

# Assignment1 report - Data Science

Instructor : 채동규

Student name / ID : 손승일 / 2018008504

Apriori algorithm

## 1. Environment

Ubuntu : 20.04.5 LTS

Python : 3.8.10

(vscode로 짜다가 터미널에서 input file을 못읽어서, Ubuntu\_linux 환경으로 옮겨보니 작동됨)

## 2. How to run this code

```
os@os-VirtualBox: ~/data_science
os@os-VirtualBox:~/data_science$ ls
apriori.py input.txt
os@os-VirtualBox:~/data_science$ python3 apriori.py 5 input.txt output.txt
os@os-VirtualBox:~/data_science$ ls
apriori.py input.txt output.txt
os@os-VirtualBox:~/data_science$ vim output.txt
os@os-VirtualBox:~/data_science$
```

Python3	apriori.py	5	input.txt	output.txt
	File name	minimum support	input file	output file

## 3. Result

```
output.txt (~/data_science) - VIM
1044 {8,16} {3,17} 5.80 19.21
1045 {3,8,16} {17} 5.80 24.17
1046 {17} {3,8,16} 5.80 24.37
1047 {3,17} {8,16} 5.80 76.32
1048 {8,17} {3,16} 5.80 48.33
1049 {3,8,17} {16} 5.80 85.29
1050 {16,17} {3,8} 5.80 44.62
1051 {3,16,17} {8} 5.80 93.55
1052 {8,16,17} {3} 5.80 65.91
1053 {3} {8,10,16} 7.00 23.33
1054 {8} {3,10,16} 7.00 15.49
1055 {3,8} {10,16} 7.00 27.13
1056 {10} {3,8,16} 7.00 24.14
1057 {3,10} {8,16} 7.00 83.33
1058 {8,10} {3,16} 7.00 52.24
1059 {3,8,10} {16} 7.00 92.11
1060 {16} {3,8,10} 7.00 16.51
1061 {3,16} {8,10} 7.00 27.78
1062 {8,16} {3,10} 7.00 23.18
1063 {3,8,16} {10} 7.00 29.17
1064 {10,16} {3,8} 7.00 52.24
1065 {3,10,16} {8} 7.00 94.59
1066 {8,10,16} {3} 7.00 81.40
1066,1 바닥
```

Minimum support를 5로 잡고 올려주신 input.txt 파일을 입력받고 실행하면 1066개의 결과값이 output 파일에 저장됨.

#### 4. Specifics of the code

1. def read\_file : Read input file

2. def generate\_first\_candidate : Generate 1- candidate item set from the db

3. def generate\_candidate : Generate K-candidate item set. Make K-candidate item set from (k-1) frequent item set

4. def prune : Check if the candidate's support matches the minimum support and if it doesn't, eliminate the candidate and if it does, insert it to k-candidate item set.

5. def get\_sup\_cnt : calculate the candidate's support count

6. def generate\_frequent : make K-frequent item set from k-candidate item set

7. def write\_file : write the frequent item sets at the output file

8. main : run till the candidate itemset and frequent item set doesn't come out. If the While true ends, function write\_file will be executed and the code will end.