a significant volume of data for accurate predictions. The product scope does not include diagnostic capabilities, and the system should not replace professional medical advice. It is intended to serve as a supplementary tool for individuals and healthcare professionals in assessing and managing cardiovascular disease risk. Future enhancements may include real-time data integration and integration with electronic health records for a more comprehensive healthcare solution.

3.3 Objectives and Goals

The Cardiovascular Disease (CVD) Prediction System is designed with a set of clear objectives and goals to address the critical need for accurate risk assessment and proactive management of cardiovascular health. The system's objectives are aligned with providing a reliable, user-friendly, and technologically advanced solution to both individuals and healthcare professionals. The overarching goals are as follows:

3.3.1 Precision in Risk Assessment:

Objective: Develop a predictive model that demonstrates a high level of precision in assessing individualized risk of cardiovascular diseases.

Goal: Achieve a predictive accuracy that surpasses conventional methods, ensuring reliable risk assessments for diverse populations.

3.3.2 Accessibility and User-Friendly Interface:

Objective: To create a user-friendly web application interface for seamless interaction.

Goal: To enable individuals, irrespective of their technological proficiency, to easily input health parameters and access personalized risk predictions.

3.3.3 Integration of Advanced Machine Learning Algorithms:

Objective: To leverage state-of-the-art machine learning algorithms, including logistic regression, random forest, and decision tree classifier, for robust risk prediction.

Goal: To employ the random forest model as the primary algorithm due to its superior accuracy, ensuring cutting-edge predictive capabilities.

3.3.4 Empowerment through Data-Driven Insights:

Objective: The objective is to empower individuals to make informed decisions about cardiovascular health.

Goal: It is to provide clear and actionable insights through visualizations, aiding users in understanding the impact of various health parameters on their CVD risk.

3.3.5 Utility for Healthcare Professionals:

Objective: Offer a practical tool for healthcare professionals involved in preventive care.

Goal: Enhance the accuracy and efficiency of cardiovascular risk management in clinical settings by providing professionals with a reliable predictive system.

3.3.6 Privacy and Security:

Objective: Implement robust user authentication.

Goal: Safeguard user information through encryption and secure data management, ensuring privacy and confidentiality.

3.3.7 Usability and Scalability:

Objective: Design the system for optimal usability and scalability.

Goal: Ensure the system's responsiveness to user inputs and its ability to handle diverse datasets, accommodating healthcare professionals.

The achievement of these objectives and goals will position the Cardiovascular Disease Risk Prediction System as a valuable and trustworthy resource, contributing to the proactive management of cardiovascular health on both individual and professional fronts.

3.4 Product Perspective

The Cardiovascular Disease Risk Prediction System fits into the broader world of healthcare technology. The system is designed to be user-friendly, ensuring an easy experience for individuals. It uses advanced technology, including machine learning algorithms, to stay up to date with the latest trends in predictive analytics. The system doesn't replace traditional medical advice but adds a proactive approach to managing cardiovascular health. It can grow and improve over time, staying relevant in the ever-changing healthcare landscape. Overall, it's a tool meant to work with healthcare professionals and enhance the way we manage cardiovascular health.

3.5 Product Functions

3.5.1 User Authentication:

Function: Allows user to log in securely.

Purpose: Ensures personalized access to the system, maintaining privacy and data security.

3.5.2 Input Interface:

Function: Provides a user-friendly interface for healthcare professionals to input health parameters.

Purpose: Enables healthcare professionals to easily enter relevant data for personalized cardiovascular risk assessment.

3.5.3 Machine Learning Models:

Function: Integrates logistic regression, random forest, and decision tree classifier models for CVD risk prediction.

Purpose: Utilizes advanced algorithms to analyze diverse clinical parameters and generate accurate risk predictions.

3.5.4 Risk Assessment:

Function: Generates personalized CVD risk assessments based on input health parameters.

Purpose: Provides users with actionable insights into their cardiovascular health risks.

3.5.5 User Management:

Function: Manages user account securely, allowing for login.

Purpose: Safeguards user information and ensures controlled access to personalized risk assessments.

3.5.6 Performance Optimization:

Function: Ensures efficient response times to user inputs, maintaining a seamless and responsive user experience.

Purpose: Enhances user satisfaction by providing timely access to risk assessments.

3.5.7 Ethical Communication:

Function: Clearly communicates the system's supplementary role to professional medical advice, emphasizing responsible use.

Purpose: Promotes ethical use of the tool and builds user trust by setting clear expectations.

3.5.8 Usability Enhancement:

Function: Designs the system for optimal usability, making it accessible to users with varying levels of health literacy.

Purpose: Ensures a user-friendly experience, encouraging widespread adoption and utilization.

3.5.9 Continuous Improvement Mechanism:

Function: Allows for future enhancements based on user feedback, technological advancements, and healthcare standards.

Purpose: Ensures the system remains relevant and effective over time, adapting to evolving healthcare needs.

3.6 Operating Environment

The software will operate in the following environment:

Hardware Platform: The software will be designed to run on standard desktop computers and laptops with modern specifications.

Operating System: The software will be compatible with multiple operating systems, including:

- Windows.
- MacOS.

3.7 Software Dependencies

Python Programming Language: Used for creating, implementing, training, and deploying machine learning models.

Jupyter Notebook: The programming interface used for developing and running the code.

Python Programming Libraries: (e.g. Scikit-learn, Seaborn, Matplotlib, Pickle, Streamlit, Pandas and NumPy etc. These libraries have been utilized for implementing and evaluating and deploying machine learning models.

3.8 Functional Requirements

The functional requirements collectively define the capabilities and behaviors of the Cardiovascular Disease Risk Prediction System, ensuring its effectiveness in providing accurate Risk assessments and contributing to proactive cardiovascular health management. Following are the functional requirements of the system.

3.8.1 User Authentication:

Requirement: The system shall provide secure user authentication, allowing user to log in.

Purpose: Ensure personalized access to risk assessments while maintaining user privacy.

3.8.2 Input Interface:

Requirement: The system shall feature a user-friendly interface for individuals to input health parameters, including age, gender, blood pressure, and cholesterol levels etc.

Purpose: Facilitate easy and intuitive entry of relevant data for personalized risk assessment.

3.8.3 Machine Learning Models:

Requirement: The system integrates logistic regression, random forest, and decision tree classifier models for cardiovascular disease risk prediction.

Purpose: Utilize advanced algorithms to analyze diverse clinical parameters and enhance the accuracy of risk predictions.

3.8.4 Risk Assessment:

Requirement: The system shall generate personalized cardiovascular disease risk assessments based on input health parameters.

Purpose: Provide users with actionable insights into their individual risk factors and overall cardiovascular health.

3.9 Non-Functional Requirements

These non-functional requirements provide the framework for the system's performance, security, usability, scalability, ethical considerations, compatibility, reliability, data integrity,

interoperability, adaptability, compliance, and documentation, ensuring that the Cardiovascular Disease Risk Prediction System meets the highest standards in terms of both functionality and user experience.

3.9.1 Performance:

Requirement: The system shall respond to user inputs within seconds.

Purpose: Enhance user satisfaction by providing a responsive and efficient experience.

3.9.2 Usability:

Requirement: The user interface shall be intuitive and accessible to user.

Purpose: Facilitate widespread adoption by ensuring a user-friendly experience.

3.9.3 Scalability:

Requirement: The system shall be capable of handling data efficiently.

Purpose: Ensure adaptability to varying data for scalability.

3.9.4 Compatibility:

Requirement: The system shall be compatible with major web browsers and devices.

Purpose: Ensure accessibility across different platforms.

3.9.5 Reliability:

Requirement: The system shall have an uptime of at least 99%.

Purpose: Ensure continuous availability for healthcare professionals.

3.9.6 Data Integrity:

Requirement: The system shall validate and maintain the integrity of user-inputted data.

Purpose: Ensure the accuracy and reliability of risk assessments.

3.9.7 Adaptability:

Requirement: The system shall allow for future enhancements and updates.

Purpose: Ensure continuous improvement and adaptability to changing healthcare standards.

3.9.8 Robustness:

Requirement: The system shall gracefully handle unexpected inputs or errors, providing meaningful feedback to healthcare professionals.

Purpose: Enhance the system's resilience in clinical settings where unexpected situations may arise.

3.9.9 Maintainability:

Requirement: The system shall be designed with well-documented code for ease of maintenance by IT professionals.

Purpose: Facilitate efficient upkeep and troubleshooting to minimize downtime and support continuous clinical use.

Chapter 4

Methodology

The Development methodology used for this kind of system is the Iterative and Incremental approach. By detailing the following methods, approaches, tools, techniques, algorithms, and other aspects, the Cardiovascular Disease Risk Prediction System aims to ensure transparency in its development process and provide a solid foundation for the understanding and utilization of the solution by healthcare professionals.

The methodology of the Cardiovascular Disease (CVD) risk prediction system embraces a structured and data-centric approach. Initially, a diverse dataset inclusive of pertinent health parameters such as age, blood pressure, and cholesterol levels are gathered. Thorough data cleaning techniques are then applied to address missing values and outliers. After the data preprocessing, which engages sampling for dataset balancing, label encoding for categorical variable conversion, and feature selection to identify significant predictors, three distinctive machine learning algorithms (Logistic Regression, Random Forest, and Decision Tree) are implemented and trained on the preprocessed data. Thorough model evaluation is conducted using validation sets to test predictive performance through metrics like accuracy, precision, recall, and F1 score.

Following the identification of Random Forest as the most effective model, a user-friendly web application is developed using Streamlit, implanting the selected model for seamless CVD risk predictions. This methodology ensures a robust and accurate risk assessment process, integrating data preprocessing and machine learning techniques for effective model training and deployment.

In pursuit of the above-mentioned methodology, an illustration is drawn which depicts the scheme of the procedure that the research is based on or commenced as. It explains the processes in blocks that what exactly is happening in each block.

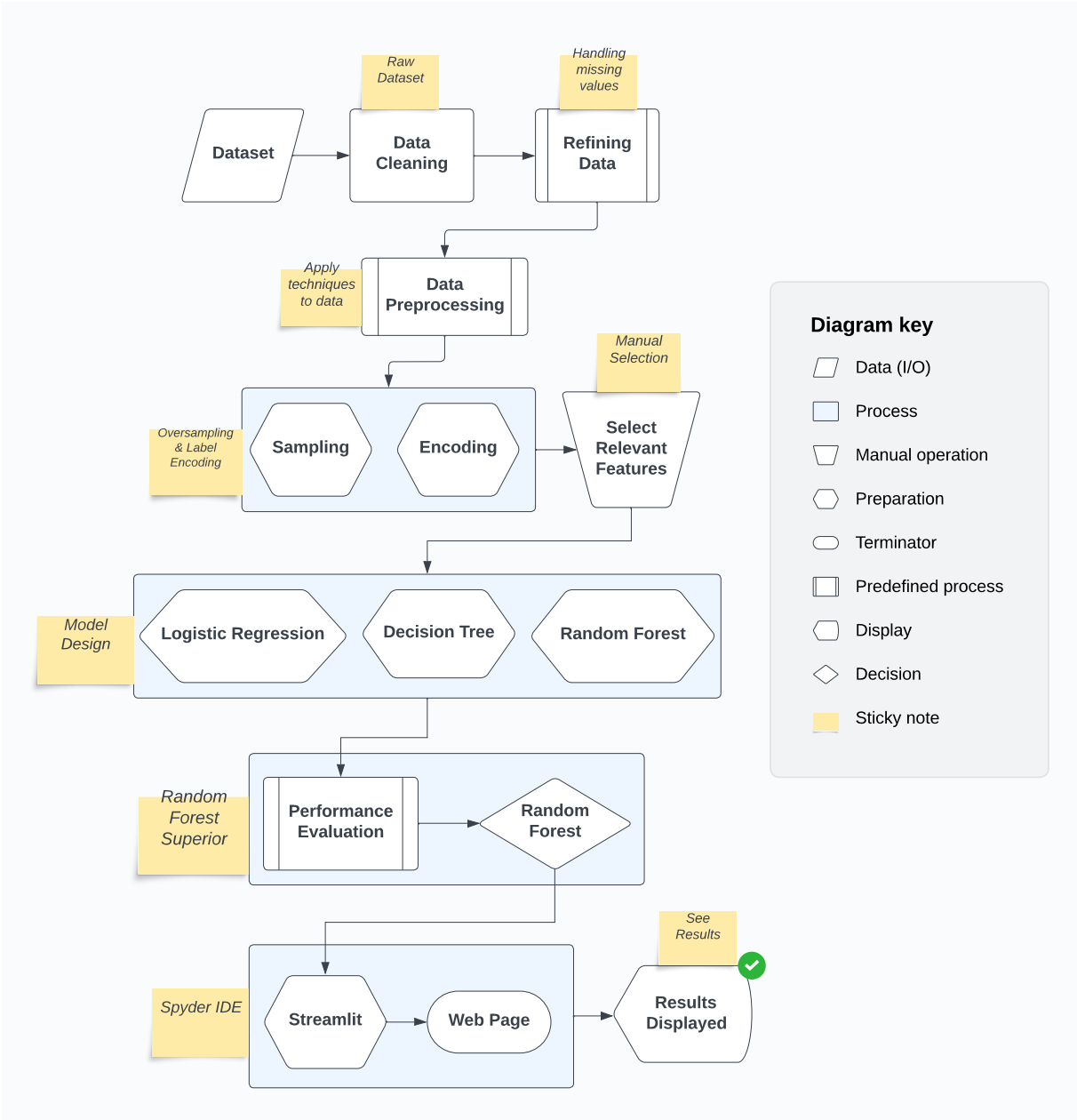


Figure 9. Methodological Course

4.1 Data Preprocessing Techniques

The following are the preprocessing techniques used in this system.

4.1.1 Importing the Dependencies:

When performing machine learning tasks, it is mandatory to import the relevant libraries. This system was built using libraries that include Pandas, NumPy, Scikit-learn and its derivatives as well. Several other libraries were also used for appropriate tasks which included saving the

developed model using Pickle, plotting the results of the model with matplotlib and seaborn, and of course using streamlit to deploy the built model on the web.

4.1.2 Data Filtering:

The filtering methods were applied to reduce the noise in the data. This involved removing the outliers with specific quantiles, and extracting the irrelevant features and making sure that the features at disposal are the best suited for the system.

4.1.3 Data Deduplication:

One of the biggest reasons of a noisy and large data is the number of duplicates in the dataset. Removing these duplicates is the main process of the preprocessing before any machine learning algorithms are applied. So, duplicates were removed from the initial dataset by using the drop_duplicates command from the pandas library.

4.1.4 Data Encoding:

To apply any sorts of machine learning algorithms on a dataset, the data must be in a format where the models produce the best predictions. Encoding is one technique that is used to ensure that data remains in a singular format. Label Encoding was used in the development of this system which converted all the categorical values in the numerical values so that it makes it easier for the model to perform prediction. The features that had categorical values were then converted into numerical values.

4.1.5 Data Profiling:

This is a very important step in making sure that the data being used adheres to the standards is validated and authenticated. Checking all the relevant information about the data such as data types, format, range, and the relationship of features with each other with the help of correlation matrix.

Subsequently, applying the above-mentioned preprocessing techniques [12], it was observed that the target class is imbalanced on which models could be inconsiderate to. So, to balance the target class, techniques pertinent to resampling were tried so that the data upon which the model is to be trained, is in the exact shape to be tried in models.

4.1.6 Class Imbalance:

Having a first look at the data, the most evident thing was the class imbalance of the target variable. The initial shape of target column was imbalanced. The instance of people not having heart disease was massively greater as opposed to the people having heart disease. This was a major issue because if we had applied machine learning algorithms to the data, it would have resulted in false predictive system that may have resulted in the underfitting or overfitting of the model. So, to balance the target variable class, resampling technique was applied, resulting in almost similar number of instances in both classes. Resampling technique involved oversampling and under sampling [13]. These enabled us to balance the classes in target variable. The before and after of the resampling is shown in the following figures.

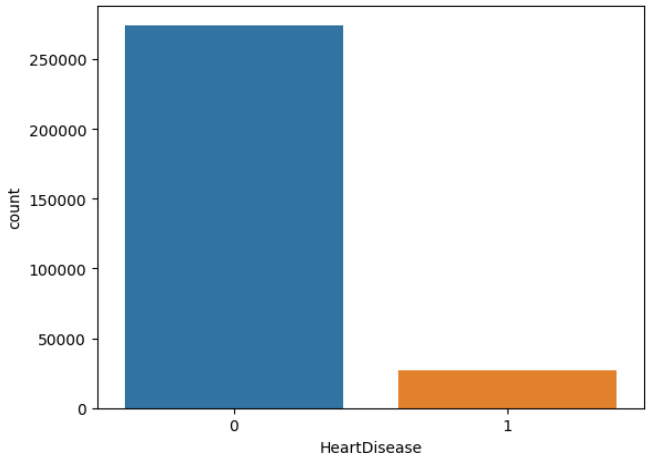


Figure 11. Imbalance Data

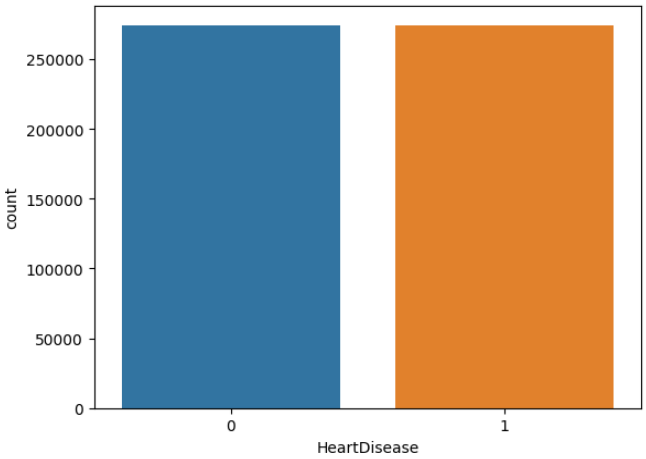


Figure 10. Balanced Data

4.2 Machine Learning Algorithms

4.2.1 Logistic Regression:

Logistic regression is a kind of classification model, which learns and predicts the parameters in the given dataset using regression analysis. The learning and prediction processes are based on measuring the probability of binary classification [14]. Logistic regression model requires class variable that should be binary classified. Likewise, in this dataset the 'HeartDisease' column has the two types of binary inputs, 'No' for the patient who has no chances of heart failure, and 'Yes' for the patient who has heart disease. It is used when the outcome variable has two possible classes. Logistic regression uses the logistic function (also called the sigmoid function) to give the output into the range (0, 1). The sigmoid function is defined as: $(S(x) =$

$1/1 + e^{-x}$, where 'e' is the base of the natural logarithm) [15]. It is simple and computationally efficient algorithm; however, it is sensitive to outliers and limited to binary classification.

Performance Metrics:

The accuracy metric is used for the measurement of the proportion of correct predictions made over the total number of predictions made. Precision refers to as how many predicted positives were actually positive, whereas recall refers to as how many of the actual positive instances were correctly predicted. F1-Score is a measure that considers both precision and recall. It is the harmonic mean of the precision and recall.

When Logistic Regression algorithm was fit on this dataset, the accuracy was 74%. The classification report shows, precision on '0' was 76% and 73% on '1', recall on '0' was 72% and 77% on '1'. Similarly, f1-score was 74% on '0' and 75% on '1'.

$$\text{Precision} = \frac{TP}{TP+FP} \ , \ \text{Recall} = \frac{TP}{TP+FN}$$

$$\text{F1-Score} = \frac{2 * Precision * Recall}{Precision + Recall} \ , \ \text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

```
[ ] lrc = LogisticRegression(C=0.1, max_iter=1000)
    lrc.fit(X_train, Y_train)

    LogisticRegression
    LogisticRegression(C=0.1, max_iter=1000)

[ ] from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

    y_pred_lrc = lrc.predict(X_test)
    accuracy = accuracy_score(Y_test, y_pred_lrc)
    conf_matrix = confusion_matrix(Y_test, y_pred_lrc)
    classification_rep = classification_report(Y_test, y_pred_lrc)


    print(f'Accuracy: {accuracy}')
    print(f'Confusion Matrix:\n{conf_matrix}')
    print(f'Classification Report:\n{classification_rep}')

    Accuracy: 0.7426924022845067
    Confusion Matrix:
    [[39408 15529]
     [12719 42127]]
    Classification Report:
                  precision    recall  f1-score   support

               0       0.76      0.72      0.74     54937
               1       0.73      0.77      0.75     54846

        accuracy                           0.74    109783
       macro avg       0.74      0.74      0.74    109783
    weighted avg       0.74      0.74      0.74    109783
```

Figure 12. LogisticRegressionClassifier

Confusion Matrix: (Logistic Regression)

The confusion matrix of this model predicts ‘42127’ samples as true positives, ‘39408’ samples as true negatives. Similarly, ‘12719’ samples as false positives, ‘15529’ as false negatives.

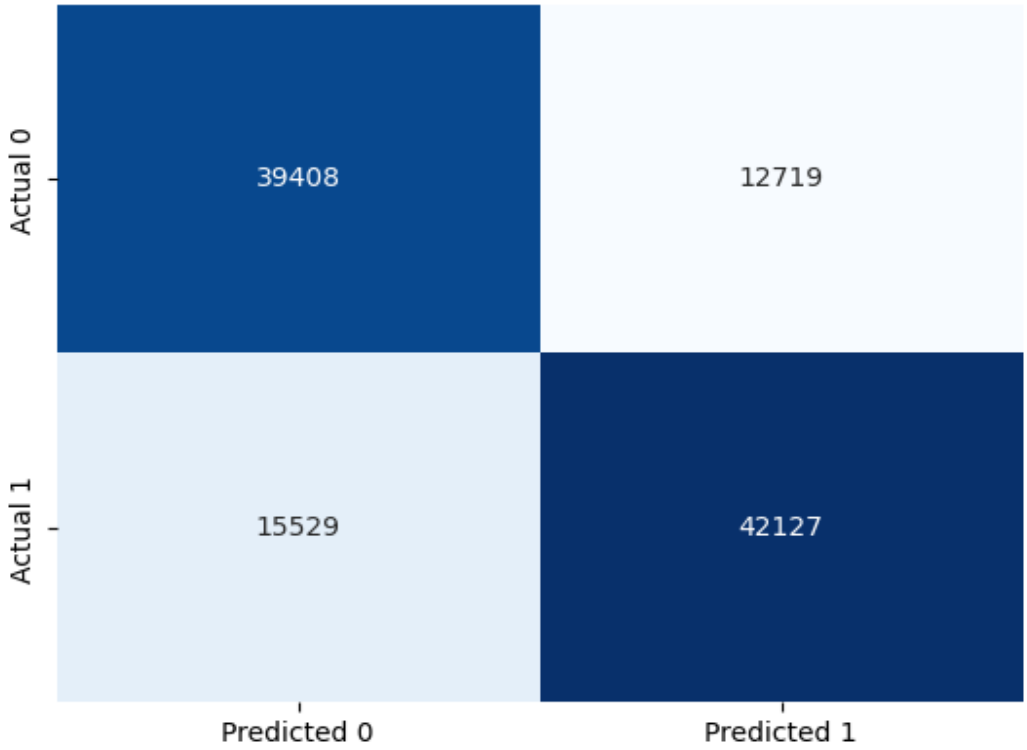


Figure 13. Confusion Matrix for LogisticRegressionClassifier

4.2.2 Decision Tree Classifier:

A supervised machine learning algorithm for classification and regression problems is called a decision tree. Recursively dividing the data into subsets according to the values of the input f eatures is how it operates. To create a treelike structure where each internal node represents a decision based on a feature, each branch represents the decision's outcome, and each leaf node represents the final predicted outcome, a series of questions about the input features are asked during the decision-making process. Decision trees are interpretable, handle non-linearity, and do not require feature scaling. However, they can overfit, be sensitive to data variations, and biased toward dominant classes. They're often used as building blocks for ensemble methods like Random Forests [16].

Performance Metrics:

When Decision Tree Classifier algorithm was fit on this dataset, the accuracy was 95%. The outcomes of classification report were precision on '0' was 100% and 91% on '1', recall on '0' was 90% and 100% on '1'. Similarly, f1-score was 95% on '0' and 95% on '1'.

```
from sklearn.tree import DecisionTreeClassifier
dtc= DecisionTreeClassifier()
dtc.fit(X_train, Y_train)

y_pred_dtc = dtc.predict(X_test)
accuracy = accuracy_score(Y_test, y_pred_dtc)
conf_matrix = confusion_matrix(Y_test, y_pred_dtc)
classification_rep = classification_report(Y_test, y_pred_dtc)


print(f'Accuracy: {accuracy}')
print(f'Confusion Matrix:\n{conf_matrix}')
print(f'Classification Report:\n{classification_rep}')
```

```
Accuracy: 0.9492635471794358
Confusion Matrix:
[[49431  5506]
 [   64 54782]]
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.90      0.95     54937
           1       0.91      1.00      0.95     54846

    accuracy                           0.95    109783
   macro avg       0.95      0.95      0.95    109783
weighted avg       0.95      0.95      0.95    109783
```

Figure 14. DecisionTreeClassifier

Confusion Matrix: (Decision Tree Classifier)

The confusion matrix of this model predicts '54782' samples as true positives, '49431' samples as true negatives. Similarly, '64' samples as false positives, '5506' as false negatives.

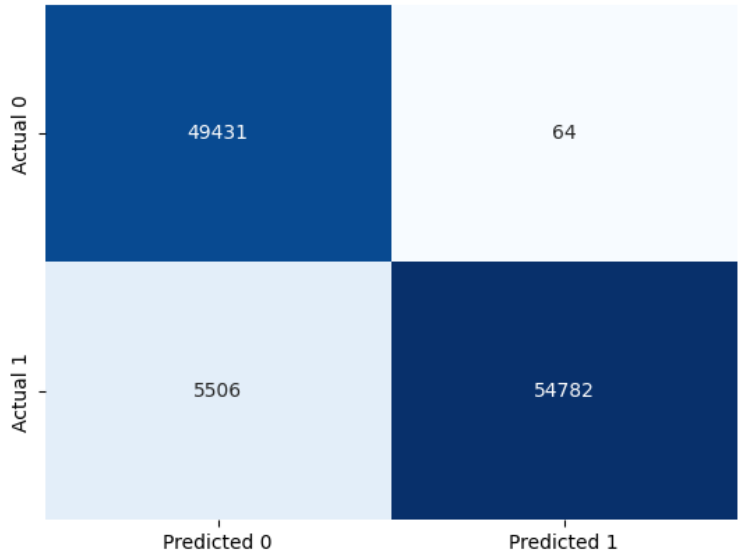


Figure 15. Confusion Matrix for DecisionTreeClassifier

4.2.3 Random Forest Classifier:

As this model is from classification family, therefore it is also known as supervised learning algorithm. This model first generates multiple random trees called a forest. For example, a dataset contains 'x' number of attributes, it first selects some feature randomly known as 'y'. Using all features; (i.e. 'y'), it produces nodes using best rift method. Furthermore, the algorithm will work for creating a complete forest by repeating the previous steps. Then during the prediction process, the algorithm tries to combine the trees using estimated outcome and voting procedure. The purpose of merging the random trees through voting in a forest is to opt out the highest forecasted tree, which can enhance the prediction accuracy for future data. It has high accuracy and robust to overfitting, however; it is computationally complex and repels imbalance data.

Performance Metrics:

When Random Forest Classifier algorithm was fit on this dataset, the accuracy was 96%. The outcomes of classification report were precision on '0' was 100% and 92% on '1', recall on '0' was 91% and 100% on '1'. Similarly, f1-score was 95% on '0' and 96% on '1'.

```
rfc = RandomForestClassifier()
rfc.fit(X_train, Y_train)


y_pred_rfc = rfc.predict(X_test)
accuracy = accuracy_score(Y_test, y_pred_rfc)
conf_matrix = confusion_matrix(Y_test, y_pred_rfc)
classification_rep = classification_report(Y_test, y_pred_rfc)


print(f'Accuracy: {accuracy}')
print(f'Confusion Matrix:\n{conf_matrix}')
print(f'Classification Report:\n{classification_rep}')
```

```
Accuracy: 0.9565415410400517
Confusion Matrix:
[[50216  4721]
 [   50 54796]]
Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.91      0.95     54937
           1       0.92      1.00      0.96     54846

    accuracy                           0.96    109783
   macro avg       0.96      0.96      0.96    109783
weighted avg       0.96      0.96      0.96    109783
```

Figure 16. RandomForestClassifier

Confusion Matrix: (Random Forest)

The confusion matrix of this model predicts ‘54796’ samples as true positives, ‘50216’ samples as true negatives. Similarly, ‘50’ samples as false positives, ‘4721’ as false negatives.

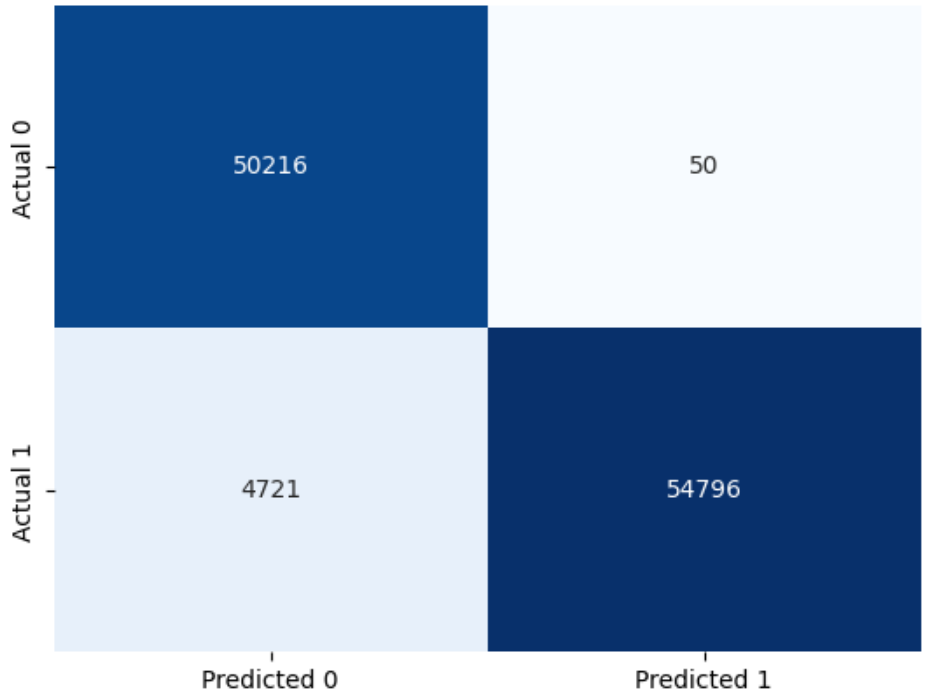


Figure 17. Confusion Matrix for RandomForestClassifier

4.3 Web Application Development Tools

Spyder IDE has been used from Anaconda Navigator Environment for the development of Web Interface using Streamlit python library and its derivatives. The implementation of responsive system for cross-browser compatibility. Spyder IDE is a versatile companion in the world of data science, offering an inviting and intuitive workspace that caters to both novices and experts. With a user-friendly interface and customizable layout, Spyder provides an ideal environment for coding. Its seamless integration with the Streamlit library adds a layer of interactivity, turning static code into dynamic, engaging applications.

Furthermore, Spyder's commitment to collaboration is evident through its version control integration and project management tools, facilitating a seamless teamwork experience. In essence, Spyder IDE with Streamlit not only simplifies the complexities of data science but also elevates the coding experience, turning it into a narrative-driven, interactive journey that invites users to explore, create, and collaborate.

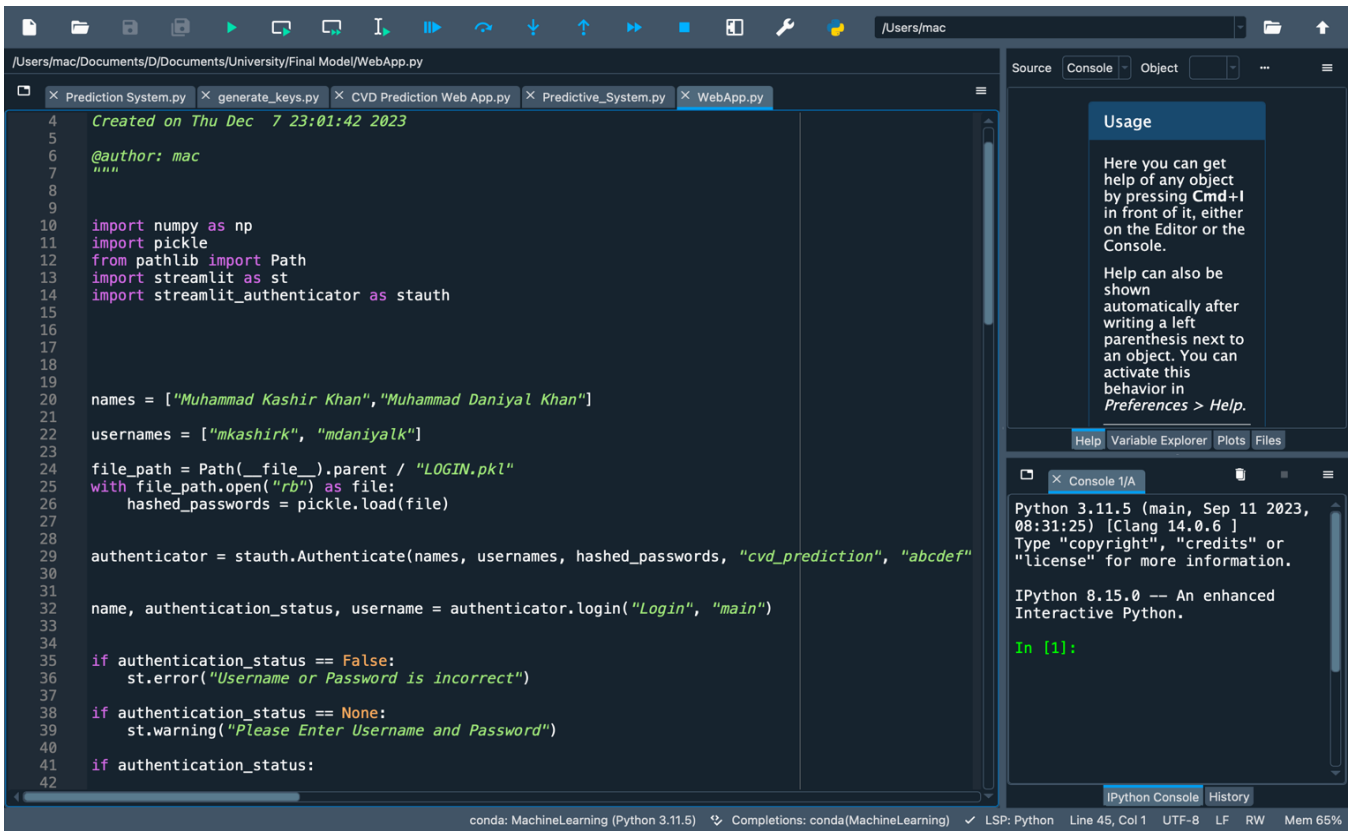


Figure 18. Spyder IDE

4.4 User Authentication and Authorization:

Implementation of secure user authentication using login details. Employed control to ensure proper authorization, allowing only healthcare professionals to access the system.

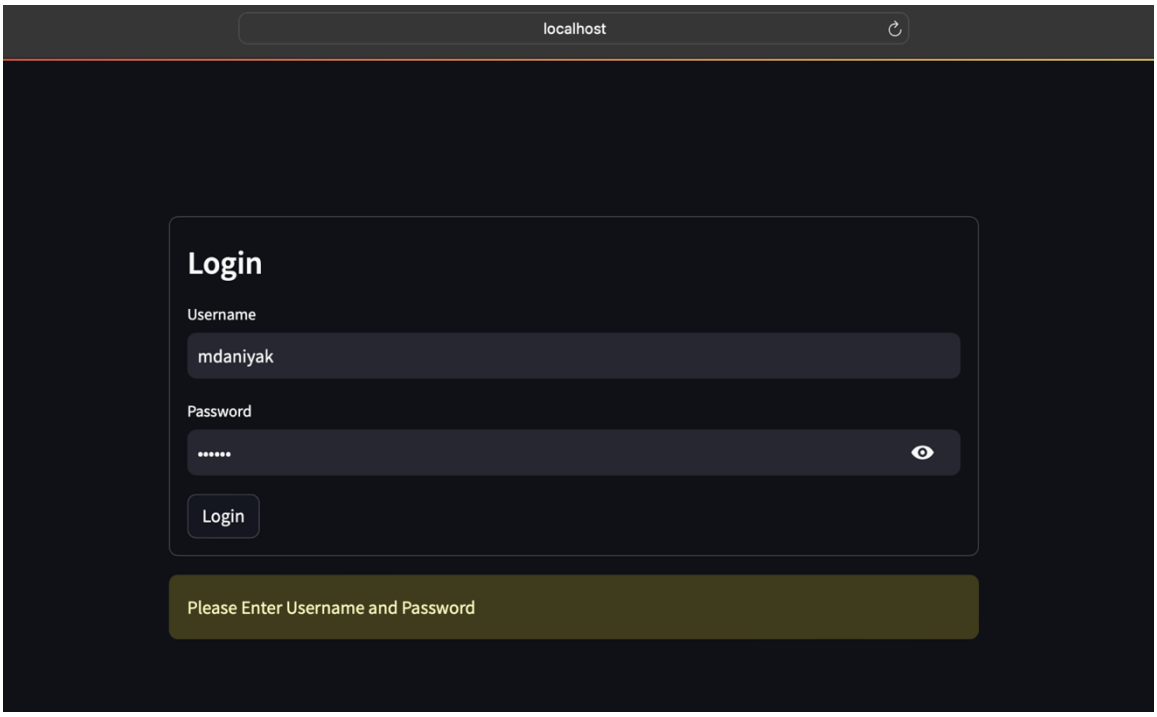


Figure 19. User-Authentication Page

4.5 Visualization Techniques

Utilized data visualization libraries (e.g., Matplotlib, Seaborn) to present graphical representations and Plots.

4.5.1 Matplotlib:

Matplotlib is a reliable data visualization library in Python. It gives you a bunch of tools to create all sorts of charts, from simple line graphs to more complex 3D plots. With Matplotlib, you have the freedom to customize your visuals, making it easy to turn your data into clear and expressive graphs. Whether you're just starting out or diving into advanced data science, Matplotlib provides a versatile canvas for painting your data stories. In the context of this system, different plots have been used and graphs have been made.

4.5.2 Seaborn:

Seaborn, a stylish companion to Matplotlib. It works seamlessly with Matplotlib, offering a simpler way to make your plots look good. Seaborn focuses on making your data visualizations not only informative but also easy on the eyes. It's like having a decorator for your charts, enhancing color palettes and simplifying the process of creating neat statistical plots. Together, Matplotlib and Seaborn make a great team, giving you both the power and the style to turn your data into visually appealing stories. Using seaborn, some aesthetics have been applied and more interaction has been added to the plots.

4.6 Continuous Improvement Mechanism

Implemented an iterative and incremental development cycle based on user feedback and emerging healthcare standards. Regularly update the system to incorporate new features, enhance performance, and address evolving healthcare needs.

4.7 Evaluation Techniques

Used evaluation techniques such as confusion matrix, classification report and accuracy score to evaluate the performance of the models. Used Cross-Validation techniques to find best parameters for the models.

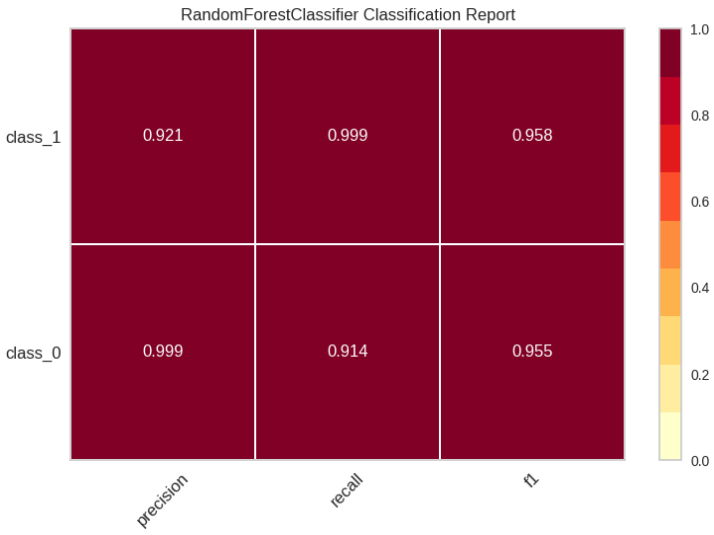


Figure 21. Classification Report of RandomForestClassifier

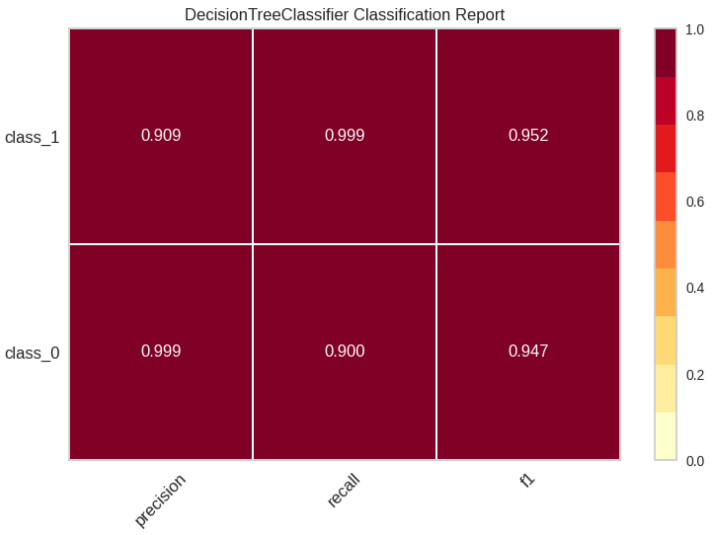


Figure 20. Classification Report of DecisionTreeClassifier

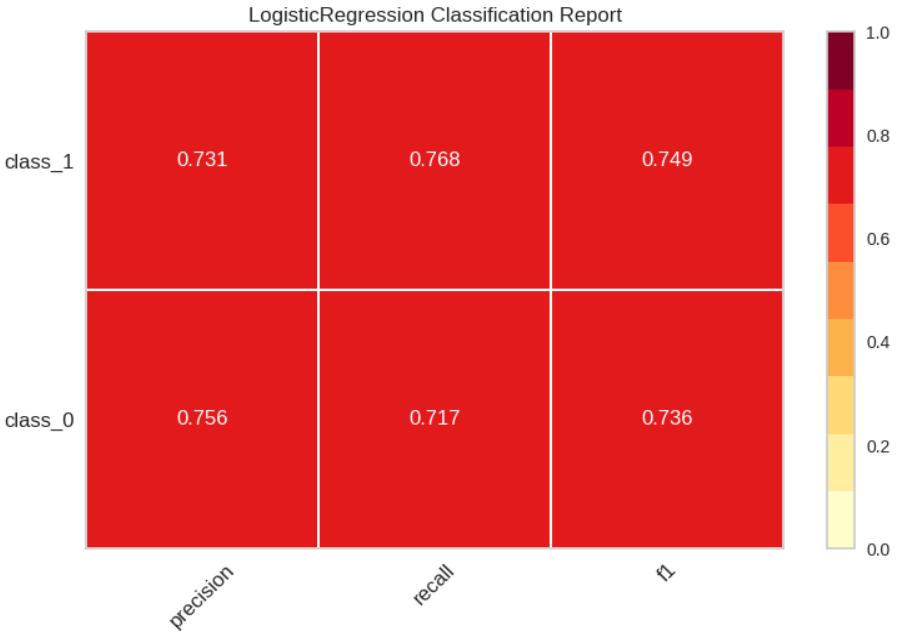


Figure 22. Classification Report of LogisticRegressionClassifier

4.8 Cross-Validation Technique

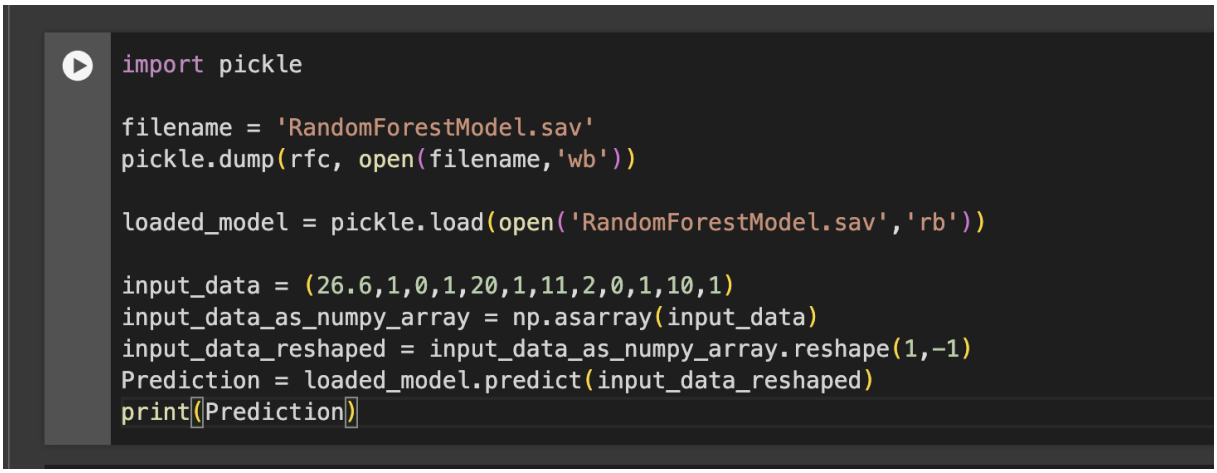
Implemented Hyperparameter tuning to find the best parameters for the respective models [17]. GridSearchCV technique has been used to find best parameters. Using GridSearchCV, the best parameters for the training of the model can be found and upon configuration the performance of the model can be enhanced [11].

Hyperparameter tuning is used as fine art of optimizing the performance of a machine learning model by adjusting the configuration settings, or hyperparameters, that are not learned during

training. Hyperparameter tuning is the key to unlocking a model's full potential, turning it from a good learner into a well-tuned, predictive expert.

4.9 Saving the trained model

Pickle in python is used to covert a python object into a byte stream to store it as a file locally. Pickle library is used to save the 'Random Forest' model locally due to its superior accuracy. 'Dump' and 'Load' were used to write and ultimately read the trained model along with the NumPy array used to input data in the model. Integrated the saved model with Spyder IDE using Streamlit to create user interface. NumPy is used in this regard to enter the data into the model in the form of an input array after reshaping it.



```
import pickle

filename = 'RandomForestModel.sav'
pickle.dump(rfc, open(filename,'wb'))

loaded_model = pickle.load(open('RandomForestModel.sav','rb'))

input_data = (26.6,1,0,1,20,1,11,2,0,1,10,1)
input_data_as_numpy_array = np.asarray(input_data)
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)
Prediction = loaded_model.predict(input_data_reshaped)
print(Prediction)
```

Figure 23. Saving the Trained Model

Chapter 5

System Design & Architecture

5.1 System Architecture/Initial Design

5.1.1 Web Application Interface:

Responsibility: User interaction, data input, and result display.

Component: The Streamlit-based web application.

Roles: (User Interface: Accepts clinical parameters from the user through a form), (Result Display: Displays the prediction result (presence or absence of heart disease) to the user), **(**Authentication: Manages user authentication through a login form).

5.1.2 Prediction Engine:

Responsibility: Utilizing machine learning models for predictions.

Components:

- Logistic Regression Model
- Random Forest Classifier Model
- Decision Tree Classifier Model

Roles: (Prediction**:** Each model is responsible for predicting the likelihood of heart disease based on input parameters).

5.1.3 Data Handling and Preprocessing:

Responsibility: Preparing user input for model consumption.

Component: Backend scripts or functions.

Roles: (Data Validation: Ensures the entered clinical parameters are valid), **(**Feature Scaling/Normalization: If required, ensures consistency in feature scales), **(**Data Transformation: Prepares the input data for model prediction).

5.1.4 Model Management:

Responsibility: Loading, managing, and updating machine learning models.

Component: Model storage and retrieval modules.

Roles: (Model Loading: Loads the trained models during system initialization), (Model Updating: Allows for periodic updates or replacement of models).

5.1.5 User Authentication and Access Control:

Responsibility: Securing access to the system.

Component: Authentication module.

Roles: (User Authentication: Verifies user identity through login credentials), (Access Control: Manages user roles and permissions).

5.2 Rationale for Decomposition

5.2.1 Separation of Concerns:

The system is decomposed to separate user interaction from the underlying prediction and model management logic. This enhances modularity and maintainability.

5.2.2 Model Independence:

Each machine learning model is treated as an independent component. This allows for easy swapping of models based on performance or updates.

5.2.3 Scalability and Maintenance:

Breaking down the system into modular components facilitates scalability. For instance, updating models or adding new features can be done without major disruptions to the entire system.

5.2.4 User Authentication:

Separating user authentication ensures that security concerns are appropriately addressed without cluttering the core functionality of the prediction system.

5.2.5 Streamlit Interface:

Streamlit is chosen for its simplicity and effectiveness in building interactive web applications. It allows for easy integration with python scripts and data processing logic. By decomposing the system in this way, the responsibilities are well-defined, allowing for easy maintenance, updates, and potential future expansions. The modular design enhances collaboration between components and facilitates a more streamlined development process.

5.3 Architecture Design Approach

The architectural design of the cardiovascular disease prediction system involves defining the structure and organization of the software components, their relationships, and the principles guiding their interactions. The chosen architecture should support the system's functionality, maintainability, scalability, and other quality attributes. Here's an overview of the architectural design approach for the described system.

5.3.1 Component-Based Architecture:

Description: The system is organized into distinct components, each responsible for specific functionalities.

Components: The components are comprised as Web-Application Interface, Prediction Engine, Model Management and User Authentication.

5.3.2 Layered Architecture:

Description: The logical layer has been applied to separate concerns and promote modularity.

Layers: Layers include Presentation (Web Application), Business Logic (Prediction Engine, Data Handling), and Data (Model Management, User Authentication).

5.3.3 Model-View-Controller (MVC) Pattern:

Description: The web application follows the MVC architecture pattern for organizing code and separating concerns.

Roles: The roles are defined as Model (Prediction Engine, Data Handling), View (Web Application Interface) and Controller (User Authentication, Model Management).

5.3.4 Security Measures:

Description: Application of secure communication protocols (Anaconda Environment) for web interactions. Encryption and security of sensitive information, especially in user authentication.

5.3.5 Scalability Considerations:

Description: The components are designed to be scaled independently if need arises. Model management may include version control and convenient swapping of models.

5.3.6 Data Flow and Interaction:

Description: User input flows through the web interface to the data handling and preprocessing component, then to the prediction engine for model predictions. Model results are sent back to the web application for display.

5.3.7 Rationale for Architectural Choices:

Modularity and Maintainability: Component-based and layered architectures promote modularity, making it easier to understand, maintain, and update specific parts of the system without affecting others.

Scalability and Flexibility: Microservices architecture and service-oriented principles allow for independent scaling of components, providing flexibility for handling varying workloads.

User Interface Separation: The MVC pattern separates the concerns of user interface design, business logic, and data handling, making the system more maintainable and adaptable to changes in user interfaces.

Security and Reliability: The security measures implemented in the architecture ensure the protection of sensitive data and reliable communication between components.

Continuous Integration and Deployment: CI/CD practices contribute to a more streamlined development process, reducing the chances of integration issues and allowing for quick and reliable deployment of updates.

This architectural design provides a flexible, scalable, and maintainable foundation for the cardiovascular disease risk prediction system, aligning with modern best practices [18].

5.4 Architecture Design

Initially the dataset was in a raw form which is not suitable for model training in machine learning. So, different operations were applied to clean and transform the data upon which the models can be best trained. The preprocessing phase involved, cleaning the data in which null values were removed, repeated values were removed. The resampling of the dataset was done resulting in the balance among the target classes. After analyzing the co-relation between the features, feature selection was done. Moreover, label encoding technique was used to transform categorical values into numerical values in the data. Before training the model onto the data, the features and target column were split into (X, Y) respectively. The model was then trained onto the data and predictions were generated.

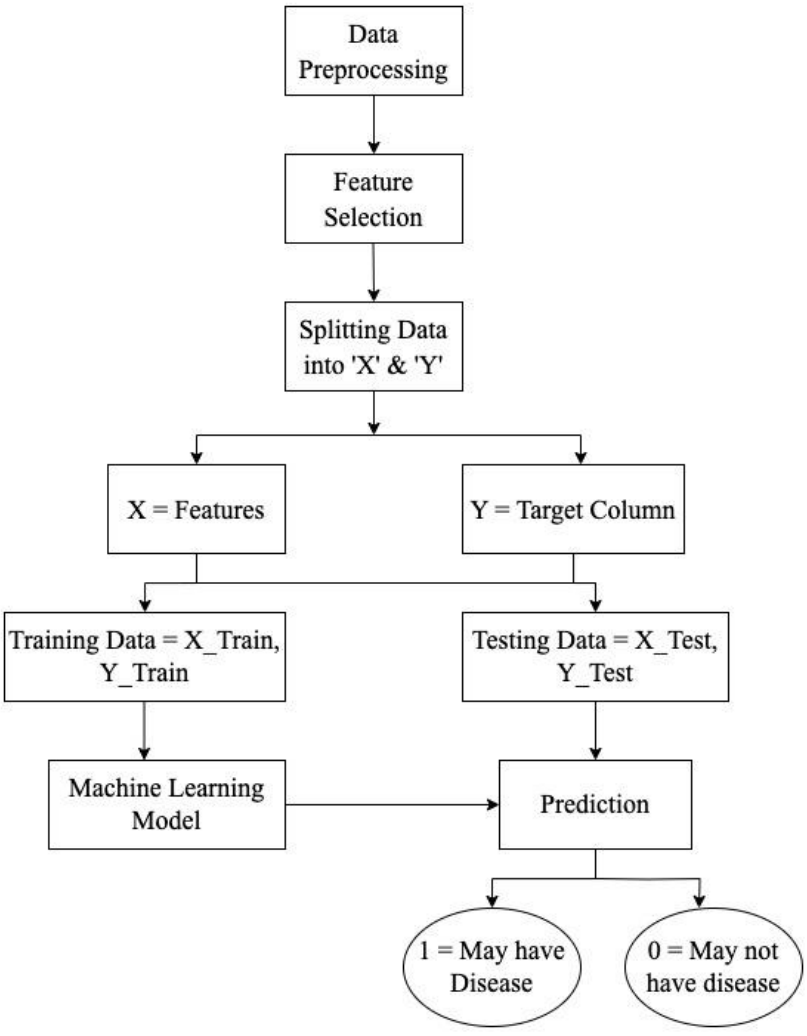


Figure 24. Model Architecture Diagram

5.5 System Flow

The system interface starts with an authentication page where the user is required to login through verified credentials. The username and password fields would require to be filled, upon validation the authentication status would pass, and the user stands authenticated. Subsequently, upon successful authentication, the user would navigate to the prediction page through a display of message that welcomes the user to the main page.

The main page is comprised of the input fields where the user must input the clinical health parameters of an individual. Every input field contains different values depending upon the containers they are kept in. These fields allow the user to input parameters in accordance with the health of the individual.

Moreover, when the parameters are entered into the input fields to generate results, there is a button provided for predicting results. Click on the predict button to see the outcomes.

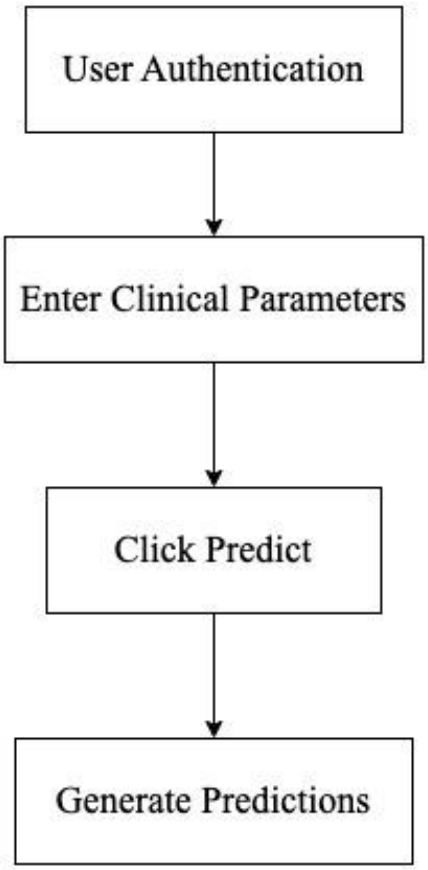


Figure 25. System Flow Diagram

5.6 Detailed System Design

5.6.1 Function: User Sign-In

Pre-Condition:

Parameters: Username (The username entered by the user). Password (The password entered by the user).

Constraints: Username and Password must adhere to the system's specified format and length requirements. The user account associated with the provided username must exist in the system.

Post-Condition:

Output: If the provided username and password are valid and match existing account, the function logs the user in and grants the access to the main module. If the credentials are invalid, an error message is displayed, and the user is prompted to try again.

5.6.2 Function: Input Clinical Health Parameters

Pre-Condition:

Parameters: Several clinical health parameters (e.g., blood pressure, cholesterol levels, age, etc.) are entered by the user.

Constraints: Each parameter must adhere to the valid range specified by the system. Data types and formats must adhere to the expected standards.

Post-Condition:

Output: The system accepts the entered clinical health parameters without any errors if they meet the specified constraints. If there are issues with the entered parameters (e.g., out-of-range values, incorrect formats), an error message is displayed, and the user is prompted to correct the input.

5.6.3 Function: Predict Cardiovascular Disease Risk

Pre-Condition:

Parameters: Clinical health parameters to be entered by the user. Every field must be filled.

Constraints: The user must be successfully logged in. The input parameters must be valid and within the acceptable range.

Post-Condition:

Output: The system processes the entered clinical health parameters and generates a prediction for cardiovascular disease risk. The prediction is displayed to the user along with relevant information, such as the confidence level or probability. In case of any errors during prediction (e.g., data processing issues, model errors), an appropriate error message is displayed.

5.7 System Operating Components

5.7.1 Component Name: System Resource Manager

Description:

The System Resource Manager is a critical software component designed to efficiently allocate, monitor, and manage system resources to ensure optimal performance and reliability of the overall system [19]. This component plays a crucial role in orchestrating the interaction between hardware and application software, enhancing the system's responsiveness, stability, and resource utilization.

Responsibilities:

Resource Allocation:

CPU Allocation: Manages the distribution of CPU processing power among running applications, preventing monopolization by any single process.

Memory Management: Allocates and deallocates memory space for running applications, ensuring efficient usage, and preventing memory leaks.

Process Scheduling: Prioritizes and schedules processes based on their priority, deadlines, and resource requirements. Implements scheduling algorithms to optimize task execution, minimize wait times, and maximize system throughput.

Error Handling: Detect and manage errors that may arise during resource allocation or process execution. Implement mechanisms for error recovery, logging, and reporting to ensure system robustness [20].

Security: Enforces access control policies and permissions to protect critical system resources. Collaborates with security modules to monitor and prevent unauthorized access or malicious activities.

Performance Monitoring: Monitors system performance metrics such as CPU usage, memory utilization, and disk I/O. Provides real-time feedback to administrators for performance optimization and troubleshooting.

Configuration Management: Facilitates the configuration of system settings and parameters, allowing administrators to adapt the system to varying workloads and requirements.

Inter-Component Communication: Enables communication and data exchange between various system components, ensuring seamless coordination and collaboration.

User Interface: Offers a graphical or command-line interface for system administrators to configure settings, monitor performance, and troubleshoot issues.

Integration: The System Resource Manager is tightly integrated with the operating system kernel and interacts with other system components, including device drivers, security modules, and application software. It collaborates with the operating system scheduler, memory manager, and I/O subsystem to ensure coordinated resource utilization.

Future Enhancements: Possible future enhancements for the System Resource Manager could include advanced machine learning algorithms for dynamic resource allocation, support for emerging hardware architectures, and improved energy efficiency measures.

To summarize, the System Resource Manager is a core component that plays a pivotal role in ensuring the efficient and reliable operation of a computer system by managing and optimizing the use of its resources.

5.8 Diagrams

5.8.1 Use case Diagram:

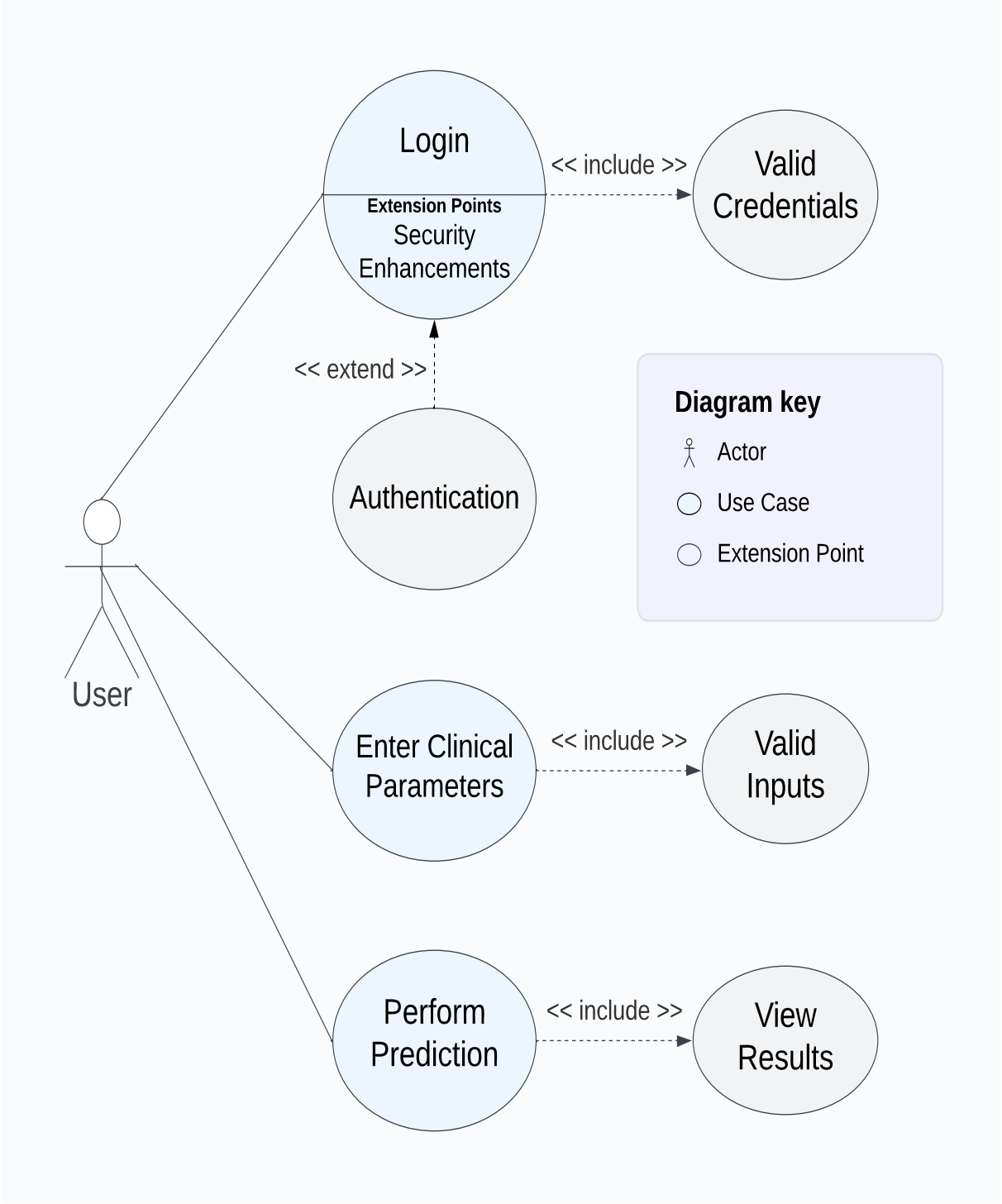


Figure 26. Use-Case Diagram

5.8.2 Activity Diagram:

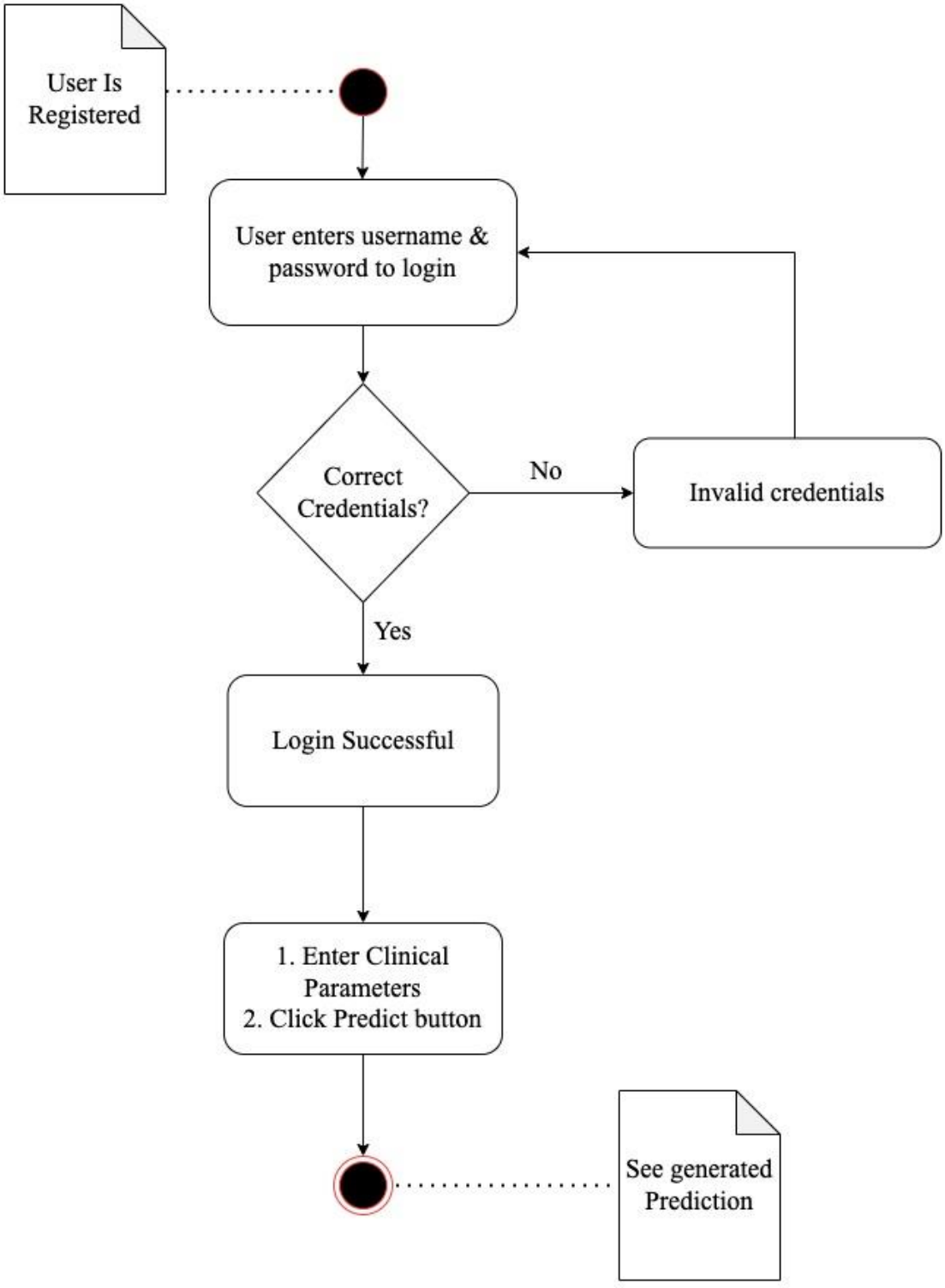


Figure 27. Activity Diagram

5.8.2 Sequence Diagram:

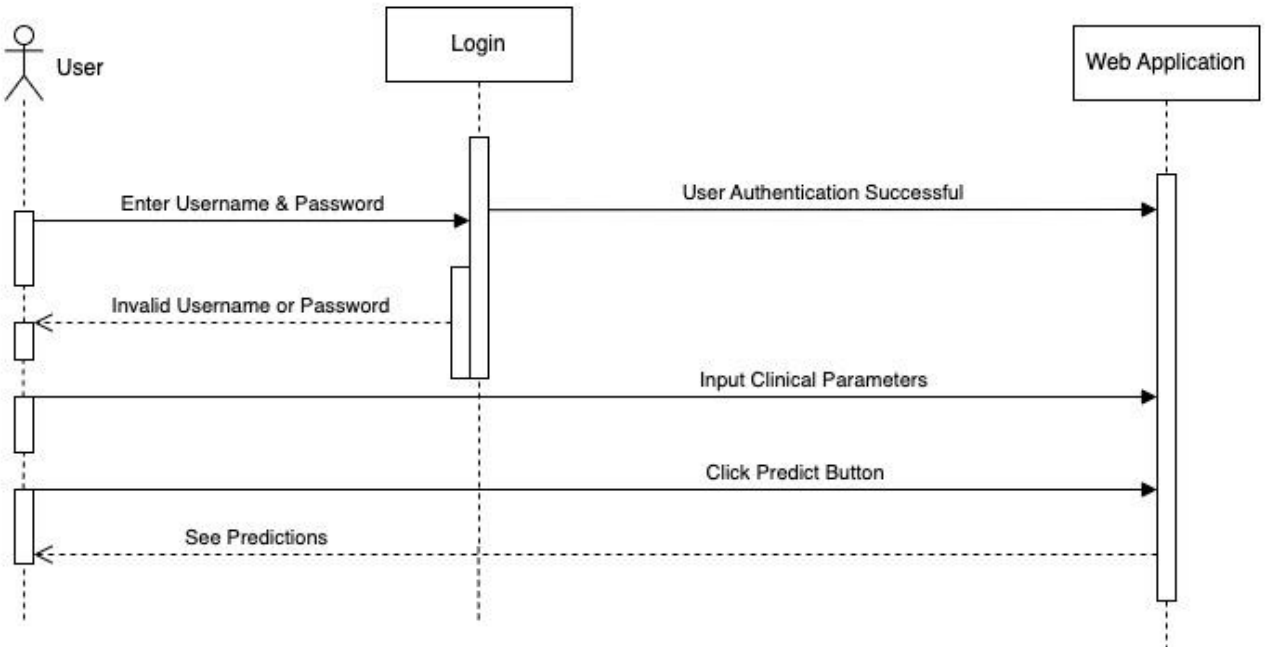


Figure 28. Sequence Diagram

5.8.4 Data Flow Diagram:

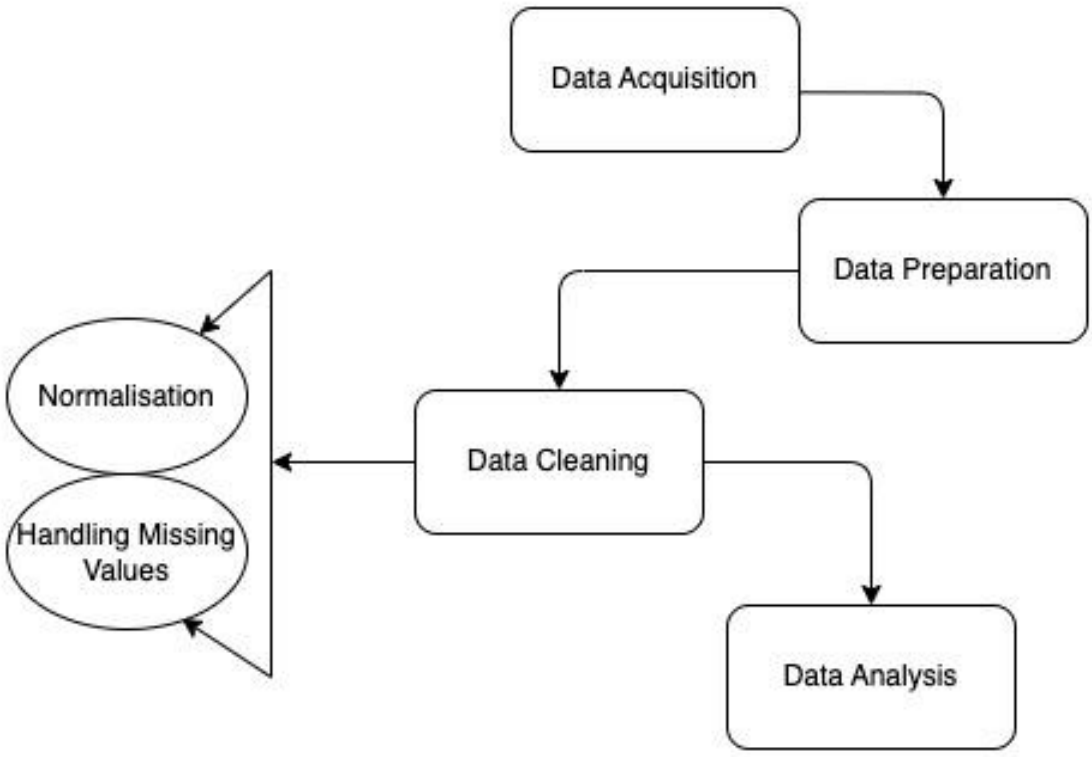


Figure 29. Data Process Flow Diagram

Chapter 6

Implementation and Testing

The Cardiovascular Disease Prediction System's web application was developed using the Spyder IDE, with the primary UI framework being Streamlit, a Python library. The development process embraced an iterative and agile methodology to ensure responsiveness to changing requirements. Key software tools and testing methodologies include:

6.1 Development Tools

IDE: Spyder IDE served as the primary integrated development environment.

Google Colab: This environment has been used for data and models development.

Backend: Python-based Streamlit handled server-side development.

Machine Learning: Scikit-learn facilitated the implementation of logistic regression, random forest, and decision tree classifier algorithms.

6.2 Testing Methodologies

Unit Testing: Ensured individual components within the Streamlit web app functioned as intended.

Integration Testing: Evaluated the seamless interaction between different components, including the Streamlit interface and the backend.

Load Testing: The system is responsive enough when there is an increased threshold.

Security Testing: Until the user authentication the access is not possible.

Performance Testing: Analyzed the web app's responsiveness and resource utilization under varying datasets.

Usability Testing: Observed smooth user experience.

6.3 Controlled Libraries and Templates

Streamlit's streamlined structure served as a controlled library, enabling rapid development and easy integration of visual elements. Templates and styling tools within Streamlit ensured a consistent and user-friendly interface.

6.4 Code Walkthroughs:

Regular code walkthroughs within the Spyder IDE & Colab facilitated observance to coding standards, readability, and addressing issues.

6.5 Evaluation and Comparison:

The following graphs and their description depict the evaluation and the eventual comparison of the models.

6.5.1 Accuracies Comparison:

As discussed earlier, there were three machine learning models used in the development process and eventually Random Forest Classifier was selected as the model which was used to build the Web Application because of its superior accuracy, which was 96% as opposed to Decision Tree Classifier having 95% and Logistic Regression having 74% accuracy. The following chart shows the comparison of the three models with their training and testing accuracies.

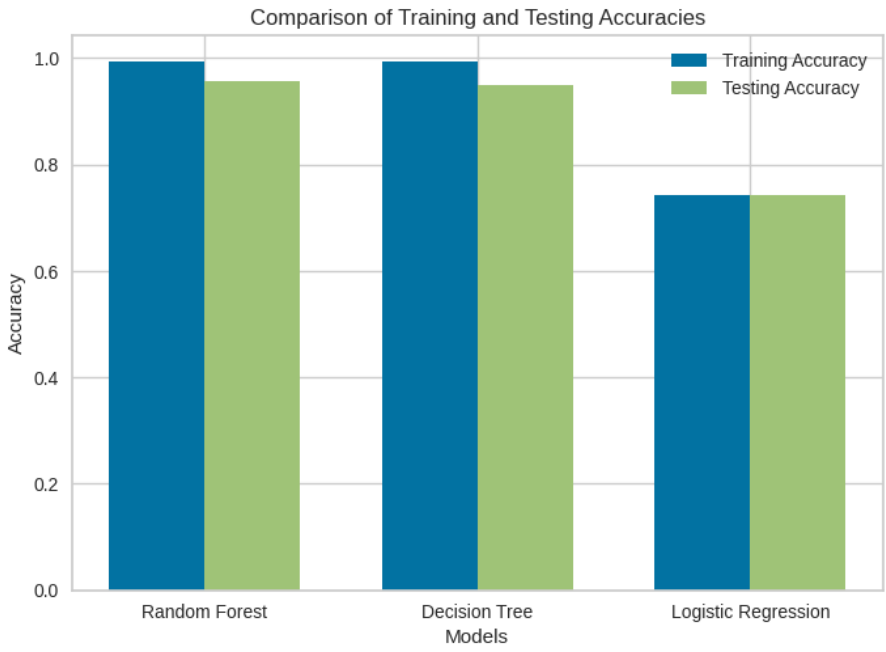


Figure 30. Comparison of Testing and Training Accuracies

Chapter 7

Results and Discussion

Principally, the method to this research involves identifying the algorithm with the superior accuracy in comparison to others. In this regard the above explained algorithms have been implemented.

The logistic regression algorithm has been a part of experiment, and it has been observed that the accuracy of the former is not up to that extent, different techniques have been applied to get an extensive accuracy but in vain. In lieu of this, another algorithm named decision tree classifier was tried, surprisingly with an explicit accuracy, it was inclined to be chosen as with the highest of the accuracy, but another refined attempt had been made but this time around, using a different classifier named random forest classifier, the observed accuracy was a bit superior to that of the decision tree.

Now, in the context of this system reviewing the objectives of the system the random forest model was chosen as the heir of the dominion. Subsequently, the outcomes, the observations and the objective analysis guided to the significance of findings, using random forest as the superior classifier.

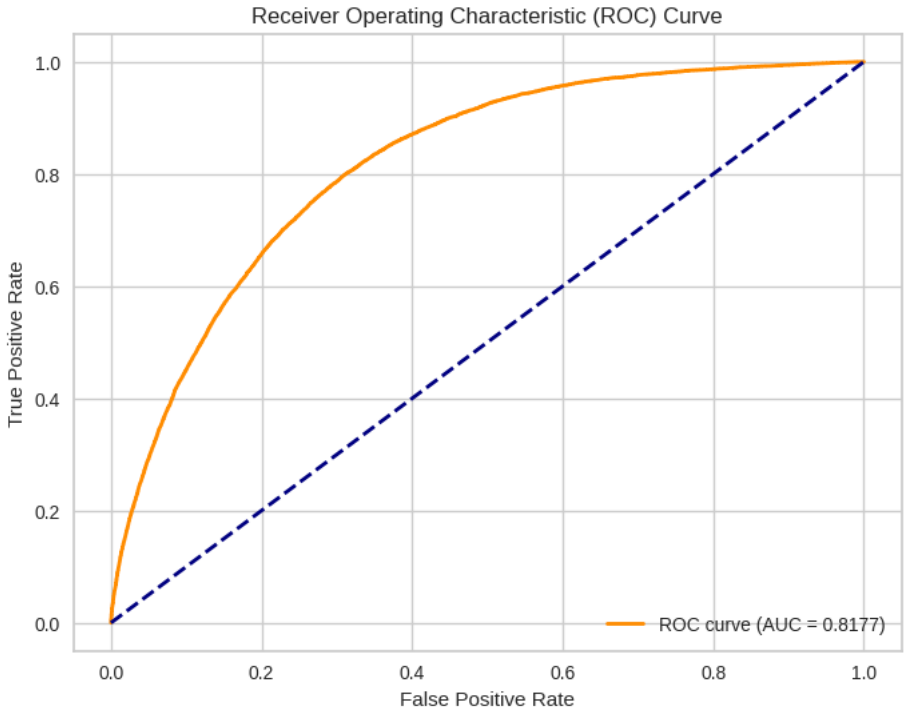


Figure 31. LogisticRegressionClassifier ROC Curve

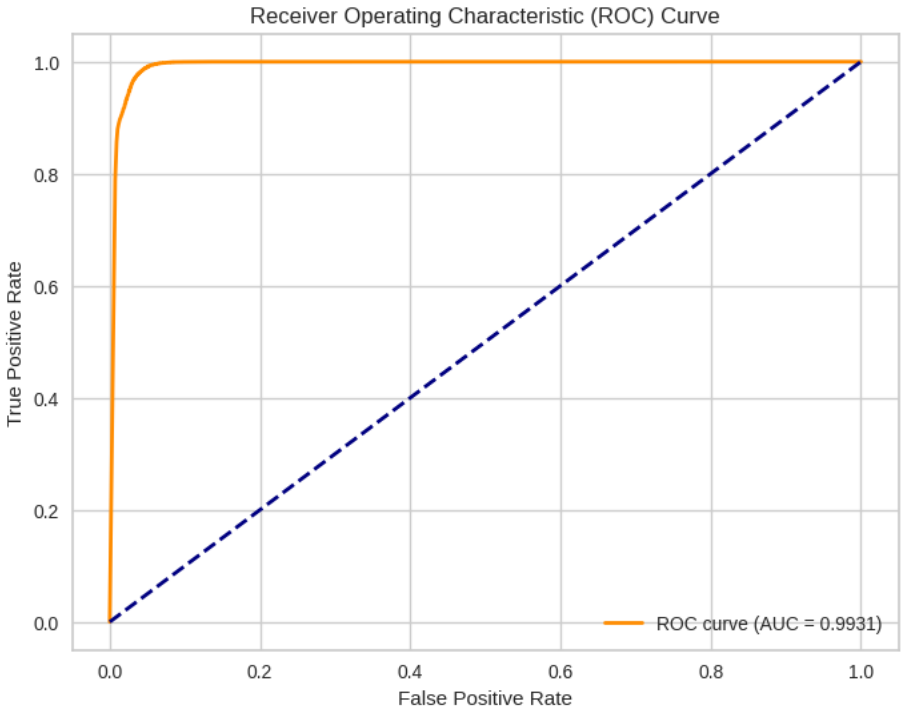


Figure 32. RandomForestClassifier ROC Curve

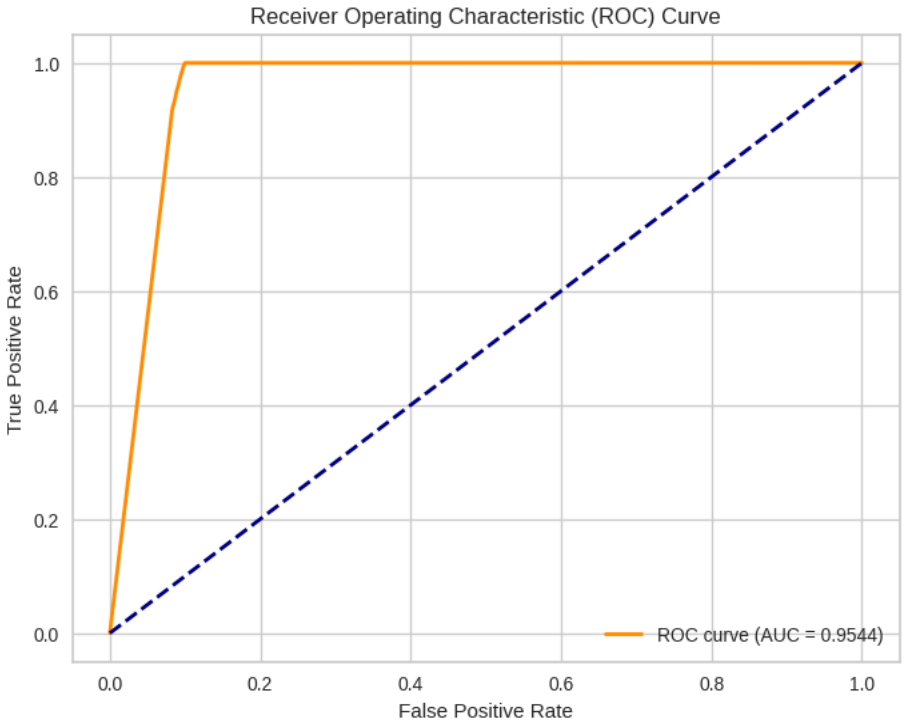


Figure 33. DecisionTreeClassifier ROC Curve

The overhead figures depict the considering analysis of the respective models used in the context of the aforementioned system.

AUC – ROC Curve:

The Receiver Operating Characteristic (ROC) curve below is a graphical representation of the performance of a binary classification model used in the system across different classification thresholds [21]. It illustrates the trade-off between the true positive rate and the false positive rate as the discrimination threshold is varied [15].

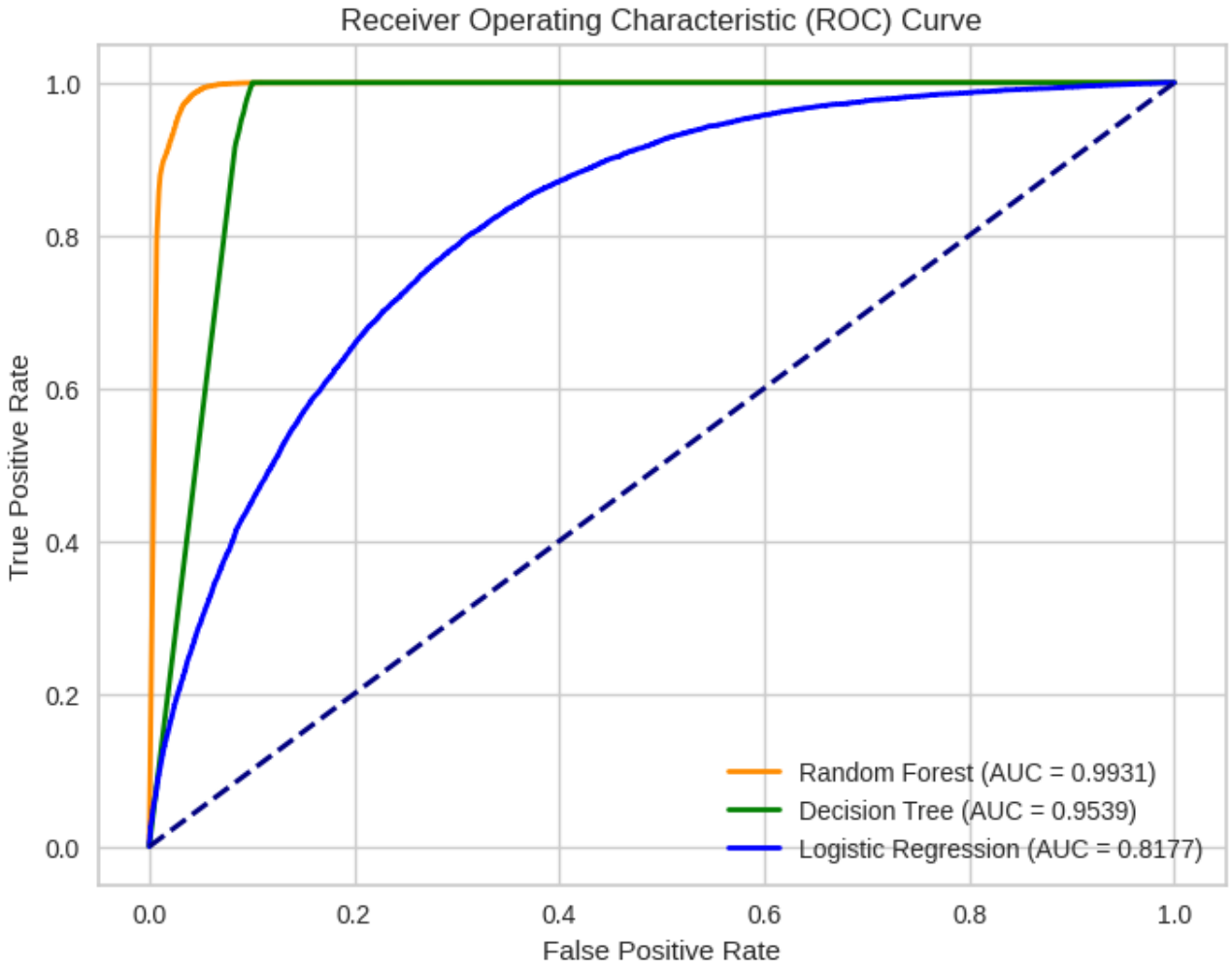


Figure 34. ROC Curve for Comparison of Models' Performance

The Above ROC Curve depicts the comparison of the models used in this research and also shows that Random Forest has the highest accuracy. This Random Forest model has then employed to create the web interface using streamlit. The application is then tested vigorously, and the following is the result when clinical parameters such as BMI, Age, Sugar levels, Activity levels and Sleep Time, including others are entered to take the probability of cardiovascular disease risk.

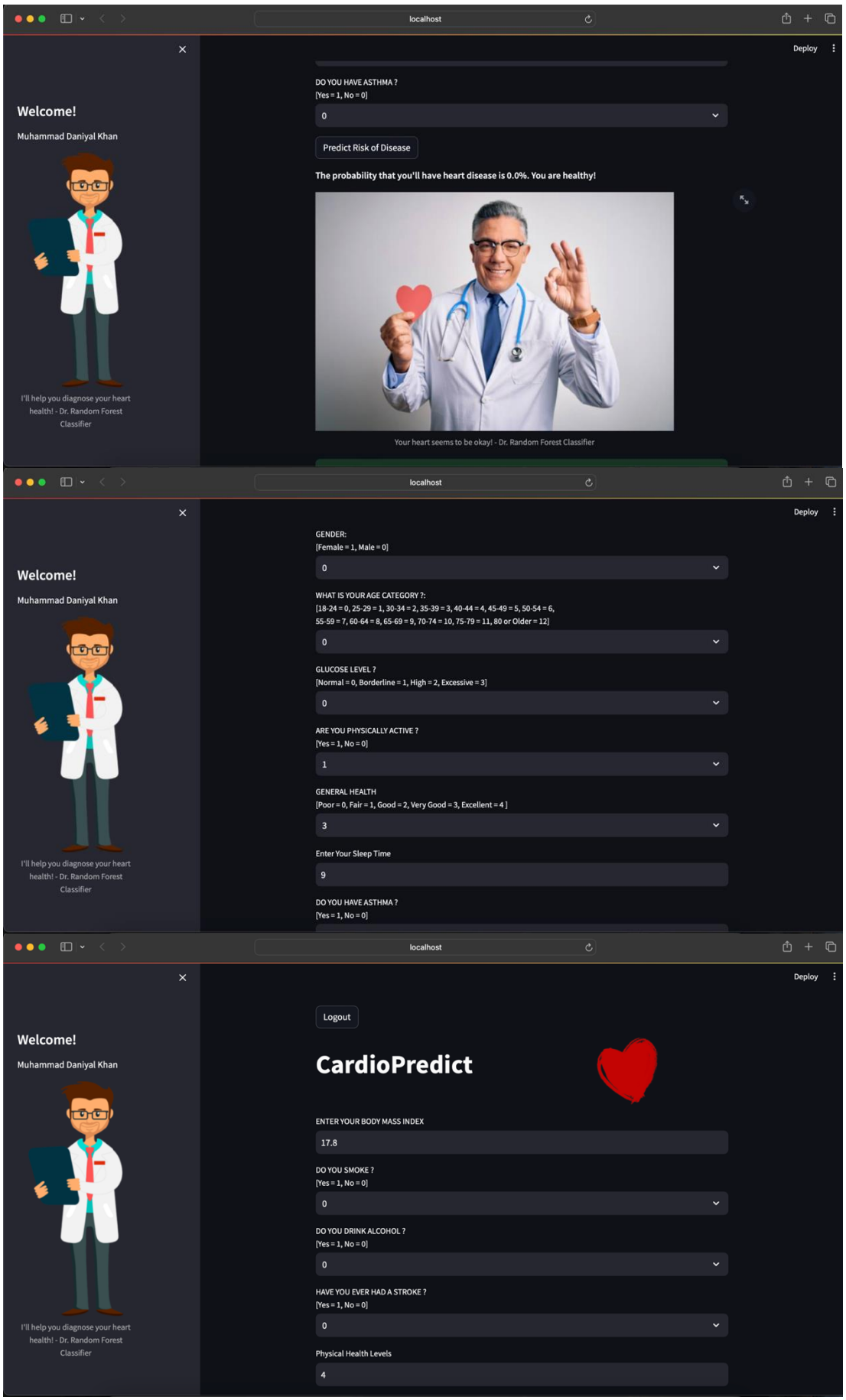


Figure 35. Prediction Results

Test Cases:

These test cases focus on fundamental aspects of the Cardiovascular Disease Risk Prediction System, certifying secure user authentication, accurate patient data input, and the effective integration of machine learning model.

In the "User Authentication" test case, the system's ability to securely authenticate healthcare professionals is evaluated by entering valid credentials and verifying that the system redirects to the Web Application with the user's name displayed.

The "Patient Data Input" test case validates the accurate entry and storage of patient information, ensuring that the system accepts complete data.

The "Machine Learning Prediction" test case assesses the system's predictive accuracy, employing known patient data to trigger machine learning predictions and validating that the outcomes align with predetermined expectations.

The "Usability Testing" test cases lays the foundation for the system's functionality, emphasizing security, accuracy, and the seamless integration of core features.

Table 3. Test Cases

Test No.	Test Objective	Test Data	Test Steps	Module	Observation	Expectations	Status
1.	User Authentication	Valid login credentials	1. Open login page. 2. Validate entries. 3. Click 'Login'.	User Authentication.	Systems redirects to the web page.	Prediction page appears.	PASS
2.	Patient data input	Clinical parameters	1. Enter the patient data. 2. Click enter.	Prediction page	System accepts input.	Data accepted.	PASS
3.	Machine Learning Prediction	Get prediction	Click 'Predict' button at the bottom.	Prediction page	System generates prediction.	Generate prediction.	PASS
4.	Usability testing	Navigate through modules	Navigate through login & prediction page.	Usability testing	Smooth UI & UX.	Seamless experience.	PASS

Chapter 8

Conclusion and Future Work

In Conclusion, the proposed Cardiovascular Disease Risk Prediction System effectively addresses the problem statement articulated in the introduction, aiming to provide healthcare professionals with a proactive tool for assessing individualized risks of cardiovascular diseases. By leveraging advanced machine learning algorithms, including logistic regression, random forest, and decision tree classifier, the system offers a comprehensive analysis of diverse clinical parameters, yielding precise risk predictions tailored to each patient's unique health profile using Random Forest Model.

To substantiate the solution's efficacy, several evaluations were conducted. The random forest model emerged as the primary algorithm due to its superior accuracy. Furthermore, extensive testing with diverse datasets was performed to evaluate the accuracy and performance of the system in predicting cardiovascular disease risks, with results consistently demonstrating high precision and reliability.

Recommendations for the system's ongoing improvement include continuous user training programs to ensure optimal utilization by healthcare professionals, regular updates to incorporate emerging research findings and evolving healthcare standards, and collaboration with healthcare institutions for broader system implementation.

Looking towards future directions, the system could benefit from the integration of real-time data for dynamic risk assessments, the incorporation of genetic data to enhance precision, and seamless compatibility to offer a comprehensive healthcare solution. These future directions aim to further enhance the system's capabilities, ensuring it remains at the forefront of cardiovascular risk prediction and continues to contribute effectively to proactive healthcare management.

References

[1] Xie, S., Z. Yu, and Z. Lv, *Multi-Disease Prediction Based on Deep Learning: A Survey.* Computer Modeling in Engineering & Sciences, 2021. **128**(2): p. 489-522.

[2] Ibrahim, I. and A. Abdulazeez, *The Role of Machine Learning Algorithms for Diagnosing Diseases.* Journal of Applied Science and Technology Trends, 2021. **2**(01): p. 10-19.

[3] Nissa, N., S. Jamwal, and S. Mohammad, *Early Detection of Cardiovascular Disease using Machine learning Techniques an Experimental Study.* International Journal of Recent Technology and Engineering (IJRTE), 2020. **9**(3): p. 635-641.

[4] Patil, M.D., R. Hasan, and V.A. Vyawahare, *Comparative Analysis of Machine Learning Algorithms for Heart Disease Prediction.* ITM Web of Conferences, 2021. **40**.

[5] Moturi, S., *Classification Model for Prediction of Heart Disease using Correlation Coefficient Technique.* International Journal of Advanced Trends in Computer Science and Engineering, 2020. **9**(2): p. 2116-– 2123.

[6] Bharti, R., et al., *Prediction of Heart Disease Using a Combination of Machine Learning and Deep Learning.* Comput Intell Neurosci, 2021. **2021**: p. 8387680.

[7] Hossen, M.D.A., et al., *Supervised Machine Learning-Based Cardiovascular Disease Analysis and Prediction.* Mathematical Problems in Engineering, 2021. **2021**: p. 1-10.

[8] Mohan, S., C. Thirumalai, and G. Srivastava, *Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques.* IEEE Access, 2019. **7**: p. 81542-81554.

[9] Mishra, S., *A Comparative Study for Time-to-Event Analysis and Survival Prediction for Heart Failure Condition using Machine Learning Techniques.* Journal of Electronics, Electromedical Engineering, and Medical Informatics, 2022. **4**(3): p. 115-134.

[10] Dissanayake, K., M.G. Md Johar, and T.W. Liao, *Comparative Study on Heart Disease Prediction Using Feature Selection Techniques on Classification Algorithms.* Applied Computational Intelligence and Soft Computing, 2021. **2021**: p. 1-17.

[11] Arghandabi, H., *A Comparative Study of Machine Learning Algorithms for the Prediction of Heart Disease.* International Journal for Research in Applied Science and Engineering Technology, 2020. **8**(12): p. 677-683.

[12] Benjelloun, F.-Z., et al., *Improving outliers detection in data streams using LiCS and voting.* Journal of King Saud University - Computer and Information Sciences, 2021. **33**(10): p. 1177-1185.

[13] Hamid, D., et al., *A Machine Learning in Binary and Multiclassification Results on Imbalanced Heart Disease Data Stream.* Journal of Sensors, 2022. **2022**: p. 1-13.

[14] Basheer*, S., R.M. Mathew, and M.S. Devi, *Ensembling Coalesce of Logistic Regression Classifier for Heart Disease Prediction using Machine Learning.* International Journal of Innovative Technology and Exploring Engineering, 2019. **8**(12): p. 127-133.

[15] Sun, W., et al., *Prediction of Cardiovascular Diseases based on Machine Learning.* ASP Transactions on Internet of Things, 2021. **1**(1): p. 30-35.

[16] V. Ramalingam, V., A. Dandapath, and M. Karthik Raja, *Heart disease prediction using machine learning techniques : a survey.* International Journal of Engineering & Technology, 2018. **7**(2.8).

[17] Mahesh, T.R., et al., *AdaBoost Ensemble Methods Using K-Fold Cross Validation for Survivability with the Early Detection of Heart Disease.* Comput Intell Neurosci, 2022. **2022**: p. 9005278.

[18] Teso, S., et al., *Leveraging explanations in interactive machine learning: An overview.* Front Artif Intell, 2023. **6**: p. 1066049.

[19] Bello, S.A., et al., *Cloud computing in construction industry: Use cases, benefits and challenges.* Automation in Construction, 2021. **122**.

[20] Kumari, P. and P. Kaur, *A survey of fault tolerance in cloud computing.* Journal of King Saud University - Computer and Information Sciences, 2021. **33**(10): p. 1159-1176.

[21] Cho, S.Y., et al., *Pre-existing and machine learning-based models for cardiovascular risk prediction.* Sci Rep, 2021. **11**(1): p. 8886.