# Software Requirements Specification

for

# Prediction of Heart Attack Using Machine Learning and Deep Learning Algorithms

**Version 1.0**

**Prepared by**


| | | |
|---|---|---|
| **K vidhyasanjana** | **1RVU22CSE074** | **vidhyask.btech22@rvu.edu.in** |


| | |
|---|---|
| **Instructor:** Prof. CVSV Reddy | |
| **Course:** | **Agile software Engineering** |
| **Lab Section:** | ***C*** |
| **Teaching Assistant:** | **Divya M** |
| **Date:** | **06-10-2023** |

# Contents



# 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 SRS document to capture the requirements and requirements analysis

## 1.2 Product Scope

Inclusion of Machine Learning Algorithms: Decision Tree, Random Forest, SVM, Logistic Regression, KNN, and Naive 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

*Developers needs this document for doing design.*
*Testers needs the document for writing Test plan*
*Test Cases*
*Test script*
*Test script Automation*
*Project manager needs this document to view requirements*
*Devops engineer needs this document to provide tools*
*Project manager needs this document for product costing and schedule.*

## 1.4 Definitions, Acronyms and Abbreviations

SRS: software Requirement specifications.
SOW: statement of work.
ML: Machine Learning
DL: Deep Learning
KNN: K-Nearest Neighbors
SVM: Support Vector Machine

## 1.5 Document Conventions

*NA*

## 1.6 References and Acknowledgments

*NA*

# 2 Overall Description

## 2.1 Product Overview

The project, "Prediction of Heart Attack Using Machine Learning Algorithms and Deep Learning." aims to create accurate prediction system for identifying the likelihood of a heart attack in individuals. the focus here is on the profound implications of early detection and intervention using machine learning algorithms and deep learning techniques.

## 2.2 Product Functionality

In the journey towards creating an accurate heart attack prediction system, we have outlined a series of core functionalities:

**Machine Learning Algorithm Implementation**

1. **Decision Tree**: We will employ a decision tree algorithm to construct a predictive model based on the characteristics derived from the Heart Attack Dataset. This model will help us understand the factors influencing heart attack occurrences.
2. **Random Forest**: Leveraging the power of ensemble learning, we will implement a random forest classification algorithm. This ensemble of decision trees will significantly enhance prediction accuracy, providing a more robust model.
3. **Support Vector Machine (SVM)**: We will develop a predictive model using Support Vector Machines (SVM). This SVM-based model will classify and predict heart attacks based on the patterns within the dataset.
4. **Logistic Regression**: By implementing logistic regression, we will create a model that estimates the probability of a heart attack occurrence. This model will offer valuable insights into the likelihood of heart attacks.
5. **K-Nearest Neighbors (KNN)**: The KNN algorithm will be utilized to classify individuals based on their proximity in the feature space. This proximity-based classification will enhance our ability to predict heart attacks accurately.
6. **Naïve Bayes**: The Naïve Bayes algorithm will be put to use to classify individuals as prone to a heart attack. This probabilistic model will aid in identifying those at higher risk.

**Deep Learning Classification**

**Deep Neural Network (DNN)**: We will implement a multi-layered neural network, optimizing critical parameters like neuron count, epochs, hidden layers, and activation functions. This fine-tuning process will result in a highly accurate heart attack prediction model.

**Data Preprocessing and Cleaning**

**Data Cleaning**: Ensuring the integrity of our data is paramount. We will handle missing or inconsistent data meticulously to guarantee that our dataset is pristine and ready for training.
**Feature Engineering**: To extract the most relevant information, we will engage in feature engineering. This process will transform our features, enhancing algorithm performance and predictive capabilities.

**Model Tuning and Optimization**

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

**Accuracy Enhancement**: Our journey towards accuracy enhancement will be continuous. We will diligently fine-tune parameters and optimize the training process to ensure the highest accuracy levels.

These functionalities form the backbone of our project, collectively contributing to the development of a robust and reliable heart attack prediction system.

## 2.3 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:**
One key challenge we face is the availability of adequate computing power, including CPUs and GPUs. This becomes especially important when we work with complex deep learning models, handle large datasets, and fine-tune various model parameters.

**Data Quality and Completeness:**
Another critical factor is the quality and completeness of our Heart Attack Dataset. The accuracy of our predictions relies heavily on the reliability of the data we use. Any issues, such as missing or incorrect data, can negatively affect the performance and trustworthiness of our prediction models. Our success in this project lies in our ability to address these challenges thoughtfully, ensuring they do not hinder our progress toward achieving our goals.

## 2.4 Assumptions and Dependencies

***Assumptions:***

1. ***Data Reliability:*** *We're assuming that the Heart Attack Dataset we use for training and testing our models is not only reliable but also accurate. We also assume that this dataset represents a diverse range of cases, ensuring our predictions are applicable to various scenarios.*
2. ***Feature Relevance:*** *We assume that the features we've chosen from the dataset are relevant for predicting heart attacks. Additionally, we trust that our feature engineering efforts will effectively enhance our models' predictive capabilities.*
3. ***Consistent Data Availability:*** *Our assumptions include a consistent supply of data throughout the project, ensuring that we can train and test our models without interruptions.*
4. ***Applicability of Tuning:*** *We assume that the tuning of parameters for both our machine learning and deep learning models will lead to improved accuracy and overall performance.*
5. ***Ethical Data Usage:*** *We're committed to ethical data usage, assuming that the data we employ for training and testing is acquired and handled ethically, adhering to strict privacy and legal guidelines.*

***Dependencies:***

1. ***Availability of Data:*** *The project relies heavily on the availability and accessibility of the Heart Attack Dataset. This dataset is a fundamental element for training and testing our machine learning and deep learning models.*

2. ***Availability of Computing Resources:*** *Adequate computing resources, including both hardware and software, are essential for the efficient training and execution of our machine learning and deep learning models.*
3. ***Expertise and Skillsets:*** *The success of the project depends on the availability of skilled professionals, including data scientists, machine learning engineers, and deep learning experts. Their expertise in implementing and fine-tuning machine learning and deep learning algorithms is crucial.*
4. ***Access to Frameworks:*** *We rely on the availability and proper functioning of machine learning and deep learning frameworks like scikit-learn, TensorFlow, or PyTorch to facilitate the development and execution of our models.*
5. ***Adherence to Guidelines:*** *We are dependent on strict adherence to regulatory guidelines and ethical considerations regarding the use of healthcare data. Ensuring that we meet all ethical and legal requirements is paramount.*
6. ***Time and Project Management:*** *The successful completion of our project relies on effective time management, coordinated task execution, and strict adherence to the project schedule and milestones.*

# 3 Specific Requirements

## 3.1 External Interface Requirements

### 3.1.1 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.*

### 3.1.2 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:*

### 3.1.3 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:*

## 3.2 Functional Requirements

***1.*** **Data Preprocessing***: The system must efficiently clean and preprocess the input data, addressing issues such as missing values, outliers, and formatting inconsistencies. This ensures that the data is in a suitable state for modelling.*
***2. Model Training:*** *Our system should be capable of training various machine learning models, including Decision Tree, Random Forest, SVM, Logistic Regression, KNN, and Naïve Bayes. These models will utilize the pre-processed data to learn patterns related to heart attack prediction.*
***3. Deep Neural Network (DNN) Training:*** *In addition to traditional machine learning models, our system should also have the capability to train a Deep Neural Network (DNN). This involves optimizing parameters such as the number of neurons, epochs, hidden layers, and activation functions to create a powerful predictive model.*
***4. Prediction Generation:*** *The system's primary purpose is to generate predictions regarding the likelihood of a heart attack occurrence. It should take user-provided data as input and employ the selected algorithm to produce accurate predictions.*

***3.2.1 Data Preprocessing :***
*The system shall clean and preprocess input data to handle missing values, outliers, and formatting inconsistencies. Data preprocessing should include:*
- *Imputation of missing values using appropriate techniques.*
- *Identification and handling of outliers to ensure they do not adversely affect model training.*
- *Consistent formatting of data to remove any inconsistencies that might hinder model performance.*

***3.2.2 Model Training :***
*The system shall be capable of training multiple machine learning models, namely Decision Tree, Random Forest, SVM, Logistic Regression, KNN, and Naïve Bayes. Model training should involve:*
- *Data partitioning into training and testing sets.*
- *Application of the selected machine learning algorithms on the training data.*
- *Optimization of algorithm-specific parameters for improved accuracy.*
- *Model evaluation on the testing data to assess performance.*

***3.2.3 DNN Training :***
*The system shall also facilitate the training of a Deep Neural Network (DNN) using the pre-processed data. DNN training should encompass:*
- *Tuning of parameters, including neuron count, epochs, hidden layer architecture, and activation functions.*
- *Proper initialization of neural network weights and biases.*
- *Monitoring and logging of training progress, including loss and accuracy metrics.*

***3.2.4 Prediction Generation (F4):***
*The primary objective of the system is to generate predictions for heart attack likelihood based on the selected algorithm and user-provided data. Prediction generation should include:*
- *Acceptance of user input data for prediction.*
- *Application of the selected algorithm to the input data.*

- *Generation of a prediction, typically in the form of a probability score or classification label.*

## 3.3 Use Case Model

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

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

- **Author**: [Author Name]
- **Purpose**: The objective of this use case is to predict the likelihood of a heart attack for an individual using the machine learning models implemented in the system.
- **Requirements Traceability**: This use case is associated with the functional requirement F4 (Prediction Generation) outlined in section 3.2.
- **Priority**: High. It's crucial for the system to accurately predict heart attacks, as it has a direct impact on patient well-being.
- **Preconditions**:
  - The user has access to the system.
  - The user has provided the necessary input data.
- **Postconditions**:
  - The system generates a prediction for heart attack likelihood.
  - The prediction result is displayed to the user.
- **Actors**:
  - User (Healthcare Professional or Individual)
- **Extends**: None

**Flow of Events**:

1. **Basic Flow**:
   - The user logs into the system.
   - The user provides input data related to the individual's health and medical history.
   - The system applies the selected machine learning or deep learning algorithm (e.g., Decision Tree, DNN) to the input data.
   - The system generates a prediction regarding the likelihood of a heart attack.
   - The prediction result is displayed to the user.
2. **Alternative Flow**:
   - If the user enters incomplete or inconsistent data:
     - The system notifies the user of the data issues.
     - The user is prompted to correct or complete the data.
   - If the system encounters technical issues during prediction:
     - An error message is displayed to the user.
     - The user is advised to try again later or contact support.
3. **Exceptions**:
   - If the user's input data contains significant outliers:
     - The system may provide a warning regarding potential data quality issues.
     - The user is encouraged to review and confirm the input data.

**3.3.2 Use Case #2: Data Preprocessing (U2)**

- **Author**: [Author Name]
- **Purpose**: The objective of this use case is to prepare input data for accurate heart attack prediction by cleaning and preprocessing it.
- **Requirements Traceability**: This use case is associated with functional requirement F1 (Data Preprocessing) outlined in section 3.2.
- **Priority**: Medium. Data preprocessing is a crucial step, but its priority is slightly lower than prediction generation.

- **Preconditions**:
  - The user has access to the system.
  - Raw input data is available.
- **Postconditions**:
  - The input data is cleaned, and preprocessing steps are completed.
  - The preprocessed data is ready for model training.
- **Actors**:
  - User (Data Scientist or System Administrator)
- **Extends**: None

**Flow of Events**:

1. **Basic Flow**:
   - The user logs into the system.
   - The user uploads the raw data for heart attack prediction.
   - The system performs data cleaning tasks, such as handling missing values, removing outliers, and ensuring consistent formatting.
   - The cleaned data is stored in a format suitable for machine learning model training.
2. **Alternative Flow**:
   - If the uploaded data is incomplete or contains severe inconsistencies:
     - The system provides feedback to the user about data issues.
     - The user may choose to address the data problems or seek assistance.
3. **Exceptions**:
   - If the data preprocessing encounters unexpected technical errors:
     - An error message is displayed to the user.
     - The user may need to retry the preprocessing or contact support.

# 4 Other Non-functional Requirements

## 4.1 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.

## 4.2 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).

## 4.3 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.

# 5 Other Requirements

*NA*

# Appendix A – Data Dictionary

*<Data dictionary is used to track all the different variables, states and functional requirements that you described in your document. Make sure to include the complete list of all constants, state variables (and their possible states), inputs and outputs in a table. In the table, include the description of these items as well as all related operations and requirements.>*

# Appendix B - Group Log

*<Please include here all the minutes from your group meetings, your group activities, and any other relevant information that will assist in determining the effort put forth to produce this document>*