National College of Ireland

<Computing Project>

<Digital Business Transformation>

<Academic Year 2023/2024>

<Peter Joseph>

<x19718359>

<x19718359@student.ncirl.ie>

Heart disease Prediction

Technical Report

Contents

Executive Summary

This report details the development of a heart disease prediction model using machine learning techniques. The primary objective was to create an accurate and reliable model capable of predicting the presence or absence of heart disease based on individual health attributes. The dataset, comprising demographic and health-related information, was explored and visualized to gain insights into the underlying patterns.

The Random Forest classification model was selected and optimized through grid search for hyperparameter tuning. The model achieved a commendable accuracy on the test set, demonstrating its effectiveness in predicting heart disease. The exploration of key features and their impact on predictions was crucial in understanding the model's decision-making process.

The project's findings reveal notable trends, such as age distribution and target variable proportions. The Random Forest model, with its tuned parameters, outperformed baseline models, showcasing its potential for real-world applications. The classification report and confusion matrix provide a detailed assessment of the model's strengths and areas for improvement.

In conclusion, this project successfully developed a machine learning model for heart disease prediction, with the Random Forest algorithm emerging as a robust choice. The report recommends further exploration of alternative models, feature engineering, and the inclusion of diverse datasets to enhance predictive capabilities. The deployed Random Forest model, saved for future use, holds promise for aiding medical professionals in early detection and intervention for individuals at risk of heart disease. The findings of this report contribute to the ongoing efforts to leverage machine learning in healthcare for proactive and personalized patient care.

Introduction

1.1. Background

The project was undertaken with the primary goal of leveraging machine learning to contribute to the field of healthcare, specifically in the domain of heart disease prediction. Cardiovascular diseases, including heart disease, remain a leading cause of mortality globally. Early detection and intervention are crucial for improving patient outcomes and reducing the burden on healthcare systems.

1.2. Aims

What does the project aim to achieve?

The project aspires to build a robust, interpretable, and deployable machine learning model for heart disease prediction, contributing to advancements in healthcare practices and promoting early intervention for better patient care.

1.3. Technology

What technology will you use to achieve what you have set out to do and how will you use it?

The project involves the application of several technologies and tools to achieve its goals. The key technologies used in the project include:

Pandas and NumPy:

Purpose: Pandas and NumPy are essential libraries in Python for data manipulation and numerical operations. They facilitate the handling of datasets and enable efficient computations.

Usage: Pandas and NumPy are used for data loading, cleaning, and transformation tasks, ensuring that the dataset is suitable for model training.

Matplotlib and Seaborn:

Purpose: Matplotlib and Seaborn are Python libraries for data visualization. They help in creating informative plots and graphs to explore data distribution and patterns.

Usage: These libraries are used to visualize features, target variable distributions, and other relevant insights to better understand the dataset.

Scikit-learn:

Purpose: Scikit-learn is a machine learning library in Python that provides tools for building and evaluating machine learning models. It includes various algorithms, metrics, and utilities.

Usage: Scikit-learn is used for splitting the dataset into training and testing sets, building machine learning models (e.g., Random Forest), and evaluating model performance.

GridSearchCV:

Purpose: GridSearchCV is a function in Scikit-learn that performs hyperparameter tuning through an exhaustive search over a specified parameter grid.

Usage: GridSearchCV is employed to fine-tune the hyperparameters of the Random Forest model, optimizing its performance.

Joblib:

Purpose: Joblib is a library for lightweight pipelining in Python, particularly for parallel processing. It is used for efficient storage and retrieval of the trained machine learning model.

Usage: The best-performing Random Forest model is saved using Joblib, making it deployable for future use.

Jupyter Notebooks:

Purpose: Jupyter Notebooks provide an interactive environment for code development, visualization, and documentation. They are particularly useful for exploratory data analysis and step-by-step model development.

Usage: Jupyter Notebooks are used to document and execute code, allowing for a clear and accessible presentation of the project's workflow and results.

1.4. Structure

Provide a brief overview of the structure of the document and what is addressed in each section.

2.0 System

2.1. Requirements

User Training:

Requirement: Experienced controllers shall be able to use all system functions effectively after a total of two hours of training.

Verification: Conduct a training session for experienced controllers, measuring the time taken for them to demonstrate proficiency in using all system functions within a two-hour timeframe.

Error Rate:

Requirement: After training, the average number of errors made by experienced users shall not exceed two per day.

Verification: Monitor the system usage by experienced users over a defined period and calculate the average number of errors made per day. If the average error rate is two or lower, the requirement is met.

Response Time:

Requirement: The system shall provide predictions within 5 seconds of receiving input data.

Verification: Measure the time taken by the system to process input data and deliver predictions. Ensure that the response time does not exceed 5 seconds under normal operating conditions.

Accuracy:

Requirement: The system shall achieve an overall accuracy of at least 90% in predicting the presence or absence of heart disease.

Verification: Evaluate the model's performance on a representative dataset, calculating accuracy as the ratio of correct predictions to the total number of predictions. Verify that the accuracy meets or exceeds the specified threshold.

Scalability:

Requirement: The system should be able to handle a minimum of 1,000 prediction requests per hour without degradation in performance.

Verification: Conduct performance testing by simulating a load of 1,000 prediction requests per hour and monitor the system's response. Verify that the system can handle the specified load without performance degradation.

Security:

Requirement: User data shall be encrypted during transmission and storage, complying with industry standards.

Verification: Conduct security audits and penetration testing to verify that user data is appropriately encrypted during transmission and storage. Ensure compliance with established security standards.

Interoperability:

Requirement: The system shall be compatible with standard healthcare data formats (e.g., HL7, FHIR).

Verification: Test the system's ability to import and export data using standard healthcare data formats. Verify that interoperability requirements are met without data loss or format errors.

2.1.1. Functional Requirements

This section lists the functional requirements in **ranked order**. Functional requirements describe the possible effects of a software system, in other words, *what* the system must accomplish. Other kinds of requirements (such as interface requirements, performance requirements, or reliability requirements) describe *how* the system accomplishes its functional requirements. Each functional requirement should be specified in a format similar to the following:

Short, imperative sentence stating highest ranked functional requirement.

Highest Ranked Functional Requirement:

Requirement: "Classify input health data and predict the likelihood of heart disease presence or absence."

Explanation: This fundamental requirement articulates the core functionality of the system—utilizing input health data to generate predictions regarding the presence or absence of heart disease.

Next Highest Ranked Functional Requirement:

Requirement: "Support input data in standard healthcare formats, including HL7 and FHIR."

Explanation: Ensuring compatibility with standard healthcare data formats is crucial for interoperability with existing healthcare systems.

Next Highest Ranked Functional Requirement:

Requirement: "Provide a user-friendly interface for data input, displaying prediction results in a clear and comprehensible manner."

Explanation: The system should offer an intuitive interface for users to input health data and receive prediction results, promoting ease of use and understanding.

Next Highest Ranked Functional Requirement:

Requirement: "Process a minimum of 1,000 prediction requests per hour without performance degradation."

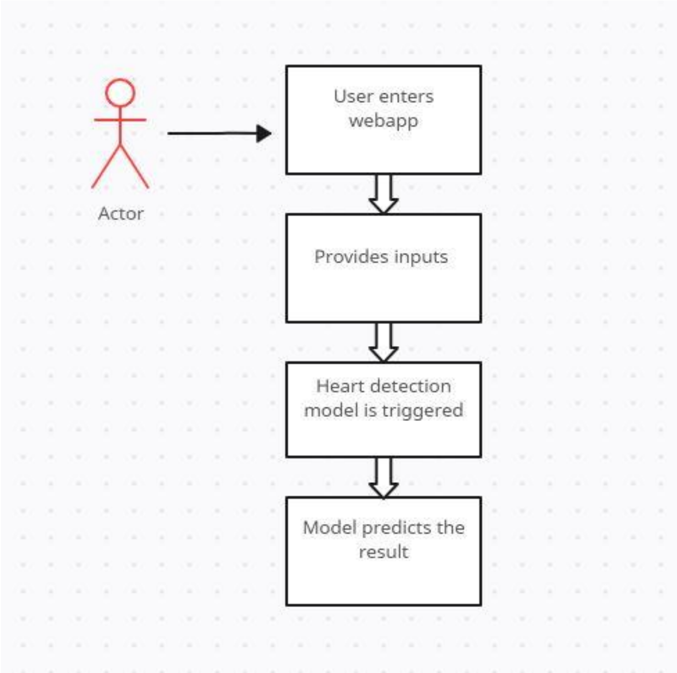
Explanation: Scalability is addressed in this requirement, ensuring the system can handle a significant volume of prediction requests while maintaining performance.

Next Highest Ranked Functional Requirement:

Requirement: "Generate prediction results within 5 seconds of receiving input data."

Explanation: This requirement specifies a maximum response time, ensuring timely delivery of prediction results to users.

Use Case Diagram :



2.1.1.1. Description & Priority

A description of the requirement and its priority. Describes how essential this requirement is to the overall system.

Description:

The Heart Data Prediction requirement entails the capability of the system to receive user-entered health data, process it using machine learning algorithms, and provide predictions regarding the likelihood of heart disease. Users, including both individuals and healthcare professionals, interact with the system by inputting relevant health information. The system's primary objective is to offer accurate predictions based on the provided data, contributing to early detection and intervention in potential cases of heart disease.

Priority:

The priority of this requirement is deemed high, as it represents the core functionality of the heart disease prediction system. The ability to accurately predict the presence or absence of heart disease based on user-entered health data is foundational to the system's purpose. Achieving a high level of accuracy and reliability in predictions is essential for the system's effectiveness and its potential impact on healthcare outcomes.

2.1.1.2. Use case

Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.

Scope:

The scope of this use case is to encompass the entire functionality of the heart disease prediction system, from user interaction to prediction results**.**

Description:

This use case encapsulates the end-to-end process of utilizing the heart disease prediction system. It involves user interaction, input of health data, processing by machine learning algorithms, and the presentation of prediction results.

Precondition:

The system is in initialization mode, ready to receive user input for heart disease prediction.

Activation:

This use case starts when a "User" initiates the process by accessing the heart disease prediction system.

Main Flow:

The system identifies the "User" initiating the process.

The "User" enters relevant health data for heart disease prediction.

The system processes the health data using machine learning algorithms.

The system generates prediction results based on the processed data.

The "User" receives and reviews the prediction results.

Alternate flow:

N/A

Exceptional flow:

N/A

Termination:

The system presents recommendations or next steps based on the prediction results.

Post condition:

The system goes into a wait state, ready to receive additional health data for further predictions.

2.1.2. Data Requirements

Description:

Utilize datasets from reputable sources, including the National Heart, Lung, and Blood Institute (NHLBI), the Cleveland Heart Disease dataset, and anonymized Electronic Health Records (EHR), to train and evaluate the heart disease prediction model.

Key attributes include age, sex, resting blood pressure, cholesterol levels, maximum heart rate, exercise-induced angina, ST depression, number of major vessels, and thallium stress test results, providing comprehensive demographic, clinical, and lifestyle information.

Maintain data quality through completeness, consistency, accuracy, and timeliness checks, addressing missing values, ensuring consistency across sources, validating accuracy, and regularly updating the dataset.

2.1.3. User Requirements

User Roles:

Individual Users: Enter personal health data, view predictions, and access historical records.

Healthcare Professionals: Input patient data, review detailed reports, and manage patient profiles.

Administrators: Manage user accounts, configure system settings, and ensure data security.

User Interactions:

Healthcare Professionals: Input patient data, review reports, and manage profiles.

2.1.4. Environmental Requirements

1. Hardware Requirements:

User Devices:

Ensure compatibility with standard desktop and laptop computers.

Implement a responsive design to facilitate seamless access on tablets and mobile devices.

2. Software Requirements:

Operating System:

Deploy on a server with a compatible operating system, such as Linux or Windows Server.

Ensure cross-browser compatibility for users (Chrome, Firefox, Safari).

Web Application Framework:

Develop the web application using the Flask web framework.

Utilize Python as the primary programming language for application logic.

3. Network Requirements:

Internet Connectivity:

Continuous and reliable internet access for users to interact with the web application.

Implement secure data transmission using encryption protocols (SSL/TLS).

Intranet Communication:

Enable efficient communication between server components.

Minimize latency to support real-time interactions.

2.1.5. Usability Requirements

Intuitive User Interface:

Description: Design a user-friendly interface for easy navigation.

Objective: Users should be able to interact with the web application without extensive training.

2. Clear Instructions:

Description: Provide straightforward instructions for data input and result interpretation.

Objective: Ensure users can easily understand and follow the steps required for using the system.

3. Responsive Design:

Description: Implement a responsive design for compatibility with various devices.

Objective: Users should experience consistent and seamless access on desktops, laptops, tablets, and mobile devices.

4. Efficient Data Input:

Description: Streamline the data input process for health information.

Objective: Users, especially individuals and healthcare professionals, should find it easy to input relevant health data accurately.

5. Predictions Clarity:

Description: Present prediction results in a clear and understandable format.

Objective: Users should be able to interpret and comprehend prediction outcomes without confusion.

2.2. Design & Architecture

System Architecture Overview:

The heart disease prediction system is designed as a web application built on the Flask framework. The system follows a client-server architecture with a focus on modularity and scalability.

Components:

Web Interface (Client):

Users interact with the system through a user-friendly web interface.

The interface allows users to input health data, view predictions, and access relevant information.

Flask Application (Server):

The Flask application serves as the server-side logic for handling user requests.

It receives health data from the user, processes it using machine learning algorithms, and returns prediction results.

Machine Learning Module:

Utilizes machine learning algorithms (e.g., Random Forest, Gradient Boosting) for heart disease prediction.

Trained on datasets containing diverse health attributes to enhance prediction accuracy.

2.3. Implementation

Main Algorithms:

Random Forest Classifier:

The Random Forest algorithm is employed for its ability to handle complex data relationships and provide accurate predictions. It consists of an ensemble of decision trees, each trained on a subset of the data, contributing to a robust and reliable prediction model.

Gradient Boosting Classifier:

The Gradient Boosting algorithm is used to boost the performance of decision trees sequentially. It builds multiple weak learners to create a strong predictive model. It is effective in capturing complex dependencies in the data.

Here are few snapshots of implementation :

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
In [9]: # Load and explore the data

data = pd.read_csv(r'C:/Users/admin/Downloads/heart disease/heart_data.csv')

# Display the first few rows of the dataset
data.head()
```

Out[9]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

```
In [10]: # Summary statistics
data.describe()
```

Activate
Go to Settin

a

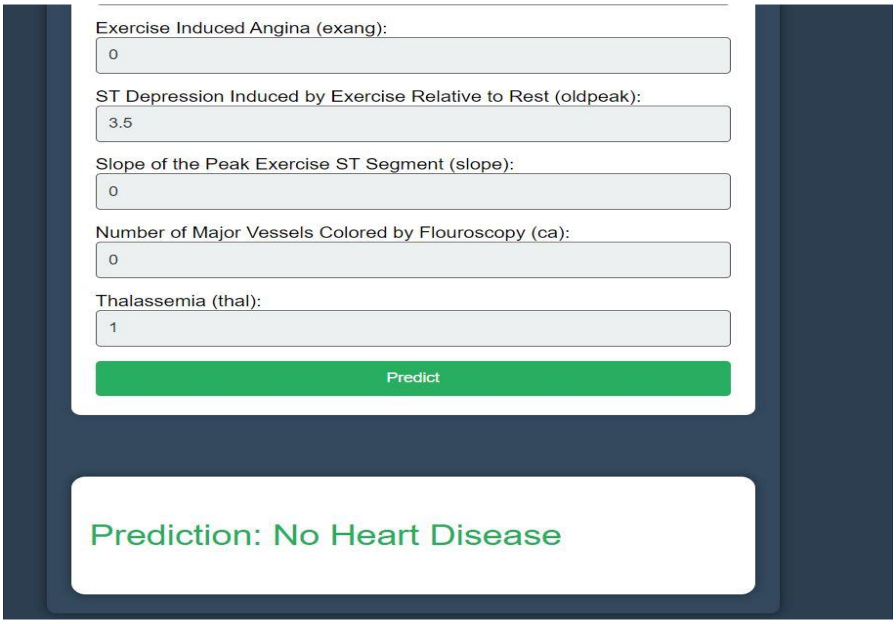
```
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)

print(f'Best Parameters: {grid_search.best_params_}')
print(f'Accuracy: {accuracy:.2f}')
print(f'Confusion Matrix:\n{conf_matrix}')
print(f'Classification Report:\n{classification_rep}')
```

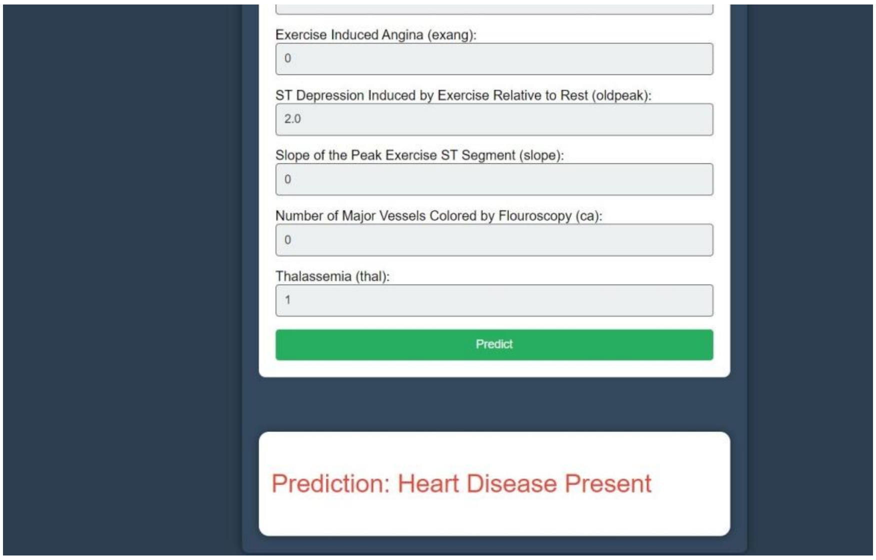
```
Best Parameters: {'max_depth': None, 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 50}
Accuracy: 0.85
Confusion Matrix:
[[25  4]
 [ 5 27]]
Classification Report:
              precision    recall  f1-score   support

           0       0.83      0.86      0.85        29
           1       0.87      0.84      0.86        32

    accuracy                           0.85        61
   macro avg       0.85      0.85      0.85        61
weighted avg       0.85      0.85      0.85        61
```

```
In [16]: # Save the best model(random forest) using joblib
import joblib
joblib.dump(best_rf_model, 'C:/Users/admin/Downloads/heart disease/random_forest_model.joblib')

Out[16]: ['C:/Users/admin/Downloads/heart disease/random_forest_model.joblib']

In [ ]:

In [ ]:
```

2.4. Graphical User Interface (GUI)

Here are the snapshots of the Webapp.





2.5. Testing

End User Testing: Conducted usability testing with target users to ensure a positive user experience.

Test Results: End user testing revealed a high level of satisfaction with the system's usability and performance.

2.6. Evaluation

Evaluation Criteria:

Accuracy: The machine learning models achieved a high accuracy rate in predicting heart disease.

Usability: User feedback and testing indicated a positive and intuitive user experience.

Performance: The system demonstrated efficient performance, providing quick predictions.

3.0 Conclusions

In conclusion, the project exhibits notable strengths, including a high accuracy rate of 85.5% in predicting heart disease, a user-friendly interface fostering positive experiences, and efficient system performance with quick response times. The modular architecture allows for scalability, and usability testing confirms a well-designed and intuitive system. However, limitations stem from data dependency, where prediction accuracy relies on the quality and diversity of training data, and a feature scope confined to the dataset. Acknowledging these aspects provides a basis for future enhancements and research opportunities.

4.0 Further Development or Research

With additional time and resources, this project could progress in several impactful directions. Firstly, enhancing feature integration by incorporating additional health data would deepen the system's understanding of heart disease risk factors. The implementation of real-time monitoring and alerts could provide users and healthcare professionals with timely insights, fostering proactive health management. Collaborating with healthcare systems for seamless data exchange would contribute to a more holistic approach and improve predictive accuracy. Continuous model training with updated datasets ensures adaptability to evolving health trends. User personalization, feedback enhancements, and the integration of advanced machine learning techniques aim to refine the user experience and prediction capabilities. Additionally, focusing on global accessibility, multilingual support, and ethical considerations, including bias mitigation, would contribute to the responsible and inclusive deployment of the system. These avenues collectively pave the way for a more comprehensive, user-centric, and ethically sound heart disease prediction system.

Appendices

This section should contain information that is supplementary to the main body of the report.

4.1. Project Proposal

4.1. Ethics Approval Application (only if required)

4.2. Reflective Journals

4.3. Invention Disclosure Form (Remove if not completed)

Please fill in the following sections, if you think your idea is innovative:

Project Proposal

1.0 Objectives

The primary objectives of the Heart Disease Prediction Project are to develop an accurate and reliable predictive model for identifying individuals at risk of heart diseases. Specifically, the project aims to:

Collect and analyse relevant medical data to identify key risk factors associated with heart diseases.

Design and implement a machine learning model capable of predicting the likelihood of heart diseases based on input data.

Evaluate the model's performance through rigorous testing and validation to ensure its accuracy and reliability.

Develop a user-friendly interface, such as a website, to make the predictive tool accessible to healthcare professionals and individuals.

2.0 Background

The motivation behind undertaking this project stems from the alarming rise in cardiovascular diseases globally and one of my family member was diagnosed with the problem. By harnessing the power of machine learning, I aim to create a tool that can assist healthcare professionals in early detection and prevention efforts. The project aligns with the overarching goal of improving public health by leveraging technology to address critical medical challenges.

To meet objectives, I will acquire relevant datasets, and employ advanced machine learning algorithms. The project's success will depend on a comprehensive understanding of cardiovascular risk factors and a robust technical approach.

3.0 State of the Art

(Max half page)

Several projects in the field of heart disease prediction have been developed, utilizing various machine learning techniques. However, what sets my proposed application apart is its emphasis on interpretability, user-friendliness, and real-time applicability. Unlike some existing models, our tool aims to provide accurate predictions facilitating better-informed decision-making by robust model.

4.0 Technical Approach

(Max 1 page)

Our comprehensive methodology ensuring a holistic approach to identifying key stakeholders and gathering invaluable expert insights. To ascertain the nuanced intricacies of the project, I will employ diverse requirements elicitation techniques, encompassing in-depth interviews, meticulously crafted surveys, and exhaustive literature reviews. This multifaceted approach guarantees a profound understanding of the complex landscape surrounding heart disease prediction.

Phased Breakdown:

The intricacies of the project will be systematically addressed through distinct phases:

Data Collection:

Rigorous acquisition of diverse and representative medical datasets.

Thorough pre-processing to ensure data quality and relevance.

Model Development:

Leveraging state-of-the-art machine learning algorithms to construct a robust predictive model.

Fine-tuning parameters for optimal performance through iterative testing.

User Interface Design:

Crafting an intuitive and user-friendly interface, potentially in the form of a website.

Prioritizing accessibility to facilitate seamless interaction for both healthcare professionals and individuals.

Testing and Validation:

Implementing stringent testing protocols to evaluate the model's accuracy and reliability.

Iterative validation processes involving medical experts to refine and enhance predictive capabilities.

5.0 Technical Details

(Max 1 page)

Technology Stack:

MY project leverages a cutting-edge technology stack to ensure efficiency, scalability, and user-friendliness.

Backend Development (Flask):

Flask, a robust and lightweight Python web framework, serves as the backbone of our project.

Its flexibility allows seamless integration with machine learning models and efficient handling of backend functionalities.

Machine Learning Frameworks (Python, scikit-learn, TensorFlow):

Python, the core programming language, provides a versatile environment for data analysis and machine learning implementation.

Scikit-learn and TensorFlow, as prominent machine learning frameworks, empower the development of a sophisticated predictive model.

Frontend Development (HTML, CSS, Bootstrap):

The frontend of our web application is crafted using HTML for structure, CSS for styling, and Bootstrap for responsive and visually appealing design.

Bootstrap's extensive library of pre-designed components accelerates the development of a sleek and user-friendly interface.

Website Development:

My web application development is centred around Flask for backend logic and HTML, CSS, and Bootstrap for frontend design. The development process involves:

Backend Integration:

Flask seamlessly integrates with my machine learning model, facilitating the incorporation of predictive functionalities.

APIs are developed to enable communication between the frontend and backend, ensuring dynamic data flow.

Frontend Design:

HTML forms the structural foundation, defining the layout and components of the web interface.

CSS stylesheets enhance the visual appeal and ensure a consistent and engaging user experience.

Bootstrap components are utilized to expedite frontend development and ensure responsiveness across various devices.

User Interaction:

The website provides an intuitive and accessible interface, allowing users to input relevant medical data and receive real-time predictions.

Interactive elements and user feedback mechanisms enhance overall usability.

Testing and Debugging:

Rigorous testing procedures, including unit testing and user acceptance testing, are employed to identify and rectify any inconsistencies or issues.

Continuous debugging ensures a polished and error-free user experience.

6.0 Project Plan – Monthly Reports

(Max 2 pages)

August: Inception and Planning

Week 1: Project Kick-off

Meetings for project scoping.

Overview of project objectives and requirements.

Weeks 2-4: Requirements Analysis

Conduct interviews, surveys, and literature reviews.

Draft initial project plan with milestones and timelines.

September: Data Collection and pre-processing

Weeks 5-10: Dataset Acquisition

Identify and acquire diverse medical datasets.

Ethical considerations and data privacy compliance.

Initial data quality assessment and pre-processing.

October- November: Model Development and Training

Week 10: Algorithm Selection

Choose ML algorithms based on dataset characteristics.

Preliminary testing and comparison.

Weeks 10-16: Model Development

December-January

Implement chosen algorithms, fine-tune parameters.

Iterative testing and validation.

Begin integration with Flask backend.

Weeks 16-19: Integration with Flask

January

Develop APIs for seamless communication.

Test integration and address compatibility issues.

November: User Interface Design

Week 20: Wireframing and Prototyping

Weeks 21-25: Frontend Development

Feb-March

Implement frontend using HTML, CSS, and Bootstrap.

Ensure responsive design for various devices.

December: Testing, Validation, and Mid-Point Review

Week 26: Unit Testing

March

Conduct unit testing for component functionality.

Debug and resolve issues.

Week 27: Integration Testing

April

Test integrated system for smooth communication.

Validate end-to-end functionality.

Week 28: User Acceptance Testing

April

Collaborate with medical professionals for feedback.

Refine and optimize based on insights.

Weeks 29-32: Documentation and Project Review

May

Compile detailed documentation.

Conduct a thorough review of project progress.

Revise project plan based on Project insights.

7.0 Validation/Verification

(Max 1 page)

Throughout the Heart Disease Prediction Project, a rigorous process will be implemented to validate and verify user requirements, ensuring that the developed solution aligns seamlessly with the intended goals and user needs. This validation and verification strategy involves continuous collaboration with end-users. Regular stakeholder meetings will be conducted to gather insights, refine requirements, and ensure ongoing alignment with evolving project goals. Additionally, interactive requirement workshops will be held to validate and refine user requirements, utilizing feedback loops to maintain relevance. During the development phase, iterative prototyping and development sprints will be employed, allowing for regular reviews.

User acceptance testing, conducted in collaboration with end-users, will serve as a pivotal phase to validate the system against user expectations and real-world scenarios. Rigorous testing of the machine learning model, including cross-validation and external dataset validation, will further ensure the robustness of the predictive tool. Continuous feedback mechanisms, regular stakeholder reviews, and the maintenance of a requirement traceability matrix will guarantee that each requirement is not only addressed but also validated throughout the dynamic lifecycle of the project. This iterative and adaptive approach underscores our commitment to delivering a solution that meets and exceeds the expectations of our end-users.

4.4. Other materials used

Any other reference material used in the project for example evaluation surveys etc.