Course name: Data Science (ITE4005)

Professor: Sang-Wook Kim (email: wook@agape.hanyang.ac.kr)

TAs: Dong-hyuk Seo (email: hyuk125@agape.hanyang.ac.kr)
Jiwon Son (email: tinybeing@agape.hanyang.ac.kr)

< Programming Assignment #2 >

4 Apr. 2022

Due Date: 25 Apr. 2022, 11:59 pm

1. Environment

- OS: Windows, Mac OS, or Linux
- Languages: C++, Java, or Python (any version is ok)

2. Goal: Build a **decision tree** with training set, and then classify the test set using it

3. Requirements

The program must meet the following requirements:

- Execution file name: **dt.exe**
- Execute the program with three arguments: training file name, test file name, output file name
  - Example:

```
C:₩>dt.exe dt_train.txt dt_test.txt dt_result.txt
```

    - Training file name: ‘dt_train.txt’, test file name: ‘dt_test.txt’, output file name: ‘dt_result.txt’
    - If using python, you are allowed to use 'dt.py' file instead of 'dy.exe'.
- Dataset
  - We provide you with 2 datasets
    - Buy_computer: dt_train.txt, dt_test.txt
    - Car_evaluation: dt_train1.txt, dt_test1.txt
  - You need to make your program that can deal with **any** datasets
  - We will evaluate your program with other datasets.
- File format for a training set

  [*attribute_name_1*]\t[*attribute_name_2*]\t … [*attribute_name_n*]\n

  [*attribute_1*]\t[*attribute_2*]\t … [*attribute_n*]\n

  [*attribute_1*]\t[*attribute_2*]\t … [*attribute_n*]\n

  [*attribute_1*]\t[*attribute_2*]\t … [*attribute_n*]\n

  - [*attribute_name_1*] ~ [*attribute_name_n*]: *n* attribute names

- [*attribute_1*] ~ [*attribute_n-1*]
  - *n-1* attribute values of the corresponding tuple
  - All the attributes are categorical (not continuous-valued)
- [*attribute_n*]: a class label that the corresponding tuple belongs to
- Example 1 (data_train.txt):

age	income	student	credit_rating	Class:buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31...40	high	no	fair	yes
>40	medium	no	fair	yes

Figure 1. An example of the first training set.

- Example 2 (data_train1.txt):

buying	maint	doors	persons	lug_boot	safety	car_evaluation
high	high	3	4	big	low	unacc
med	high	2	2	small	med	unacc
low	med	5more	2	big	high	unacc
low	high	2	4	med	low	unacc
med	vhigh	4	2	med	med	unacc

Figure 2. An example of the second training set.

  - Title: car evaluation database
  - Attribute values
    - Buying (Buying price): vhigh, high, med, low
    - Maint (Price of Maintenance): vhigh, high, med, low
    - Doors (Number of Doors): 2, 3, 4, 5more
    - Persons (Capacity of persons to carry): 2, 4, more
    - Lug_boot (Size of Luggage Boot): small, med, big
    - Safety (Safety of car): low, med, high
  - Class labels: unacc, acc, good, vgood
  - Number of instances: training set - 1,382; test set - 346

- Attribute selection measure: **information gain, gain ratio, or gini index**
- File format for a test set

[*attribute_name_1*]\t[*attribute_name_2*]\t … [*attribute_name_n-1*]\n

[*attribute_1*]\t[*attribute_2*]\t … [*attribute_n-1*]\n

[*attribute_1*]\t[*attribute_2*]\t … [*attribute_n-1*]\n

[*attribute_1*]\t[*attribute_2*]\t … [*attribute_n-1*]\n

- The test set does not have [*attribute_name_n*] (class label)
- Example 1 (dt_test.txt):

age	income	student	credit_rating
<=30	low	no	fair
<=30	medium	yes	fair
31...40	low	no	fair

Figure 3. An example of the first test set.

- Example 2 (dt_test1.txt):

buying	maint	doors	persons	lug_boot	safety
med	vhigh	2	4	med	med
low	high	4	4	small	low
high	vhigh	4	4	med	med
high	vhigh	4	more	big	low
low	high	3	more	med	low

Figure 4. An example of the second test set.

- Output file format

[*attribute_name_1*]\t[*attribute_name_2*]\t … [*attribute_name_n*]\n

[*attribute_1*]\t[*attribute_2*]\t … [*attribute_n*]\n

[*attribute_1*]\t[*attribute_2*]\t … [*attribute_n*]\n

[*attribute_1*]\t[*attribute_2*]\t … [*attribute_n*]\n

  - Output file name: **dt_result.txt (for 1th dataset), dt_result1.txt (for 2nd dataset)**
  - You must print the following values:
    - [*attribute_1*] ~ [*attribute_n-1*]: given attribute values in the test set
    - [*attribute_n*]: a class label predicted by your model for the corresponding tuple
  - Please **DO NOT CHANGE the order of the tuples** in each test set.
    - You should print your outputs to match the order of correct answers.
  - **Please be sure to use \t to identify your attributes.**
  - **Be sure to match the output format!**
    If the format is not correct, you can't get any score.

5. Submission

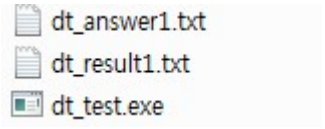
- Please submit the program files and the report **to *GitLab***
  - Report
    - File format must be *.pdf.
    - Guideline
      - ✓ Summary of your algorithm
      - ✓ Detailed description of your codes (for each function)
      - ✓ Instructions for compiling your source codes at TA's computer (e.g. screenshot) (*Important!!*)
        - If TAs read your instructions but cannot compile your program, you will get a penalty. Please write the instructions carefully.
      - ✓ Any other specification of your implementation and testing
  - Program and code
    - An executable file (.exe or .py)

- ✓ If you are not in the following two cases, please submit alternative files (e.g., jar file, makefile, …)

- All source files

6. Testing program

- Please put the following files in a same directory: Testing program, your output files (dt_result.txt, dt_result1.txt), an attached answer file (dt_answer.txt, dt_answer1.txt)



- Execute the testing program with two arguments (answer file name and your output file name)

```
DM_assignment2>dt_test.exe dt_answer1.txt dt_result1.txt
```

- Check your score for the input file

```
346 / 346
```

  - the number of your correct prediction / the number of correct answers

- The test program was built with program 'mono'. So, even if you are using mac or linux instead of window, you can run dt_test.exe using C# mono.

7. Penalty

- Late submission
  - 1 week delay: 20%
  - 2 weeks delay: 50%
  - Delay more than 2 weeks: 100%
- Requirements unsatisfied
  - Significant penalty up to 30% will be given when the requirements are not satisfied