Data Science

Assignment 2 Decision tree

Prof. Sangwook Kim  
TAs. Jangwan Koo, Yeonchang Lee

2013011424  
Yedarm Seong  
Division of Computer Science and Engineering

1. Environment

OS: Windows 10  
Language: Python 3.5.2

2. Summary of algorithm

(1) Calculate cross entropy

for features in dataset  
 label.count += 1  
for each label  
 entropy calculate  
return entropy

(2) Choose best feature

base entropy = calculate cross entropy(dataset)  
for each features except label  
 for feature in features  
 sub dataset = split(data\_set)  
 entropy = calculate cross entropy(sub dataset)  
 info gain = base entropy – entropy  
return feature is max entropy

(3) Create tree

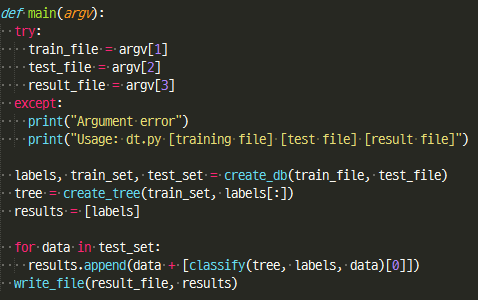
Check if every item in the dataset is in the same class  
 if so return the class label  
 else  
 Find best feature  
 Split the dataset  
 Create a branch node  
 for each split  
 Create tree, add the result to the branch node  
 return branch node

(4) Classify

go down the recursively created tree and check the list that came in as input.

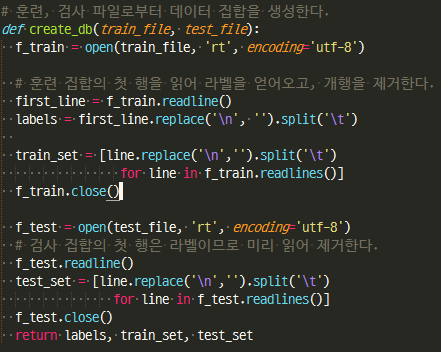
3. Detailed description of codes (for each function)

Main function



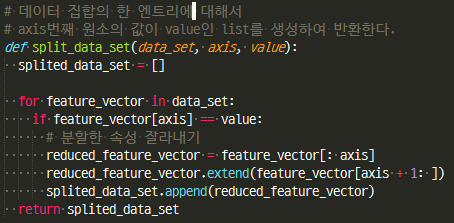
The training file, the test file, and the result file.  
If any one of them has an invalid path, an error message is output.  
Create training sets, test sets, and labels from the read training files and test files.  
Create a tree from the training set and test the test set.  
Write the results in a file according to the assignment type.

Function for create database and label



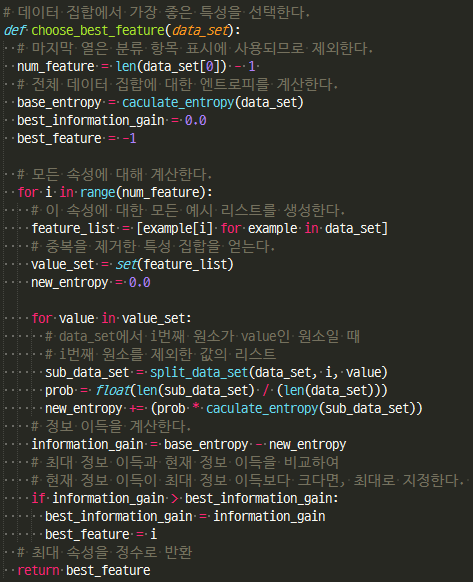
Read the file data create a python list for the set.  
Create a label.

Function for calculate entropy and gini index



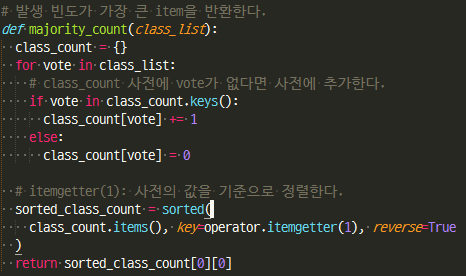
For a single entry in the dataset, create a list whose value is the element of the axis index.

Function for choose best feature



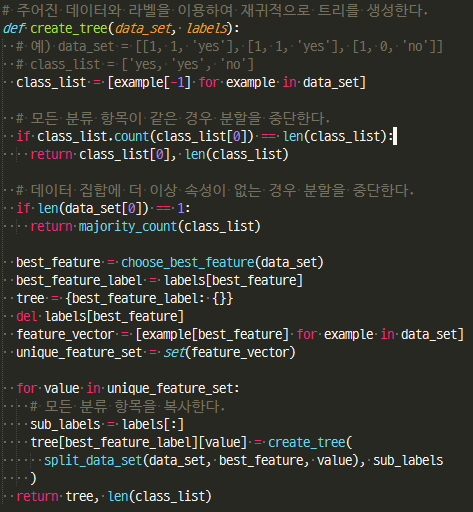
Calculate the base entropy for the entire data set.  
Computes each entropy for all attributes, and calculates the information gain.  
Returns the property with the highest information gain.

Function for majority vote count



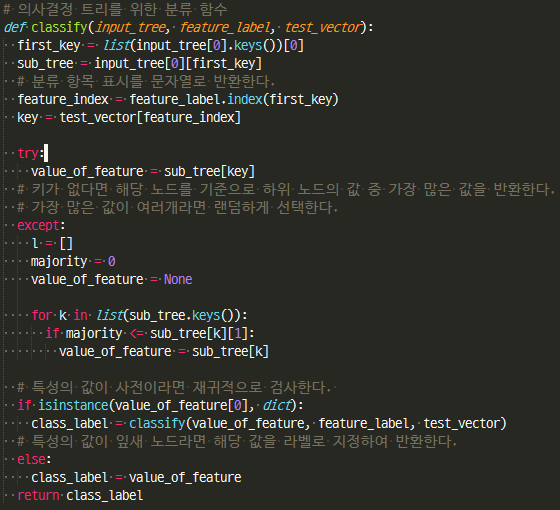
If there are no more attributes in the dataset, return the most frequent class label.  
Calculate the frequency for each class label, sort, and return the label with the highest value.

Function for create tree



The most important function is a function that recursively creates a tree.  
A tree is created from input data sets and labels.  
If all the classification items are the same or there is no more attribute to divide, the node becomes a leaf node.  
Select the best property to divide, and create a subtree with the label as key.  
It recursively creates the tree in a top-down fashion.  
The result of the tree is the python dictionary, or tuple (dictionary, occur count), which stores the number of occurrences as value.

Function for classify



Tests and classifies test\_vector from input\_tree.  
If the subtree of input\_tree is a python dictionary type, it calls classify recursively until it is not a dictionary.  
If it is a leaf node, it returns the corresponding label because it is no longer a dictionary type.  
If there is more than one of the split labels, it will be selected randomly to avoid overfitting.

4. Instructions for compiling source codes at TA's computer (e.g. screenshot) (*Important!!*)

**Python does not need a compile process**

If you already install pyinstaller for python3, in the directory where the dt.py file is located, type the following command:

Windows:  
pyinstaller -F dt.py

Ubuntu:  
pyinstaller -F dt.py

Then the dist folder is created in the directory where the command is executed, and there is ‘dt.exe’ file in it.(For linux, the ‘dt’ file)

Usage: dt.exe [training data file] [test data file] [result file]

After using pyinstaller, there is an exe file in the dist folder:  
ex) dist/dt.exe dt\_train.txt dt\_test.txt dt\_result.txt

If you move the exe file to the same location as dt\_train.txt and dt\_test.txt you can use it like this:  
ex) dt.exe dt\_train.txt dt\_test.txt dt\_result.txt

5. Any other specification of implementation and testing

Another execution method:  
Running as a Python file

ex) dt.py [training data file] [test data file] [result file]