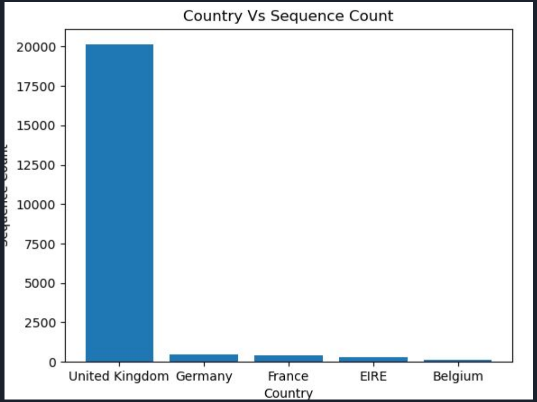
Implementation of SPADE Algorithm for Online Retail Dataset in Python

By
KAILASH KOLLURU
2016AAPS0210H

COUNTRY v/s Sequence Count

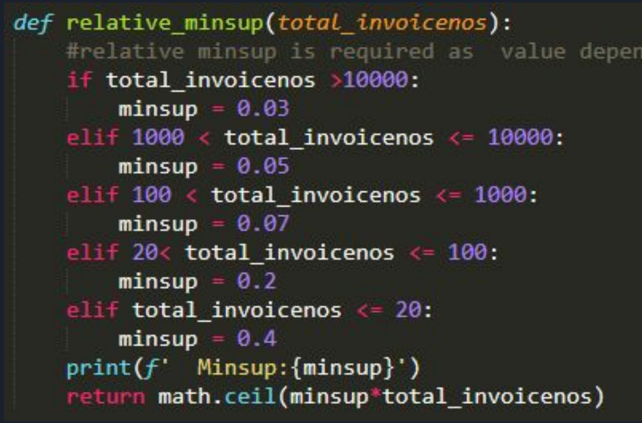
- Data has different Countries but not of same data.
- To prevent the biasing of one country we divided the data to country bucket and mine for Frequent Sequence Patterns.



Relative Minsup

For a data set ,

- if its large, then the Minsup will always be low
- If its small, then Minsup will be large



```python
def relative_minsup(total_invoicenos):
    #relative minsup is required as  value depen
    if total_invoicenos >10000:
        minsup = 0.03
    elif 1000 < total_invoicenos <= 10000:
        minsup = 0.05
    elif 100 < total_invoicenos <= 1000:
        minsup = 0.07
    elif 20< total_invoicenos <= 100:
        minsup = 0.2
    elif total_invoicenos <= 20:
        minsup = 0.4
    print(f'  Minsup:{minsup}')
    return math.ceil(minsup*total_invoicenos)
```

So, We choose minsup on basis on number of sequence in an country

As Minsup varies from country to country, Support Threshold will also be different .

Thus , this **Improves the performance** in finding the patterns specific to a country.

OverView on Implementation

- We Identify all the 1-item sequence, whose count of occurrence >= Support Threshold . Then using Temporal Join we make 2-item sequence.
- We make a Temporal Join for 2 sequences where
  - Both Items must be on same sequence
  - Higher EventID is given as EventID for that join 2-sequence.
  - If EventId are same, last item of one of sequence will be added to other

Later we call Enumerate frequent function which is a recursive function to form (n+1) sequence from (n)th sequence

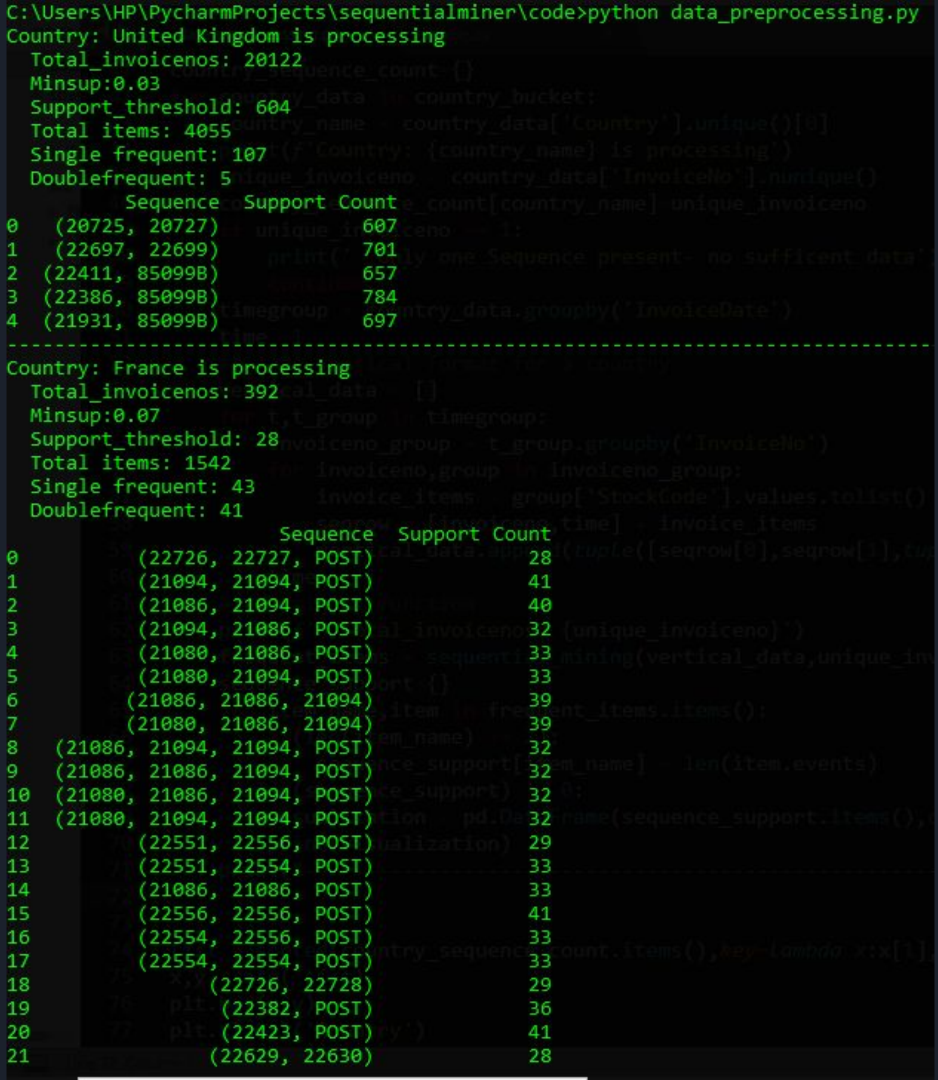
Final Result

- In Terminal, on Running
    - python data_preprocessing.py

We get result as shown.

This shows the corresponding

- Minsup and Support Threshold
- Single & Double frequent Items
- List of 2 or more Frequent Patterns. Along with support count.



```
C:\Users\HP\PycharmProjects\sequentialminer\code>python data_preprocessing.py
Country: United Kingdom is processing
  Total_invoicenos: 20122
  Minsup:0.03
  Support_threshold: 604
  Total items: 4055
  Single frequent: 107
  Doublefrequent: 5
          Sequence  Support Count
0   (20725, 20727)            607
1   (22697, 22699)            701
2  (22411, 85099B)            657
3  (22386, 85099B)            784
4  (21931, 85099B)            697
------------------------------------------------------------------------------
Country: France is processing
  Total_invoicenos: 392
  Minsup:0.07
  Support_threshold: 28
  Total items: 1542
  Single frequent: 43
  Doublefrequent: 41
                       Sequence  Support Count
0          (22726, 22727, POST)             28
1          (21094, 21094, POST)             41
2          (21086, 21094, POST)             40
3          (21094, 21086, POST)             32
4          (21080, 21086, POST)             33
5          (21080, 21094, POST)             33
6         (21086, 21086, 21094)             39
7         (21080, 21086, 21094)             39
8   (21086, 21094, 21094, POST)             32
9   (21086, 21086, 21094, POST)             32
10  (21080, 21086, 21094, POST)             32
11  (21080, 21094, 21094, POST)             32
12         (22551, 22556, POST)             29
13         (22551, 22554, POST)             33
14         (21086, 21086, POST)             33
15         (22556, 22556, POST)             41
16         (22554, 22556, POST)             33
17         (22554, 22554, POST)             33
18               (22726, 22728)             29
19                (22382, POST)             36
20                (22423, POST)             41
21               (22629, 22630)             28
```

GITHUB REPOSITORY:

https://github.com/kailashkolluru/SPADE-Alogrithm

THANK YOU