Data Mining 2015

Report Assignment 1
Classification Tree

Ferdinand van Walree - 3874389
Rizkiyanto - 4234634

1. Introduction

This report gives an analysis of a classification tree experiment using Heart Disease Dataset from UCI Machine learning repository. The goal of this experiment is to create a model predicts the value of a target variable based on available data.

2. Analysis

This chapter presents the activities that is used to analyze the data. The Heart Disease data has been pre-processed, and all rows with missing value have been removed. The final dataset contains 297 observations of 19 Variables. The attribute Information concerning the data set as follows :

- Age : Age in years
- Sex : 1 = male ; 0 = female
- ChestPain.asymptomatic : 1 = yes ; 0 = no
- ChestPain.nonanginal : 1 = yes ; 0 = no
- ChestPain.nontypical : 1 = yes ; 0 = no
- ChestPain.typical : 1 = yes ; 0 = no
- RestBP : resting blood pressure (in mm Hg on admission to the hospital)
- Chol : serum cholestoral in mg/dl
- Fbs : (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
- RestECG: resting electrocardiographic results (0 = normal, 1 = having ST-T wave abnormality, 2 = showing probable or definite left ventricular hypertrophy)
- MaxHR : maximum heart rate achieved
- ExAng : exercise induced angina (1 = yes; 0 = no)
- Oldpeak : ST depression induced by exercise relative to rest
- Slope : the slope of the peak exercise ST segment (1 = upsloping, 2 = flat, 3 = downsloping)
- Ca : number of major vessels (0-3) colored by flourosopy
- Thal.fixed : 1 = yes ; 0 = no
- Thal.normal : 1 = yes ; 0 = no
- Thal.fixed : 1 = yes ; 0 = no
- AHD : diagnosis of heart disease (Value 0 = < 50% diameter narrowing, Value 1 = > 50% diameter narrowing)

2.1. Preparation

The experiment started with the splitting randomly the data into two parts, the Training data and test data. The training data that consist of 200 records is used to create the model while the remaining 97 records is the test data and is used to verify the model.

2.2. 10-Fold Cross Validation

After splitting the data, 10-Fold cross validation was performed on the training data. The training data partitioned into 10 equal size sub-sample, and a single sub-sample is used as the test set and the other 9 sub-samples are combined to create a classification tree. This activity repeated for every single sub-sample.

Creating the classification tree, there are other parameters that are used, that is nmin and minleaf. Nmin is the minimum number of data that allowed to be split while minleaf is the minimum number of data required for a leaf node. In order to get the best parameter for the model, different settings of nmin and minleaf was performed.

The resulting predicted class labels of the tree were then compared to the true class labels to get the estimated error-rate.

2.3. Choosing the nmin and minleaf

To get the best nmin and minleaf, we first tried to find boundaries wherein the error-rate was clearly lower. After that we observed that values for nmin that are between 10 and 25 produce lower error-rates, as shown in table 1. We then focused on those values combined with different minleaf values. The result is shown in table 2. Moreover, the results indicate that the combination of nmin = 15 and minleaf = 7 generates the lowest error-rate with the value of 0.255.

NO	Nmin	Minleaf	Error-rate
1	1	1	0.31
2	100	100	0.45
3	50	50	0.345
4	70	50	0.345
5	70	25	0.38
6	25	25	0.38
7	25	10	0.28
8	10	5	0.285
9	15	5	0.27

10	25	5	0.285

Table 1. nmin and minleaf

Nmin	**Minleaf**									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
10	0.265	0.275	0.275	0.29	0.285	0.28	0.26	0.27	0.28	0.28
11	0.27	0.28	0.275	0.28	0.29	0.28	0.26	0.27	0.28	0.28
12	0.27	0.28	0.275	0.28	0.29	0.28	0.26	0.27	0.28	0.28
13	0.27	0.27	0.275	0.28	0.29	0.28	0.26	0.27	0.28	0.28
14	0.27	0.275	0.285	0.285	0.285	0.275	0.26	0.27	0.28	0.28
15	0.275	0.285	0.28	0.28	0.27	0.265	0.255	0.27	0.28	0.28
16	0.27	0.28	0.275	0.275	0.265	0.265	0.255	0.27	0.28	0.28
17	0.285	0.29	0.285	0.285	0.275	0.275	0.26	0.275	0.28	0.28
18	0.28	0.285	0.28	0.28	0.275	0.275	0.265	0.275	0.28	0.28
19	0.275	0.28	0.275	0.275	0.27	0.275	0.27	0.275	0.28	0.28
20	0.27	0.275	0.27	0.27	0.265	0.27	0.265	0.275	0.28	0.28
21	0.27	0.275	0.27	0.27	0.265	0.27	0.265	0.275	0.28	0.28
22	0.29	0.295	0.29	0.29	0.29	0.295	0.29	0.295	0.295	0.29
23	0.28	0.285	0.285	0.285	0.285	0.285	0.28	0.285	0.285	0.28
24	0.28	0.285	0.285	0.285	0.285	0.285	0.28	0.285	0.285	0.28
25	0.28	0.285	0.285	0.285	0.285	0.285	0.28	0.285	0.285	0.28

Table 2. nmin and minleaf

3. Results

Applying the test data to the model, that is nmin = 15 and minleaf = 7, we calculated the following error rate: 0.1340206. Clearly this is significantly lower than the error rate of 0.255 that we calculated from our training data. This shows that our model is most likely overfitted. Figure 1 shows the resulting classification tree that was constructed by using the final model on the test data.

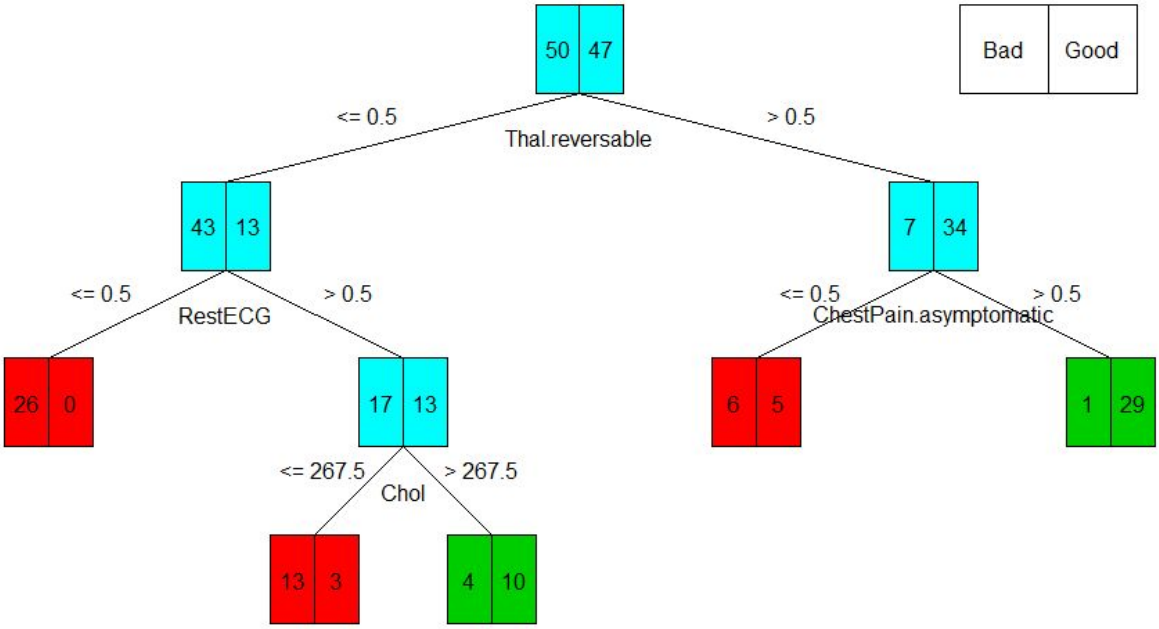

Figure 1. Classification tree of sample data

4. Remark(s)

- In the assignment it is said that tree.simplify is not needed for the above experimentations. For us however, it is needed. In the case that a node has no majority class, we then randomly choose the class label assigned to the node. This means that by not removing such nodes the resulting classification can differ. This then means that the error rate can differ as well. To remove this randomness we use tree.simplify on our tree.
- We actually calculated two combinations of nmin and minleaf values that produced the same error rate. Namely 15 and 7, and 16 and 7. We have used 15 and 7 to further our experiments.