BIG DATA ANALYTICS

LAB 5

Aim: - Implement custom map reduce program.

Definition:- Find Association Rules using Apriori Algorithm with help of Map-Reduce.(Market Basket Analysis).

What is Market Basket Analysis?

* It is one of the key technique used by large retailer to uncover association b/w items.
* Benefit : to increase revenue.

What is Association Rule Mining?

* Association rule learning is a rule-based machine learning method for discovering interesting relations between variables in large databases. It is intended to identify strong rules discovered in databases using some measures of interestingness. [Wikipedia]
* e.g. If item A is bought by customer c1 then chances of being bought item B by same customer on same transaction.
* A == > B, if A then B; A = antecedent and B = consequent.
* If we have huge dataset(transaction) then we can get so many combination of type

A => B , A&B => C, A&B&C => D and so on so to find promising rules we need some metrices.

Those metrices are follows:

1. Support :- support is frequency of item or items combination.

supp(X)=Number of transaction in which X appears / Total number of transactions

1. Confidence: - How often items collection B occur given occurrence of items collection A.

conf(X⟶Y)=supp(X∪Y) / supp(X)

1. Lift:- it’s Strength of any rule.

lift(X⟶Y)=supp(X∪Y) / supp(X)∗supp(Y)

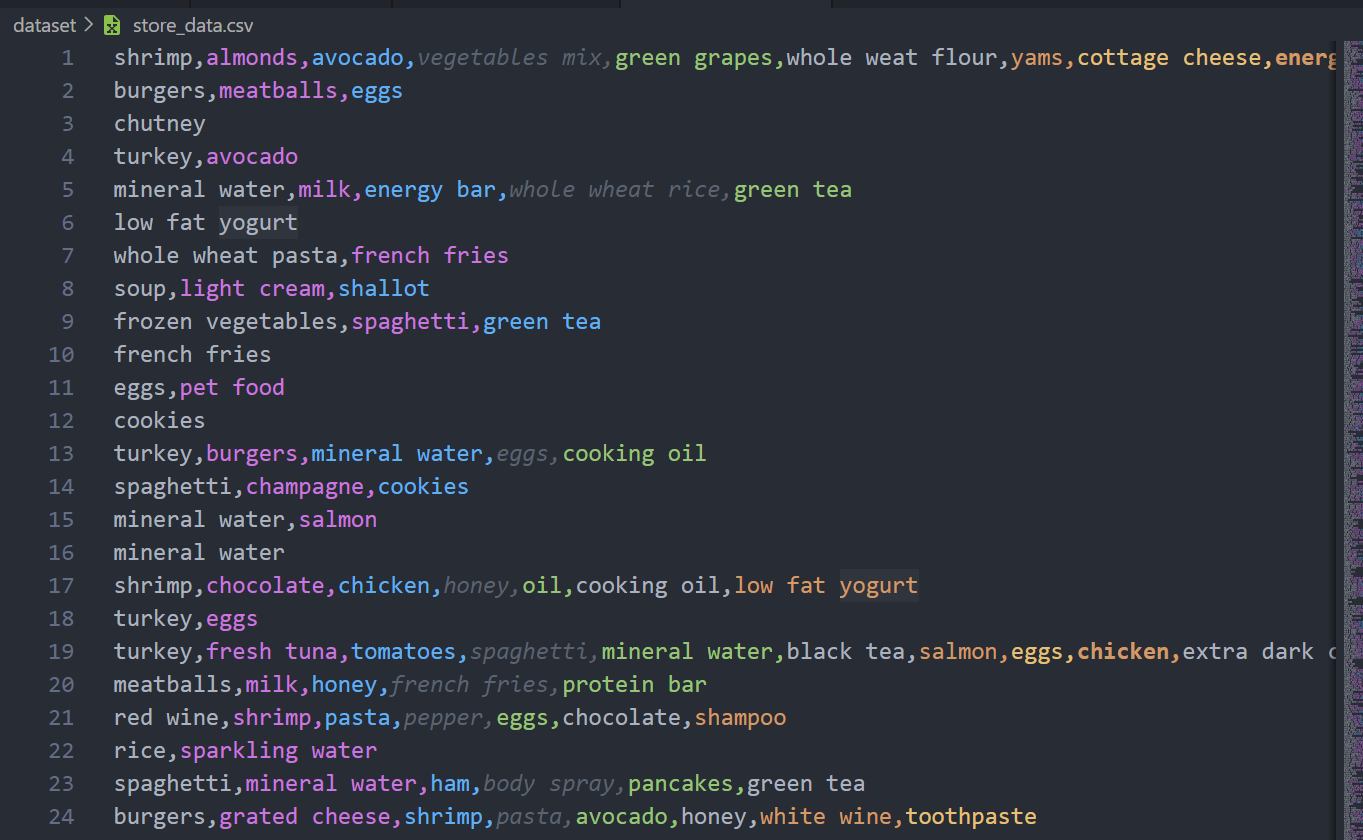
Where Map Reduce come into Picture?

* We can use map reduce to find frequency of itemset(combination of items)(itemset can also have single item).

Dataset:

Each line in dataset is representing single transaction(bill) items.

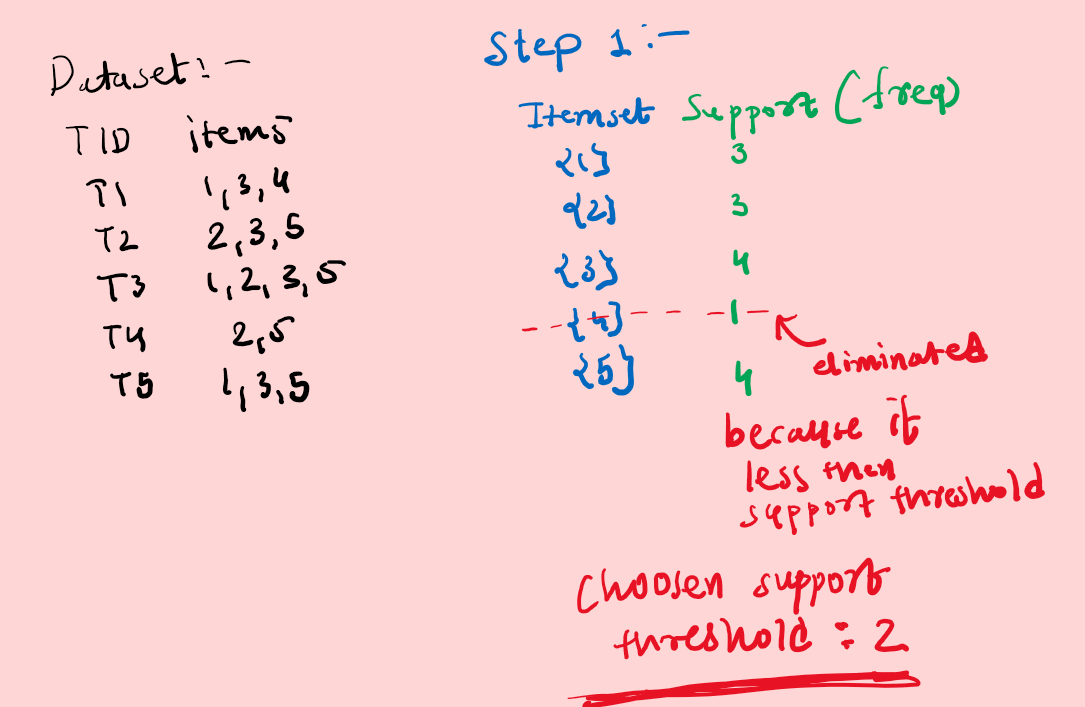
It has total 7501 transactions.

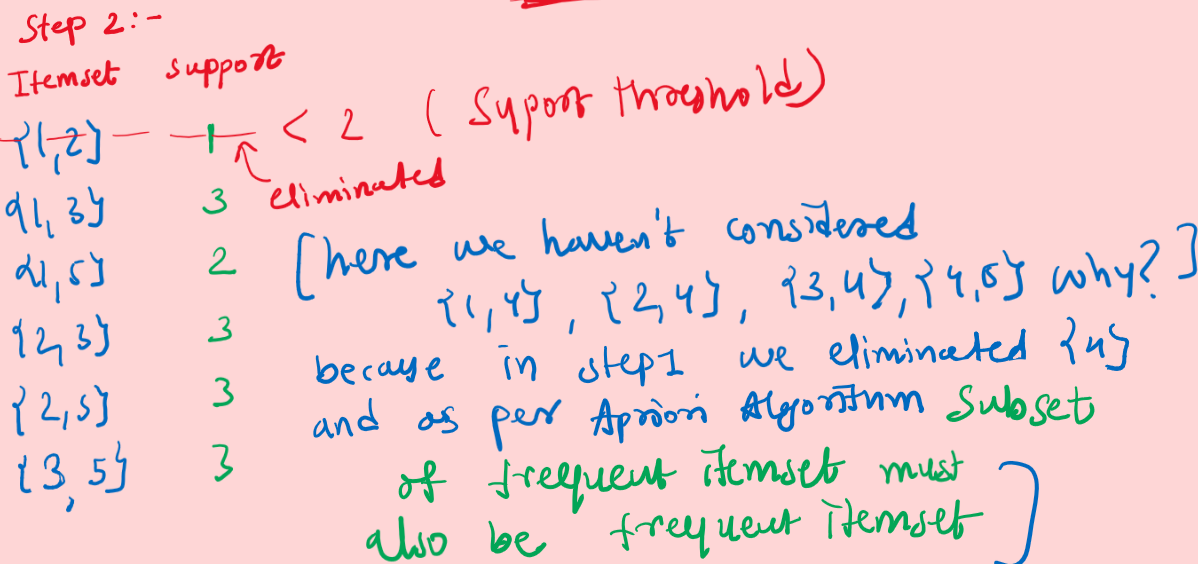


(Dataset taken from: <https://github.com/gitganeshnethi/Datasets/blob/master/store_data.csv>).

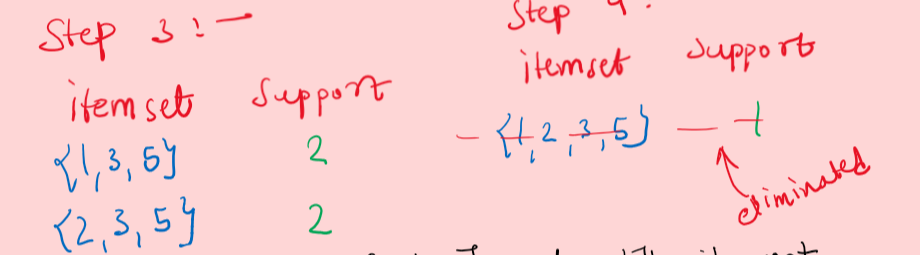
### Apriori Algorithm ###

* Apriori algorithm is a classical algorithm in data mining. It is used for mining frequent itemsets and relevant association rules. It is devised to operate on a database containing a lot of transactions, for instance, items brought by customers in a store.
* It uses frequent item sets to generate association rules. It is based on the concept that **subset of frequent itemset must also be frequent itemset**.
* Consider below example:





* Observation: - we need to use previous step map reduce o/p to generate itemset of current step.



* We have to find itemset till it not become empty. in our case it became empty when itemset has size four. We have to consider previous itemset to find rules.
* So we have to find rules using {1,3,5} and {2,3,5}

For I = {1,3,5} subsets are

S = {1,3},{1,5},{3,5},{1},{3},{5} ( empty set and I itself is not considered)

If( Support(I)/Support(S) >= min\_confidence\_val)

S -> (I-S) means S recommends I-S

e.g. {1,3} 🡪 ({1,3,5} – {1,3}) means 1 & 3 🡪 5

support({1,3,5})/support({1,3}) = 2/3 = 66.66% > 60% so rule is valid.

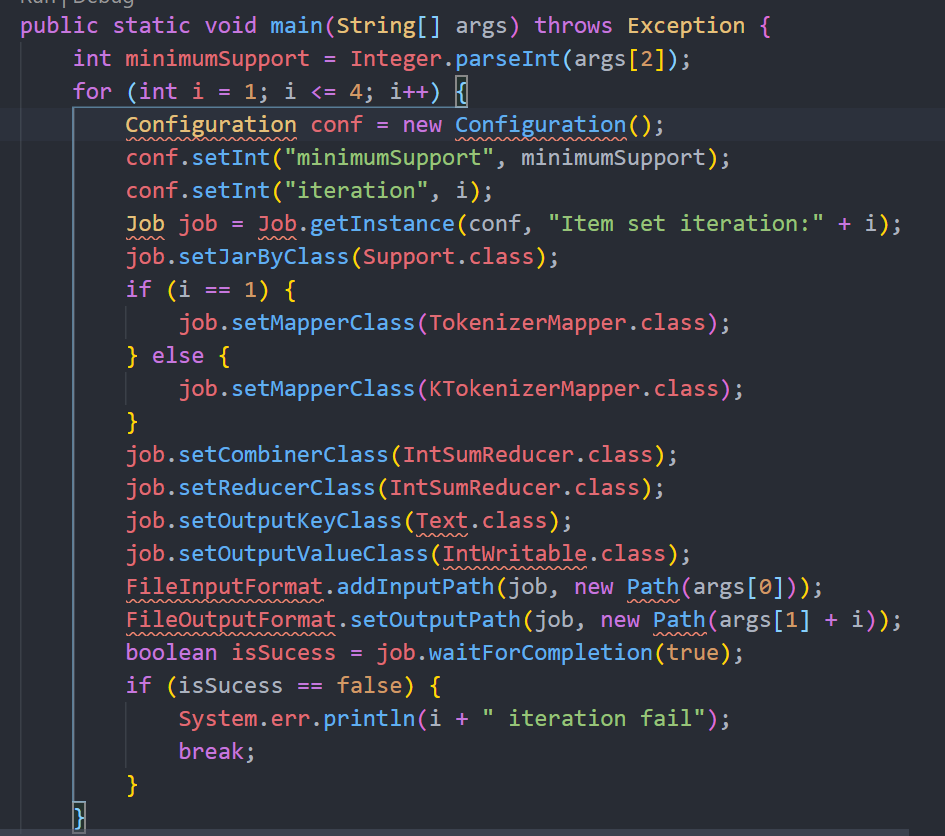
{3} 🡪 (1,3,5} – {3}) means 3 🡪 1 & 5

support({1,3,5})/support({3}) = 2/4 = 50% < 60% so rule is rejected.

* We have to choose value of min\_conf\_value if we choose higher value then we find less rules with more confidence, If we choose small value we find so many rule but with less confidence.

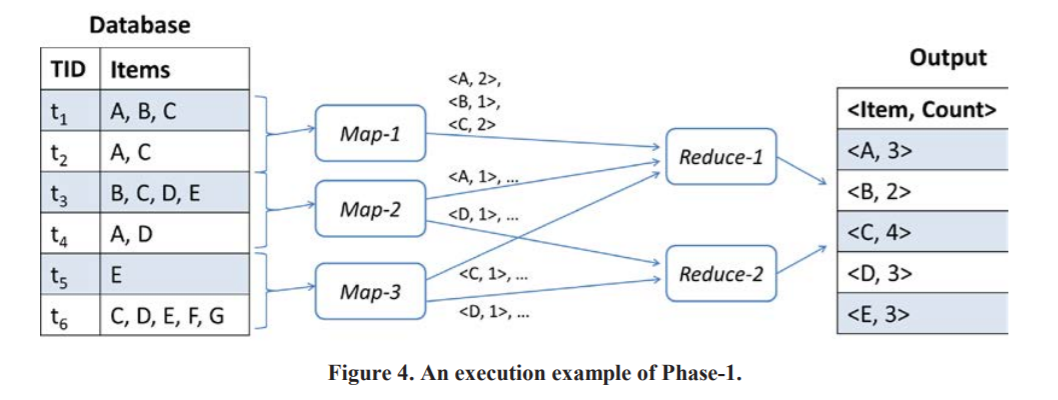
Code and explanation:

* We have taken 4 map reduce task because by trial error I found out in iteration four itemset become empty for given support threshold(minimum Support).
* minimum support is passed as command line argument, to use in mapper and reducer class it set in conf.
* if i==1 TokenizerMapper else KTokenizerMapper
* because TokenizerMapper we considering single item(it’s logic is same as wordcount mapper).
* for KTokenizerMapper we have to consider subset of size i.

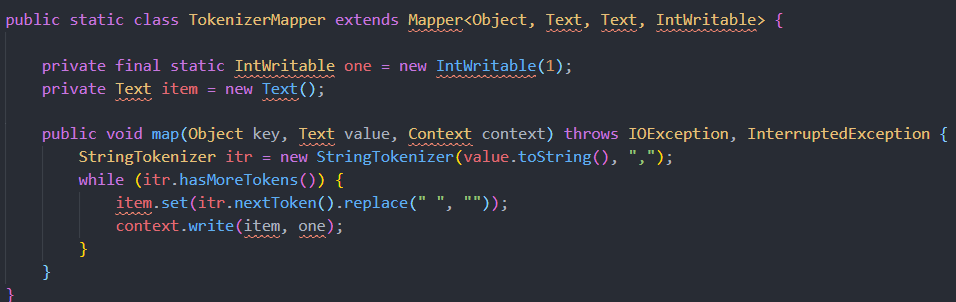




Iteration 1 Mapper:

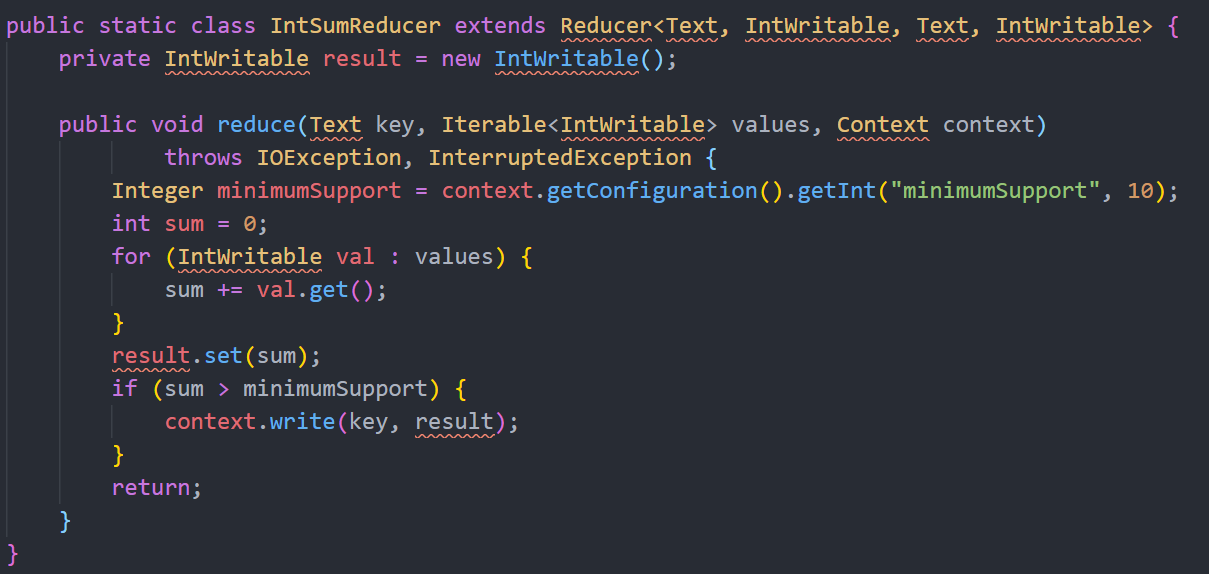


* key input: byte offset
* value input: line from our dataset.
* Key output: word
* value output: 1
* for each line splited using “,” so we get all words.



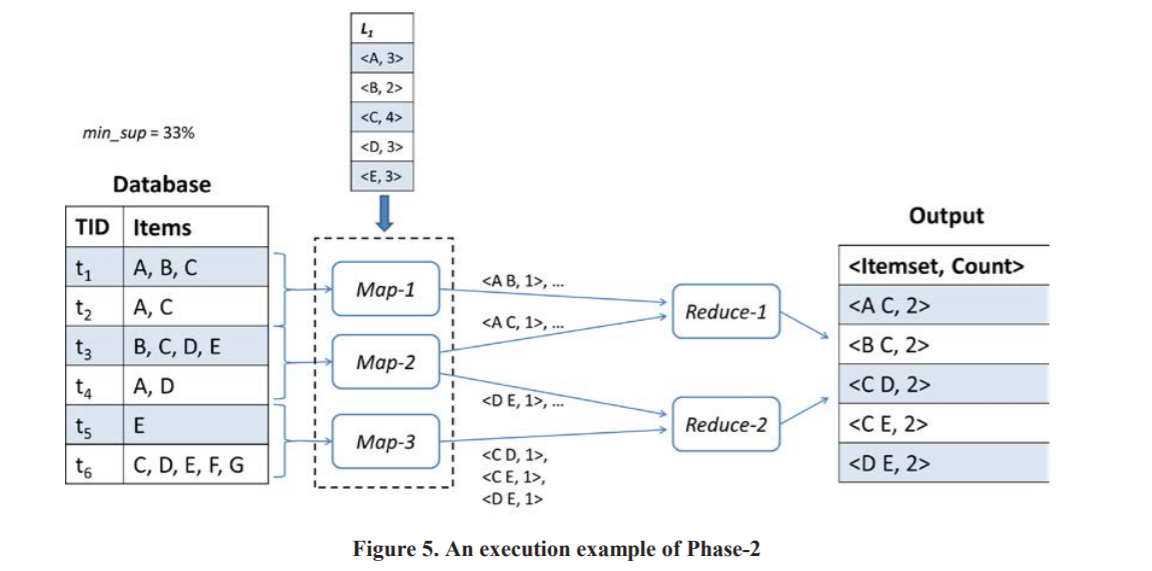
Reducer of Both Mapper:

* output key = output key of map phase
* output value = total occurrence of item
* same logic as wordcount reducer.
* Just before writing checking for it’s support(frequency) is greater than minimum support.

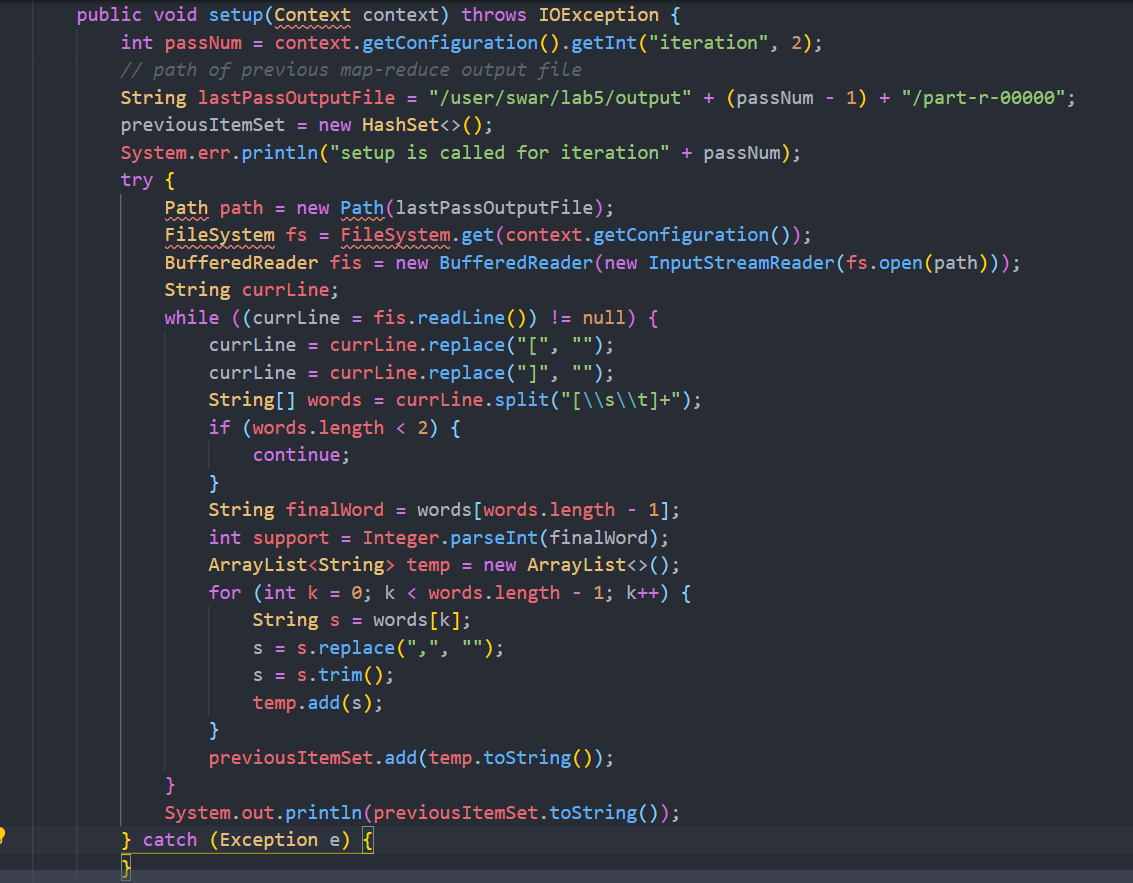




Mapper for rest of iteration:



* Setup phase:
* We use this phase to read o/p of previous phase and put it into hashset and use that for finding itemset of current step.





Map function:

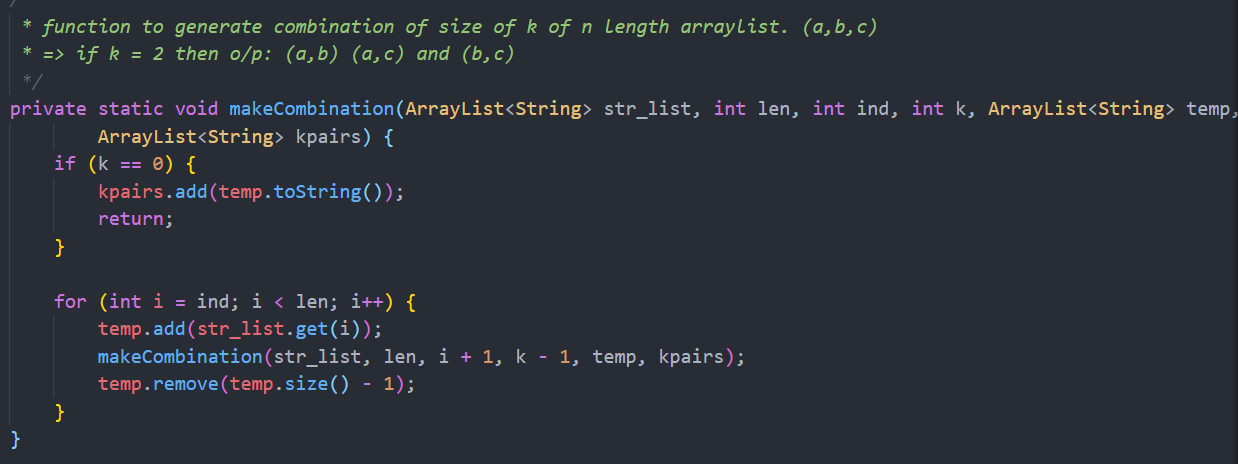
* Here str\_list contains words of given line.
* Sorting of that word require so that (milk,bread) and (bread,milk) never consider different.

due to sorting always we get set in which items are in lexicographic order.

* Pair\_size has value about in which iteration we are currently in according that we have to generate subsets.
* makeCombination function generates subset of size pair\_size.
* isValidItemSet is used for validating itemset from previous mapper o/p.







* isValidItemSet function is used for validating itemset using previous map reduce phase

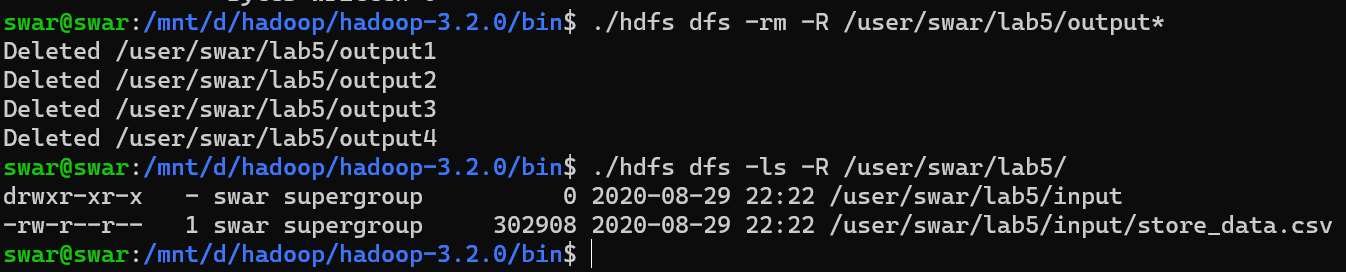
why we do this step?  
because it reduces number of itemset so less number of itemset is passed to reducer so less data need to traverse from network.

isValidateItemSet has logic of apriori algorithm: subset of frequent itemset must also be frequent itemset.

If we don’t do this step still it works fine because in reducer it going to filter out.

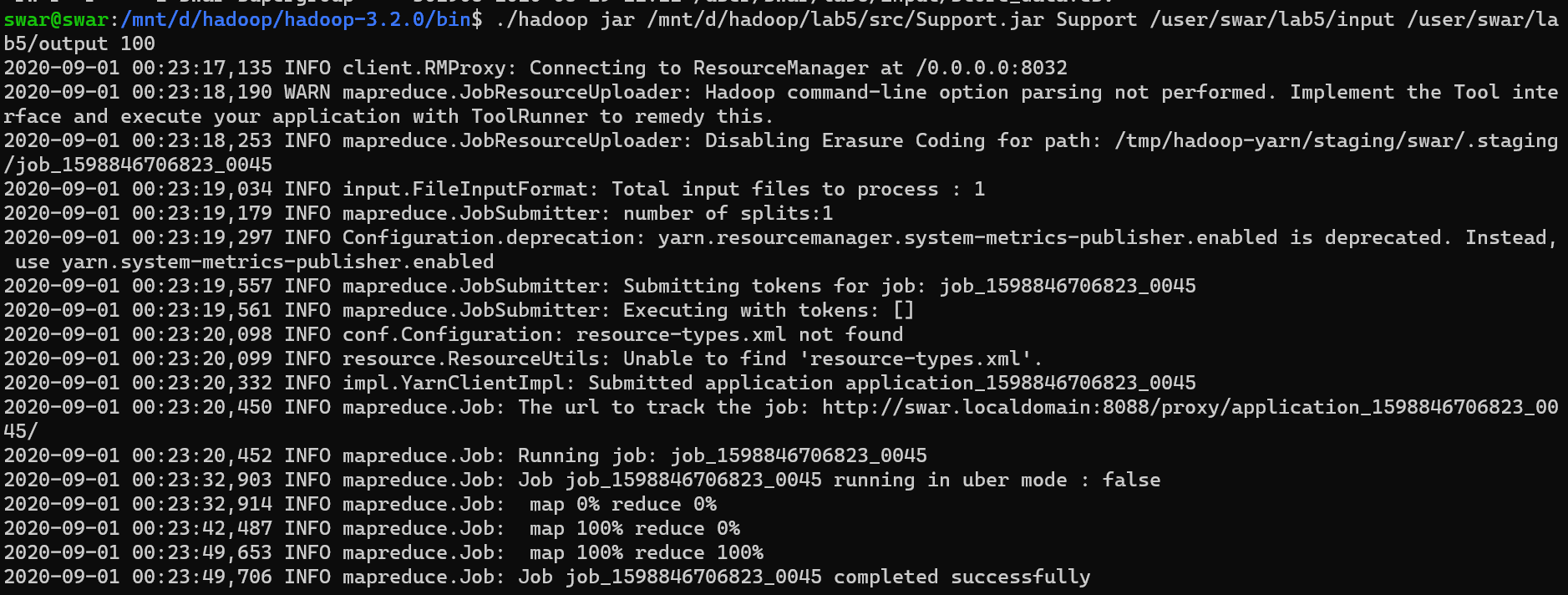


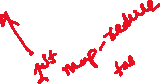
* current state of hdfs.

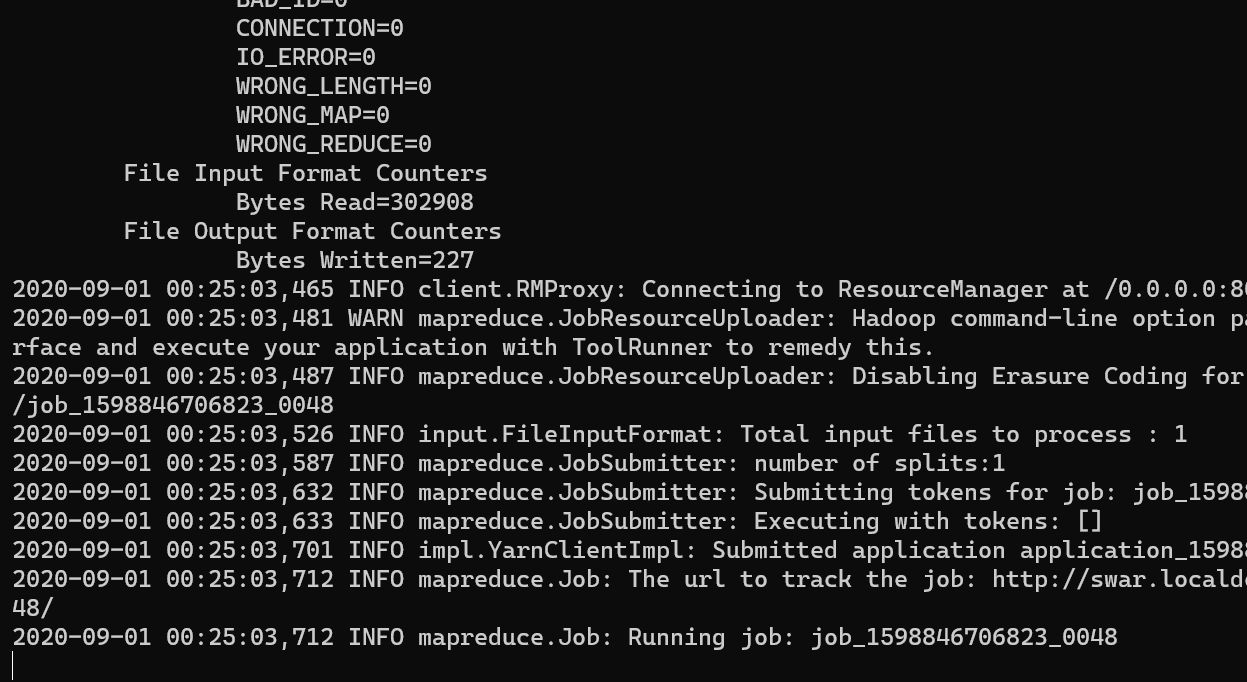


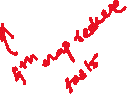
* running map reduce task.



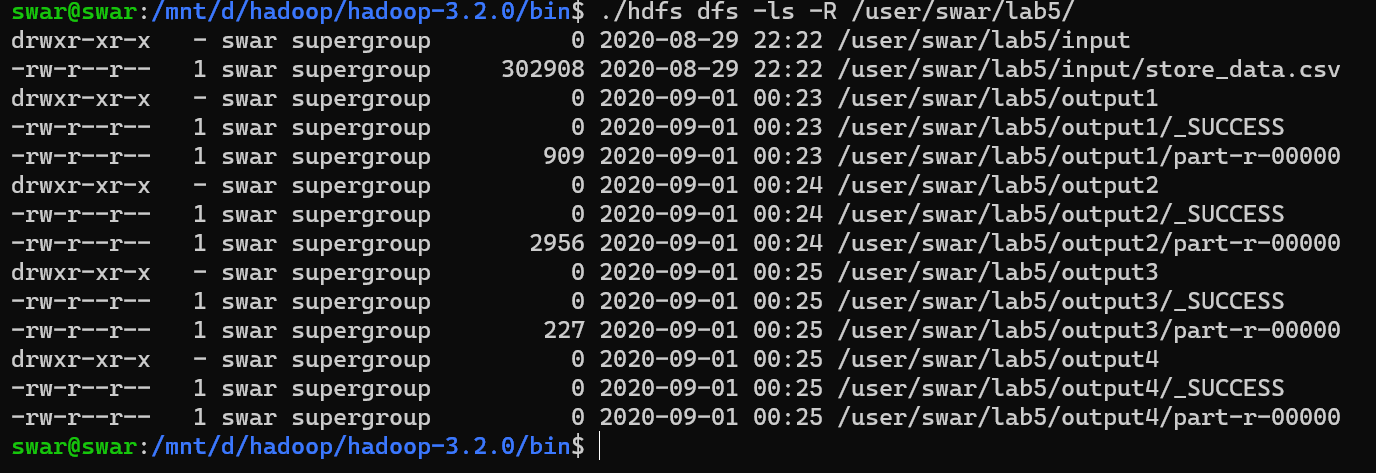








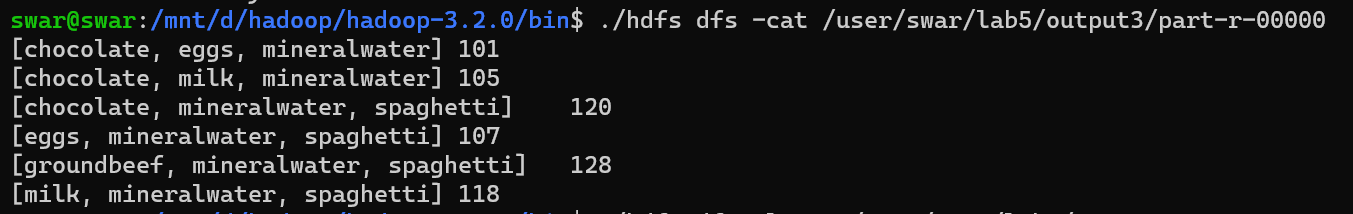
* After Running.



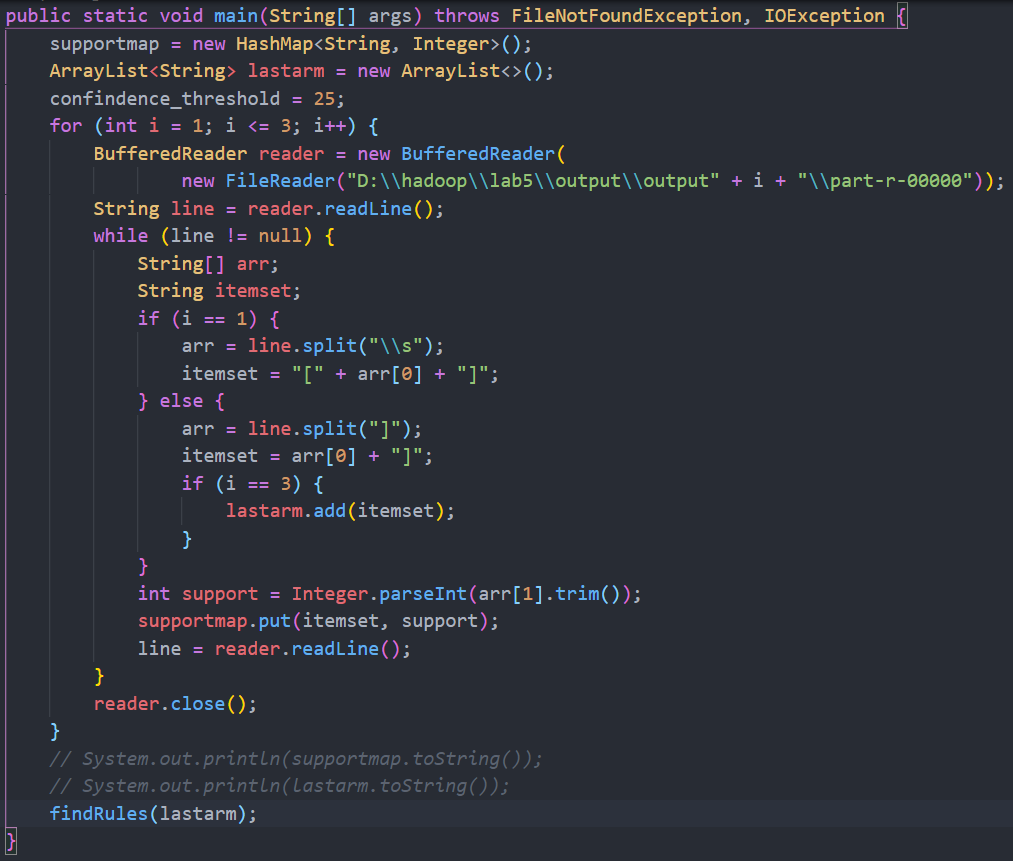






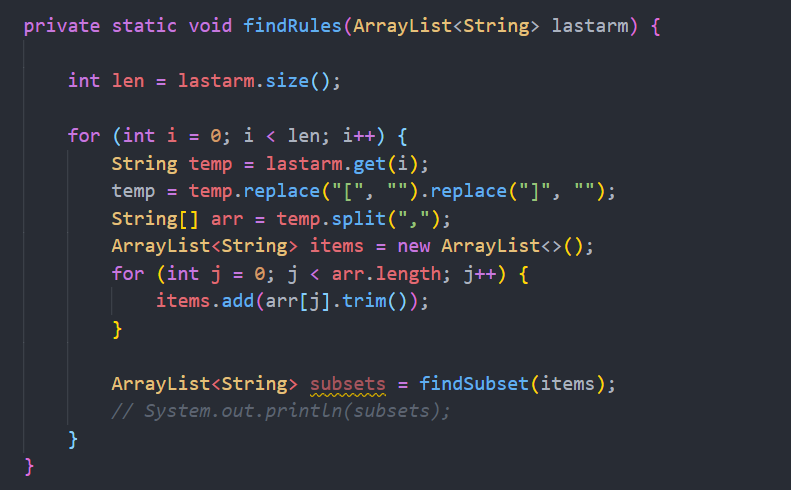


* Now finding confidence using for rules:
* So for that I transferred this output file from hdfs to local file sytem.
* Read those file and put it into hashmap. Key is itemset and value is support.





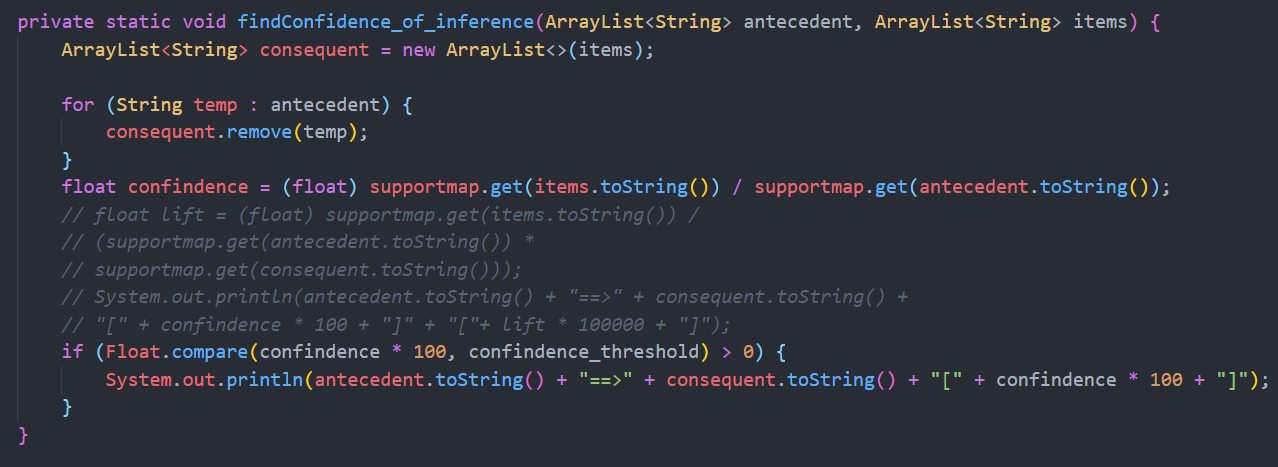
* Last arm has itemset of 3rd iteration using which we are creating rules.



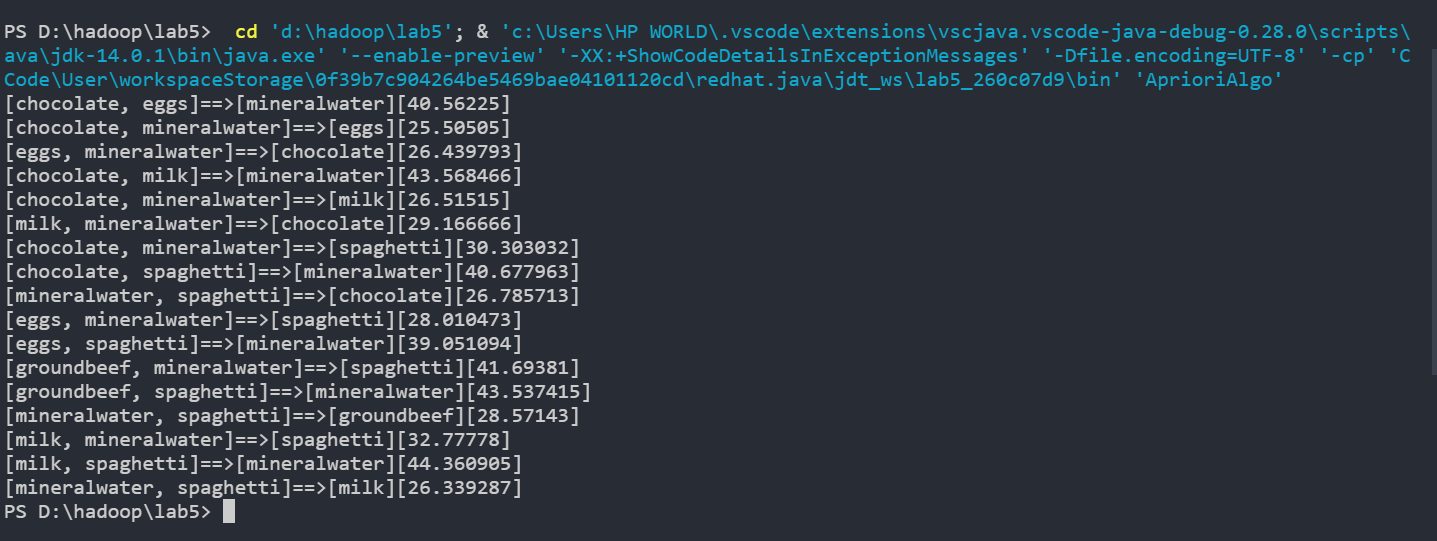


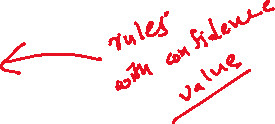












Advantages of Apriori algorithm:-

1. Easy to implement

2. Use large itemset property

Disadvantages of Apriori algorithm

1. Requires many database scans

2. Very slow

References:

* <https://dl.acm.org/doi/pdf/10.1145/2184751.2184842?casa_token=8kk1kzUT4TAAAAAA:S4rtH1FUiHCLjVMnK_a9li1m3Hjq-TabgVa43NsQ6ziX8sTCEzrwf3uDqax4KwgdFilE08qsALqSAEQ>
* <https://www.youtube.com/watch?v=guVvtZ7ZClw>