To run the program-

bin/spark-submit Ankitha_Radhakrishna_SON.py <caseNo> <CSV_FILE> <support_threshold>

Spark version- 2.2.1

Methodology-

- Apriory algorithm is used to find the frequent items.
- Phase1 of SON-
    - Data chunks are processed by map partitions.
    - Pass1 Apriory : Singletons are created by doing a flatMap on the RDD that contains the file data. A collection counter keeps a count of the single items. The counts are compared with support threshold and those that equal or exceed support threshold are considered frequent.
    - Threshold is for each partition is calculated as-
      Adjusted_threshold = (number of baskets in the data chuck)/(total number of baskets in the dataset) * support_threshold_provided
    - Pass2 Apriory : frequent itemsets of all sizes are created on map partitions.
    - Candidate item sets of size k are created by looping over frequent itemsets of size k-1.
    - The above list is filtered to match the adjusted support threshold.
    - Map1 = > Output => (F,1) where F is a frequent itemset
    - Reduce1 outputs all frequent itemsets
- Phase2 SON-
    - Map2 –
        - Input – (all candidate items, a chunk of baskets)
        - Output(C,v) where C is the frequent candidate itemset, v= count of the itemset in the baskets in that chunk

      Reduce2-

        - Groups candidate itemsets across chunks and does a summation of each of the counts. If count >= support_threshold, that itemset is a part of the output.

ml-latest-small dataset

CASE 1		CASE 2	
120	53 sec	180	650 sec
150	9.5 sec	200	480 sec

ml-20m dataset

CASE 1		CASE 2	
30000	700 sec	2800	460 sec
35000	430 sec	3000	370 sec