Heart Disease Diagnosis

Amir Alansary

May 2020

Data was collected from the Cleveland Clinic Foundation. The original databases have 76 raw attributes, and the processed one use only 14 of them (we use processed here). The "goal" field refers to the presence of heart disease in the patient (integer valued from 0 "no presence" to 4)

Main task

1. Classify patients with heart failure
2. Identify correlated features

Data Attributes (All attributes are numeric-valued, number of instances is 303)

1. **age**: in years
2. **sex**: (1 = male; 0 = female)
3. **cp**: chest pain location (1 = substernal; 0 = otherwise)
4. **trestbps**: resting blood pressure (in mm Hg on admission to the hospital)
5. **chol**: serum cholestoral in mg/dl
6. **fbs**: fasting blood sugar > 120 mg/dl (1 = true; 0 = false)
7. **restecg**: resting electrocardiographic results: 0 normal, 1 having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV), and 2 showing probable or definite left ventricular hypertrophy by Estes' criteria
8. **thalach**: maximum heart rate achieved
9. **exang**: exercise induced angina (1 = yes; 0 = no)
10. **oldpeak**: ST depression induced by exercise relative to rest
11. **slope**: of the peak exercise ST segment: 1 upsloping, 2 flat, and 3 downsloping
12. **ca**: number of major vessels $(0 - 3)$ coloured by flourosopy
13. **thal**: 3 = normal; 6 = fixed defect; 7 = reversable defect
14. **num** (the predicted attribute): diagnosis of heart disease (angiographic disease status). Value 0: $< 50\%$ diameter narrowing, and value 1: $> 50\%$ diameter narrowing

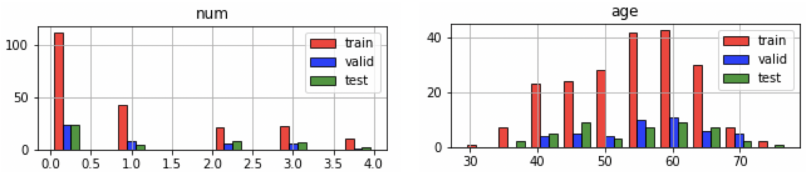
1 Solution

1.1 Pre-processing

- Pre-processing code with visualisation and results is in the notebook file **'data_preprocessing.ipynb'**
- Download data processed.cleveland.data file, which is available from the Data Folder
- Download the data description file heart-disease.names
- Load data processed.cleveland.data, and add header (column names) from heart-disease.names
- Inspect and clean data: Some of the cell values are missing (value = '?') - 6 instances. These instances are removed, another way is to replace the missing entries with the mean value of this attribute from the training set. Check all attributes are numerical with the correct type.
- Split the data into (70% = 207) train - (15% = 45) validation - (15% = 45) test, (another way is to do k-fold cross-validation). The split is stratified on the 14 features to ensure a similar distribution of these features in each subset.
- Normalise data: Re-scale feature values either to $[-1, 1]$ or $[0, 1]$. Another way is to standardise the values according to the mean and std of training data.

1.2 Data Analysis

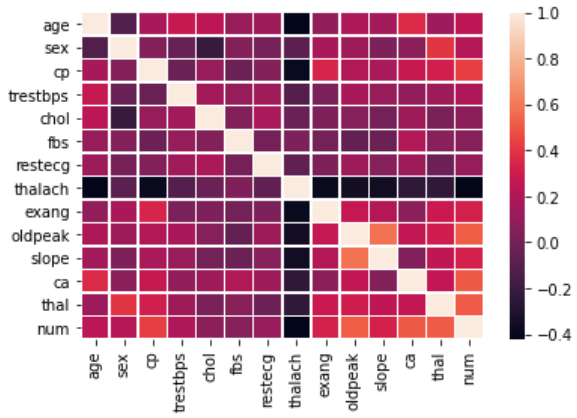
- **Data distribution**: Figure 1.2 shows the distribution of age and target label (num) in the different data splits, see the data processing notebook for the plots of the other attributes. The class distribution also shows a class imbalance, where the dominant class is 0.



- **Feature correlation**: Figure 1.2 shows the pair-wise correlation matrix between the 14 attributes as a heatmap. You can notice that the heart failure level (num) has a negative correlation with the maximum heart rate achieved (thalach), and a positive correlation with number of major vessels (ca) and depression induced by exercise relative to rest (oldpeak)

1.3 Prediction Models

- **Logistic Regression (LR)**: It is similar to linear regression, where the prediction is a linear combination of input feature, but followed with a sigmoid function. The main advantage of this model is its simplicity and works as a white box. However, it can underfit because of the low model complexity (low number of parameters).
- **Neural Networks (NN):** It can be seen a stacked logistic regression layers. It allows to increase the model complexity, hence stronger model with better performance. But it works as a black box. It is more difficult to find which features from the input contributed to the prediction output.



- **Conventional ML models**: Some examples are Decision Trees (DT), Support Vector Machines (SVM), and Random Forests (RF). The majority of these models can be used as strong prediction models, as well as provide interpretations about their decisions. However, Deep Neural Networks (DNN) have shown a significant superior performance in many of applications regardless the lack of the interpretability of the model.

1.4 Experiments and Implementation

I used the scikit-learn for Decision Trees, Random Forests, AdaBoost, and SVM models, see the notebook file **'test_ml.ipynb'**. I have implemented a logistic regression and neural network models using pytorch, see **'models.py'**. **'main.py'** contains most of the code inclduing the trainer and validator. **'metrics.py'** contains evaluation metrics including accuracy. **'dataset.py'** is responsible for data loading.

I have experimented using a weighted sampler and weighted loss with neural network to handle the class imbalance. However, both of these experiments performed almost similar to the normal random sampler.

A general notice, models overfit whenever I have increased their complexity (e.g. max depth in trees, or hidden layers size in neural networks). I also tried using a dropout layer, but validation accuracy decreased.

1.5 Results

As shown in Table 1, the two layers neural network performed the best with test accuracy 64.44%. I have run several experiments to grid search the best hyperparameters for these models, such as learning rate, size of hidden layers, data normalisation, batch size, and sampling methods. Given the limited space, I am reporting only the results from the best configurations.

Feature Selection (importance): Not all the model used in this report can provide a straightforward approach for extracting the most important features from the input, which contributed to the final predictions. Table 2 shows the weights generated from the ML models per each feature. Slope and thal were commonly used with high weights for all the shown models. Trestbps was the most important feature for the AdaBoost Classifier. It's worth mentioning that logistic regression model also provides an array of weights [input_size x ouptut_size], which in our case will be weights

Model	Train Accuracy %	Validation Accuracy %	Test Accuracy %	Model Info
Decision Trees	66.67	46.67	53.33	max_depth=3
Random Forests	**89.86**	55.56	60.00	max_depth=5
AdaBoost	62.80	53.33	60.00	learning_rate=0.03
SVM	70.53	55.56	62.22	kernel='linear'
Logistic Regression	52.17 (top2 71.50)	53.33 (top2 71.11)	55.56 (top2 68.89)	learning_rate=0.0002 batch_size=20 optimiser=sgd
Neural Network	70.05 (top2 88.89)	**57.78 (top2 71.11)**	**64.44 (top2 80.00)**	learning_rate=0.0002 batch_size=20 optimiser=adam hidden_size=[32,16]

Table 1: Accuracy results in % from all trained models.

for each class label. Because of the limited space of this report, the weights are kept as an option to print out in the code.

Model	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
Decision Trees	0	0.08	0.39	0	0.0.8	0	0	0	0	0	0.30	0.06	0.09
Random Forests	0.07	0.09	0.03	0.15	0.08	0.10	0.02	0.02	0.12	0.04	0.14	0.04	0.11
AdaBoost	0	0	0	0.44	0	0	0	0	0.04	0	0.32	0	0.20

Table 2: Weights given by trained models for each feature from the input. Important features have higher weights.

2 Conclusion and Discussion

Neural networks have shown potential results, which can be further investigated in the future. This can be done by customising specific architectures and widen the search space of the used hyper-parameters. The downside is the lack of the interpretability of these models. However, advanced research methods such attention may allow these models to draw some insights from their decision making process.

The evaluation section can be extended to include accuracy per class, and using more metrics such as: sensitivity, specificity, ROC, and confusion matrix. These metrics can extend our understanding of the performance of each model. Furthermore, data augmentation can play an important role in improving the generalisation and performance.