Analyze Heart Disease Pathogeny by Machine Learning

EE 660 Course Project

Project Type: (2) Experimental or theoretical exploration of machine learning

Number of student authors: 1

Hongrui Cai and hongruic@usc.edu

08/12/2019

1. Abstract

This is a project that tries to analyze the relationship between heart disease diagnosis and many other features. The published papers has shown that there are 14 most related features which serves as our data source. Though we know what factors may influence heart disease, it is still attractive to know how they specifically influence the heart disease, how each feature weighs in the influence. In this project, I use several Machine Learning methods to explore this problem. And I finally achieve a prediction accuracy of 97.6% served by Classification and Regression Tree.

2. Introduction

2.1. Problem Type, Statement and Goals

This is so interesting as there might be a way to predict how heart disease is related some related features. So that we could prevent it in time as there will be some "observable symptom" before the heart disease occurs.

This is a classification problem as the predicted variable contains only two values, 0 and 1, which will be explained later.

2.2. Overview of Our Approach

I use the logistic regression, Tree model and random forest model in this project. The main criteria I use here is the prediction accuracy.

3. Implementation

3.1. Data Set

The dataset that is used in this project contains fourteen features. As this is a realistic problem, it's important to understand what each feature means. They are shown below:

Data	Type	Description
Age	Integer	age in years
Sex	Binary	(1=male,0=female)
cp	Categorical	Chest pain type 1:typical angina 2:atypical angina 3:non-anginal pain 4:asymptomatic
Trestbps	Float	Resting blood pressure
Chol	Float	Serum choletoral in mg
Fbs	Binary	Fasting blood sugar>120mg/dl 1:true 0:false
Restecg	Categorical	Resting electrocardiographic results 0:normal 1:having ST-T wave abnormality 2:showing probable or definite left ventricular hypertrophy
Thalach	Float	Maximum heart rate achieved
Exang	Binary	Exercise induced angina 1:yes 0:no
Oldpeak	Float	ST depression induced by exercise relative to rest
Slope	Categorical	The slope of the peak exercise ST segment 1:upsloping 2:flat 3:downsloping
Ca	Integer	Number of major vessels colored by fluoroscopy

Data	Type	Description
Thal	Categorical	3:normal 6:fixed detect 7:reversible defect
pre_attribute	Binary	The predicted diagnosis of heart disease 0:<50% diameter narrowing 1:>50% diameter narrowing

3.2. Preprocessing, Feature Extraction, Dimensionality Adjustment

We need firstly to deal with the missing data like “?”. This would otherwise cause the whole row of data to be a string when we read it in pandas. The way I use here is to use the mean value of this feature of neighboring points to fill in the blank. As there are less than 10 points that has missing values, I just adjust them in the original csv file.

The number of the first and the second class are well balanced where the first class has 499 points and the second with 526 points, so there is no need to consider unbalance of these two classes.

And then we are doing a standardize method which means standardize each feature so that they have zero mean and variance of one.This would be helpful when we use the gradient descent, which is however useless when we apply the decision tree related algorithm.

As for the feature extraction part, it is stated there are fourteen related features in all collected 64 features [1].

Besides, there is no need to do the dimensionality reduction or sparse coding method as the number of feature is 14, and the number of points is over 1000.

3.3. Dataset Methodology

There are totally 1023 data points in this problem. I use the function sklearn.model_selection.train_test_split to split the data in a proportion of 4:1 for train and test.

Then for the training set, we continue splitting it into training set and validation set with proportion of 4:1. The validation set is especially used for choosing the parameters for the model.

3.4. Training Process

Logistic Regression

The first method we use is the logistic regression. The logistic regression is a Classification method which uses the regression idea. We firstly convert the feature data through the sigmoid function. So that they are converted into a range (0,1) variable.

$$g(z) = \frac{1}{1+e^{-z}},$$

(This is sigmoid function, where z is w.T*x in our problem)

Based on the Maximum Likelihood Probability method, the likelihood of a set of parameter w could be written as:

$$L(\beta) = \prod_{i=1}^{n} P(y_i \mid x_i; \beta)$$
or
$$L(\beta) = \prod_{i=1}^{n} (h(x_i))^{y_i} (1 - h(x_i))^{1-y_i}$$

(each y is 0 or 1 in this problem)

Notice that the P(y|x, w) is assume as a binary distribution here as we have only two classes.

Take the log form, and add a negative symbol, we then get cost function. Take the derivative of the cost function and as it is hard to directly make it to zero, we then use gradient descent to lead it toward the descending direction.

In formula it is like:

$$\beta_j := \beta_j - \alpha \sum_{i=1}^{n} (h(x_i) - y_i) x_{ij}$$

(alpha is the learning rate here)

We have 1023 data points, which is far bigger than 10 times of feature numbers(14). So this is not a sparse data set.

The specific model I use is the sklearn.linear_model.LogisticRegression, where I try different parameter C (regularization item) to avoid overfitting. And the validation set shows when we choose C bigger than 1, the result converges. But if we take C less than one, then the regularization part weighs too much and causes bad result.

The penalty I used is l1, actually there is no big difference of using l1 or l2. And I choose tolerance as 1e-4. The pre-process result shows the best way is using lib linear solver.

CART

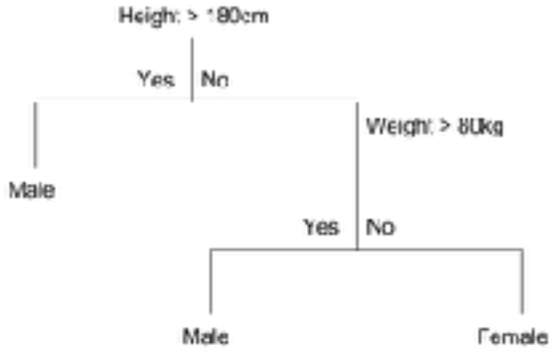
Generally, the decision tree is like a partitioning of the input space where each input value is one dimension and totally forms a p-dimension space. Then the decision tree split it into many “hyper-rectangles”. The problem then becomes how to choose the order of features and the threshold in each feature.

For this problem, the criteria we use is the Gini coefficient.

Gini(p)=p*(1-p)

As we are using binary tree in this problem, for each feature, we find the threshold that minimize the Gini coefficient, and for all the features, we set the one that has minimum Gini coefficient as the node.

The general idea is like the figure shown below:



The model I use for the decision tree is sklearn.tree. To avoid overfitting, we try different value of maximum number of height and nodes.

The result of validation set shows that when the number of nodes and heights are limited too small, the decision tree can not fully grow, and achieve a low prediction accuracy. When we limit the max_depth at and max_leaf_nodes as , it achieves a good result.

The tree model may fits this problem better. As this problem, the heart disease, seems like a logic chain of all factors.

Random Forest

The random forest is developed based on the fact that the decision tree is easy to have an over-fitting problem. And this is why I would like to try it here. In random forest, we combine several decision trees so that each tree has less contribution. It is described as: A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models

We need firstly choose the samples. We choose n samples where each serves for training a tree. Then we develop the CART based on the samples provided.

The model I use here is sklearn.ensemble.RandomForestClassifier, as it is generally a balanced model which is not easy to have overfitting problem, I just set the coefficient as the heuristic set. And I want to have a try whether it works better than the Tree model.

3.5. Model Selection and Comparison of Results

Logistic Regression

The best prediction result of logistic regression is 0.814, and through logistic regression we could have the weight vector w as shown below:

```
[ -1.12862138 -19.52684926  21.7927679   -6.39172298  -8.53394479
  0.70527378   4.066003     12.87339172 -11.21168916 -21.42851353
 10.35075657 -24.68148305 -16.3180217 ]
```

CART

As the prediction accuracy of logistic regression is not good enough, we then try the second model, Classification and Regression Tree.

The validation set shows that it works well with CART model. The best result with tree model is 0.976, which means this model fits well with the problem.

Random Forest

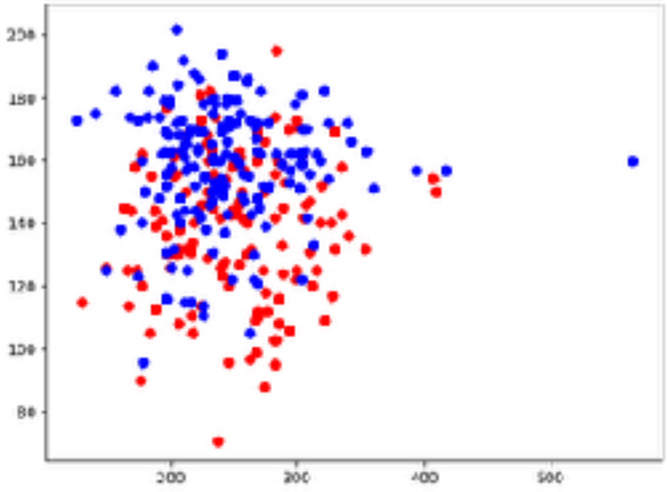
I tried the random forest to find out if the tree model is good enough and if there is any benefits to use random forest which balance the weights of each tree to avoid overfitting.

The result is almost same with the tree model actually, which means the tree model does not have an overfitting problem, just as what we show in its own part.

And I list the comparison of the prediction accuracy of three models below:

Model	Prediction Accuracy(validation set)	Prediction Accuracy(test set)
Logistic Regression	0.882	0.795
Tree	0.951	0.976
Random Forest	0.971	0.966

besides, it is meaningless in this problem to extract two features and draw a decision boundary. Because there are not two features that greatly split the data points. Generally the figure is like :



4. Final Results and Interpretation

The final system I use in this project is the CART model where I set the max_depth as 11 and max_leave_nodes as 60. The resulting tree is shown below. It finally achieves a prediction accuracy for test set of 97.6%.

What is interesting here is that the random forest does not work better than the decision tree in this problem, which means there is no over-fitting phenomenon when applying tree model. So the formed tree reveals some logic that hides behind the heart disease.

More specifically, there are actually explicit "path" to the heart disease, I mean, people can even judge his/her own heart condition based on these factors. However, as shown in the figure, it is not like a "single-path" to heart disease, but there are many possibilities that someone may have such a disease. It makes it more difficult to be applied for prevention of heart disease. Nevertheless, this could still provide some idea when judge someone's heart condition.

From another perspective, we can also consider the weight factor, which is shown below, that we derived from the logistic regression. Although it only achieves a 79.5% prediction accuracy, it provides an intuitive idea how each factor related with heart disease. For example, it shows that the diagnosis is greatly related to chest pain type, number of major vessels and, sex. It is so interesting that sex has occupied such a big weight, and it seems strange. All in all, through logistic regression it only provides with a 79.5% prediction accuracy, so it could not form a definite solution to this problem, but just provide with some rough idea how each feature weighs.

```
[ -1.12 -19.5  21.7   -6.39  -8.53  0.705   4.06    12.8 -11.2 -21.4 10.3 -24.6
-16.3 ]
```

5. Summary and conclusions

The best prediction model for this problem has been proved to be tree model, specifically Classification and Regression Tree which achieved 97.6% prediction accuracy. And the figure of the training tree which is shown above provides some idea about the “logic path” to heart disease. We also derive a weight vector which shows how each feature weighs when influence heart disease.

6. References

[1] "Using machine learning to understand, predict, and prevent cardiovascular disease" 2018. [Online]. Available: https://www.kaggle.com/iamkon/ml-models-performance-on-risk-prediction/comments