A Report

On

Implementation of

Sequential Pattern Discovery using Equivalence classes (SPADE) algorithm

To mine for Sequential Patterns on

Online Retail

By

KOLLURU KAILASH

2016AAPS0210H

INTRODUCTION:

Sequential Mining is a technique of Data mining in dealing the sequential data. It’s can be interpreted as using Mining theories and methods in finding the most occurring patterns in the given data sequence. This Sequential Mining is mainly done on Time series where values are presumed to be discrete or categorical which can be closely related. And Finding such Correlations or Patterns is the goal in sequential mining. They are many Algorithms out of which some stand globally and scalable like GSP, FreeSpan, PreSpan, MaPress. And one of the recent Algorithm developed algorithm was SPADE (Sequential Pattern Discovery using Equivalence classes).

The best previous Algorithm was found to be not the best after the arrival of SPADE. And also, SPADE was seen to be a highly linear scalable wrt no of input sequences.

SPADE takes only one database scan for analysis when compared with GPS algorithm it takes more than one database scan. SPADE follows a vertical format for ease to analyse unlike GPS. This vertical format is in form as <itemset: sequence\_id, Event\_id>.

DATABASE INFO:

Online Retail Database has 8 columns i.e. InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitPrice, CustomerID, Country. Of which InvoiceNo acts a transaction Id which is unique to each transaction and if there is any credit it indicates cancellation of transaction. StockCode is unique to item. Description is the name of item. So, for analysis Descriptions are redundant. InvoiceDate indicates the date of transaction. UnitPrice shows the price of that corresponding StockCode item.

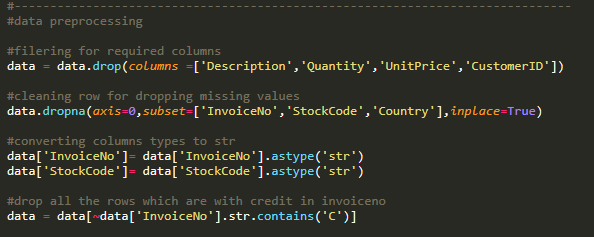
PROCESS & APPROCH:

General process in applying SPADE on a Database. First the database is converted to Vertical format as EventID, SequenceID, ItemSet.

IMPLEMENTATION:

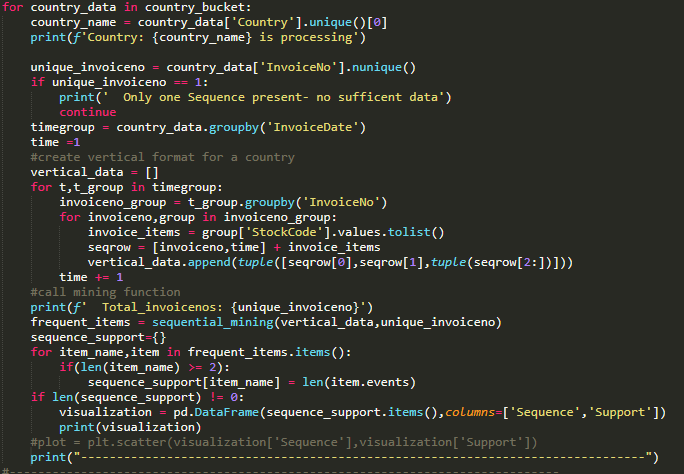
**Step-1**: First starting with DataBase import which is done by pandas’ library. Reading Excel Sheet with pandas.read\_excel and create DataFrame

**Step2**: Then **Data Pre-processing** was done on imported data where the redundant columns and missing values of any row are removed from dataframe. And also, the Transaction where cancelled or credit are removed by InvoiceNo which have ‘C’ in the code.



As the data has different countries and most of the data is of the country ‘United Kingdom’ which makes a bias in finding pattern sequence more favouring to UK. So, we divide the whole data into country buckets and also as the data for every country are relatively may be more or less of data columns. Therefore, used a Relative Minsup for support threshold.

**Step3**: Coverting country buckets to Vertical Format Data by first grouping on InvoiceDate then iterate on each timegroup then groupby InvoiceNo which helps in obtaining in desired format <sequence\_id , event\_id: itemset>. And Iterating over every country data, call mining function.

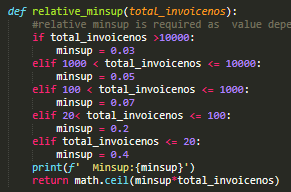


**Step4**: In**Mining Function**, intitally we will make a class object for each item which will have the sequence and a Event tuple of ‘sid’,’eid’. Here SequenceID was InvoiceNo and EventID was based on InvoiceDate.

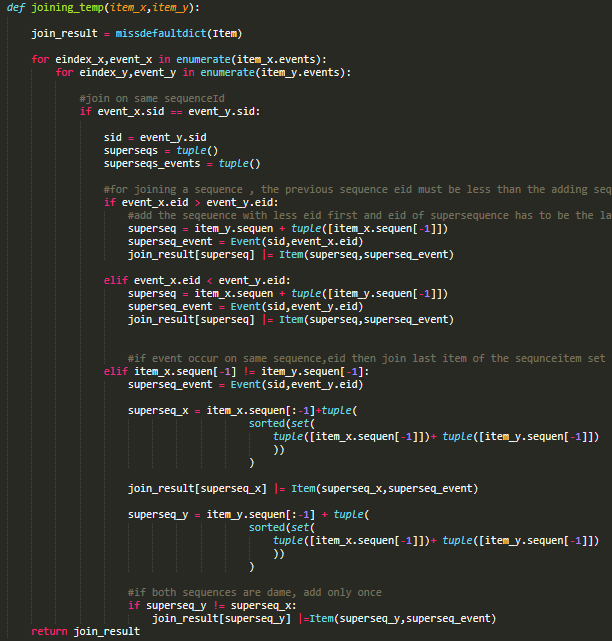


1. We set a **Relative Minsup** because the database has many country but country data can be more or less. So, followed a method

‘For Large dataset, Minsup will be small and For Small dataset, Minsup will be large’.



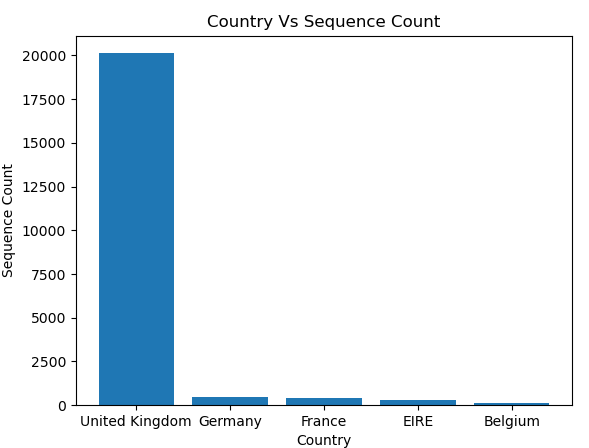
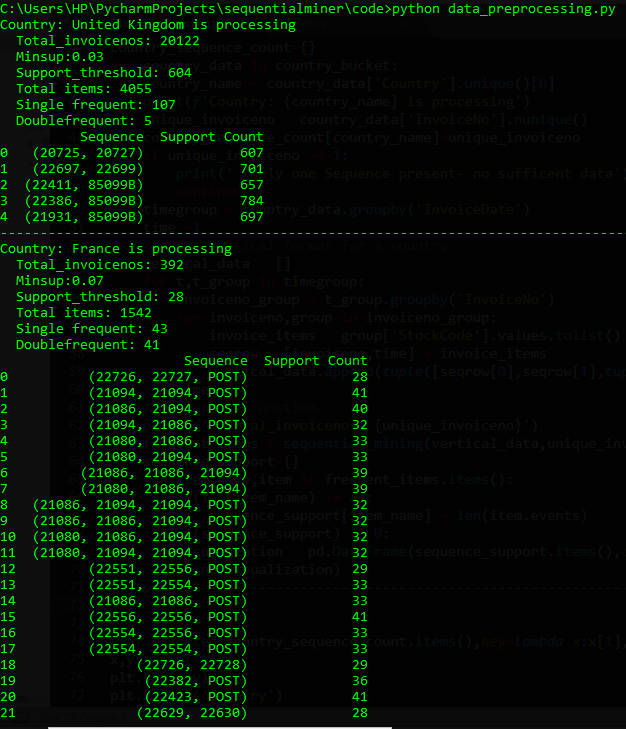
1. We call **Singlefrequent function** will returns only the frequent items of single item objects.
2. Therefore we call **Doublefrequent function** which returns only the two sequence which are frequenct are identified.
3. To make Item class object for two sequence. We make a **Temporial join** on each items of two sequence and check for support greater than support threshold.



1. We **Emunerate on frequent two sequence object** untill there are no frequent sequence found.

**Step5:** On running the code, The processing of each country is shown as follows for sample. This shows Total Invoice, Support Threshold, Single & Double Frequent Items.

**For Data Visualization, I have created a DataFrame which will have Sequence and its Support Count. And the DataFrame shows only Sequence of length 2 or more and its Count of occurrence** (sample picture)



GitHub Repository:

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

REFERENCE:

1. <https://www.researchgate.net/publication/225266300_Zaki_MJ_SPADE_An_efficient_algorithm_for_mining_frequent_sequences_Machine_Learning_421_31-60> (Main reference)
2. [https://www.geeksforgeeks.org/](https://www.geeksforgeeks.org/implementing-apriori-algorithm-in-python/)
3. <http://www.philippe-fournier-viger.com/spmf/SPADE.pdf>
4. <https://link.springer.com/content/pdf/10.1023/A:1007652502315.pdf>
5. <https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Sequence_Mining/SPADE>