

MarKET BASKET ANALYSIS AND RECOMMENDATION SYSTEM FOR INSTACART



CPE-646-A

Shengyu Hu

**Introduction**

Recommendation systems have played a crucial way in redefining our modern lives in many ways through solving the problem of information overload by searching through a large volume of dynamically generated content to provide personalized services. It is most commonly recognized for video and music services, product recommenders and content recommenders for social media platforms.

During the special period of global pandemic, groceries stores are experiencing an exponential growth in customer orders. Instacart, one of the most popular grocery delivery services providers in the United States and Canada, has expended its shopper network by 250% in order to meet the increasing customer demand by hiring 0.3 million new full-service shoppers. Following its business model requires the need of recommender system to make a user’s experience better and in turn encourage them to return purchasing with Instacart.

**Objectives**

Instacart uses the order data from the customers to be able to understand their purchase patterns. It uses the data to learn about the products a user will buy again or add to their cart next during a particular shopping session. The project is proposed to investigate user behaviors and end up with recommending the best possible products for each user through collaborate filtering technique association rule mining. Behavior-related problems are going to be solved such as which department or aisle will have the greatest number of orders or how frequent a customer places the order on the Instacart app.

**Dataset**

Source: <https://www.kaggle.com/c/instacart-market-basket-analysis/data>

The Instacart dataset provides over 3 million grocery orders in 2017 by more than 200,000 users, each with 4 to 100 orders. Information contain products details and their corresponding departments and aisles. Additional characteristics also include the time of day of purchase and if the product has been reordered by customer before. There are no missing values or outliers.

**Exploratory Data Analysis**

* Customer Clustering (Appendix A)

Then users are clustered through PCA and K-means clustering. I used the elbow method and the dendrogram to figure out an optimal number of clusters. It turns out to be anywhere between 3 and 5. Final decision was made on 5 clusters. From now on, customer behaviors can also be analyzed on different clusters.

* Order distribution by User (Appendix B)

The maximum number of orders per user ID is 99. This is an exponential distribution, intuitively this make sense. The average number of orders is 17 per user and 50% of the customer order less than 10 times. Assuming everything is equal, the customer purchasing behavior is sub-optimal if measured by number of repeated purchases. Perhaps marketing can boost their promotional efforts towards a subset of customers who order less than 16 times but more than 9 times, in a hope to close the gap.

* Order distribution by Department (Appendix C)

Some departments have infrequent sales, which could help eliminating some of the departments from recommending. Produce, snacks and dairy eggs contribute to almost 50% transactions.

* Order distribution by Week/Day/Hour (Appendix D)

In general, orders are placed everyday mostly between 10AM - 4PM. Sunday and Monday are the busiest day for order placement. The hottest timing for order placement is Sunday between 1 - 3PM and Monday between 9 - 11AM. Wednesday and Thursday are less busy for order placement.

* Re-order frequency by Cluster (Appendix E)

There seems to be a capped variable (30 days). The most common number of days between orders is 11 days. Users in Cluster 1 and 2 tend to have a wider time gap between purchases. Users in cluster 3 and 4 are the most recurrent buyers.

* Re-order ratio by Department (Appendix F)

As expected, produce and diary eggs enjoyed the highest re-order ratio whereas pantry was experiencing the worst situation. Although snack and beverage have similar proportion of order sales, beverage has a positive re-order ratio while snack has a negative one.

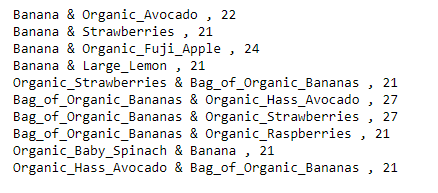
* Top Product/Department/Aisle (Appendix G)

Clearly, fresh fruits and veggies are mostly purchased and most of the popular products are organic. There are no obvious differences of the most popular products among different clusters. Among all products, bananas, strawberries, baby spinach and avocado rank the highest in customer preferences.

**Recommender**

**A. Product Bundle**

It usually happens that some products are more often bought together than others. Product bundle can be used to predict which product the customer will buy next. Once a customer adds one product to cart, a list of recommended products will be offered to be bought together.



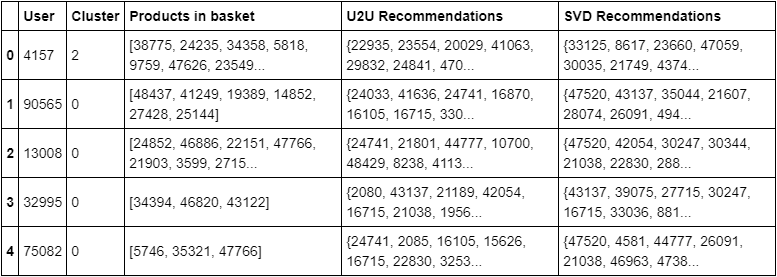
Not surprisingly, bananas are bought mostly with avocado, apples and strawberries. Then a simple recommender is generated based on frequencies in the bundles by firstly sorting the bigram frequencies in descending order and then returning merely the corresponding product names in the same order. Take chocolate sandwich cookies for example, a recommending list in a size of either 5 or 15 (Appendix H). If a customer adds the cookies into his basket, then reduced fat milk or semi-sweet chocolate morsels will be recommended to him.

**B. Neighborhood-based Method**

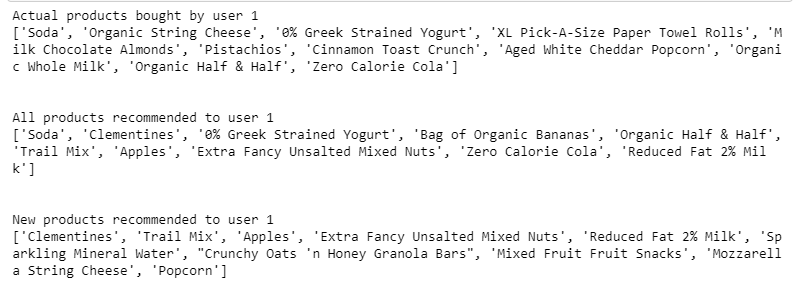
Purchases of all users are split into training and testing with a test size of 0.2. Prior purchases are used to build utility matrix, with products as rows, users as columns and entries as purchase frequencies.

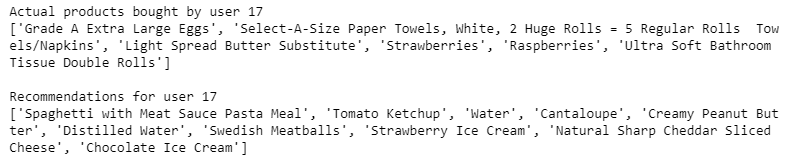
It is a user-to-user recommender. After figuring out the cluster he belongs to, top similar users are generated based on cosine similarities, and the similarity is tested through checking user purchasing history. Top similar users of User ID 1 give a recall value of 0.333 indicating a high similarity. Product recommendations are then generated [Appendix I].

**C. Latent-factor Method**

Since the utility matrix is almost fully sparse, it will uncover latent features through matrix factorization. SVD factorization is applied and I sticked with the example user ID 1. The utility matrix is factorized using SciPy’s SVD. The list of recommended products is generated but it may contain products that already in the user’s basket which needs to be removed before the final recommendation [Appendix J]. 

The recall for intersection of U2U and SVD is around 9%. Nearly 1 in 10 products recommended by each system is recommended too by the other for each user. For User 4157 as listed in the above output table, product 47059 is indeed repeated in both recommendations.

Example recommendations for User ID 1 using SVD are illustrated above. However, SVD may reveal some drawbacks when tackling with implicit data. SVD considers explicit data where the user has rated both products they like and dislike, while provided the count of product purchased by a user, low values in the utility matrix or a user not buying a product cannot be treated as dislike. As the Instacart dataset consists only of implicit feedback in the form of past and current grocery orders, there is a need to robust the approach with, for example, Alternating Least Squares.

Example recommendations for user ID 17 is shown above. There are a few similarities between actual product bought and recommendations. It is might due to that, for example, ‘Select-A-Size White Paper Towels’ are a very suitable alternative to ‘Select-A-Size Paper Towels, White, 2 Huge Rolls = 5 Regular Rolls Towels/Napkins’.

**Results**

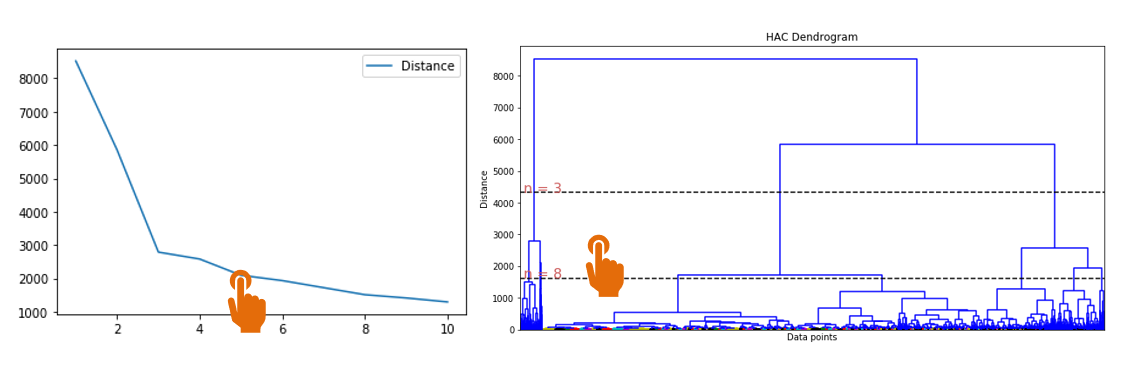
A list of products is generated to each Instacart user and a predix of the list is presented as the recommended products. It is important to realize that reliable feedbacks regarding which products are disliked are not available, and not purchasing a product can stem form multiple different reason. In this case, precision based metrics are not very appropriate for model evaluation. With the test size 0.2, recommendations are generated for every user. As a result, matrix factorization using either SVD or ALS outperforms the baseline model by 2 times better. Different number of latent factors are tested as well, and it can be concluded that the accuracy of 100-factor SVD is slightly lower than the baseline model probably due to overfitting. According to the neighborhood-based model, users seem to be interested at items that other similar users have purchased. Last but not least, Alternating Least Square runs much faster than other models because the linear running time is optimized under the settings.

**Discussion & Futher Works**

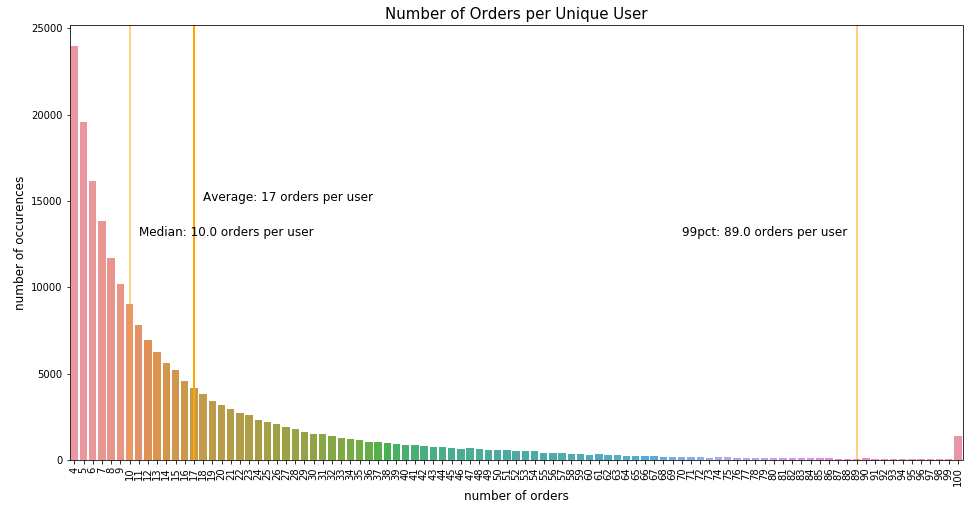
It is an quite interesting area to explore. The evaluation of the recommender systems, even a simple SVD model, performs better than just recommending the popular products. Personalized recommendations based on purchase history are more indicative of Instacart user purchase patterns. However, if new updated data with outliers or missing values are introduced, there will be a cold start problem. In this case, hybrid recommender with, for example, content-based models or deep neural networks will definitely product a better outcome. The ‘Banana Mystery’ is also another topic to discuss for the future works.

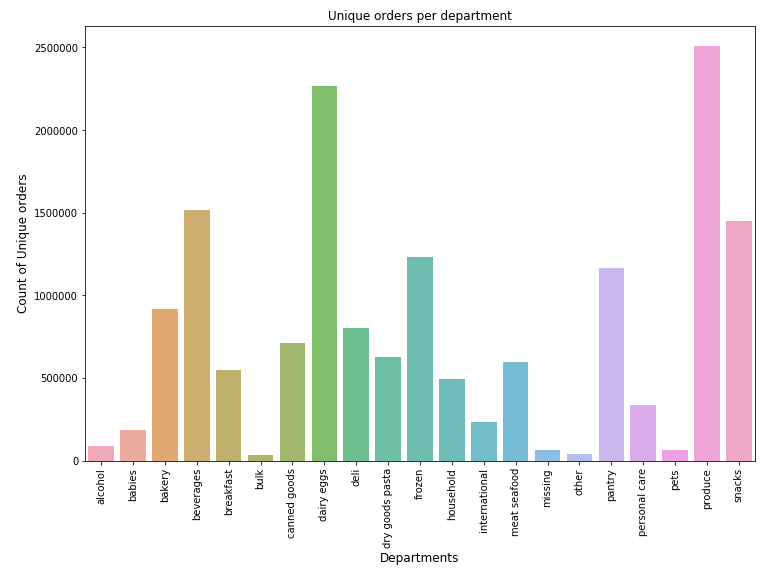
**Appendices**

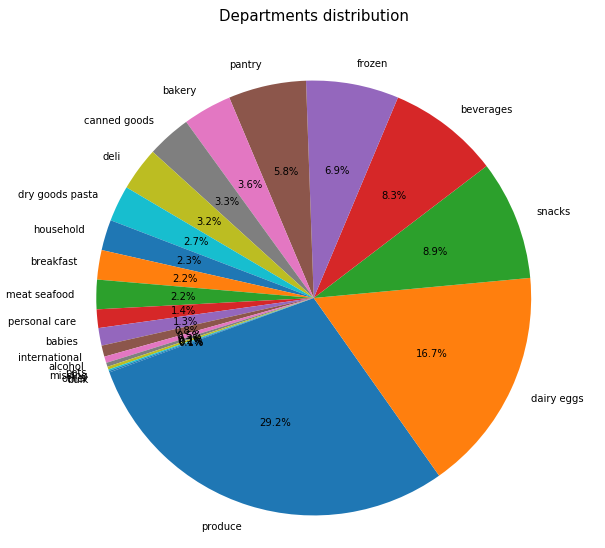
A. Customer clustering



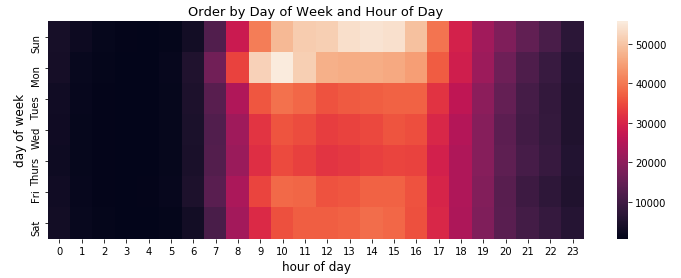
B. Order distribution by User



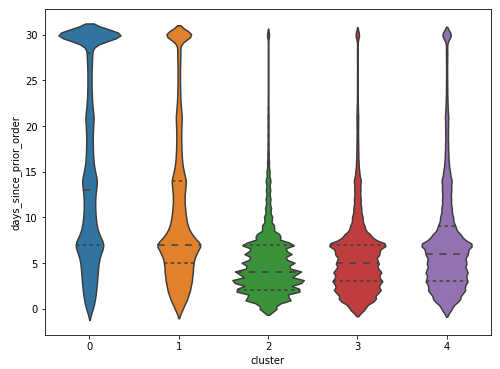
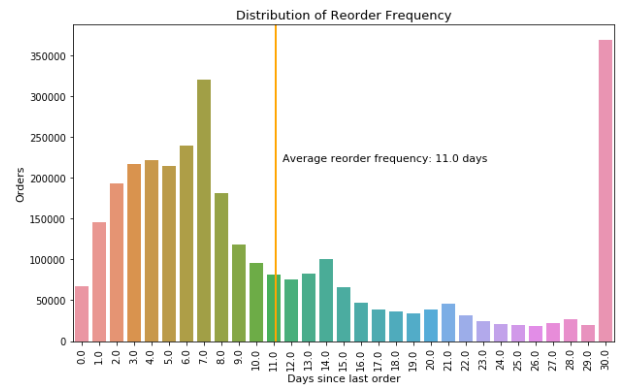
C. Order distribution by Department



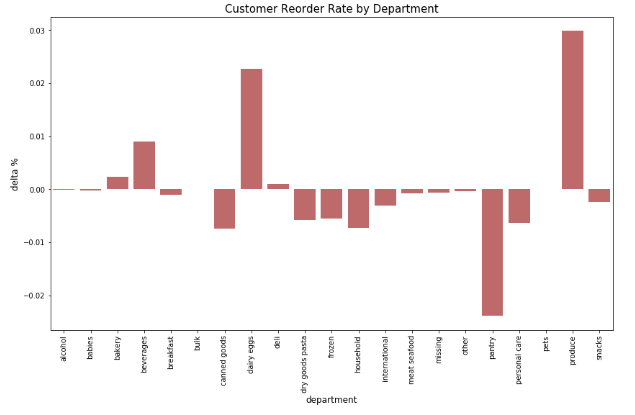
D. Order distribution by Week/Day/Hour



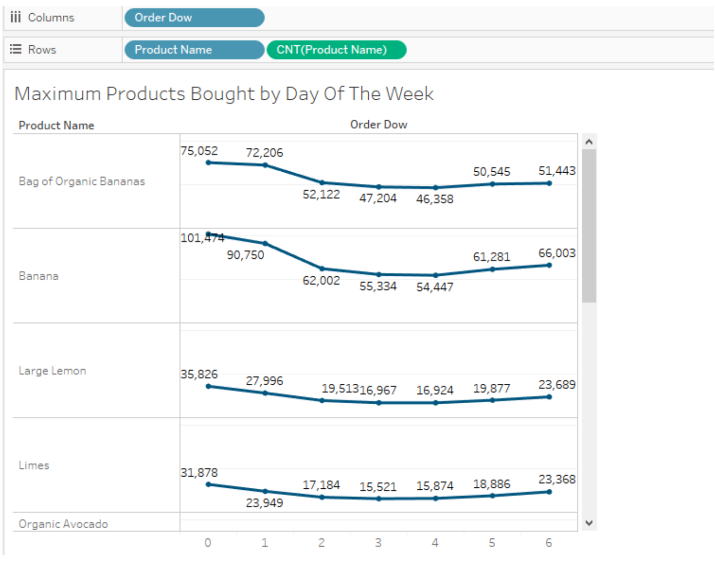
E. Re-order distribution by Cluster

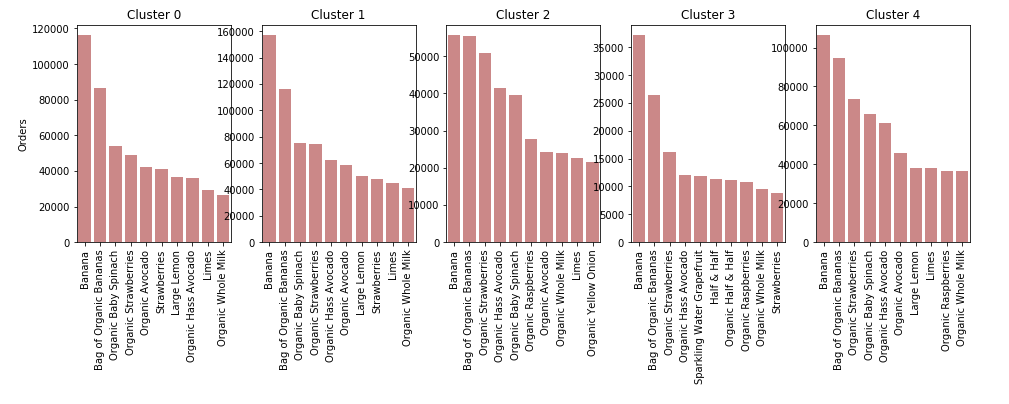
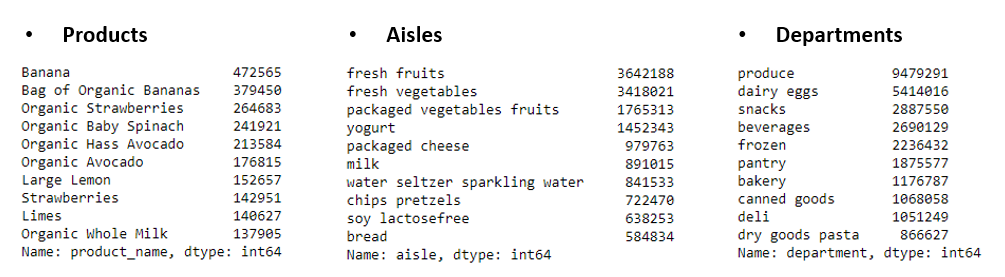


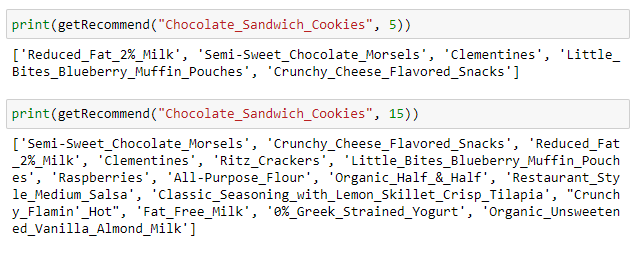
F. Re-order ratios by Department



G. Top Product/Department/Aisle





H. Recommended products for ‘chocolate sandwich cookies’

I. U2U Recommender

J. Remove products that already exist in the basket