Final Project: Machine Learning in Biomed

Eldar Tarighat 40033044

July 6, 2025

Abstract

This report details a machine learning project focused on predicting heart failure using a publicly available dataset. The project involves data exploration, preprocessing steps such as handling missing values, one-hot encoding, outlier detection, and data normalization. Feature selection is applied to reduce dimensionality, and a RandomForestClassifier is trained and evaluated. The aim is to demonstrate a complete machine learning pipeline for a classification problem in the biomedical domain.

1 Introduction

This project addresses a classification problem within the biomedical field, specifically the prediction of heart disease. The objective is to implement a comprehensive machine learning pipeline, from data acquisition and preprocessing to model training and evaluation, leveraging concepts learned in the Machine Learning in Biomedical course. The chosen dataset is related to heart failure prediction, and the goal is to classify individuals based on various medical attributes as either having or not having heart disease.

2 Dataset Description

The dataset used for this project is the "Heart Failure Prediction" dataset, which is available on Kaggle. This dataset is suitable for classification tasks and contains various features related to patient health metrics. Based on the code analysis, the dataset attributes include:

- **Age**: age of the patient [years]
- **Sex**: sex of the patient [M: Male, F: Female]
- **ChestPainType**: chest pain type [TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic]
- **RestingBP**: resting blood pressure [mm Hg]
- **Cholesterol**: serum cholesterol [mm/dl]
- **FastingBS**: fasting blood sugar [1: if FastingBS $>$ 120 mg/dl, 0: otherwise]
- **RestingECG**: resting electrocardiogram results [Normal: Normal, ST: having ST-T wave abnormality, LVH: showing probable or definite left ventricular hypertrophy]
- **MaxHR**: maximum heart rate achieved [Numeric value between 60 and 202]
- **ExerciseAngina**: exercise-induced angina [Y: Yes, N: No]
- **Oldpeak**: oldpeak = ST [Numeric value measured in depression]
- **ST_Slope**: the slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping]
- **HeartDisease**: output class [1: heart disease, 0: Normal]

Initially, the dataset has 918 samples and 12 columns.

3 Methodology

3.1 Imports and Initial Setup

The project utilizes standard Python libraries for data manipulation, visualization, and machine learning, including ‘numpy‘, ‘pandas‘, ‘seaborn‘, ‘matplotlib‘, ‘plotly‘, ‘sklearn‘, and ‘kagglehub‘ for data access. Warnings were suppressed, and pandas display options were set for convenience.

3.2 Data Loading and Initial Exploration

The "Heart Failure Prediction" dataset was downloaded via ‘kagglehub‘ and loaded into a pandas DataFrame. Initial exploration involved viewing the first few rows (‘df.head()‘), checking the dataset's shape (‘df.shape‘), and generating descriptive statistics (‘df.describe()‘). Object type columns were converted to string type to facilitate further processing. Histograms were plotted for all columns to visualize their distributions.

3.3 Missing Value Imputation

A specific issue identified was the presence of zero values in the 'Cholesterol' column, which are considered missing or erroneous measurements in a medical context.

- The number of data points with zero Cholesterol was identified.
- The mean of the non-zero Cholesterol values was calculated.
- These zero values were then imputed with the calculated mean.
- Histograms before and after imputation were generated to show the effect of this correction.

3.4 Pairplot Analysis

A pairplot was generated with 'HeartDisease' as the hue. This visualization helps in understanding the relationships between different features and how they correlate with the target variable, providing insights into potential patterns for heart disease prediction.

3.5 One-Hot Encoding

Categorical features, identified as those with string data types, were converted into numerical format using one-hot encoding. ‘pd.get$_dummies$‘$wasapplied, and$‘$drop_first = True$‘$wasusedtoavoidmulticollinearity.Thisstepisc$

3.6 Correlation Matrix

A correlation matrix of the numerical features was computed and visualized using a seaborn heatmap. This helps in understanding the linear relationships between variables and identifying highly correlated features, which can sometimes indicate multicollinearity issues or provide insights into feature importance.

3.7 Outlier Detection and Removal

Outliers in the dataset were detected and removed using the Local Outlier Factor (LOF) algorithm. LOF measures the local deviation of density of a given data point with respect to its neighbors. Data points identified as outliers (with a LOF score indicating deviation) were removed from the dataset to improve model robustness. The dataset shape before and after outlier removal was printed to confirm the change.

3.8 Data Splitting

The preprocessed data was split into features (X) and the target variable (y), which is 'HeartDisease'. Subsequently, the data was divided into training and testing sets using ‘train$_test_split$‘$, with an 80\% - 20\% ratio and a fixed$‘$random_state$‘$ for reproducibility.$

3.9 Normalization

The numerical features in both the training and testing sets were normalized using 'StandardScaler'. This process scales the data to have a mean of 0 and a standard deviation of 1, which helps in preventing features with larger values from dominating the learning process.

3.10 Feature Selection

Feature selection was performed using 'SelectKBest' with '$f_c lassif$'$as the scoring function, selecting the top 5 features. This$

3.11 Model Training

A 'RandomForestClassifier' was chosen as the classification model. Hyperparameter tuning was performed using 'GridSearchCV' to find the best combination of '$n_e stimators$'$(number of trees) and$'$max_d epth$'$for the Random$ $validation (cv = 3) was used, and the model was evaluated based on accuracy.$

3.12 Results

After training, the best parameters found by 'GridSearchCV' were printed. The model's performance on the test set was evaluated using the '$classification_r eport$', $which provides precision, recall, f1-score, and support for each cl$

4 Conclusion and Discussion

The implemented pipeline successfully addresses the heart disease prediction problem. The various preprocessing steps, including handling missing values, encoding categorical features, and outlier detection, were crucial for preparing the data. Feature selection helped in focusing on the most informative attributes. The RandomForestClassifier, with optimized hyperparameters, demonstrated its performance through the classification report and confusion matrix. Further improvements could involve exploring other machine learning algorithms, more advanced feature engineering techniques, or ensemble methods.