Heart Disease Prediction

Namit Patel,Smit patel,Yogi Patel

April 22, 2024

1 Problem Definition

Heart disease remains one of the leading causes of morbidity and mortality worldwide. Early detection and accurate prediction of heart disease are crucial for effective management and prevention of adverse outcomes. The goal of this project is to develop a predictive model capable of accurately diagnosing the presence and severity of heart disease based on clinical and demographic features. By leveraging machine learning algorithms, the model aims to analyze patterns within a dataset containing patient information such as age, sex, chest pain type, blood pressure, cholesterol levels, and other relevant attributes. The predictive model will classify individuals into different severity levels of heart disease, ranging from no disease to critical conditions. The ultimate objective is to assist healthcare professionals in making timely and informed decisions, leading to improved patient outcomes and reduced cardiovascular morbidity and mortality rates.

2 Dataset Description - Heart Disease Prediction Dataset

2.1 Overview

The Heart Disease Prediction dataset is a collection of clinical data from three different locations: Hungary, Cleveland, and Switzerland and VA long Beach. It contains information about individuals who have been diagnosed with various cardiac conditions, with the aim of facilitating predictive modeling for heart disease. The dataset used has 921rows and 30 columns.

2.2 Numerical Columns

1. ID
2. Age
3. Trestbps (Resting Blood Pressure)
4. Chol (Serum Cholesterol)
5. Thalch (Maximum Heart Rate Achieved)
6. Oldpeak (ST Depression Induced by Exercise Relative to Rest)
7. CA (Number of Major Vessels Colored by Fluoroscopy)
8. Num (Diagnosis)

2.3 Categorical Columns

1. Sex
2. Dataset
3. CP (Chest Pain Type)
4. FBS (Fasting Blood Sugar)
5. Restecg (Resting Electrocardiographic Results)
6. Exang (Exercise Induced Angina)
7. Slope (Slope of the Peak Exercise ST Segment)
8. Thal (Thallium Stress Test Result)

2.4 Target Column

The target column is "Num," which represents the predicted attribute for heart disease diagnosis. Here's the mapping of the unique values in the "Num" column:

1. 0: No heart disease
2. 1: Mild heart disease
3. 2: Moderate heart disease
4. 3: Severe heart disease
5. 4: Critical heart disease

These values provide the severity levels of heart disease, ranging from no disease to critical condition. This information will be essential for training predictive models to classify the severity of heart disease based on the input features.

3 Attributes

- **ID**: Unique identifier for each patient.
- **Age**: Age of the patient in years.
- **Sex**: Gender of the patient (Male/Female).
- **Dataset**: Location where the data was collected (Hungary, Cleveland, or Switzerland).
- **Chest Pain Type (CP)**: Type of chest pain experienced by the patient, categorized as typical angina, atypical angina, non-anginal, or asymptomatic.
- **Resting Blood Pressure (Trestbps)**: Resting blood pressure measured in mm Hg.
- **Serum Cholesterol (Chol)**: Serum cholesterol level measured in mg/dl.
- **Fasting Blood Sugar (FBS)**: Indicates whether the fasting blood sugar level is above 120 mg/dl (TRUE/FALSE).
- **Resting Electrocardiographic Results (Restecg)**: Results of the resting electrocardiogram, indicating the presence of left ventricular hypertrophy or normal.

- **Maximum Heart Rate Achieved (Thalch)**: Maximum heart rate achieved during exercise.
- **Exercise Induced Angina (Exang)**: Indicates whether angina was induced by exercise (TRUE/FALSE).
- **ST Depression Induced by Exercise Relative to Rest (Oldpeak)**: ST depression induced by exercise relative to rest.
- **Slope of the Peak Exercise ST Segment (Slope)**: Slope of the peak exercise ST segment, categorized as upsloping, flat, or downsloping.
- **Number of Major Vessels Colored by Fluoroscopy (CA)**: Number of major vessels colored by fluoroscopy (0-3).
- **Thallium Stress Test Result (Thal)**: Result of the thallium stress test, categorized as normal, fixed defect, or reversible defect.
- **Diagnosis (Num)**: Diagnosis of heart disease (0: No presence, 1-4: Presence).

4 Purpose

The dataset serves as a valuable resource for predicting the presence or absence of heart disease based on various clinical and demographic factors. By analyzing this dataset, we can develop predictive models to identify individuals at risk of cardiovascular disorders and implement preventive measures accordingly.

5 Usage

We can utilize this dataset for: Developing machine learning models for heart disease prediction. Investigating the effectiveness of different diagnostic tests and risk factors in predicting heart disease. Understanding geographical variations in cardiac health and disease prevalence. Informing public health policies and interventions aimed at reducing the burden of cardiovascular diseases.

6 Novelty

- **Data Import**: The notebook begins by importing the necessary libraries, such as pandas, numpy, and matplotlib.
- **Data Loading**: The dataset is loaded using pandas' `read_csv` function.
- **Data Cleaning**: The notebook includes code for cleaning the data, such as removing unnecessary columns, handling missing values, and converting categorical data to numerical data.
- **Exploratory Data Analysis (EDA)**: The code conducts EDA by examining the dataset's characteristics, such as the age distribution, sex distribution, and unique values in the 'dataset' column. This EDA helps in understanding the data before proceeding with predictive modeling.
- **Gender Analysis**: A unique aspect is the calculation of heart disease percentages based on gender. The code calculates and compares the percentage of males and females with heart disease in the dataset, highlighting the gender disparity in heart disease prevalence.
- **Visualization**: The code includes a histogram plot to visualize the distribution of ages in the dataset. This visual representation enhances the understanding of the age distribution within the dataset.
- **Dataset Analysis**: By exploring the unique values and value counts in the 'dataset' column, the code provides insights into the different datasets included in the analysis, such as 'Cleveland', 'Hungary', 'Switzerland', and 'VA Long Beach'.

- **Age and Gender Analysis**: The code further delves into the dataset by examining the value counts of the 'age' column grouped by 'sex', offering a detailed breakdown of age distribution based on gender.
- **Missing Values and Outliers**: The code utilizes various techniques to handle missing values, such as imputation methods like SimpleImputer, KNNImputer, and IterativeImputer. It also explores the presence of missing values in different columns like 'trestbps', 'chol', 'fbs', 'restecg', 'thalch', 'exang', 'oldpeak', 'slope', 'ca', and 'thal'.
- **Machine Learning Models**: The code incorporates a range of machine learning models for heart disease prediction, including Logistic Regression, K-Nearest Neighbors, Support Vector Machines, Naive Bayes, Decision Trees, Random Forest, AdaBoost, Gradient Boosting, and XGBoost classifiers. It also covers metrics like accuracy score, R2 score, confusion matrix, mean squared error, and mean absolute error to evaluate the performance of these models. Additionally, the notebook utilizes a pipeline approach for model building and includes the MLxtend library for frequent pattern mining using Apriori algorithm.
- **Model Evaluation and Interpretation:**: There is a huge importance of model evaluation and interpretation in assessing the predictive model's performance and reliability. We have used the metrics for evaluating model performance, such as accuracy, R-squared coefficient, and model complexity plots, and interpreted their implications in the context of graduate admission prediction.

Overall, the code showcases a comprehensive analysis of the heart disease dataset, incorporating EDA, gender-based analysis, visualization, and dataset-specific insights to gain a deeper understanding of the data and potentially aid in predictive modeling for heart disease.

7 Algorithms

Algorithm 1 Data Cleaning

Input: Data from a CSV file
Output: Cleaned data

1: **procedure** CLEAN DATA(data)
2: data ← data without unnecessary columns
3: data ← data with missing values replaced with NaN
4: **end procedure**

Algorithm 2 Removing Outliers

Input: Data with Outlier
Output: Data without Outlier

procedure REMOVE OUTLIER(data)
    Analysis of Boxplot
    if data has outlier
    if data has outlier **then**
        Compute $Q1$ and $Q3$
        $IQR \leftarrow Q3 - Q1$
        $min \leftarrow Q1 - 1.5 \times IQR$
        $max \leftarrow Q3 + 1.5 \times IQR$
        for value in data **do**
            if value ¡ min **then**
                value $\leftarrow$ min
            end if
            if value ¿ max **then**
                value $\leftarrow$ max
            end if
        end for
    end if
end procedure

Algorithm 3 Pre-processing the Data

Input: Data
Output: Processed Data

procedure PRE-PROCESSING(data)
    if feature is categorical **then**
        Impute data by mode of the column
        Make a copy of the imputed columns
        Drop the original columns
        Transform the categorical data using `LabelEncoder` method
    end if
    if feature is numeric **then**
        Round off the data to 3 decimal places
        Impute the data by mean of the column
        Apply PCA to Data
    end if
end procedure

Algorithm 4 Exploratory Data Analysis (EDA)

procedure EDA(data)
    1. Summary Statistics
        Compute basic summary statistics (mean, median, mode, standard deviation, min, max, quartiles, etc.) for each variable.
    2. Data Visualization
        Plot histograms for each variable to visualize the distribution.
        Plot boxplots for each variable to identify outliers and compare distributions.
        Create scatter plots for pairs of variables to explore potential relationships.
        Create correlation heatmaps to examine correlations between variables.
    3. Missing Values Analysis
        Identify missing values in the data.
        Compute the percentage of missing values for each variable.
    4. Data Cleaning
        Handle missing values (e.g., imputation, deletion).
        Handle outliers (e.g., capping, transformation, removal).
    5. Data Transformation
        Normalize or standardize numerical variables, if necessary.
        Encode categorical variables (e.g., one-hot encoding, label encoding).
    6. Feature Engineering
        Generate new features from existing ones, if applicable.
    7. Preliminary Insights
        Identify key trends, patterns, or anomalies in the data.
        Document key findings and insights for further analysis.
end procedure

Algorithm 5 Algoritham for Feature Importance

Input: Processed Data
Output: Reduced Selected Data

procedure FEATURE IMPORTANCE(data)
    Split the data into train and test sets.
    Initialize a Random Forest Tree Classifier model.
    Train the model on the training set.
    Evaluate the model's performance on the test set.
    Retrieve the importance of features from the trained model.
    Plot a graph of the feature importance.
end procedure

Algorithm 6 Impute Categorical Missing Data

procedure IMPUTE_CATEGORICAL_MISSING_DATA($passed\_col$)
    1. Split the DataFrame into two parts: df_null containing rows with missing values in $passed\_col$, and df_not_null containing rows without missing values in $passed\_col$
    2. Define X as the features from df_not_null without $passed\_col$, and y as the target column $passed\_col$
    3. Define $other\_missing\_cols$ as a list of columns with missing values except $passed\_col$
    4. Initialize $label\_encoder$ as a LabelEncoder instance
    5. Encode categorical features in X using $label\_encoder$
    6. Instantiate an IterativeImputer with RandomForestRegressor as estimator
    7. Impute missing values in $other\_missing\_cols$ in X using IterativeImputer
    8. Split X and y into training and testing sets using train_test_split
    9. Instantiate RandomForestClassifier as $rf\_classifier$
    10. Fit $rf\_classifier$ on X_train and y_train
    11. Predict y_test using $rf\_classifier$
    12. Calculate $accuracy\_score$ between y_test and y_pred
    13. Print the $accuracy\_score$
    14. Replace missing values in $df\_null[passed\_col]$ with predictions from $rf\_classifier$
    15. Concatenate df_not_null and df_null to get $df\_combined$
    16. Return $df\_combined[passed\_col]$
end procedure

Algorithm 7 Heart Disease Prediction: Model Training and Evaluation

procedure MODEL TRAINING AND EVALUATION(Data)
    1. Train Models
    Train Logistic Regression
    Train K-Nearest Neighbors (KNN)
    Train Support Vector Machine (SVM)
    Train Naive Bayes
    Train Decision Tree
    Train Random Forest
    Train AdaBoost model
    Train Gradient Boosting model
    Train XGBoost model
    2. Evaluate Models
    Evaluate each model using the testing data and the following metrics:
        Accuracy score
        Confusion matrix
        R2 score
        Mean squared error (MSE)
        Mean absolute error (MAE)
    3. Model Comparison and Selection
    Compare the performance of each model using evaluation metrics.
    Select the model with the best performance for deployment.
    4. Frequent Pattern Mining (using MLxtend)
    Utilize the Apriori algorithm from the MLxtend library to mine frequent patterns from the data.
    Analyze the patterns to identify associations between different features.
    5. Document Findings
    Summarize the performance of different models and chosen model.
    Document frequent patterns and potential insights from the mining process.
end procedure