Software Requirements Specification

for

Prediction of Heart Attack Using Machine Learning Algos and Deep Learning

Version 1.0

Prepared by

Bindu Devidas

Instructor: *<place your instructor's name here>*

Course: Agile Software Engineering

Lab Section: *<place your lab section here>*

Teaching Assistant: *<place your TA's name here>*

Date: 6th October,2023

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
Draft Type and Number	Full Name	Information about the revision. This table does not need to be filled in whenever a document is touched, only when the version is being upgraded.	00/00/00

1 Introduction

<TO DO: Please provide a brief introduction to your project and a brief overview of what the reader will find in this section.>

1.1 Document Purpose

The purpose of this document is to capture the functional requirements and requirement analysis

1.2 Product Scope

Inclusion of Machine Learning Algorithms: Decision Tree, Random Forest, SVM, Logistic Regression, KNN, and Naïve Bayes implementations for heart attack prediction.

Deep Learning Component: Implementation of a Deep Neural Network (DNN) for heart attack prediction, incorporating tuning of parameters such as neuron count, epochs, hidden layers, and activation functions.

Model Tuning: Experimentation with various parameters (e.g., training and test data sizes, epochs, activation functions) to achieve maximum accuracy for both machine learning and deep learning models.

1.3 Intended Audience and Document Overview

- *Product Manager need for reviewing requirements*
- *Devops engineer need this doc for providing tools*
- *Project manager need this to do project plan, project costing and scheduling*
- *Developers need this document for doing design*
- *Testers need this document for writing test plan,test cases, test scripts and test script automation*

1.4 Definitions, Acronyms and Abbreviations

Definitions:

- ***Heart attack**: A medical condition that occurs when blood flow to the heart muscle is blocked, often resulting in chest pain, shortness of breath, and other symptoms.*
- ***Machine learning**: A field of artificial intelligence where computer systems are trained to learn and improve performance on a specific task without being explicitly programmed.*
- ***Deep Learning (DL):** A subset of machine learning based on artificial neural networks with multiple layers, allowing the system to learn complex patterns and representations.*
- ***Data Preprocessing:** The process of cleaning, transforming, and organizing raw data into a format suitable for analysis and modeling.*
- ***Feature Extraction:** The process of selecting and transforming relevant features from the raw data to be used in model training.*
- ***Prediction Model:** A mathematical or computational model developed through machine learning or deep learning to predict outcomes based on input parameters.*

- ***User Interface (UI):*** *The graphical or textual interface through which users interact with the heart attack prediction system.*

Acronyms and Abbreviations

- *SRS - System Requirements Specification*
- *UI - User Interface*
- *ML - Machine Learning*
- *DL - Deep Learning*
- *QA - Quality Assurance*
- *DB - Database*
- *API - Application Programming Interface*
- *CSV - Comma-Separated Values (file format)*

1.5 Document Conventions

NA

1.6 References and Acknowledgments

NA

Overall Description

1.7 Product Overview

The product, "Prediction of Heart Attack Using Machine Learning Algorithms and Deep Learning," aims to create a robust and accurate prediction system for identifying the likelihood of a heart attack in individuals. This involves utilizing both traditional machine learning algorithms and deep learning techniques to provide early detection and intervention opportunities.

1.8 Product Functionality

The product, "Prediction of Heart Attack Using Machine Learning Algorithms and Deep Learning," encompasses several key functionalities aimed at accurate heart attack prediction using diverse machine learning and deep learning approaches.
TO DO:
Machine Learning Algorithm Implementation

Decision Tree: Utilize a decision tree algorithm to create a predictive model based on features from the Heart Attack Dataset

Random Forest: Implement a random forest classification algorithm to enhance prediction accuracy through an ensemble of decision trees.

Support Vector Machine (SVM): Develop a SVM-based predictive model to classify and predict heart attacks based on the dataset.

Logistic Regression: Implement logistic regression to model the probability of a heart attack occurrence, providing valuable insights.

K-Nearest Neighbors (KNN): Utilize the KNN algorithm to classify individuals based on their proximity in feature space for heart attack prediction.

Naïve Bayes: Implement the Naïve Bayes algorithm to classify individuals as prone to a heart attack using probabilistic models.

Deep Learning Classification

Deep Neural Network (DNN): Implement a multi-layered neural network, optimizing parameters like neuron count, epochs, hidden layers, and activation functions for accurate heart attack prediction.

3. Data Preprocessing and Cleaning:

Data Cleaning: Handle missing or inconsistent data, ensuring the dataset is clean and ready for training.

Feature Engineering: Process and transform features to extract relevant information, enhancing the performance of the algorithms.

4. Model Tuning and Optimization:

Parameter Tuning: Experiment with various parameters such as training and test data sizes, epochs, hidden layers, and activation functions to optimize model performance.

Accuracy Enhancement: Continuously improve model accuracy by fine-tuning parameters and optimizing the training process.

1.9 Design and Implementation Constraints

While developing the "Prediction of Heart Attack Using Machine Learning Algorithms and Deep Learning" product, there are certain design and implementation constraints that need to be considered to ensure the feasibility, efficiency, and effectiveness of the project.

Hardware Limitations:
Availability of sufficient computing power (CPU, GPU) may be a constraint for running complex deep learning models, especially with large datasets and extensive parameter tuning.

Dataset Quality and Completeness:
The availability of a comprehensive and accurate Heart Attack Dataset is crucial for developing reliable prediction models. Incomplete or inaccurate data can adversely affect the performance and validity of the models.

1.10 Assumptions and Dependencies

1. Assumptions:
Data Reliability:

Assuming that the Heart Attack Dataset used for training and testing the models is reliable, accurate, and represents a diverse range of cases.
Feature Relevance:

Assuming that the selected features in the dataset are relevant to predict heart attacks and that feature engineering has been appropriately performed to enhance predictive capabilities.
Consistent Data Availability:

Applicability of Tuning:

Assuming that parameter tuning for both machine learning and deep learning models will result in improved accuracy and performance.
Ethical Data Usage:

Assuming that the data used for training and testing is acquired and utilized ethically, adhering to privacy and legal guidelines.

2. Dependencies:
Availability of Data:

The project is dependent on the availability and accessibility of the Heart Attack Dataset, which is essential for training and testing the machine learning and deep learning models.
Availability of Computing Resources:

Availability of adequate computing resources, including hardware and software, is crucial for training and running the machine learning and deep learning models efficiently.
Expertise and Skillsets:

The project depends on the availability of skilled data scientists, machine learning engineers, and deep learning experts with expertise in implementing and tuning machine learning and deep learning algorithms.
Access to Machine Learning and Deep Learning Frameworks:

The project is dependent on the availability and proper functioning of machine learning and deep learning frameworks like scikit-learn, TensorFlow, or PyTorch.
Adherence to Regulatory Guidelines:

The project depends on adherence to relevant regulatory guidelines and ethical considerations concerning the use of healthcare data.
Time and Project Management:

Successful completion of the project is dependent on effective time management, task coordination, and adherence to the project schedule and milestones.

2 Specific Requirements

2.1 External Interface Requirements

2.1.1 User Interfaces

The user interface (UI) of the "Prediction of Heart Attack Using Machine Learning Algorithms and Deep Learning" product plays a crucial role in ensuring a seamless and user-friendly experience for users interacting with the system. The UI should be intuitive, informative, and allow users to efficiently interact with the prediction system. Here are the key user interface requirements

Data Entry:
The UI should provide a section for users to input the relevant data features required for heart attack prediction.

Data Validation:
Validate user inputs to ensure they meet the required format and constraints.

Feature Explanation:
Display brief explanations or tooltips for each feature, assisting users in providing accurate data.

2.1.2 Hardware Interfaces

The "Prediction of Heart Attack Using Machine Learning Algorithms and Deep Learning" product requires specific hardware interfaces to support the processing, storage, and efficient functioning of the system. These hardware interfaces ensure compatibility and optimal performance. Here are the hardware interface requirements:

2.1.3 Software Interfaces

The "Prediction of Heart Attack Using Machine Learning Algorithms and Deep Learning" product requires integration with various software components and platforms to support the development, deployment, and functionality of the system. These software interfaces are crucial for data processing, model development, user interface design, and more. Here are the software interface requirements:

2.2 Functional Requirements

1. *Data Preprocessing: The system should clean and preprocess the input data, handling missing values, outliers, and formatting inconsistencies.*
2. *Model Training: Train machine learning models including Decision Tree, Random Forest, SVM, Logistic Regression, KNN, and Naïve Bayes using the preprocessed data.*
3. *Deep Neural Network (DNN) Training: Train a Deep Neural Network (DNN) using the preprocessed data, optimizing parameters such as neuron count, epochs, hidden layers, and activation functions.*
4. *Prediction Generation: Generate predictions for heart attack likelihood based on the selected algorithm and user-provided data.*

2.3 Use Case Model

TO DO: Provide a use case diagram that will encapsulate the entire system and all actors.

2.3.1 Use Case #1 (use case name and unique identifier – e.g. U1)

TO DO: Provide a specification for each use case diagram

Author – Identify team member who wrote this use case

Purpose - What is the basic objective of the use-case. What is it trying to achieve?

Requirements Traceability – Identify all requirements traced to this use case

Priority - What is the priority. Low, Medium, High. Importance of this use case being completed and functioning properly when system is deployed

Preconditions - Any condition that must be satisfied before the use case begins

Post conditions - The conditions that will be satisfied after the use case successfully completes

Actors – Actors (human, system, devices, etc.) that trigger the use case to execute or provide input to the use case

Extends – If this is an extension use case, identify which use case(s) it extends

Flow of Events

1. Basic Flow - flow of events normally executed in the use-case
2. Alternative Flow - a secondary flow of events due to infrequent conditions
3. Exceptions - Exceptions that may happen during the execution of the use case

Includes (other use case IDs)

Notes/Issues - Any relevant notes or issues that need to be resolved

3 Other Non-functional Requirements

3.1 Performance Requirements

Performance requirements outline the expected performance characteristics and constraints that the "Prediction of Heart Attack Using Machine Learning Algorithms and Deep Learning" product should meet. These requirements ensure that the system operates efficiently and provides a satisfactory user experience. Here are the performance requirements:

1. Response Time:
1.1 Algorithm Selection:

The system should respond to the user's algorithm selection within 1 second.
1.2 Prediction Generation:

The prediction generation process, including model inference, should take less than 2 seconds.
2. Scalability:
2.1 Concurrent Users:

The system should support at least 100 concurrent users without a significant degradation in response time.
2.2 Scalability with Data Size:

The system should handle datasets of varying sizes, from a minimum of 100 records to a maximum of 10,000 records, while maintaining acceptable response times.
3. Model Training:
3.1 Machine Learning Model Training:

Training the machine learning models should not take more than 5 minutes for any given dataset.
3.2 Deep Learning Model Training:

Training the deep learning models should not exceed 30 minutes for any given dataset.
4. Prediction Accuracy:
4.1 Machine Learning Model Accuracy:

The accuracy of machine learning models should be at least 85% on average.
4.2 Deep Learning Model Accuracy:

The accuracy of deep learning models should be at least 90% on average.

3.2 Safety and Security Requirements

Regulatory Compliance:
Ensure compliance with relevant healthcare data privacy laws and regulations, such as the Health Insurance Portability and Accountability Act (HIPAA) in the United States.

Code Security:

Adhere to secure coding practices to mitigate potential vulnerabilities such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).

3.3 Software Quality Attributes

Software quality attributes, also known as non-functional requirements, describe the overall quality characteristics that the "Prediction of Heart Attack Using Machine Learning Algorithms and Deep

Learning" product should possess. These attributes are crucial for evaluating the system's performance, reliability, usability, and maintainability. Here are the software quality attributes:

1. Reliability:
1.1 System Stability:

The system should operate reliably and consistently, ensuring minimal downtime or disruptions.
1.2 Error Handling:

Implement robust error handling mechanisms to gracefully handle unexpected errors and failures.
2. Performance:
2.1 Responsiveness:

The system should respond promptly to user interactions and requests for predictions.
2.2 Throughput:

Ensure that the system can handle a significant number of requests simultaneously without performance degradation.
2.3 Efficiency:

Optimize system performance to utilize computing resources efficiently and minimize response times.

3. Usability:
3.1 Intuitive User Interface:

Design a user-friendly interface that is easy to navigate and understand for a diverse user base.
3.2 Learnability:

Ensure that users can quickly learn how to use the system effectively without extensive training.
3.3 Accessibility:

Design the system to be accessible to users with disabilities, following relevant accessibility standards.
4. Scalability:
4.1 Horizontal Scalability:

Design the system to scale horizontally to accommodate a growing user base and increasing data volume.
4.2 Vertical Scalability:

Design the system to scale vertically to handle increased computational requirements, such as model complexity.
5. Maintainability:
5.1 Modularity:

Design the system with clear and distinct modules, promoting ease of maintenance and future enhancements.

5.2 Code Readability:

Adhere to coding standards and practices that enhance code readability, aiding maintainability.
6. Security:
6.1 Authorization and Authentication:

Ensure that access to the system and data is controlled through secure and robust authentication and authorization mechanisms.
6.2 Data Encryption:

Implement encryption protocols to protect sensitive user data during storage and transmission.
6.3 Secure API:

Secure APIs to prevent unauthorized access and ensure data integrity.