**Programming Assignment #1**

컴퓨터소프트웨어학과 2017029470 김종원

1. **Environment**

* Mac OS (Monterey) M1 chip
* Python 3.10.3 (release March 16, 2022)

1. **How to compile**

Before compiling apiori.py, python version 3 must be installed in your system.



* Execution file name: apriori.py
* Minimum support: 5%
* Input file name: input.txt
* Output file name: output.txt

1. **Summary of algorithm**

Apriori algorithm is for finding frequent item sets in a dataset for association rules. I apply an iterative approach where k-frequent item sets are used to find k+1 item sets.

* Initially, scan database once to get frequent 1-itemset.
* Generate candidate item sets of length k+1 from frequent item sets of length k by self-joining. (Self-join is a regular join, but the table is joined with itself.)
* And pruning the candidates to prevent huge number of candidates.
* After checking the support for the remaining candidates, those exceeding the minimum support are classified into a frequent pattern.
* Terminate when no frequent or candidate set can be generated.

1. **Detailed description of codes**

텍스트이(가) 표시된 사진

자동 생성된 설명

* The apriori function receives two variables: transaction and minimum support as arguments. The purpose of this function is to generate association rules by applying the apriori algorithm.
* First, I scan the transaction database to generate frequent patterns of length 1. (line 15 ~ line 24)
* As described above, from frequent pattern of length k, frequent pattern of length k+1 is created, and if no more creation is possible, the loop is terminated. (line 25 ~ line 43)
* Using the obtained frequent pattern, check whether an association rule can be created. (line 45)

텍스트이(가) 표시된 사진

자동 생성된 설명

* The get\_support function receives two variables: itemset and transaction as arguments. The purpose of this function is to return the number of transactions which include itemset.

텍스트이(가) 표시된 사진

자동 생성된 설명

* The self\_join function receives two variables: frequent patterns and length as arguments. The purpose of this function is to return joined candidate set, which is made from previous frequent patterns.
* After putting the items constituting all patterns in the frequent pattern into joined candidates list and using Python’s built-in itertools to create tuples and store them in a set. (line 59 ~ line 67)

텍스트이(가) 표시된 사진

자동 생성된 설명

* The prune function receives three variables: candidates, previous frequent patterns and length as arguments. The purpose of this function is to return pruned candidate set, which is reduced form of candidates made through self-joining.
* If a certain pattern is to become a frequent pattern, all subset of the corresponding pattern must be included in the existing frequent pattern. In the prune function, the case in which the pattern length is 1 and the case in which the pattern length exceeds 1 were considered separately, because the result of combination is in the form of tuples.
* If a subset of a certain pattern is not included in the existing frequent pattern, it is excluded from the candidate. (line 72 ~ line 90)

텍스트이(가) 표시된 사진

자동 생성된 설명

* The test support function receives three variables: candidates, minimum support as number, and transaction as arguments. The purpose of this function is to return frequent patterns.
* First, deep copy pruned candidates in frequent pattern set. If the number of transactions containing the itemset is less than the minimum support, the itemset is removed from the frequent pattern set. (line 97 ~ line 99)

텍스트이(가) 표시된 사진

자동 생성된 설명

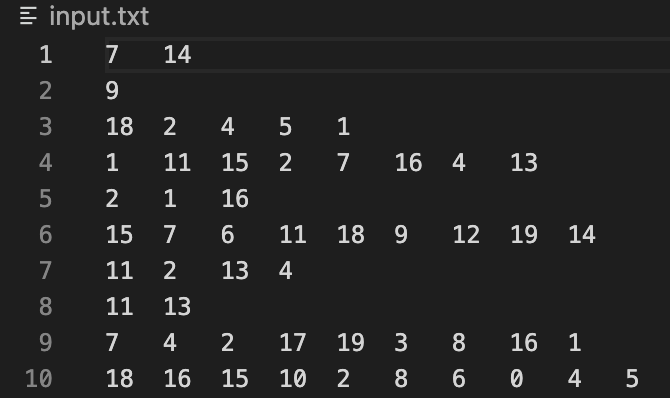
* The association rule function receives three variables: frequent patterns, length, transaction as arguments. The purpose of this function is to generate association rules using frequent patterns.
* For a frequent pattern, after obtaining a subset from length 1 to length of pattern – 1, I made an association rule by calculating support and confidence value. (line 105 ~ line 115)

텍스트이(가) 표시된 사진

자동 생성된 설명

* If a subset of a certain pattern is not included in the existing frequent pattern, it is excluded from the candidate. (line 72 ~ line 90)
* The above picture is the main function. As explained in the compilation method, the first argument is minimum support, the second argument is the name of input file, and the third argument is name of output file. The input file was read line by line to create a list containing transactions (line 123 ~ line 128), and the association rules created as a result of the apriori algorithm were written to the output file. (line 129 ~ line 132)

1. **Testing result**

텍스트, 점수판, 영수증이(가) 표시된 사진

자동 생성된 설명

* The above pictures are part of input file and output file captured. The row in the input file means one transaction, and the row in the output file means one association rule.
* As stated in the task specification, there is no duplication of items in each transaction.
* The order of association rules in the output file is random.
* The value of support and confidence are rounded to two decimal places.