Heart Disease Prediction Analysis Using Machine Learning

May 8, 2025

Contents

1 Introduction

This document provides a comprehensive explanation of a machine learning project aimed at predicting heart disease using a dataset containing patient health metrics. The analysis involves data preprocessing, model building, and performance evaluation using Logistic Regression and k-Nearest Neighbors (kNN) algorithms. The dataset, referred to as `heart.csv`, includes features such as age, sex, cholesterol levels, and others, with a binary target variable indicating the presence or absence of heart disease.

The objective is to clean the data, transform it for modeling, train two machine learning models, and compare their performance based on accuracy. This explanation is structured to aid in preparing for an oral presentation, providing clear insights into each step of the process.

2 Dataset Overview

The dataset contains 1025 patient records with 14 columns, including 13 features and 1 target variable. The features are:

- `age`: Patient's age in years
- `sex`: Gender (1 = male, 0 = female)
- `cp`: Chest pain type (0–3)
- `trestbps`: Resting blood pressure (mm Hg)
- `chol`: Serum cholesterol (mg/dl)
- `fbs`: Fasting blood sugar > 120 mg/dl (1 = true, 0 = false)
- `restecg`: Resting electrocardiographic results (0–2)
- `thalach`: Maximum heart rate achieved
- `exang`: Exercise-induced angina (1 = yes, 0 = no)
- `oldpeak`: ST depression induced by exercise
- `slope`: Slope of the peak exercise ST segment (0–2)
- `ca`: Number of major vessels colored by fluoroscopy (0–3)
- `thal`: Thalassemia (1–3)
- `target`: Heart disease presence (1 = yes, 0 = no)

The dataset initially has 1025 rows, and the goal is to preprocess it to ensure data quality before modeling.

3 Data Preprocessing

Data preprocessing is critical to ensure the dataset is suitable for machine learning. The following steps were applied:

3.1 Handling Missing Values

The dataset was checked for missing values using the `dropna()` function in pandas. No missing values were found, as the dataset retained all 1025 rows after this step.

3.2 Ensuring Numeric Data

To confirm that all columns contain numeric data, a loop was used to convert each column to numeric values, dropping any rows with non-numeric entries. No rows were dropped, indicating that all data was already numeric.

3.3 Removing Negative Values

Negative values in numeric columns (e.g., age, cholesterol) are biologically implausible. All numeric columns were checked, and rows with negative values were removed. No rows were dropped, suggesting no negative values were present.

3.4 Outlier Detection and Removal

Outliers can skew model performance. A Z-score method was used to detect outliers, where values with an absolute Z-score greater than 3 were considered outliers. After applying this filter, the dataset was reduced from 1025 to 969 rows, indicating that 56 rows with extreme values were removed.

3.5 Data Transformation

The data was transformed to prepare it for modeling:

- **Separating Features and Target**: The `target` column was separated as the dependent variable (y), and the remaining 13 columns were used as features (X).
- **Feature Scaling**: Features were standardized using `StandardScaler` to ensure all features have a mean of 0 and a standard deviation of 1, which is essential for algorithms like Logistic Regression and kNN.
- **Data Splitting**: The dataset was split into training (80%) and testing (20%) sets using `train_test_split` with a random state of 42 for reproducibility.

4 Model Building and Evaluation

Two machine learning models were trained and evaluated: Logistic Regression and k-Nearest Neighbors (kNN).

4.1 Logistic Regression

Logistic Regression is a linear model suitable for binary classification. The model was trained on the scaled training data with a random state of 42. Predictions were made on the test set, and the accuracy was calculated as:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

The Logistic Regression model achieved an accuracy of 0.8814 (88.14%).

4.2 k-Nearest Neighbors (kNN)

kNN is a non-parametric algorithm that classifies data points based on the majority class of their k nearest neighbors. The model was configured with `n_neighbors=5` and trained on the same training data. The kNN model achieved an accuracy of 0.9072 (90.72%).

4.3 Model Comparison

The accuracies were compared:

- Logistic Regression: 0.8814
- kNN: 0.9072

Since 0.9072 > 0.8814, the kNN model outperformed Logistic Regression. The result was printed as: “kNN performs better.”

5 Results and Discussion

The kNN model achieved higher accuracy (90.72%) compared to Logistic Regression (88.14%). This suggests that kNN is better suited for this dataset, possibly due to its ability to capture non-linear relationships in the data. However, kNN is computationally more expensive and sensitive to the choice of k. Logistic Regression, while slightly less accurate, is faster and more interpretable.

The preprocessing steps ensured data quality, and the reduction to 969 rows after outlier removal likely improved model performance by eliminating extreme values. Feature scaling was crucial, as both models rely on distance-based calculations or coefficient optimization.

6 Conclusion

This analysis demonstrates a systematic approach to predicting heart disease using machine learning. The kNN model outperformed Logistic Regression, achieving 90.72% accuracy. Future work could explore hyperparameter tuning (e.g., optimizing k for kNN), testing other algorithms (e.g., Random Forest), or incorporating feature selection to improve performance.

The project highlights the importance of data preprocessing, model selection, and evaluation in building effective machine learning models for healthcare applications.

7 Presentation Tips for Oral Explanation

For an oral presentation, consider the following tips to explain this project effectively:

- **Start with Context**: Briefly introduce the problem of heart disease prediction and the dataset's relevance.
- **Explain Preprocessing Clearly**: Use simple terms to describe missing value handling, outlier removal, and scaling. For example, "We removed extreme values to ensure the data was reliable."
- **Describe Models Visually**: Use diagrams or slides to show how Logistic Regression (linear boundary) and kNN (neighborhood-based) work.
- **Highlight Results**: Emphasize the accuracy difference (90.72% vs. 88.14%) and why kNN performed better.
- **Engage the Audience**: Ask rhetorical questions, e.g., "Why do you think scaling is important for these models?"
- **Practice Key Terms**: Be comfortable with terms like Z-score, standardization, and accuracy to sound confident.
- **Conclude with Impact**: Summarize the project's success and mention potential improvements to show forward-thinking.