PROJECT REPORT

Heart Attack Predictor

NAME:
Asaad Iqbal
Roll Number:
SU92-BSDSM-F23-020
Institute:
Superior University
Department:
Software Engineering
Semester:
3rd
Section:
BSDS-3A
Submitted to:
Prof. Rasikh Ali

1. Problem Statement

Heart Attack Predictor: This is a machine learning-based standalone desk-application that predicts the probability of having a heart attack. The tool provides a risk prediction along with a probability score based on a Random Forest classifier that determines the user-provided health metrics.

Implementation Environment

The overall system is implemented in Python and uses the Tkinter GUI.

2. Goals

- Given medical data, to help users evaluate their risk for having a heart attack.
- To give an interface that is intuitive for non-technical users.
- Utilizing machine learning for precise and reliable predictions.

3. Methodology

3.1 Data Preprocessing

User inputs are standardized using a pre-trained scaler, scaling feature values before prediction.

1. Medical attributes considered:
2. Age
3. Sex
4. Chest Pain Type (CP)
5. Resting Blood Pressure (trestbps)
6. Cholesterol (chol)
7. Fasting Blood Sugar (fbs)
8. Resting Electrocardiographic Results (restecg)
9. Maximum Heart Rate Achieved (thalach)
10. Exercise-Induced Angina (exang)
11. Oldpeak (Exercise-Induced ST Depression)
12. Slope of Peak Exercise ST Segment
13. Number of Major Vessels (ca)
14. Thalassemia (thal)

3.2 Model

Utilizes a pre-trained Random Forest model heart_model_rf.pkl

A scaler scaler_rf.pkl ensures consistent input feature scaling.

3.3 GUI Implementation

The GUI, as implemented using Tkinter is created with labeled fields for inputting each attribute of medical nature

A "Predict Risk" button initiates the risk prediction

Results are shown in a pop-up window, including the level of risk and the probability score

4. Features

User-Friendly GUI: makes it easier to input data and to view results

Integration of machine learning model: Random Forest used for robust prediction.

Scalability: Easily extendable to accommodate more complex models or features.

5. Technical Specifications

5.1 Dependencies

Python Libraries: tkinter, numpy, joblib, pandas, matplotlib, seaborn

Machine Learning Framework: Scikit-learn (for model training and exporting)

5.2 Input Processing

Correct data types and acceptable value ranges are checked before proceeding with input validation.

User error is handled elegantly with informative error messages.

5.3 Prediction Pipeline

- Users can input medical information using the GUI.
- Input is scaled using the scaler that has already been trained.
- The Random Forest makes the prediction:
  - Risk class: Heart Attack Risk or No Heart Attack Risk
  - Probability score for the class that has been predicted.
  - Results are displayed to the user.

6. Results

- The system successfully predicts heart attack risk based on user-provided data.

- GUI design ensures usability and accessibility.