Description

Both the codes are implementing SON Algorithm using Apache Spark Framework. The goal is to find all the possible combinations of the frequent item sets in any given input file within the required time.

FrequentBusinessUserSets.py:

Here there are two cases:

1. Calculating the combinations of frequent businesses (as singletons, pairs, triples, etc.) that are qualified as frequent given a support threshold.
2. Calculating the combinations of frequent users (as singletons, pairs, triples, etc.) that are qualified as frequent given a support threshold

Input format:

1. Case number: Integer that specifies the case.
2. Support: Integer that defines the minimum count to qualify as a frequent itemset.
3. Input file path: has information on user, business, ratings, etc.
4. Output file path

Output:

1. Duration of execution
2. The frequent item set candidates and the final set of frequent item sets.

```
Candidates:
('100'),('101'),('102'),('103'),('105'),('97'),('98'),('99')

('100', '101'),('100', '98'),('100', '99'),('101', '102'),('101', '97'),('101', '98'),('101', '99'),('10

('100', '101', '98'),('100', '101', '99'),('101', '97', '99'),('102', '103', '105'),('102', '103', '98')

('102', '103', '105', '98'),('102', '103', '105', '99'),('102', '103', '98', '99'),('102', '105', '98',

('102', '103', '105', '98', '99')

Frequent Itemsets:
('100'),('101'),('102'),('103'),('97'),('98'),('99')

('100', '101'),('100', '98'),('101', '102'),('101', '97'),('101', '98'),('101', '99'),('102', '103'),('1

('100', '101', '98'),('101', '97', '99'),('102', '103', '99'),('97', '98', '99')
```

FrequentProductSets.py:

Here the goal is:

1. Doing data preprocessing to make user ID concatenated with the purchase date as the key.
2. Appling the SON algorithm to find the frequent item sets of product IDs from the point of view of the users buying the products considering only those customers with purchases greater than a specified filter threshold.

Input format:

1. Filter threshold
2. Support: Integer that defines the minimum count to qualify as a frequent itemset.
3. Input file path: having user ID, purchase date, product IDs, etc.
4. Output file path

Output:

3. Duration of execution
4. The frequent item set candidates and the final set of frequent item sets.