Project Requirement Document

Heart Disease Prediction System

Project Overview:

Develop a machine learning-based web application to predict the likelihood of a person developing heart disease within ten years based on their health data. The application will utilize classification algorithms and provide detailed performance metrics including confusion matrix and classification report. The system will be integrated with a Flask web application where users can input their details to get predictions.

Objective:

To create a predictive model that can accurately identify the risk of heart disease using various health-related features and to integrate this model into a web-based application for easy user access.

Problem Statement:

Heart disease is a leading cause of mortality globally. Early prediction and prevention are crucial for improving health outcomes. By leveraging machine learning, we aim to create a tool that helps individuals assess their risk based on specific health parameters, thereby enabling timely medical interventions.

Dataset Description:

The dataset contains the following features:

- male: Gender of the individual (1 = male, 0 = female)
- age: Age of the individual in years
- education: Education level (1 = less than high school, 2 = high school, 3 = some college, 4 = college graduate)
- currentSmoker: Whether the individual is a current smoker (1 = yes, 0 = no)
- cigsPerDay: Number of cigarettes smoked per day
- BPMeds: Whether the individual is on blood pressure medication (1 = yes, 0 = no)
- prevalentStroke: Whether the individual has had a stroke (1 = yes, 0 = no)
- prevalentHyp: Whether the individual has hypertension (1 = yes, 0 = no)
- diabetes: Whether the individual has diabetes (1 = yes, 0 = no)
- totChol: Total cholesterol level
- sysBP: Systolic blood pressure

- diaBP: Diastolic blood pressure
- BMI: Body Mass Index
- heartRate: Heart rate
- glucose: Glucose level
- TenYearCHD: 10-year risk of coronary heart disease (1 = yes, 0 = no)

Machine Learning Requirements:

Data Preprocessing:

- Handle missing values if any.
- Standardize/normalize the data as required.
- Split the data into training and testing sets.

Modeling:

- Use classification algorithms such as Logistic Regression, Decision Trees, Random Forest, and others as deemed appropriate.
- Train models on the training dataset.
- Evaluate models using performance metrics.

Evaluation Metrics:

- Confusion Matrix
- Classification Report (Precision, Recall, F1-Score, Accuracy)

Integration:

- Develop a Flask web application.
- Create a form for user input for each of the features.
- Implement functionality to pass user inputs to the trained model and display the prediction result.

Flask Application Requirements:

Form Creation:

- Develop a form to capture user details corresponding to the features in the dataset.
- Ensure form validation and error handling.

Prediction Endpoint:

- Create an endpoint to receive form data and process it through the trained model.
- Display the prediction result (risk of heart disease) to the user.

User Interface:

- Simple and user-friendly interface.
- Instructions for users on how to input their details.