**Programming Assignment #1 Report**

Computer Science and Engineering

2010003970

Namwoo Kim

1. **Summary of algorithm**

This program implemented basic apriori algorithm using python. First, iterating all transactions, make a frequency table for each length-1 itemset and count all transaction records to calculate support values (Percent of occurrence in transactions).

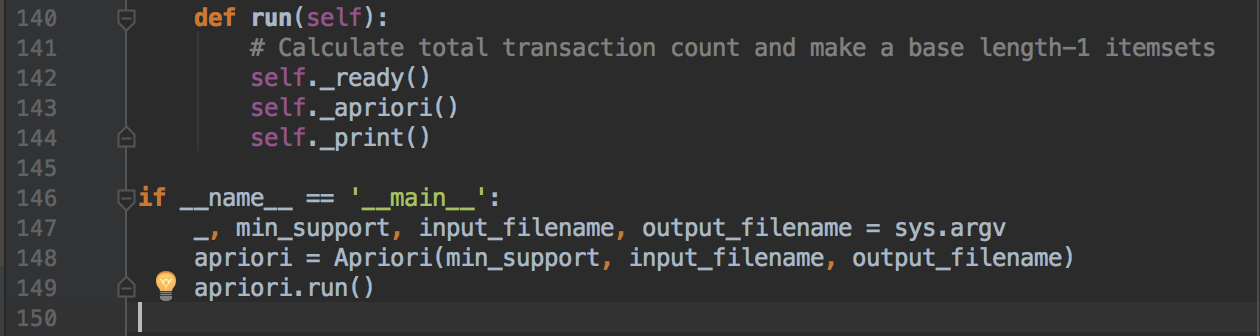
Then, generate candidates of incremented length (2 at this point) itemsets from frequency table which has support value over minimum threshold.

Again, scan all transactions and check if transaction is superset of one of candidates. If so, add candidates into frequency table to calculate support. Also add candidate into frequency table which itemset is subset of candidate (To print out association later)

Repeat those steps until there are no candidates. Then calculate support and confidence to print out.

1. **Detailed description of my codes**

**<Entry point>**



This is entry point of my code. getting parameters from console and build a class to run apriori. The ‘run’ function consisted of three simple functions.

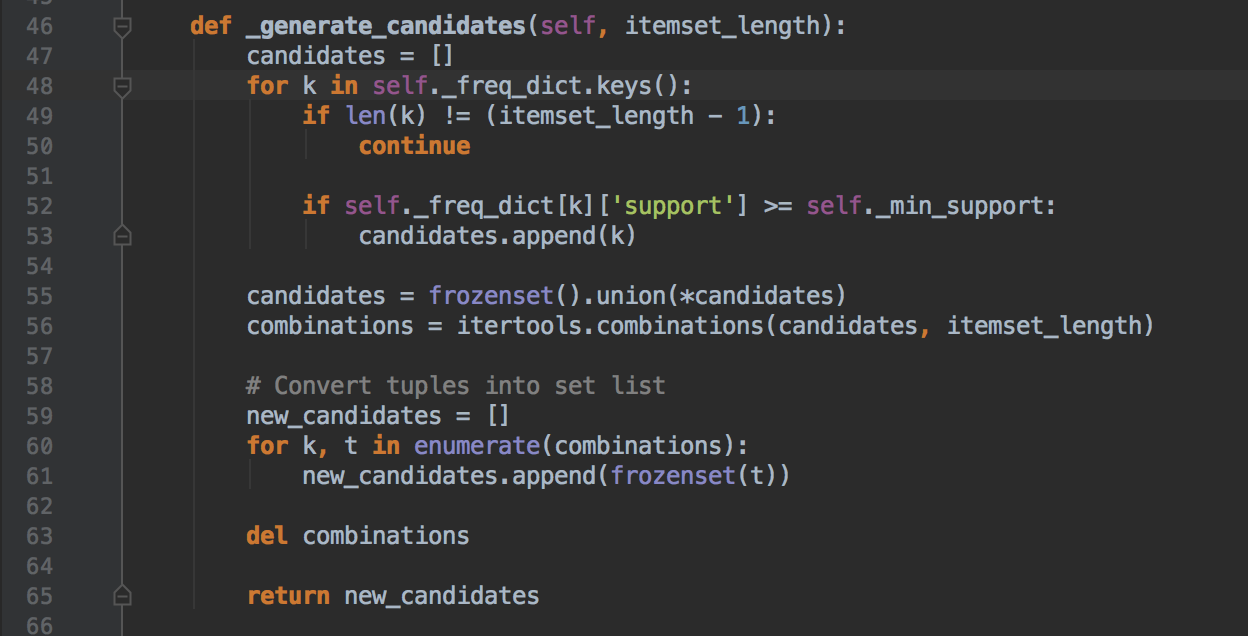
**<self.\_ready() function>**



As I mentioned before, \_ready function is initial process to run apriori. First, read from “input\_filename” to iterate all transactions. (line 20-22) Second, looping every lines and convert each line into integer type of set. (line 25) Third, iterating item in transactions and check whether item is in \_freq\_dict (which is a frequency table) (line 27-29) Fourth, Increase frequency if there’s an item in \_freq\_dict. If not, make a bare data type to increase frequency for further use. (line 30-42)

Last, save total transaction count. (line 44)

**<self.\_generate\_candidates() function>**



This function generates candidates with given itemset length. This function finds all itemsets which length is just subtracted one from requested. (we need length-1 itemsets to generate length-2 itemsets) Then, if some itemsets satisfies minimum support, it will be a candidate for candidates. Finally, use a function to generate combinations with given candidates. Then we can get itemsets which length is same as requested.

**<self.\_apriori() function>**

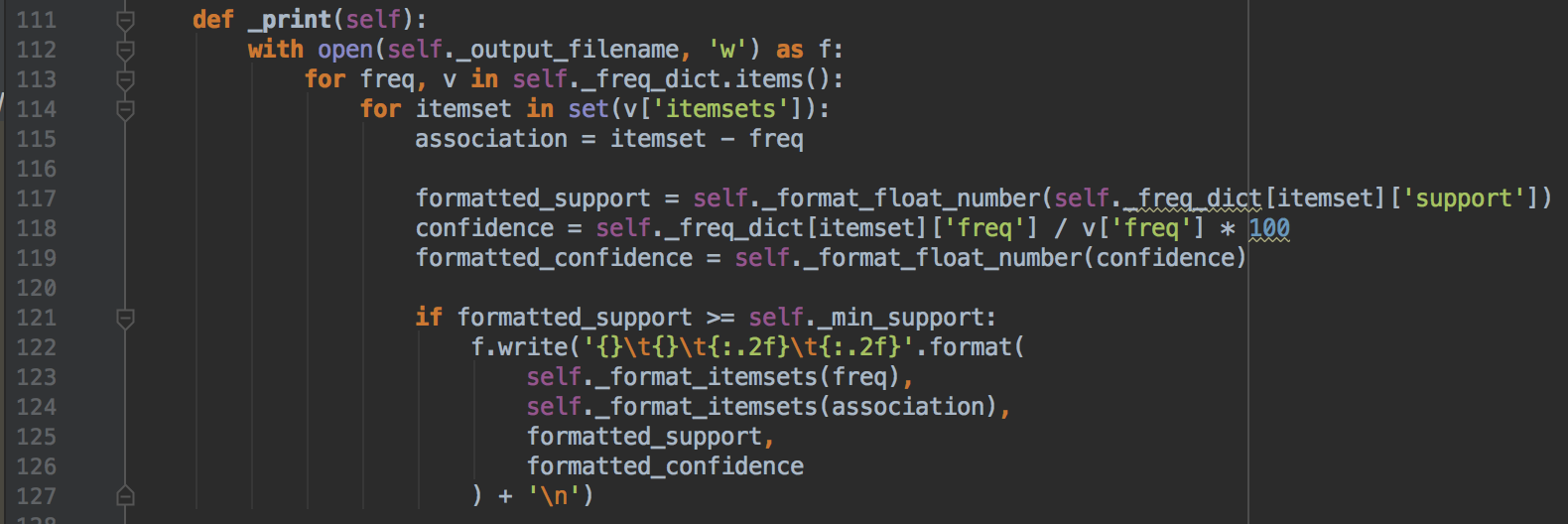


This function is also same as in self.\_ready() function. There are two most different things in this code.

First, in line 87, we can simply check candidate whether this is subset of transaction or not. Then increase frequency If candidate is subset of current transaction.

Second, retrieve all itemsets in the frequency table and add a current transaction as a superset of candidates. This is quite important to show associations later.

**<self.\_print() function>**

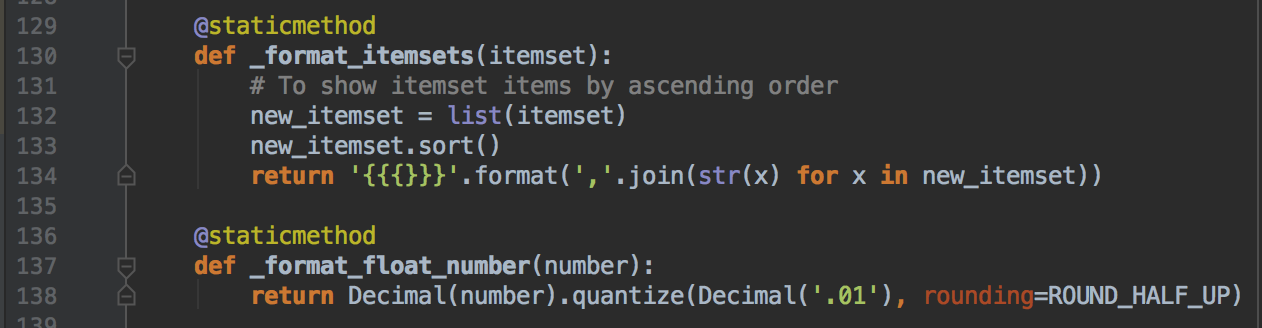


This function generates support and confidence and print itemsets.

In line 115, itemset – freq is a simple feature can calculate this form {0, 1, 2, 3} – {1, 2} = {0, 3}. Since I simply stored superset of current itemset in “itemset” so I use this “–“ command to show itemsets which is not include current itemset items named “freq”.

To generate confidence, we can calculate by dividing current itemset’s frequency by superset itemset’s frequency. (line 118)

**<Remain simple static functions>**



\_format\_itemsets function does sorting items in each itemset.

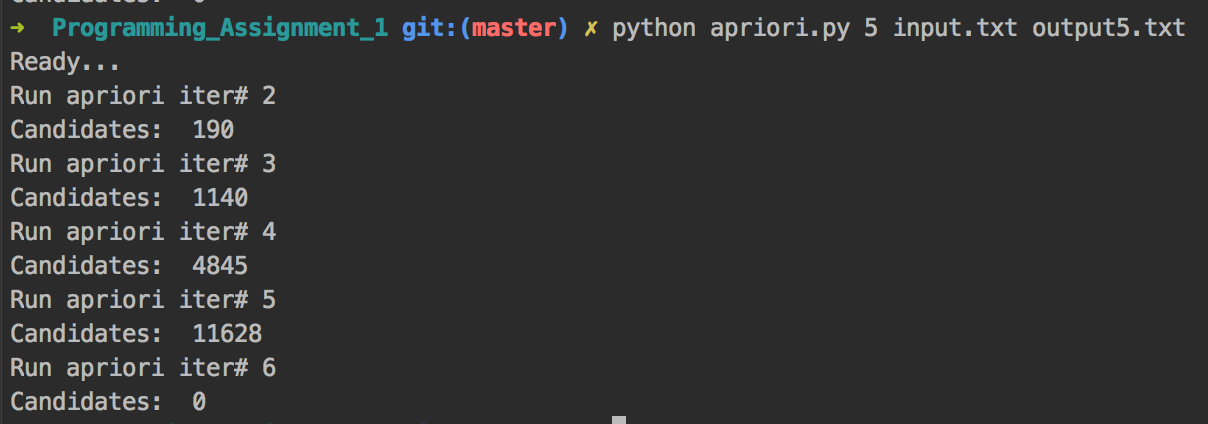
\_format\_float\_number function does quantize numbers to calculate numbers properly and make numbers as same as given sample output. (there was a simple difference between my output and TA’s output when using round() functions)

1. **Instructions for compiling my source code at TA’s computer**

This program tested in Python 2.7 and 3.6. You can easily run this program in any other environments which has python because python program doesn’t need to compile for run.

In mac OS, there is a preinstalled python.



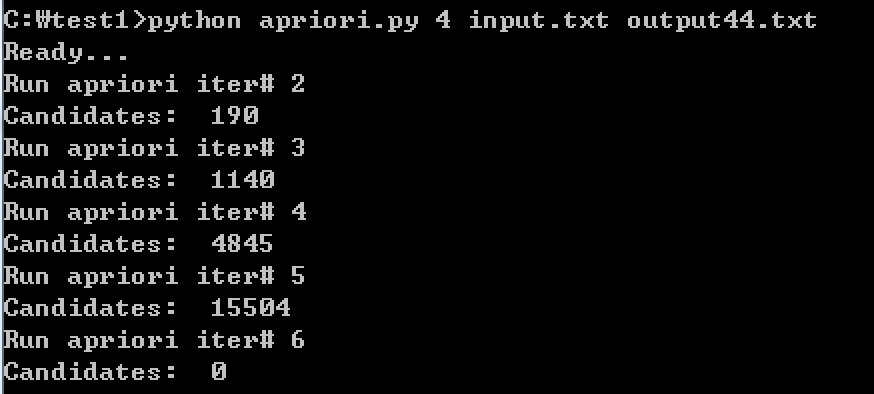


OK for both ways.

In windows, Install python 2.7 first and then execute this in your shell.

setx PATH "C:\Python27;C:\Python27\Scripts;C:\Python27\Lib\site-packages"

This is the most important part. Adding a PATH variable in the shell and quit and re-open terminal. If you enter python in terminal and It works, you can run my program.



1. **Any other specification of my implementation and testing**

I’ve tested both mac OS and Windows environments and successfully tested with the program that TA given to me. I think there’s nothing to mention for any specifications.