**Decision Tree Report**

2013011082 이상옥

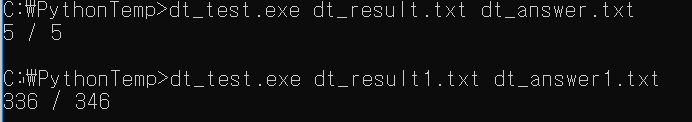
**1. Result & How to run program**

**How to run program**

(python version 3.6)



**Output result**

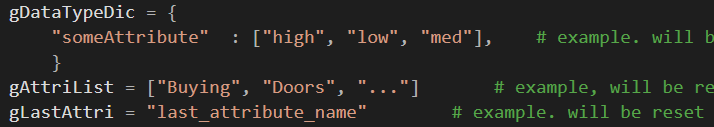


Case 0: 5/5 correct

Case 1: 336/346 correct

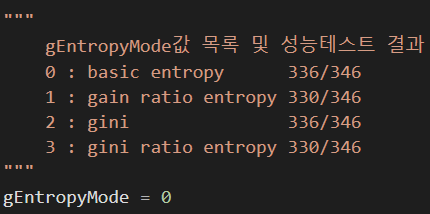
**2.** **Flexibility, Decision making rule**

Decision Tree in *DecisionTree.py* have higher flexibility than required



It can get any kinds of input **except infinite range number input** (of course it regards number as string and try to calculate, but performance will be poor) because it registers input on *gDataTypeDic* on runtime, data don’t need to be ordered.

If you only follow tab(‘\t’) restrict on assignment input rule, it will take **any input you give**.



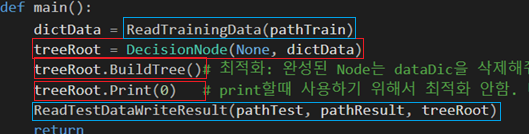
You can change how this program make decision by changing global value.

In default, it used basic entropy gain mode.

**3. Major public class, function, global values**

Some important **codes that user need to know** will be call **Public** in this report

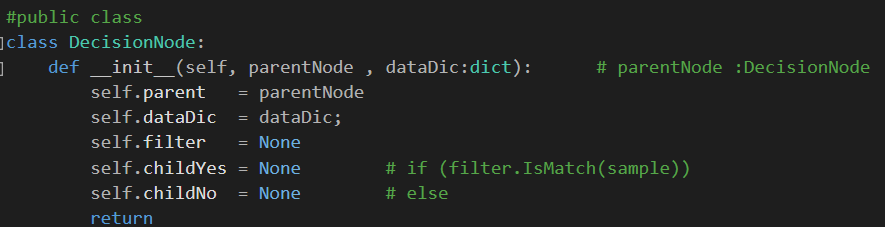
In *main()*, you can see example of how public code used. All you need to know is these.



**3.1 public class: *DecisionNode***

*DecisionNode* class is node for decision tree.

It has 1 filter, 2 child nodes, data dictionary they got when first constructed.



Also supporting 3 methods.







*rootNode.BuildTree()* build whole decision tree from *self.dataDic.*

*rootNode.Print(0)* will show how and why decision filters are set

*rootNode.GuessWhat((“val0a”, “val1c”, “val2b”, …))* will return guessed final value by decision tree structure.

**3.2 public function: *ReadTraingData(pathTrain), ReadTestDataWirteResult(…)***



Read training data from file, returned value will be used for root node of *DecisionNode*



Read testing data from test path, write answer on result path