Case Study of Retail Transactions via Apriori Algorithm in Python

Israel Nolazco
Introduction to Data Mining and Analytics
Lewis University
Romeoville, Illinois
israelnolazco@lewisu.edu

***Abstract*— Association Rule Mining is one of several way to find patterns in Data. This approach typically aims to find the likelihood of an event occurring given an initial starting point. This document will primarily focus on the Apriori Algorithm. The algorithm assumes frequent subset will have a certain level of certainty in future events. However, the algorithm itself will look at item sets that are common and uncommon with one and another. More so, this document will explore the full capabilities of the Rule Mining via the Apriori Algorithm using the programing language Python whilst utilizing an item set of purchase items for about ten thousand transactions.**

Keywords

Apriori Algorithm, Data Analytics, Market Analytics, Retail

I. Introduction

In the retail industry it is crucial to maintain an accurate inventory stock. Whether the store is small or big; there are too many risks the store could face if its inventory is inaccurate or outdated. Some of the problems could be, the dead stock of an item. This is the result from an item that is difficult to sell and has been in the shelve an excess amount of time. Or, the Out of Stock of an item. This is typically the result from the high demand of an item [1]. In order to avoid any of the problems above, the use of Apriori Algorithm can be implemented in order to discover item set that are typically purchase together and items that are not purchase together. This data could lead a store operator to organize and properly stock items based on consumer behavior. This document will look at one thousand transactions and analyze consumer behavior based on its results. Furthermore, this document will make a case for products that are yet to be release and how previously obtain data could be used to make an appropriate forecast.

II. Methods

In order to understand the capabilities of Apriori Algorithm there are some keywords that are important to understand before proceeding [2]:

Support: Is the ratio of the number of times an item is present over the total number of transactions.

$$Support(X) = \frac{number\ of\ times\ (X)\ is\ present}{Total\ number\ of\ transactions}$$

Figure 1. Support equation.

Confidence: Is the ratio when the transactions contain x and y over all the transactions where x is present.

$$Confidence(X \rightarrow Y) = \frac{Transactions\ containing\ (X\ \&\ Y)}{Transactions\ containing\ X}$$

Figure 2. Confidence Equation note: The equation only works when items are represented as set. NAN values will provide no insight in this equation.

Lift: Is the likelihood of transactions containing x and y over transactions only containing x.

$$Lift(X \rightarrow Y) = \frac{Confidence\ (X \rightarrow Y)}{Support(X)}$$

Figure 3. Lift Equation note: A Lift of 1 means there is no association between products X and Y. Lift of greater than 1 means products X and Y are more likely to be bought together. Lift of less than 1 refers to the case where products X and Y are unlikely to be bought together

The data utilize in this case study is obtain from Kaggle and its tittle ***Groceries Market Basket Dataset [3].*** The dataset contains nine thousand eight hundred and thirty-five transactions ranging from one to thirty-two items per transaction. The data is under course Association rules mining using Apriori algorithm. Course Assignment for CS F415- Data Mining @ BITS Pilani, Hyderabad Campus. Done under the guidance of Dr. Aruna Malapati, Assistant Professor, BITS Pilani, Hyderabad Campus [4]. Each item is represented by a column. This adjustment further increases the utilization of the data for analysis. In this case study we are looking at all thirty-two possible items in a set of one thousand transactions because, the Apriori Algorithm takes is a lengthy process thus the need to decrease the number of transactions down to one thousand from the original nine thousand plus in order to conduct this study in a timely manner.

Additionally, the installation of the following library is necessary in order to facilitate the use of Apriori library.

```
pip install apyori

from apyori import Apriori
```

The following code was utilized in order to obtain the results necessary for this case study:

```
#reads csv file and give it a reference name
retail_info = pd.read_csv('groceries - groceries.csv')


#creates an empty list and then uses a for loop ranging from 1 to 1000 rows and using column 1 to 32
records_1 = []
for i in range(1, 1000):
    records_1.append([str(retail_info.values[i,j]) for j in range(1,32)])


#utilizing the list contents the apriori function is applied with the following parameters
association_rules = apriori(records_1, min_support=0.01, min_confidence=0.3, min_lift=3, min_length=2)
association_results = list(association_rules)


#shows the total number of results
print(len(association_results))


#for loop creates a visually pleasant result
for item in association_results:


    # first index of the inner list
    # Contains base item and add item
    pair = item[0]
    items = [x for x in pair]
    print("Rule: " + items[0] + " -> " + items[1])


    #second index of the inner list
    print("Support: " + str(item[1]))


    #third index of the list located at 0th
    #of the third index of the inner list


    print("Confidence: " + str(item[2][0][2]))
    print("Lift: " + str(item[2][0][3]))
print("=====================================”)
```

With the addition of the Apriori library, it is possible to set a parameter that meet certain criteria. For this case study, the following parameters were selected.

min_support=0.01 – This allows for items that were seen at least ten different times over the one thousand transactions to be taken upon consideration.

min_confidence=0.3 → This represent a minimum thirty percent chance that the items were seen together over all off the one thousand transactions.

min_lift=3 → This represents a three times likely hood of transactions having said pair over purchasing the product by itself.

min_length=2 → This number represents how big our item set should be. In this case we are only looking for a minimum of two items purchase together.

III. Analysis

The results showed forty-four potential results given the parameters. The following are the result for the two potential sets with the highest lift index.

```
Rule: whipped/sour cream -> berries
Support: 0.01001001001001001
Confidence: 0.45454545454545453
Lift: 6.136363636363637
=====================================
Rule: nan -> whipped/sour cream
Support: 0.01001001001001001
Confidence: 0.45454545454545453
Lift: 6.136363636363637
```

Noticed that nan (missing values) are mixed within in the result thus it is crucial to further analyze the results before creating rules or any application. Additionally, we see that whipped/sour cream is six time more likely to be bought together and it seems the support for whipped/sour cream is only 0.01 which means out of ten thousand transactions a total of ten times was whipped/sour cream seen. Finally, the confidence reports a forty five percent of transactions containing whipped/sour cream also have berries in the transaction.

The following results reflect the lower possible lift. Items that are only three times more likely to be bought together.

```
Rule: nan -> whipped/sour cream
Support: 0.013013013013013013
Confidence: 0.8125
Lift: 3.017425650557621
=====================================
Rule: whipped/sour cream -> whole milk
Support: 0.013013013013013013
Confidence: 0.8125
Lift: 3.017425650557621
```

As we mentioned before, it is crucial we ignore the missing values as they are also the most common result in this research. Moving forward, whipped/source cream and whole milk shows one of the highest values for confidence. This could more than likely conclude that dairy products such as cream and milk are typically bought together in a greater frequency. However, as the lift only shows an index value of three. This means that although dairy products are typically bought together and because they are frequent requested items; the chances of needing both items at the same time decreases over time as the consumer may already have a product at home and need to restock from time to time.

Lastly, the following results reflect the lowest possible support. Items that were seen only about ten different times over all the one thousand transactions.

```
Rule: newspapers -> oil
Support: 0.01001001001001001
Confidence: 0.3333333333333333
Lift: 3.7415730337078648
=====================================
Rule: root vegetables -> tropical fruit
Support: 0.01001001001001001
Confidence: 0.38461538461538464
Lift: 3.025439127801333
```

These results tell us something new and rather interesting about the behavior of the customer given this data set. For one the newspapers and oil are very unlikely matches. However, the best possible explanation to this conundrum is a customer is more than likely going to forget oil as a crucial piece while cooking and will have it as its last item in their list. Additionally, newspapers and magazines are typically located at the register, thus concluding why both items are seen together. Additionally, as expected the results tell us that although the support for root vegetables is low there is a thirty percent chance fruit will be purchase as well.

IV. Conclusion

Apriori Algorithm is quite possible best utilize when dealing with market research analytics. There were multiple insights by utilizing a small set of one thousand transactions. Certainly, it is unexpected to encounter so many different behaviors. The data demonstrated that dairy product will typically be purchase along with berries. A store owner could potentially predict the seasons when pastries are in high demand and create a combination of a dairy sale whilst increasing the prices of berries, thus minimizing an impact on daily budget and still advertising itself as an affordable solution for consumers. Additionally, with this information a store owner could relocate certain items close to one another in order to create a higher demand; via setting dairy and berries products closer to the one another or news paper and oil closer to one another. However, let us not forget that these discoveries were encounter by analysis items that were only present ten out of one thousand transactions. The true potential of this algorithm will shine when it comes to forecasting growth and demand of products over time. Seasonal behavior such as holidays can be used to analyze what type of product will be in higher demand than other and predict when its best to provide lower prices to lower the risk of over stock. Lastly, given the accessibility of Python it is clear such potential insights via Association Rule Mining are accessible to small and big retail store.

References

[1] Williams, K. (2019). How to Manage Inventory in a Retail Store. [online] ShopKeep. Available at: https://www.shopkeep.com/blog/how-to-manage-inventory-in-a-retail-store#step-1 [Accessed 24 Sep. 2019].

[2] Malik, U. (2019). Association Rule Mining via Apriori Algorithm in Python. [online] Stack Abuse. Available at: https://stackabuse.com/association-rule-mining-via-apriori-algorithm-in-python/ [Accessed 24 Sep. 2019].

[3] Nasrullah, I. (2019). Groceries Market Basket Dataset. [online] Kaggle.com. Available at: https://www.kaggle.com/irfanasrullah/groceries [Accessed 24 Sep. 2019].

[4] Malapati, A. (2019). shubhamjha97/association-rule-mining-apriori. [online] GitHub. Available at: https://github.com/shubhamjha97/association-rule-mining-apriori [Accessed 24 Sep. 2019].