Market Basket Analysis of Instacart

Group members: Suhasini Kalaiah Linagiah, Bharath kumar karre

Abstract

The utility of Big Data is transforming the shopping experience in remarkable ways, particularly by suggesting items to users based on their prior purchases and similarity to others. For this project, we used the Instacart dataset which has over 3 million online grocery purchases. From this data, we generated frequent item sets and association rules with the Apriori algorithm, Recommendation based on clustering users using K-means algorithm, and finally made recommendations of Product Bundles using the bigram frequencies of items. The combination of these three steps allows us to predict the item bundles that a user will buy next time, offering immense benefit to businesses.

Introduction

The shopping experience has been transforming with advances in computational abilities and techniques. With the advent of Big Data analytics, similar items can now be grouped near each other to facilitate the shopping experience of customers. For example, without Big Data analytics, grocery stores would not put diapers and beer near each other because these seem to be wholly separate items on the surface until associations are found between them in the data.

The impacts of Big Data analytics are not restrained to brick and mortar stores though. Many companies are now delivering food directly to the front door of customers' homes. One of these companies, Instacart, provides a publicly available, large dataset of 3 million customer purchases along with the day of the week, time of purchase, and relative time in between purchases for each user [4]. The size and nature of this dataset make it an ideal candidate to test Big Data approaches to discern patterns and potentially find creative ideas to improve profits for food delivery companies.

This project focuses on the ability to predict and recommend items for users. These will be based on the association rules based on the similarity of items as well as the similarity in features of users, to give as accurate and personalized of a prediction for purchases with the goal of increasing profits for a prospective company.

Related work

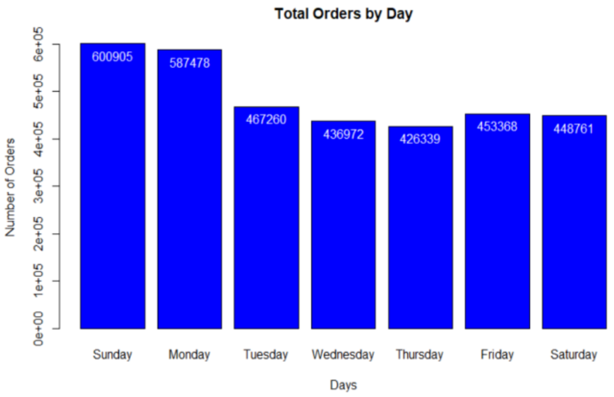
The inspiration to do this project came from a mixture of class lectures and research papers that explore methods to form association rules and make item predictions. The conceptual framework that inspired the use of Apriori for market basket analysis came from the foundational paper by Brin *et al.* [3]. They explored the implications of using Apriori and other dynamic algorithms for forming association rules. They found that Apriori works better than others on high support thresholds where support is the number of occurences of an itemset. Most importantly, this paper showed that the use of market basket analysis is much more suited for grocery store transactions than for census data and similar datasets with high levels of correlations and redundancies. This narrowed our focus to the Instacart dataset since they proved Market Basket analysis is well-suited for this type of dataset.

Having decided on the dataset, we searched for papers that performed market data analysis on retail purchases. The approach used by Annie *et al.* provided a practical framework upon which our approach was built. They utilized a K-Apriori approach to form association rules and then clustered the users in groups to predict future items [2]. Similar to the notes from the CIS 5570 lecture on Frequent Itemsets [1], Annie *et al.* took advantage of the monotonic property of support measures in which the support of an itemset never exceeds the support of its subsets. This allowed Annie *et al.* to form frequent itemsets using the Map/Reduce architecture in an efficient manner even on sparse datasets. The customers were then grouped into clusters based on the similarity of their purchased items using the K-means algorithm and then recommending items to certain clusters [2].

Exploratory Data Analysis

To have deeper understanding of data we performed EDA, we found few interesting patterns in data.

1. Sunday and Monday have highest orders in a week.
2. Orders are more from 8 AM to 6 PM. As we approach night, the customers inflow decreases.
3. We identified few products are reordered in first week of purchase. While few items are ordered after a month
4. Majority of healthy products are ordered during the daytime and unhealthy products are ordered in the evening.



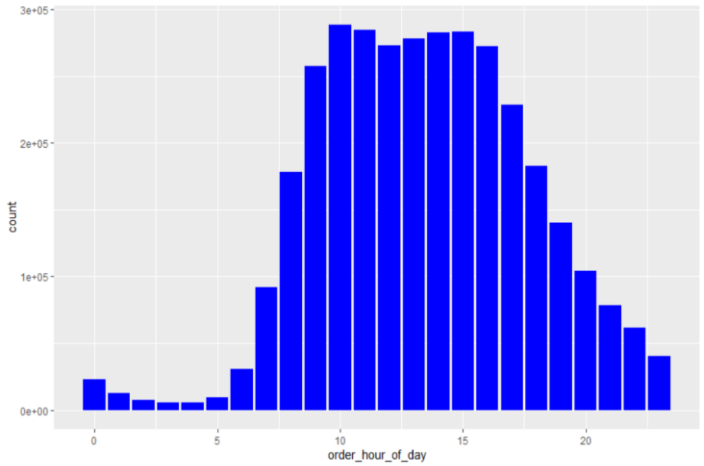
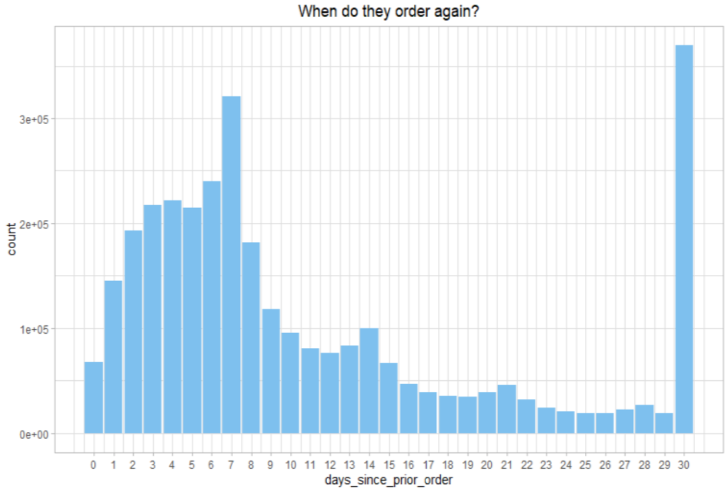


Figure 1: No of orders by Day

Figure 2: Number of orders by time of day



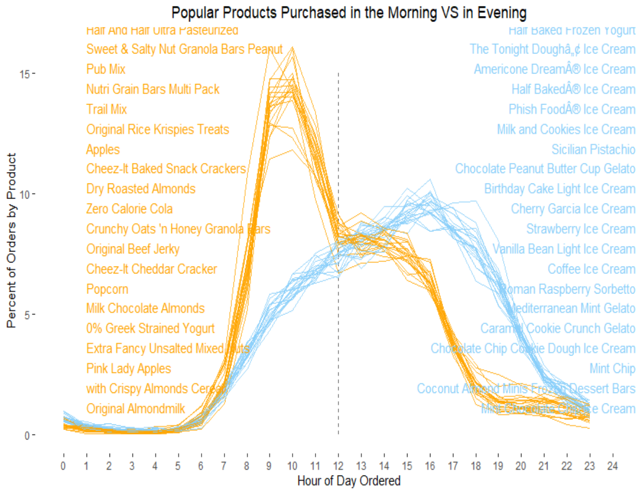


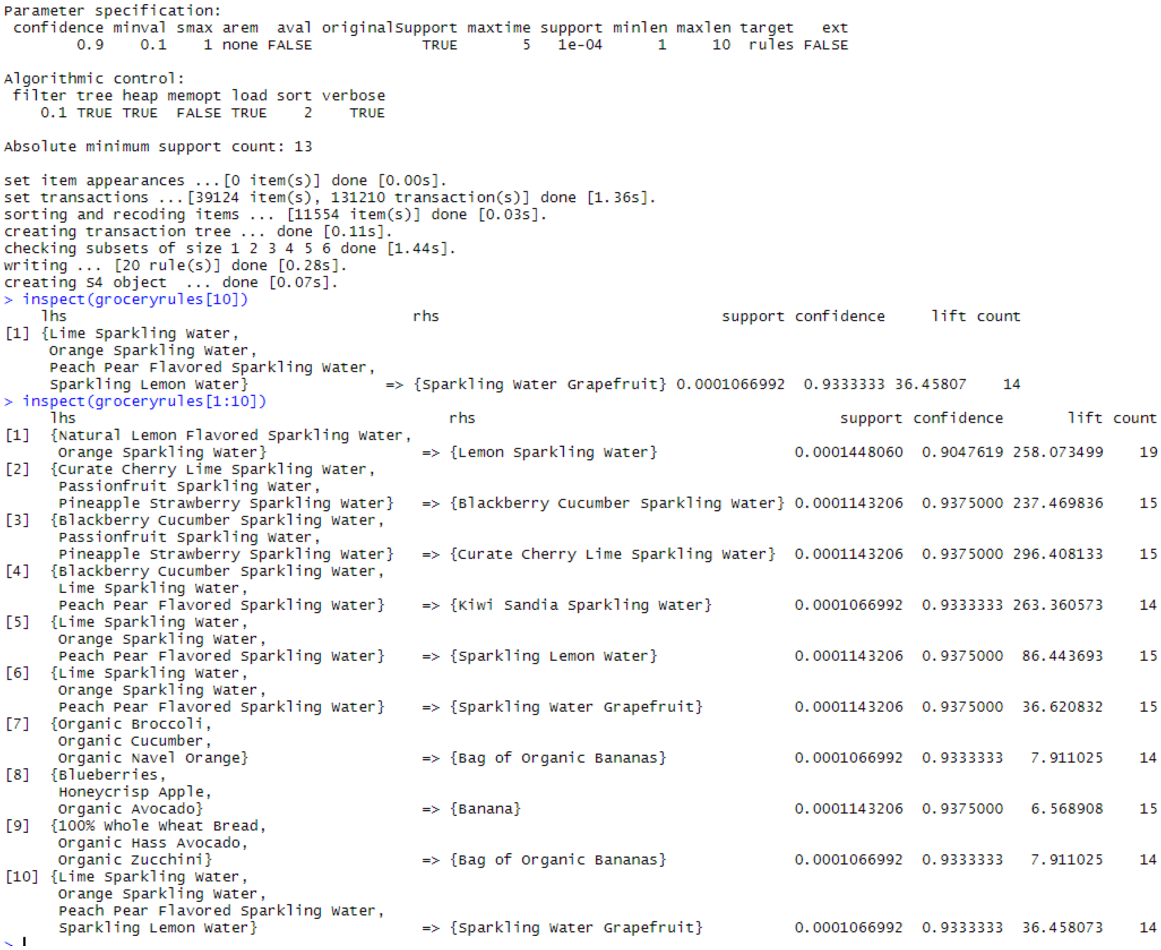
Figure 3: No of orders people reorder products

Figure 4: Popular Products purchased during Morning & Evening

Methodology

This project analyzed the Instacart dataset in four phases. The first phase explored the features of the Instacart dataset to find general patterns.

The second phase of our project performed association rule mining on the entire dataset using the Apriori algorithm. We have implemented the Apriori algorithm in R programming language and tested with different support and confidence values. It accomplishes this by iteratively generating candidate frequent itemsets and pruning non-frequent itemsets. This repeats *k* times for whatever size *k* itemsets are desired. Here is the output of the apriori algorithm for confidence of 0.9 and support of 0.0001



```
Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen maxlen target   ext
        0.9    0.1    1 none FALSE            TRUE       5   1e-04      1     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 13

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[39124 item(s), 131210 transaction(s)] done [1.36s].
sorting and recoding items ... [11554 item(s)] done [0.03s].
creating transaction tree ... done [0.11s].
checking subsets of size 1 2 3 4 5 6 done [1.44s].
writing ... [20 rule(s)] done [0.28s].
creating S4 object  ... done [0.07s].
> inspect(groceryrules[10])
    lhs                                  rhs                                support confidence     lift count
[1] {Lime Sparkling Water,
     Orange Sparkling Water,
     Peach Pear Flavored Sparkling Water,
     Sparkling Lemon Water}           => {Sparkling Water Grapefruit} 0.0001066992  0.9333333 36.45807    14
> inspect(groceryrules[1:10])
     lhs                                      rhs                                 support confidence       lift count
[1]  {Natural Lemon Flavored Sparkling Water,
      Orange Sparkling Water}             => {Lemon Sparkling Water}             0.0001448060  0.9047619 258.073499    19
[2]  {Curate Cherry Lime Sparkling Water,
      Passionfruit Sparkling Water,
      Pineapple Strawberry Sparkling Water}   => {Blackberry Cucumber Sparkling Water} 0.0001143206  0.9375000 237.469836    15
[3]  {Blackberry Cucumber Sparkling Water,
      Passionfruit Sparkling Water,
      Pineapple Strawberry Sparkling Water}   => {Curate Cherry Lime Sparkling Water}  0.0001143206  0.9375000 296.408133    15
[4]  {Blackberry Cucumber Sparkling Water,
      Lime Sparkling Water,
      Peach Pear Flavored Sparkling Water}    => {Kiwi Sandia Sparkling Water}         0.0001066992  0.9333333 263.360573    14
[5]  {Lime Sparkling Water,
      Orange Sparkling Water,
      Peach Pear Flavored Sparkling Water}    => {Sparkling Lemon Water}               0.0001143206  0.9375000  86.443693    15
[6]  {Lime Sparkling Water,
      Orange Sparkling Water,
      Peach Pear Flavored Sparkling Water}    => {Sparkling Water Grapefruit}          0.0001143206  0.9375000  36.620832    15
[7]  {Organic Broccoli,
      Organic Cucumber,
      Organic Navel Orange}                   => {Bag of Organic Bananas}              0.0001066992  0.9333333   7.911025    14
[8]  {Blueberries,
      Honeycrisp Apple,
      Organic Avocado}                        => {Banana}                              0.0001143206  0.9375000   6.568908    15
[9]  {100% Whole Wheat Bread,
      Organic Hass Avocado,
      Organic Zucchini}                       => {Bag of Organic Bananas}              0.0001066992  0.9333333   7.911025    14
[10] {Lime Sparkling Water,
      Orange Sparkling Water,
      Peach Pear Flavored Sparkling Water,
      Sparkling Lemon Water}                  => {Sparkling Water Grapefruit}          0.0001066992  0.9333333  36.458073    14
> 
```

When customers purchase Curate Cherry, Passion fruit, pineapple, strawberry sparkling water, then they also buy Blackberry cucumber Sparkling water.

When a customer purchases Lime, Orange, Peach Pear, Lemon then they also bought Grapefruit sparkling water. The customer is preferring to buy all citrus flavors.

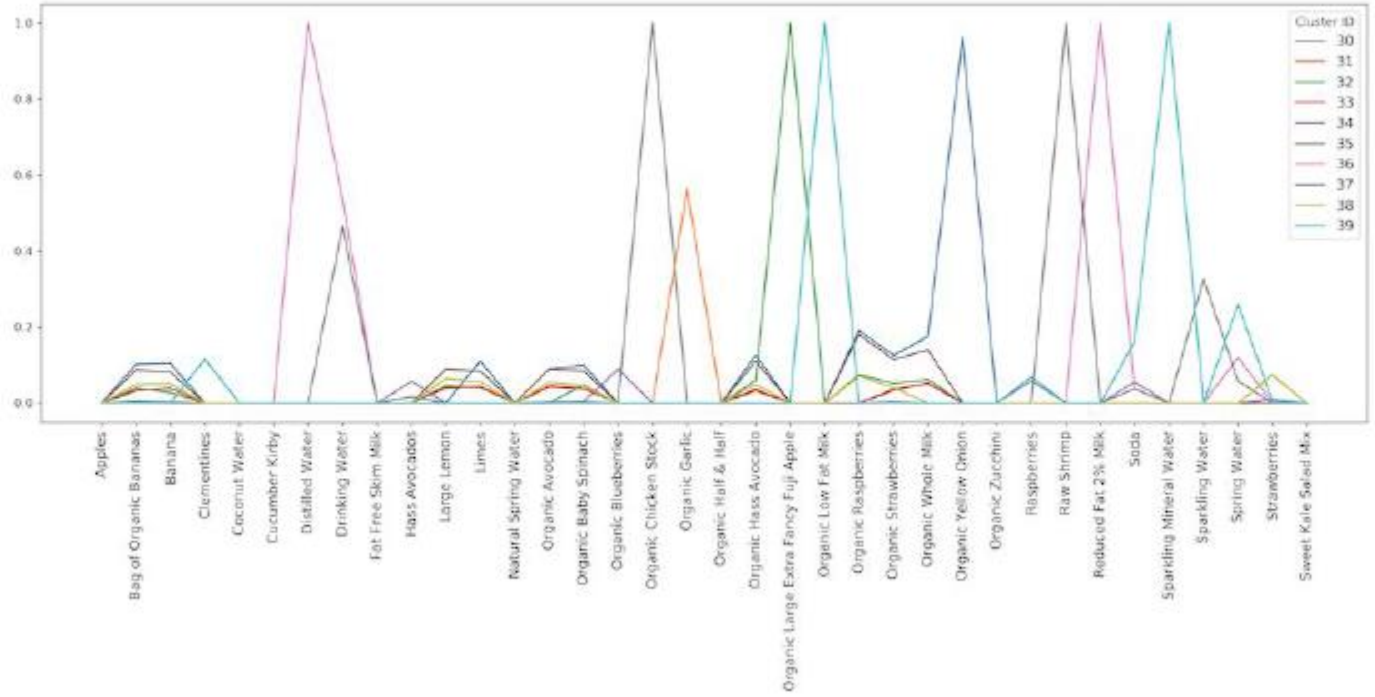
Word2Vec to convert words into vector form of features which can then be used to visualize patterns and further used in prediction models. Examples can be seen in Figures 1 and 2 in which the number of sales per department and number of sales per day can be visualized.

The third part of this project was to cluster the users based on similar habits that were found from part 1, the exploratory data analysis phase. The K-means algorithm was employed to measure the similarity of users and cluster them into groups. The K-means algorithm we used was adopted from Spark.mlib, which is a parallelized version of the standard K-means algorithm adapted for use in Big Data analytics. We decided to cluster the users based on

- Day of week a customer is purchasing
- Hour is customer purchasing
- Days since last order
- Number of total orders
- Number of total products
- Preferred products (We converted this data to word2vec format)

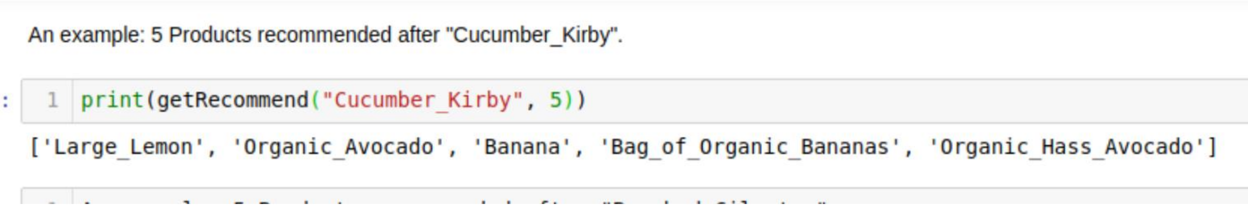
By implementing the elbow method, we found that we can have 40 clusters(k = 40). From these clusters , we identified the purchasing patterns, and we can group them.

From the image below, we can observe the buying patterns of the users of different clusters for the top 20 products.



Since we used our own machines, we were restricted to using 10% and 20% of the dataset for clustering.

The fourth part of this project aimed to recommend product bundles based on bigram frequencies from the dataset. After clustering users into groups with K-means, we then recommended bundles of 5, 10, and 15 items to groups based on their frequently purchased items. Finally, to evaluate the accuracy of the recommended items the number of recommended items found in the order are divided by the total number of items in the order. For example, if an order contains 10 products, we start by recommending a bundle of 10 items similar to item 1 based on the top 10 associations for item 1. Then, for each occurrence of the recommended item in the original order, the score is incremented by 1. This process is repeated for each item in the order and the final score is divided by the size of the order. This process is then repeated for all orders and an average score is reached.



An example: 5 Products recommended after "Cucumber_Kirby".

```
print(getRecommend("Cucumber_Kirby", 5))
['Large_Lemon', 'Organic_Avocado', 'Banana', 'Bag_of_Organic_Bananas', 'Organic_Hass_Avocado']
```

Figure 3: Example of a bundle of 5 items recommended based on bigram frequencies of “Cucumber_Kirby”

Experimental Discussion

The goal of our experiment was to project accurate association rules among the clusters and recommend the items. Then we determine the accuracy of recommending bundles of items to clusters of users based on their past purchases. We began by computing the association rules for the entire Instacart dataset. Then, we clustered the users into groups of 10 based on their similarities in number of transactions. Following this, we iterated through orders placed in different clusters, generated a recommendation bundle of items similar to the size of each order based on the top associations (bigrams), and then computed the accuracy of recommended items being in the order. We produced a final accuracy of 17.94% after iterating through the different clusters and orders.

Conclusion

The approaches used produced an accuracy of 17.94% for recommended items being purchased. This shows promise with this approach and potential applicability to companies, particularly online retailers that simply need to recommend items instead of stock different items near each other. Going forward, more computers can be utilized to nullify the bottleneck that our computer’s memory played and the Apriori algorithm could be expanded for trigrams and larger itemsets than just bigrams.

References

1. Abouelenien, Mohamed; Retrieved from CIS 5570 Introduction to Big Data Lecture, Chapter 6 Frequent Itemsets. (Dec 2019), Dearborn, Michigan, USA

2. Annie, Loraine Charlet M C; Kumar, Ashok D.; Market Basket Analysis for a Supermarket based on Frequent Itemset Mining. International Journal of Computer Science Issues (IJCSI); Mahebourg Vol. 9, Iss. 5, (Sep 2012): 257-264.

3. Brin, Sergey; Motwani, Rajeev; Ullman, Jeffrey D.; Tsur, Shalom; Dynamic itemset counting and implication rules for market basket data, Proceedings of the 1997 ACM SIGMOD international conference on Management of data, p.255-264, (May 11-15, 1997), Tucson, Arizona, USA [doi>10.1145/253260.253325]

4. “The Instacart Online Grocery Shopping Dataset 2017”, Accessed from https://www.instacart.com/datasets/grocery-shopping-2017 on 12/7/2019