Classification Tree Analysis of Pima data

Ruoyuan Li (001223313)

Pima data Description

My data set called **pima** located at **R** package “**faraway**”, it is a diabetes survey on 768 adult pima indian women. There are nine variables in total, eight predictor variables and one binary response variable. I did some changes to original data, like cleaning, mutating, and selecting for making classification tree building process easier. Then the original **pima** ended with following:

```
## 'data.frame':    768 obs. of  5 variables:
##  $ bmi       : Factor w/ 2 levels "abnormal","normal": 1 1 1 1 1 1 1 1 1 1 ...
##  $ age       : Factor w/ 2 levels "older","yonger": 2 2 2 2 2 2 2 2 1 1 ...
##  $ pregnant_f: Factor w/ 3 levels "low","middle",..: 3 1 3 1 1 2 2 3 1 3 ...
##  $ test      : Factor w/ 2 levels "negative","positive": 2 1 2 1 2 1 2 1 2 2 ...
##  $ diabetes  : num  0.627 0.351 0.672 0.167 2.288 ...
```

where “bmi” is body mass index, it is a categorical variables, “normal” indicates bmi ranges from 18 to 25, otherwise “abnormal”. “age” is categorical variable, ages from 20 to 50 considered to be relatively “younger”, 50 and over are considered to be relatively “older”. “pregnant_f” is categorical variable, **Low** referred to pregnant [0,2] times, **Middle** referred to pregnant (2,5] times, **High** referred to pregnant (5,17] times. “diabetes” is diabetes pedigree function. “test” is response which “negative” means non diabetes patients, and “positive” means diabetes patients.

In this analysis, classification tree is the technical tool, it can visualized the partition of the space of observations. The goal is create a suitable classification tree then use the tree predicting test data, at the end validating the classification performance.

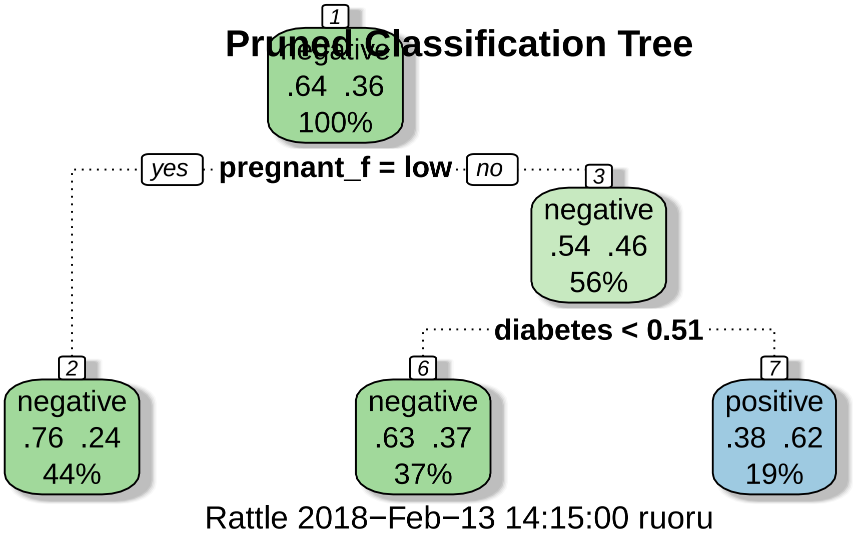
Classification Tree and Predicting

Taking 75% of data as train data, the rest 25% are test data. Using the train data building the classification tree by following:

```
pima_tree <- rpart(test~., data=pima1, subset = pima_train, method="class", minsplit=25)
```

It ends with ten terminal nodes at the end, at it splitting over eight times. Under this situation, pruning the tree may be a better choice. Pruning process is based on “cp” which is “Complexity Parameter” from the **rpart** object. The following function returns the optimal cp value associated with the minimum error which complete pruning process directly.

```
ptree<- prune(pima_tree,
                 cp= pima_tree$cptable[which.min(pima_tree$cptable[,"xerror"]),"CP"])
```



The following tree diagram visually shows the result of classification tree, the root node contain 64% of the training set from test result is negative, and 36% of the training set from the test results is positive. “pregnant_f” categorized to low level is the first splitting occurs. It means the test results of patients for low level pregnant times are classified into negative test results. It formed one of three final terminal node. The first right terminal contain 56% of training data, 54% of them are correctly classified into negative test results, 46% of them misclassified into positive test results, so will keep splitting. “diabetes” bigger than 0.51 is second splitting, it means the test results patients for diabetes less than 0.51 are classified into negative results, otherwise classified into positive test results. There are total 3 terminal nodes at the end after two splitting. Their interpretations are following:

- the first terminal node from left contain 44% of the training data, 76% of the observations are correctly classified into negative test results, and 24% of the observations are misclassified into positive test result.

- the second terminal node from the left contain 37% of the training data, 63% of the observations are correctly classified into negative test result, and 37% of the observations misclassified into positive test results.
- the final terminal node from the left contain 19% of the training data, 62% of the observations are correctly classified into positive test results, 38% of the observations are misclassified into negative test results.

Then using the pruned classification tree from training data to predict test data, and make comparison based on predicting results. First check the table compared between predicted results and actual results.

Table 1: prediction with actual table

	negative	positive
negative	117	46
positive	15	14

From this table, there are total 61 misclassified out of 192 observations. 15 of positive results are misclassified into negative test results, 46 pf negative results are misclassified into positive test results. The misclassification rate is around 32% which calculating by **e1071::classAgreement**.

In conclusion, the classification tree is simple and easy to understand. However, based on performance of predicting test data, the classification tree from training data is not as good as we expected. It may be caused by selecting variables, I may need to include all the variables from the original data. Further analysis can be explore more variables of the classification tree. Also it may be caused by data itself, since there are more negative test results (500 observations in total) than positive test results (268 observations in total). Based on limit variables, the classification tree and performance are okay.