Sensyne Health Problem

S. Rathee

Last modified: 11 Apr 2020

Overview

In this report, I explained the workflow for heart disease dataset analysis App. It contains:

- Plots for Original Dataset
- Plots for Clean Dataset
- Correlation Matrix
- Principal Component Analysis
- ML Model

Dataset

I'll be working with the Cleveland Clinic Heart Disease dataset which contains 13 variables related to patient diagnostics and one outcome variable indicating the presence or absence of heart disease. The data was accessed from the UCI Machine Learning Repository in April 2020. The goal is to be able to accurately classify as having or not having heart disease based on diagnostic test data. There are 14 variables provided in the data set and the last one is the dependent variable that we want to be able to predict. Here is a summary of what the other variables mean:

- Age: Age of subject
- Sex: Gender of subject: 0 = female 1 = male
- Chest-pain type: Type of chest-pain experienced by the individual: 1 = typical angina 2 = atypical angina 3 = non-angina pain 4 = asymptomatic angina
- Resting Blood Pressure: Resting blood pressure in mm Hg
- Serum Cholesterol: Serum cholesterol in mg/dl
- Fasting Blood Sugar: Fasting blood sugar level relative to 120 mg/dl: 0 = fasting blood sugar <= 120 mg/dl 1 = fasting blood sugar > 120 mg/dl
- Resting ECG: Resting electrocardiographic results 0 = normal 1 = ST-T wave abnormality 2 = left ventricle hyperthrophy
- Max Heart Rate Achieved: Max heart rate of subject
- Exercise Induced Angina: 0 = no 1 = yes
- ST Depression Induced by Exercise Relative to Rest: ST Depression of subject
- Peak Exercise ST Segment: 1 = Up-sloaping 2 = Flat 3 = Down-sloaping
- Number of Major Vessels (0-3): Number of visible vessels under flouro
- Thal: Form of thalassemia: 3 3 = normal 6 = fixed defect 7 = reversible defect

- Diagnosis of Heart Disease: Indicates whether subject is suffering from heart disease or not: 0 = absence 1, 2, 3, 4 = heart disease present

Tools and Environment

I choose R (3.6.3) as a coding environment. I found a lot of blogs and r codes to analyze above dataset. Instead of just copy the code from these blogs, I tried to implement my own App. The app can be accessed directly from the web or locally.

1. Web Access: Use browser to open https://sanjaysinghrathi.shinyapps.io/zcompant-problem/
2. Local Access: Use local machine having R 3.6 and Rstudio 1.2 to run two lines of code:

```
* library(shiny)

* runGitHub( "Heart-Disease-Problem", "sanjaysinghrathi")
```

Original Data Visualization

I plotted every attribute alone and w.r.t. diagnosis of heart disease. A closer look at the data identifies some NA and "?" values that will need to be addressed in the cleaning step. We also want to know the number of observations in the dependent variable column to understand if the dataset is relatively balanced.

Data Cleaning and Visualization

Since any value above 0 in 'Diagnosis_Heart_Disease' (column 14) indicates the presence of heart disease, we can lump all levels > 0 together so the classification predictions are binary – Yes or No (1 or 0). The total count of positive heart disease results is less than the number of negative results so the fct_lump() call with default arguments will convert that variable from 4 levels to 2.

The data cleaning pipeline also deals with NA values, converts some variables to factors, lumps the dependent variable into two buckets, removes the rows that had "?" for observations, and reorders the variables within the dataframe. After plotting clean data, I observe that following conditions are associated with increased prevalence of heart disease (note: this does not mean the relationship is causal).

- Asymptomatic angina chest pain (relative to typical angina chest pain, atypical angina pain, or non-angina pain)
- Presence of exercise induced angina
- Lower fasting blood sugar
- Flat or down-sloaping peak exercise ST segment
- Presence of left ventricle hypertrophy
- Male
- Higher thelassemia score
- Higher age
- Lower max heart rate achieved
- Higher resting blood pressure
- Higher cholesterol
- Higher ST depression induced by exercise relative to rest

Correlation Matrix

Highly correlated variables can lead to overly complicated models or wonky predictions. The ggcorr() function from GGally package provides a nice, clean correlation matrix of the numeric variables. The default method is Pearson which I use here first. Pearson isn't ideal if the data is skewed or has a lot of outliers so I'll check using the rank-based Kendall method as well.

There are very minor differences between the Pearson and Kendall results. No variables appear to be highly correlated. As such, it seems reasonable to stay with the original 14 variables as we proceed into the modeling section.

Principal Component Analysis

I plotted some pca components to see the separation between patients having or not having a heart disease. The pca1 shown a high variance of 24% along with pca2 with 12% variance. It gives a hint about a good separation between the two groups.

Machine Learning Model

The plan is to split up the original data set to form a training group and testing group. The training group will be used to fit the model while the testing group will be used to evaluate predictions. The initial_split() function creates a split object which is just an efficient way to store both the training and testing sets. I used 80:20 ratio to split the data.

Generally, researchers pick up their favorite ML model to predict the response on the testing set. I like to compare a few most suitable models for a problem. Here, I used elastic net, support vector machines (Linear), neural networks and random forest. I used the "Boruta" package for feature selection.

At last, I choose an elastic net model with an internal cross-validation (leave-one-out) accuracy of 85.57%, external accuracy of 83.05% and F1 84.84%. The app shows the confusion matrix, ROC plot and a table having various measures for ML model like accuracy, kappa, p-value, F1. My app online version struggle with "out of memory" issue due to 1 GB memory availability for free user. Please prefer to use github version if possible.

Note: Please refresh your browser if the app show error during ML model training. It is because of the low memory available for a free app developer account. Results for online version can be different due to out of memory issue during training.

References

- https://www.kaggle.com/snogard/heart-disease-uci-using-r
- https://www.kaggle.com/duttanishtha/heart-disease-dataset-analysis
- https://www.r-bloggers.com/heart-disease-prediction-from-patient-data-in-r/
- https://archive.ics.uci.edu/ml/datasets/Heart+Disease