Data Science

Assignment #1

|  |  |
| --- | --- |
| Course name | Data science |
| Assignment | Apriori Algorithm |
| Major | Computer |
| Student ID | 2011004028 |
| Name | Seonghun Lee |

1. Summary of my algorithm

There are two parts of my algorithm. First, make frequent item-sets and then using this frequent item sets, find association rules.

1. Make Frequent Item Set

In Apriori Algorithm, using self-join frequent k item set, I can make k+1 candidate item set. And check the support of the k+1 candidate item set then make frequent k+1 item set.

1. Find association rules

It is really simple if you have all frequent item set. Find all combination of frequent item set, and calculate how many time they have appeared in the transactions and store its support and confidence.

1. Detailed description of my codes (for each functions)

|  |
| --- |
| Def check\_minsup |
| * this function checks minimum support of candidate and return frequent set and its support |
| Def eval\_sup |
| * this function evaluates support of candidate and return the number of frequent set |
| Def selfjoin |
| * this function self-join of Frequent Set k and make Candidate Set k+1 |
| Find all Frequent item sets |
| * the while sentence above is the important part of this program to find all Frequent Set |
| Find all combinations of the frequent set |
| * find all possible association rules in every frequent set |
| Write to the output file |
| Write to the output file adjusting to the format |

1. Instructions for compiling my codes at TA’s computer (e.g. screenshot)

|  |
| --- |
| 1. Put the Apriori.py and input.txt files into same folder |
|  |
| 1. Execute command prompt and go to this folder |
|  |
| 1. Execute Apriori.py with command line arguments |
| * The format is Apriori.py, minimum support, input file name, output file name |
| 1. Check the output file in your folder |
|  |

1. Any other specification of my implementations and testing

|  |
| --- |
| Test result |
| * In minimum support 4 and 5 the result is same with test data. |