Assignment#2 - Decision Tree

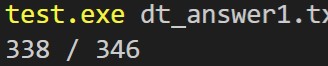
2014004066 서왕규

<develop enviroment>

Window10

Python3.8

<consequence>



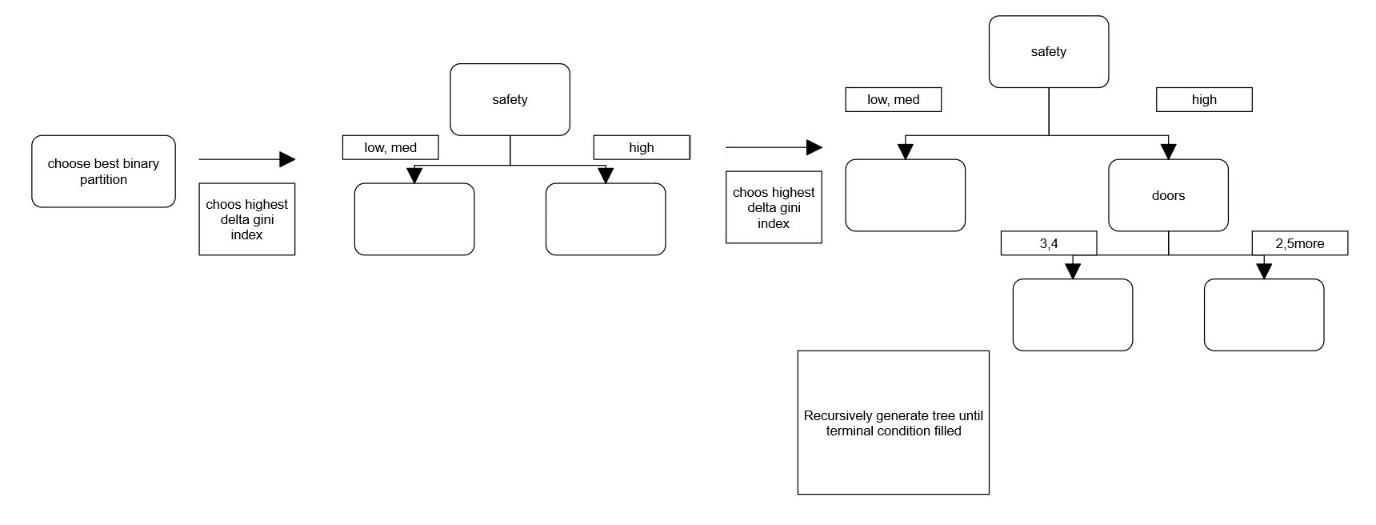
1. Algorithm



<measurement> Gini index

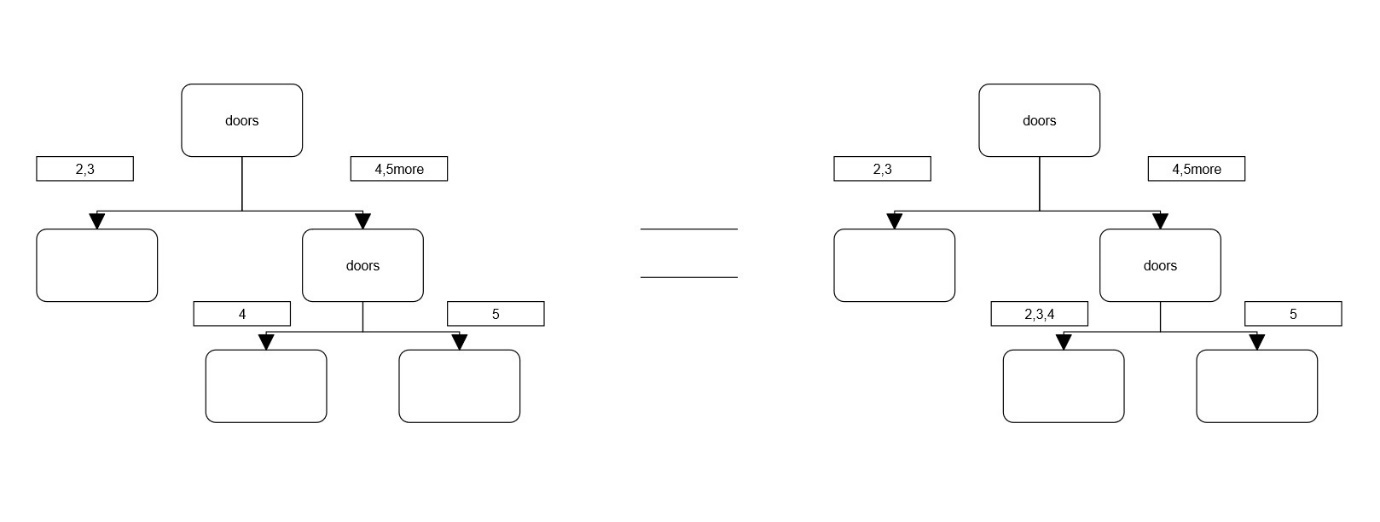
This program use measurement of **gini index**, but my code also has implementation of information gain + gain ratio. **If tester wants to test which measurement is better, please substitution parameter ‘gini’ to ‘gain’ in output function in main.** I choose gini index measurement because it has more accuracy to test program.

[1] Making all binary partition in every attribute.

 [2] Calculating gini index starting at root, find highest among set of binary partition recursively

Gini(D) = 1 –

Find Highest

- < considerable problem situation in implementation > Doors : 2,3,4,5more

If first chosen binary partition is doors(2,3)/doors(4,5more), but, still highest delta gini of binary partition is doors(4)/doors(5more) in doors(4,5more) group?

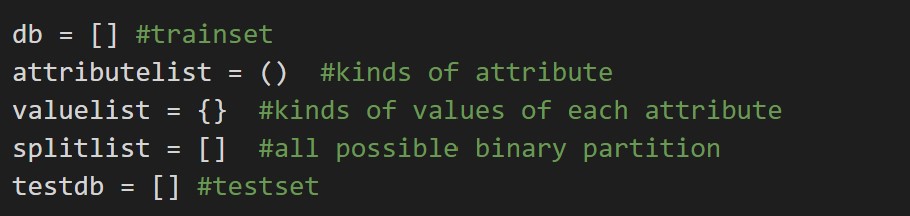
-> For solving this situation, [1]step make all binary partition in all attribute. And choose highest among them. So, if that situation, next chosen will be gini:doors(2,3,4)/gini:doors(5more) partitioning.

It is same consequence of doors(2,3)/doors(4,5more) -> doors(4)/doors(5more) in . Because there are already no candidates who have 2,3 of doors in doors(4,5more) group.

- <terminal condition> when no more benefit of or no sample in leaf node, or reach predefined depth, recursive tree making terminate.

[3] post pruning : choose appropriate depth from statistic data in testing

2. detailed description of my codes

<global variable>

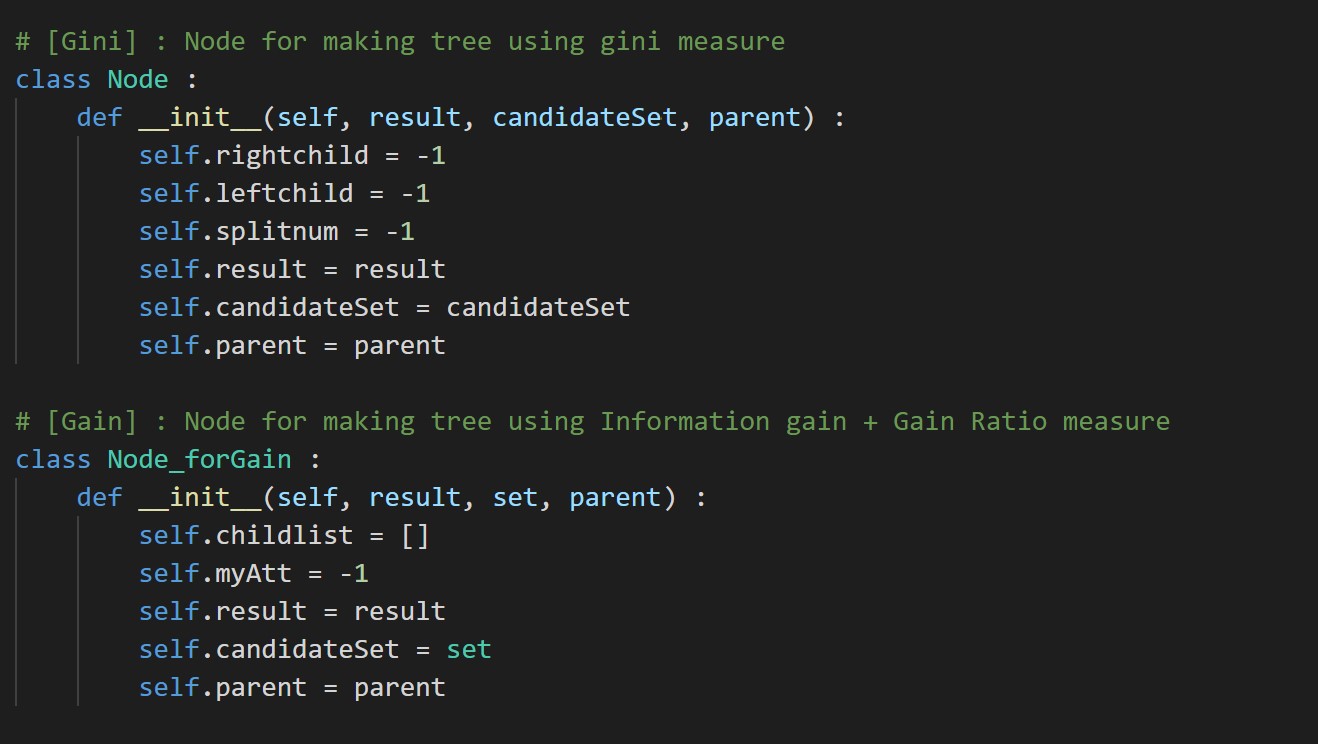
Db : It is list for train data

Attributelist : A tuple that has kinds of attribute

Valuelist : A dictionary for saving kinds of values of each attribute

Splitlist : list for saving all possible binary partition

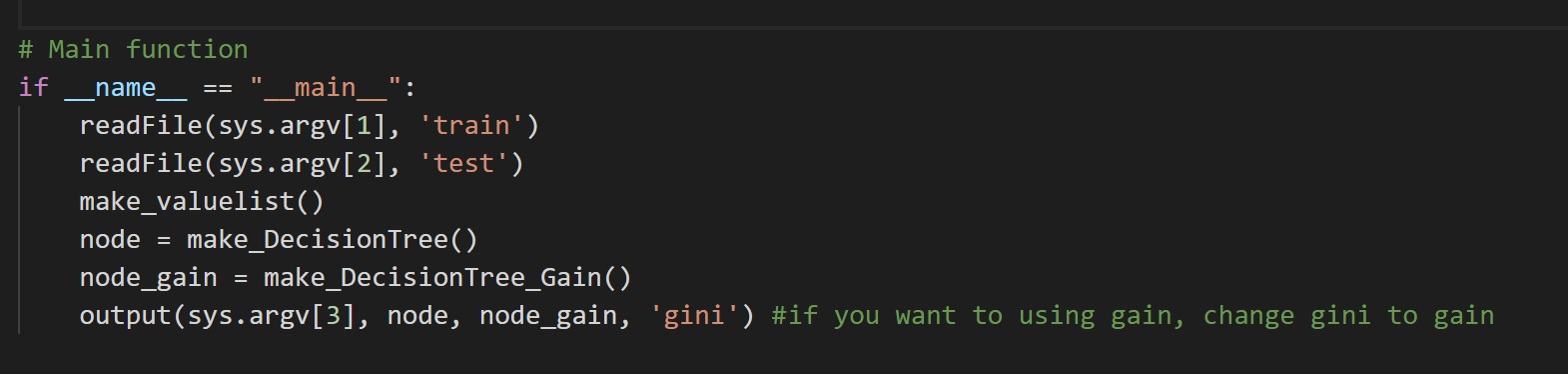
Testdb : list for saving test data

<class>

Node : It is class for representing a node composed of decision tree that made by gini index measure

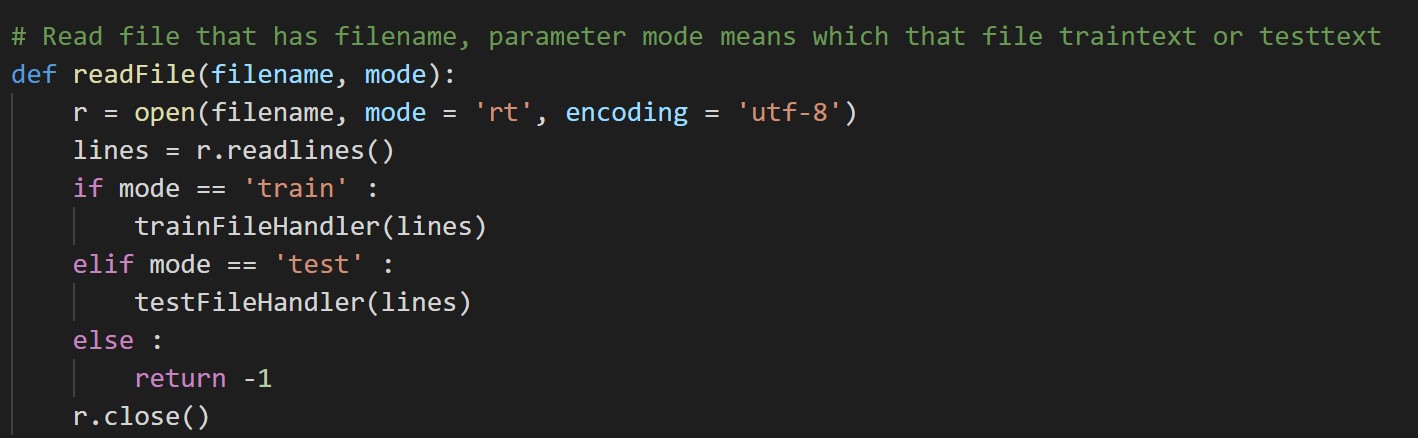
Node\_forGain : It is class for representing a node composed of decision tree that made by information gain and gain ratio measure. It is not used in my program in final section, it is for test.

**If tester wants to test which measurement is better, please substitution parameter ‘gini’ to ‘gain’ in output function in main.**

[1] main

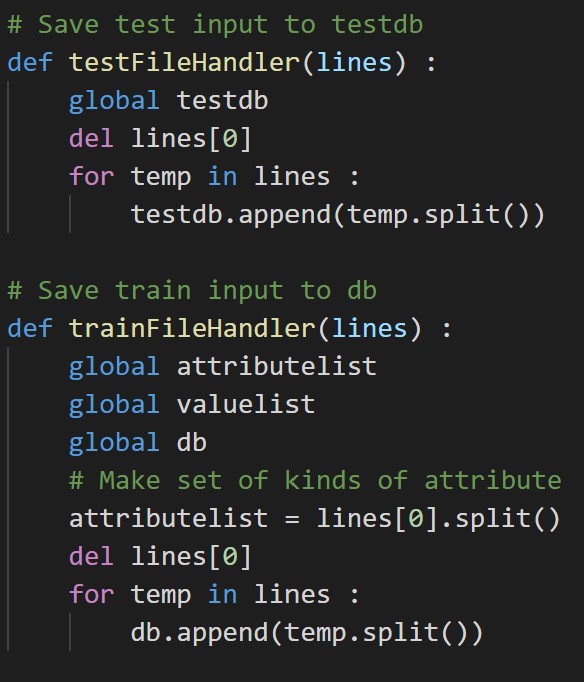
That is main function call read argument1,2 text file.

Make 2 kinds of decision tree, for finding more appropriate model taking high score on test program. After trying, I choose Gini index measurement. Detail is on 3. Statistic.

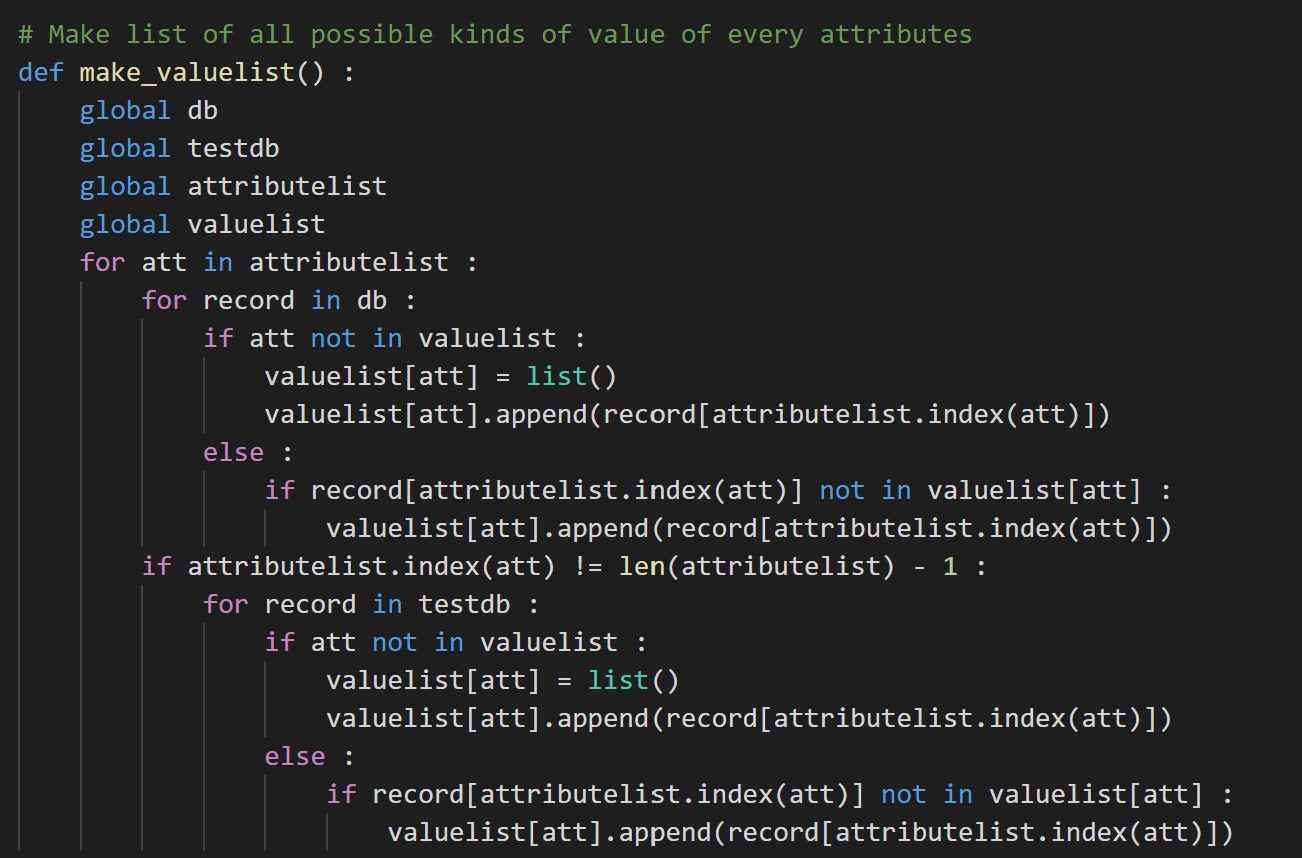
 [2] readFile

Read file and mode means that file is test file or train file

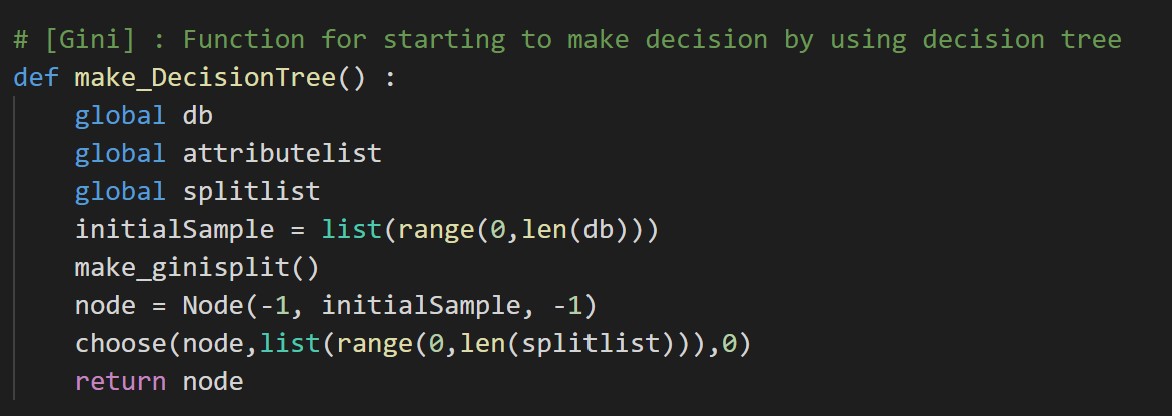
[3] file Handler functions

Read test file and train file. And save to list its data.

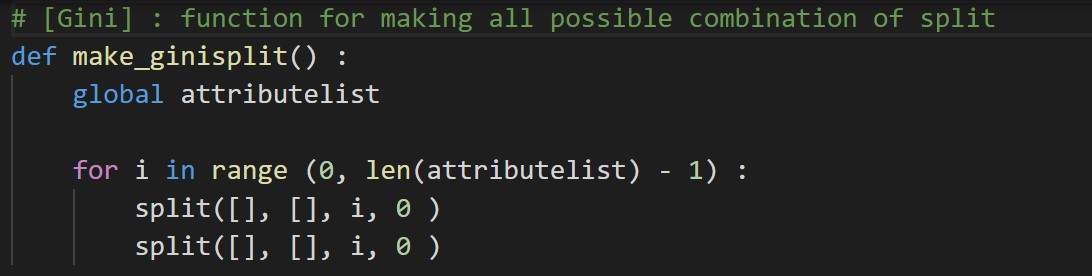
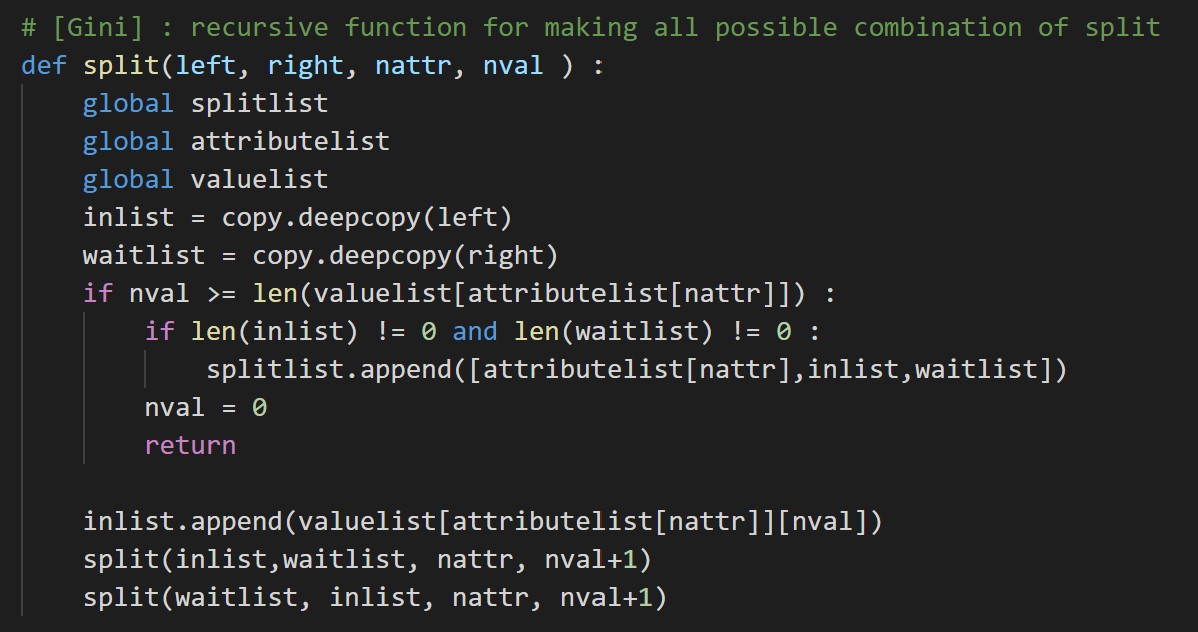
In trainFileHandler, attributelist is generated

[4] make\_valuelist

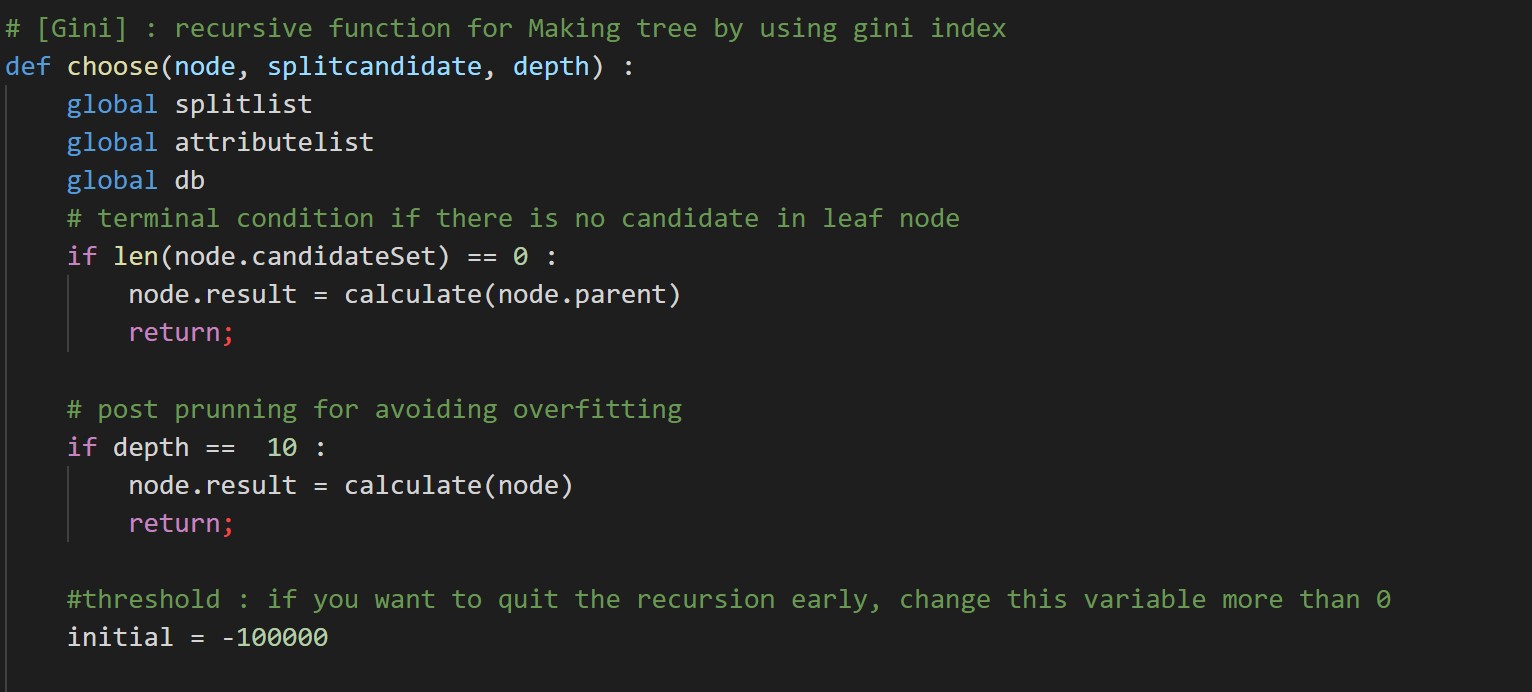
Making list of all possible kinds of value of every attributes from train set and test set.

[5] make\_DecisionTree

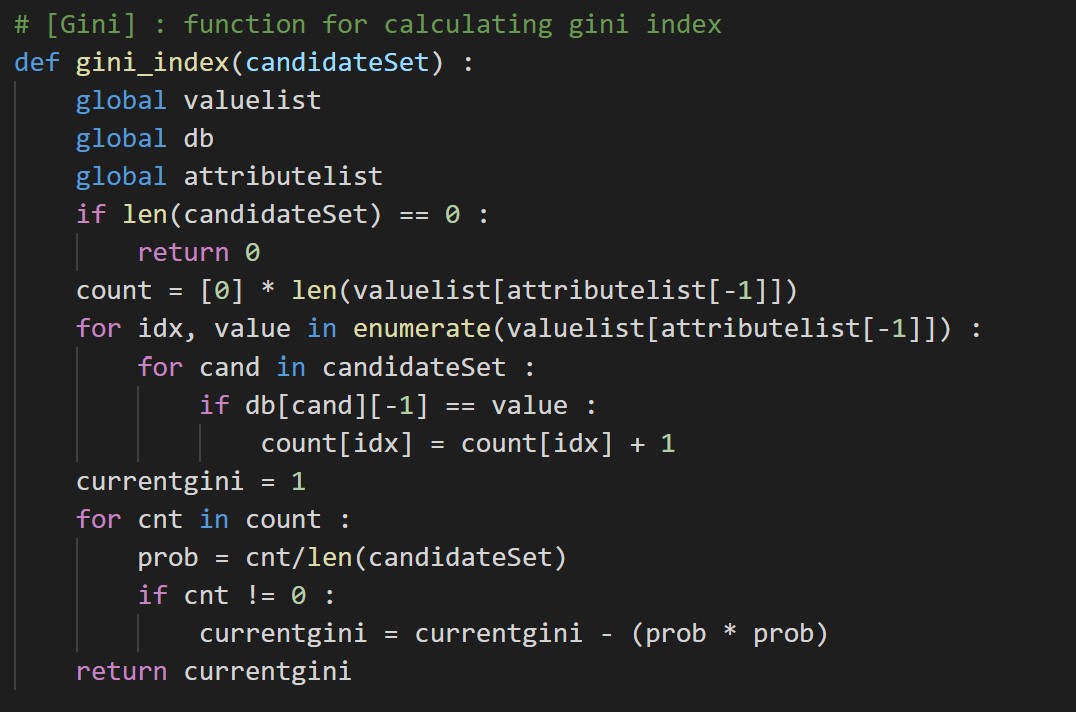
This function call split function that makes all possible binary partition at all attribute, and make a root node of decision tree, and then call recursive function(choose) complete making tree.

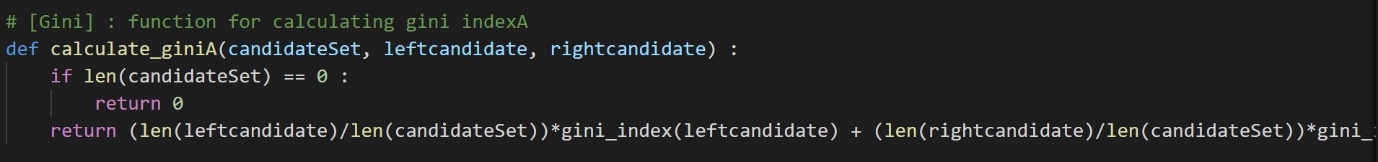
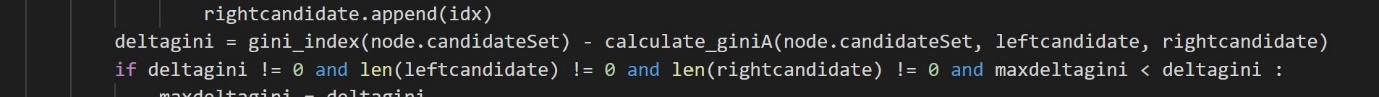
[6] split

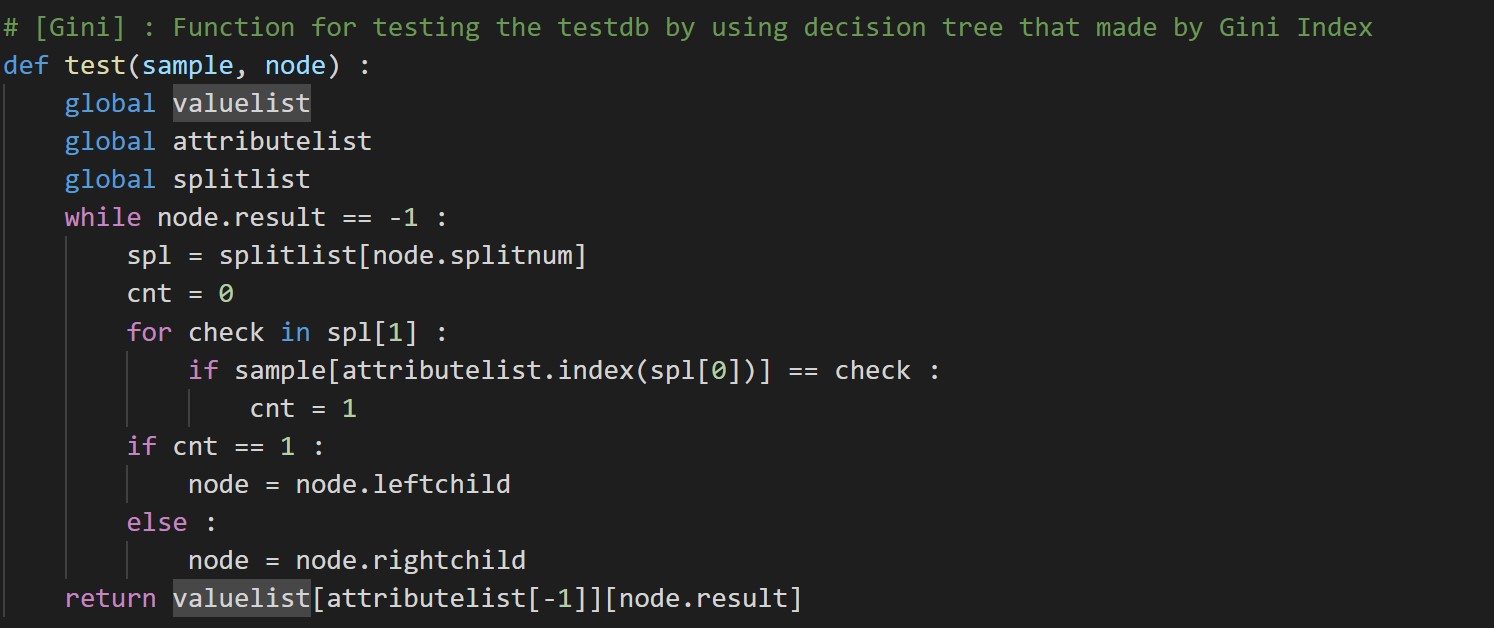
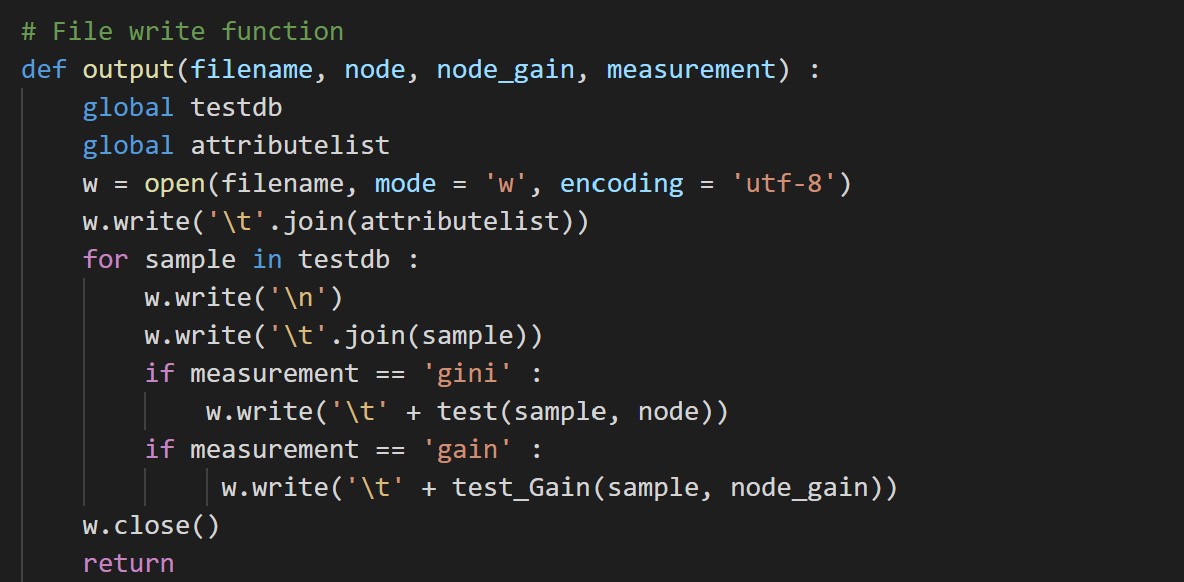
Making all possible combination of binary partition from all attribute and save it to splitlist. It remove skewed to one side partitioning that one part has no member.

[7] choose

It make a whole tree by using gini index measurement. There is depth for post pruning and initial variable can be act as threshold for pruning

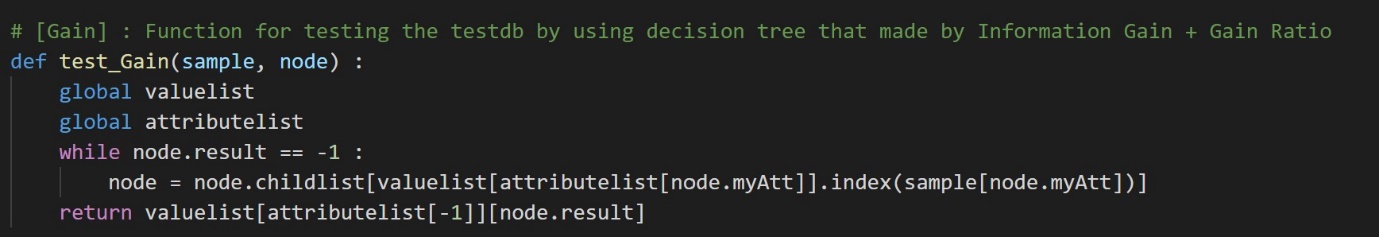
[8] gini index functions

Gini(D) = 1 –

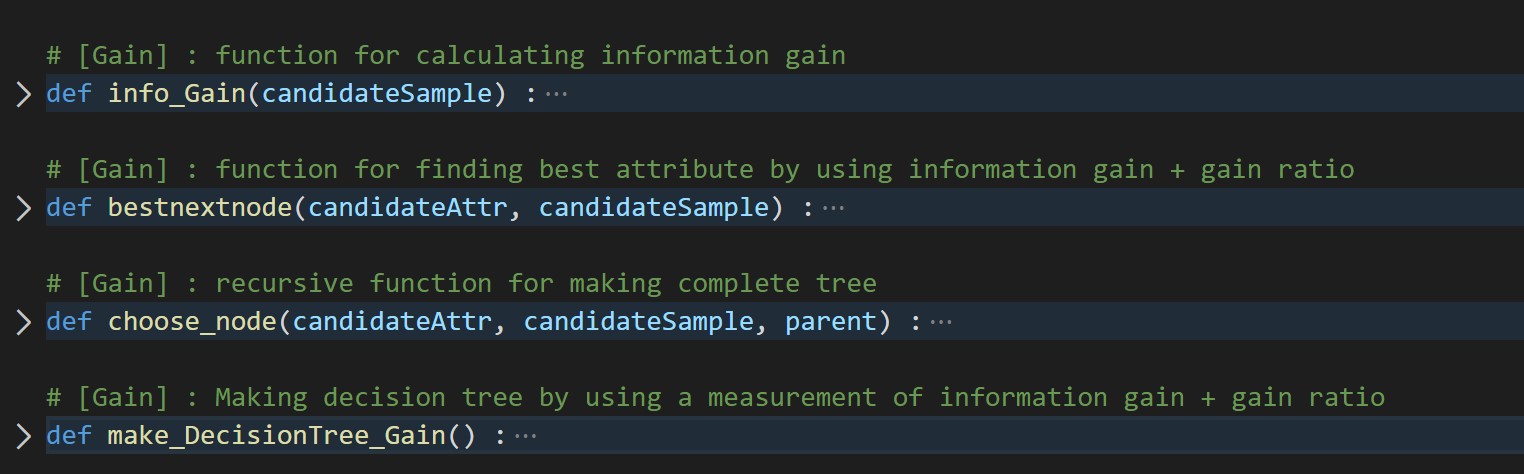
[9] output and test function

Put test data set to tree and find their class from there’s departure of tree

File write function, if 4th parameter ‘gain’ or ‘gini’, you can choose measurement.

[10] information gain functions

Put test data set to the tree that made by gain ratio and then find class label

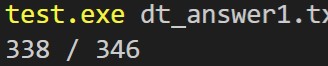
Info\_gain : function for calculating information gain

bestnextnode : function for calculating information gain

choose\_node : recursive function for making complete tree

make\_DecisionTree\_Gain : Making decision tree by using a measurement of information gain + gain ratio

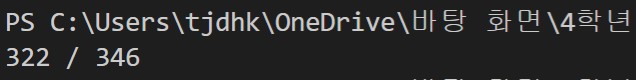
3. Statistic(about prunning) for finding best measure and threshold

depth 10 is higher score in test program

|  |  |
| --- | --- |
| Depth | Test result |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |
| 12 |  |
| 13 |  |

More than 12, no pruning in this trainset

Under capture is Gain ratio consequence.



4. Instruction for compiling



My source code is using language of python. So, using python interpreter,

Syntax > Python dt.py (train file name) (test file name) (output file name)