Instacart Market Basket Analysis

Xiaolong Wei	Ying Li	Zhiheng Dong
GTid: 903650392	GTid:903565910	GTid: 903650936
xwei91@gatech.edu	yli3313@gatech.edu	zdong88@gatech.edu

May 01,2021

Problem Statement

Consumers around the country have been sheltering in place which lead to the increased demand for online grocery shopping. Instacart is a grocery ordering and delivery app that helps customers to shop in their favorite stores. In order to provide a delightful shopping experience, Instacart has to understand the purchase behavior of the customer, such as combinations of products that most frequently occur together in their orders. For example, customers have a higher chance to order creamer if they had coffee in their basket.

For this project, our team is interested in predicting which previously purchased products will be in the customer's next order and recommending new products based on their association. We will use the past transactions dataset containing order information about detail items and times. This problem is important because Instacart needs to optimize their recommendation system for each customer, to improve customer online shopping experience.

Data - Overview

All the data are open source and can be found on kaggle's "Instacart Market Basket Analysis" [1] competition. This dataset contains 5 tables: departments, orders, order_products, products, and aisles. The ER diagram below is a graph presentation of the table relationships:

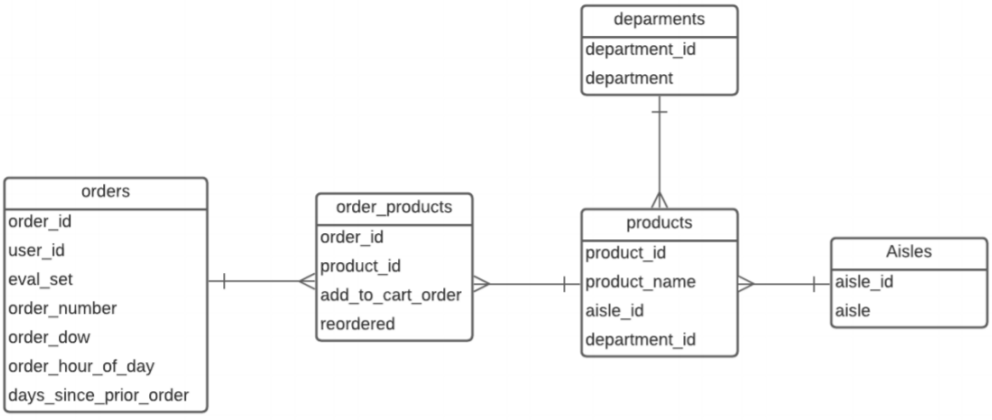


Figure 1: Graph presentation of original dataset

Methodology

Data Wrangling:
The initial dataset contains multiple tables with columns that are not pertinent for building the predictive model. In addition to that, there are multiple orders for each customer that require combining their transactions result in one row. Based on the original dataset "eval_set" that contain either "prior" or "train", we used the data that labeled with "prior" as our training set, and data labeled with "train" as the testing dataset.

Data Standardization:
Features scaling is important before building predictive models that help with reducing bias. We found that some user IDs tend to purchase more items than others, and this might contribute unequally to the predictive model. Thus, we standardize each product purchase count by dividing the total purchase count of that customer before fitting the data into models.

- *Customer Product Purchase Ratio:* we construct a sparse matrix (COO matrix) using the Use_ID as rows and the Product_ID as columns and Product_Purchase_Ratio as corresponding values. Each row represents the unique customer's purchase history and is treated equally during the analysis process. The data is saved in covariance matrix format for space efficiency
- *Customer Reorder Ratio:* We found out that a customer might purchase a product multiple times, so we generate the ratio of reordered products by dividing individual product reorder count by the total number of reorders. Also, considering customers might have preference changes, we also generate the ratio using only the customer's last 3 orders. The data are saved as individual pickle file labeled with customers user_id for faster loading

Principal Component Analysis:
Principal Component Analysis (PCA) is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller set that still contains most of the information in the large set. PCA aims to remove correlations within the data and ranks coordinates by importance. In our experiment, we transformed and reduced the data into 2 dimensions for exploratory data analysis.

K-means clustering:
K-means clustering is an unsupervised machine learning method that aims to partition *n* observations into *k* clusters, which each observation is assigned to a cluster with the nearest mean distance to the cluster centroid. Before building the predictive model, we tried to cluster customers into 12 groups based on their purchased product ID.

- We performed k-means clustering over unique User_ID with full product purchases
- We also performed k-means clustering over unique User_ID with PCA results

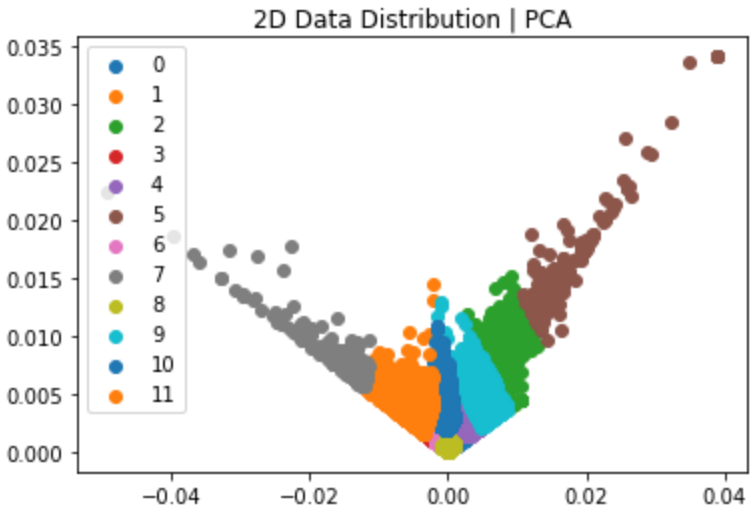


Figure 2: Data distribution of clusters using PCA

Association rules learning:
Association rule learning is a rule-based machine learning algorithm that aims to discover the associations between data points for large scales of dataset. By using association rules learning methods, we tried to discover the items that are frequently bought together. There are three common ways to measure association: support, confidence, and lift.

Item-item collaborative filtering (Item-CF)
Item-CF uses the purchase frequency of the item pair to determine their association score using the association rules. For instance, milk might be frequently purchased together with bread, so we can use the total occurrence of milk and bread purchases to compute the likelihood for purchasing the pair when customers purchase either of the products within the pair [3].

- Item-to-item similarity mappings with no regularization:

$$W_{ij} = \frac{|N(i) \cap N(j)|}{|N(i)|}$$

  To compute the conditional probability of purchasing item i and item j together, given the occurrences for purchasing N(i) and purchasing both item i and j at the same time.
  This method has one downside: the score will be too extreme when the denominator is relatively small or large.
- Item-to-item similarity mappings with regularization:

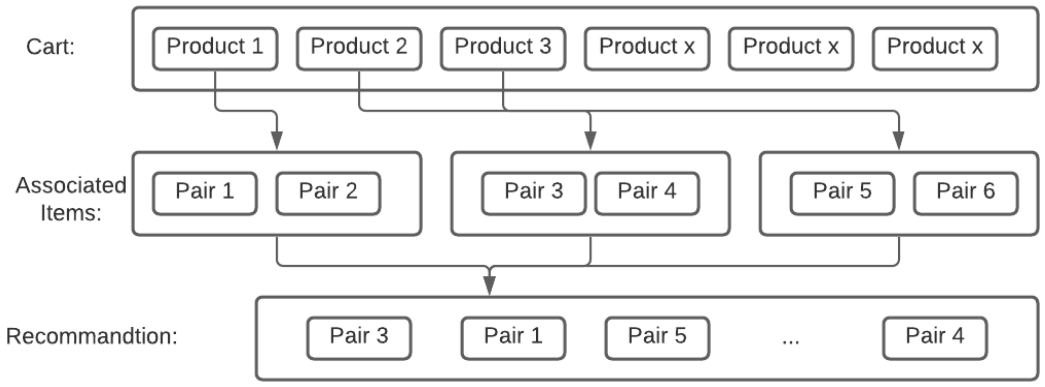
$$W_{ij} = \frac{|N(i) \cap N(j)|}{\sqrt{|N(i)|\,|N(j)|}}$$

  To penalize the extreme behavior of the similarity, a new denominator is introduced, that aims to equally weigh frequently and seldom purchased items, we regularize the denominator by taking the square root of total item i and item j purchases.

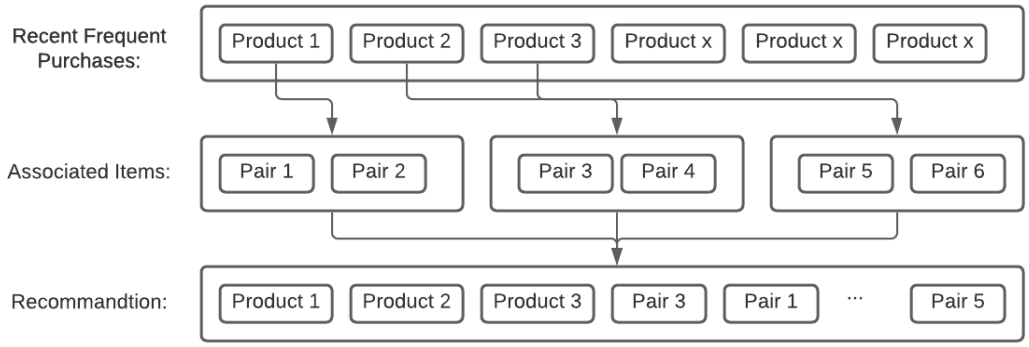
Recommendation Methods

Based on the k-means clustering algorithm and pairwise association rule, we come up with 3 approaches to make product recommendations. Each method will give 10 product recommendations.

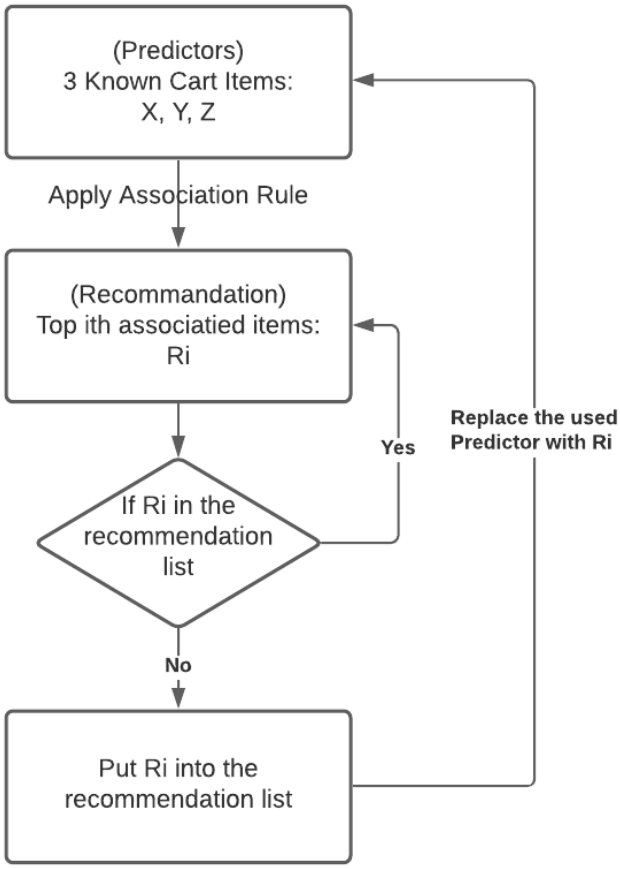
- Pairwise Recommendations using 3 known cart items - use 3 known cart items, to recommend top 10 associated items using cluster based association rules.



- Frequent purchase items & top 5 associated pairwise items - recommend 5 items from customer recent frequent purchases, then use those 5 items to find another 5 associated items using cluster based association rules.



- Dynamic pairwise recommendations with 3 known cart items: Initially have 3 known cart items in the predictor bucket to make the 4th recommended item by picking the top 1st associated item as recommendation. Then, update the predictor bucket for the next run of recommendations.



Scoring Method

To evaluate our model performance, we make 10 product recommendations used to compare with customers' last purchases (testing dataset).

- If number of recommended products greater than the number of last purchases by customers:
- If number of recommended products less and equal to the number of last purchases by customers:

$$S(X) = \begin{cases} \frac{|A(X) \cap P(X)|}{|A(X)|} & : if \ \ |A(X)| < |P(X)| \\ \frac{|A(X) \cap P(X)|}{|P(X)|} & : otherwise \end{cases}$$

** S(X): accuracy score, A(X): actual purchases, P(X): predicted purchases

Evaluation and Results

K-means clustering:

After tuning and determining the dataframe, we run the k-means clustering algorithm on the *Customer Product Purchase Ratio* covariance matrix in order to find the optimal k values. As shown in Figure 3, the k value of 12 is chosen.

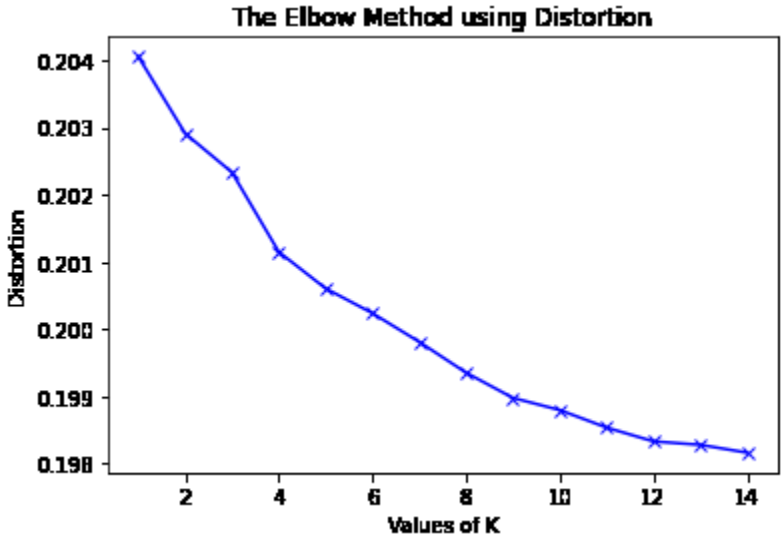


Figure 3: K-means clustering elbow curve

Association rules learning:

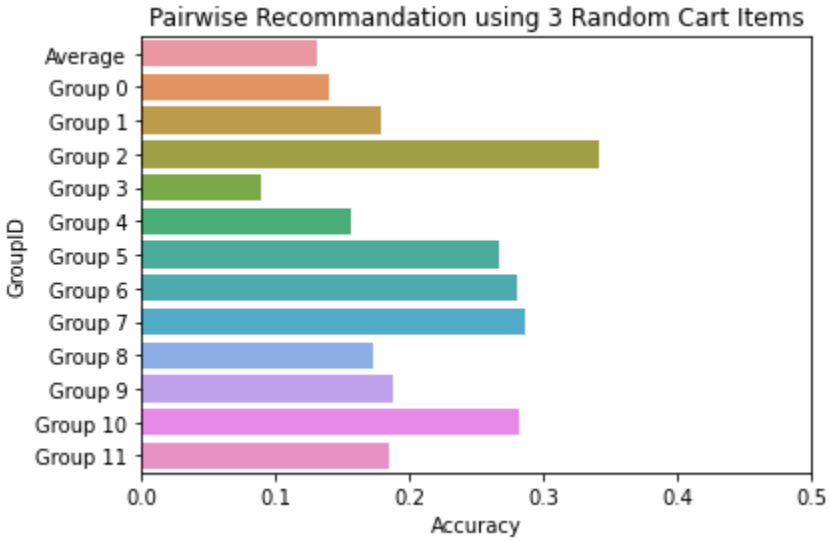


Figure 4: customer groups prediction accuracy with 3 random cart items model

The prediction accuracy for pairwise recommendation model using 3 random cart items is shown in Figure 4, the average accuracy of all customer groups is near 0.15, while Group 2 has the highest prediction accuracy which is 0.35, and Group 3's performance is worst which is less than 0.1.

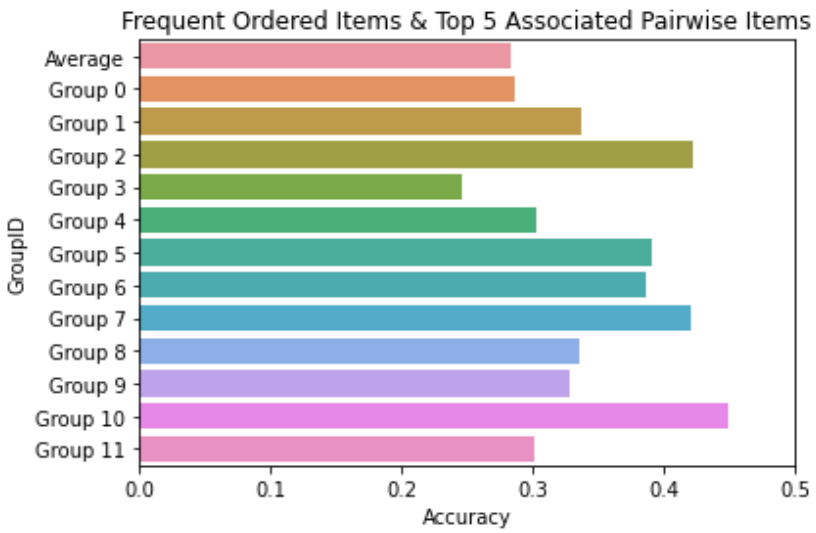


Figure 5: customer groups prediction accuracy with frequent ordered items and top 5 associated pairwise items model

In Figure 5, the prediction accuracy of analysis using the combination model of frequent ordered items and top 5 associated pairwise items is shown. In general, the accuracy of all customer groups have improved, with the average accuracy is near 0.3. The highest accuracy is Group 10 with a score of 0.45, and the lowest score is Group 3 which is 0.23.

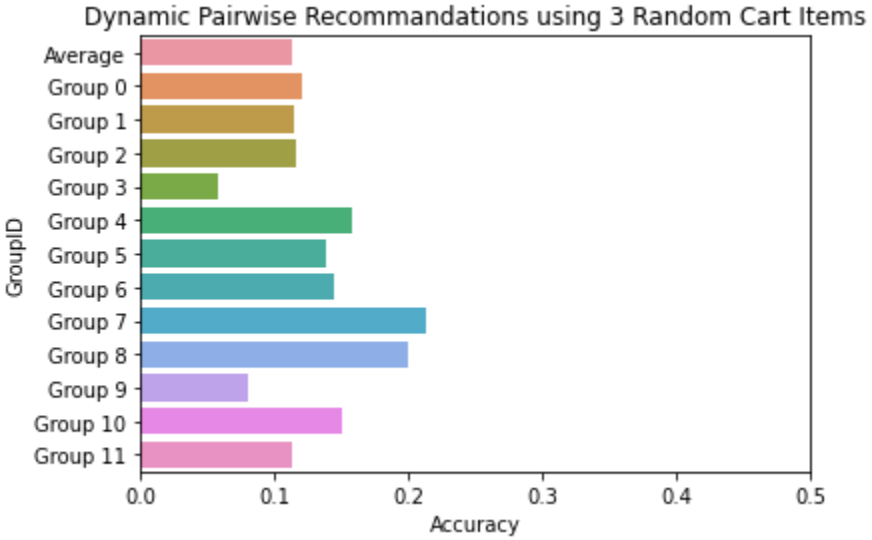


Figure 6: customer groups prediction accuracy with frequent ordered items and top 5 associated pairwise items model

The prediction accuracy of the model using dynamic pairwise recommendation with 3 random carts items is shown in Figure 6. The distribution of the prediction accuracy is more even compared to the previous models, and the average accuracy of this method is 0.12, with highest 0.22 of Group 7 and lowest 0.05 in Group 3.

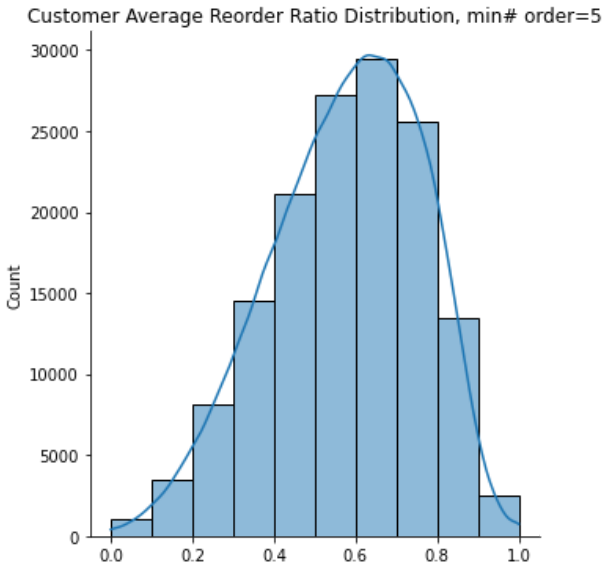


Figure 7: customer average reorder ratio distribution

Figure 7 shows that in average nearly 56% of customers transactions contain recorder items.

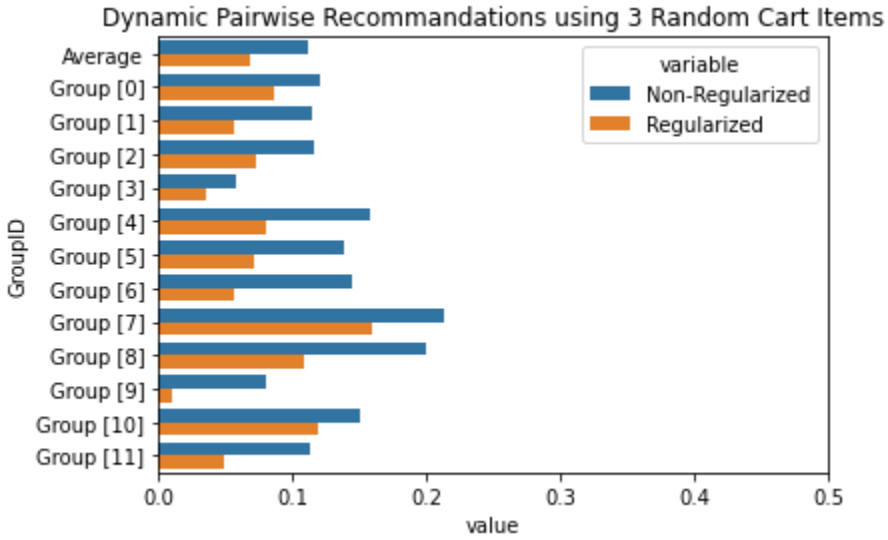


Figure 8: prediction accuracy of dynamic pairwise recommendations model using 3 random cart items

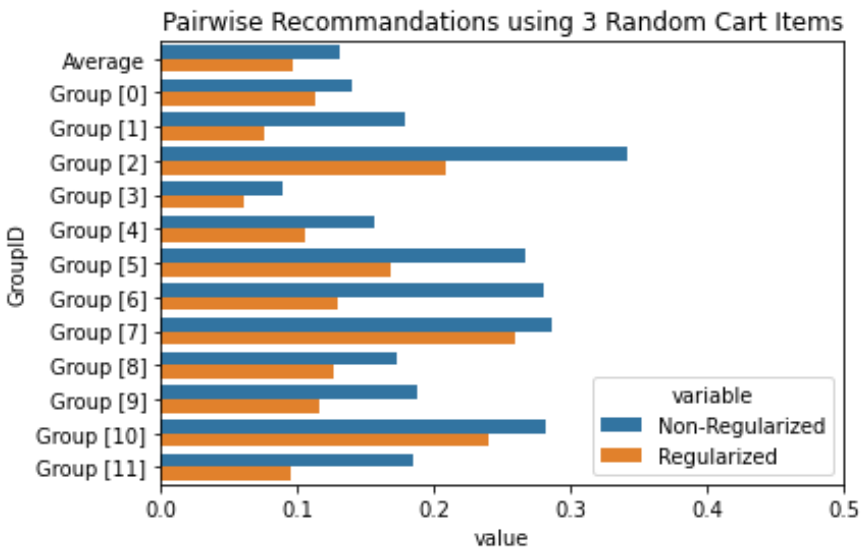


Figure 9: prediction accuracy of pairwise recommendations model using 3 random cart items

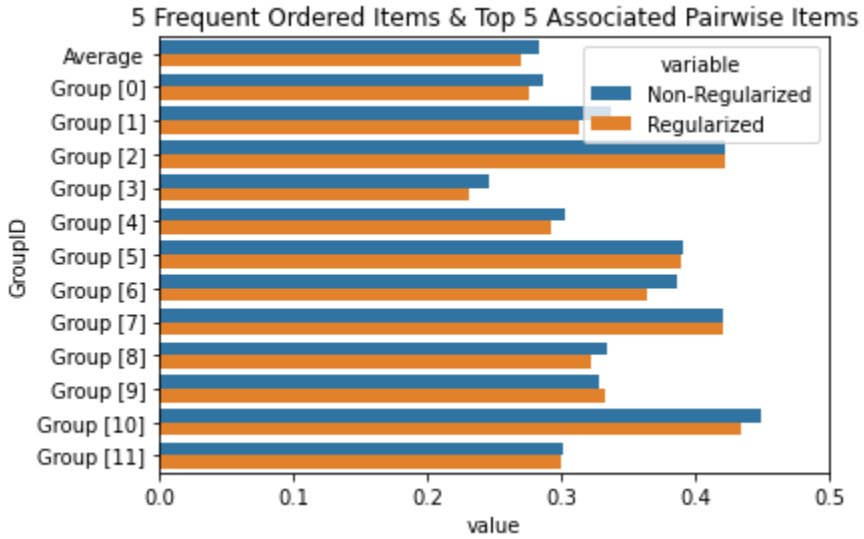


Figure 10: prediction accuracy of combination model using 5 frequent ordered items and top 5 associated pairwise items

In Figure 8,9,10, the prediction accuracy of non-regularized and regularized models have shown. In general, the non-regularized models have higher prediction accuracy compared to regularized models; while in the method using 5 frequent ordered items and top 5 associated pairwise items, the prediction accuracy is very similar in both non-regularized and regularized models. The model in Figure 9 shows more fluctuation in prediction score compared to the rest.

Discussion

After spending more time looking into the case, we have made some changes to our approaches which are different from what we initially planned in the project proposal. We found out that we are not going to use ISOMAP and Spectral Clustering which require the distance matrix, considering that we are having a fairly large dataset and limited computation resources. Also, Logistic regression is not going to be used since we have more than one predicted results in our model.

We are inspired by the Amazon's item-to-item (ItemCF) method, and we have decided to use it as the foundation of the rest of our works. Even though we are focusing on finding the associations between products, we still incorporate some user-to-user (UserCF) ideas by clustering users based on their purchasing behaviors, then generate corresponding item-item association rules for each cluster, which are used when making recommendations for the customers based on their cluster group.

We have come up with 3 recommendation methods and compare the prediction accuracy using each method. By far, the combination of user historical purchases and top 5 associated items method gives us the best result, in figure 5, the method results in an average accuracy rate of 26%. According to figure 7, we have noticed that most of the customers are having fixed purchasing behavior toward their orders. On average, each new transaction contains 57.56% of reordered products. Despite the higher accuracy rate, using frequent purchase products for recommandations is very limited to expose novel products to the customers. In contrast, even though the dynamic approach results in lower accuracy rate, it is likely to allow customers to browse less popular but associated products.

In order to penalize popular products, we have also tested the models using regularized association rules. In both Figure 8 and 9, there are significant differences between non-regularized and regularized method accuracies on both pairwise recommendation and dynamic pairwise recommendation; the non-regularized methods have much higher prediction accuracies. All recommendations for these 2 methods are based on the products with highest association scores, by adding penalized terms, the scores for popular items decrease and scores for less-popular items increase. It is very possible that most of the instacart customers tend to purchase popular items, which leads to lower accuracy for the regularized methods. However, as we have mentioned above, lower accuracies are not necessarily bad, for a good recommendation system, the drive in sales for less-popular products could be one of the goals for better coverage.

There is another limitation on the data we used to make product recommendations. The Instacart dataset contains only past orders. Our prediction models are heavily based on customers' purchasing behavior, and we also only use the most recent purchase to test our models accuracy. By doing this way, our models are only evaluated against customers' past purchasing habits, and our recommendations are not executed to see how customers get intervened after recommendations as well as the changes from their next new purchases. Therefore, low model accuracy doesn't necessarily mean a bad model. In the real life scenario, we believe that recommendation methods should be tested using A/B testing with real time customer purchases, in this way we will have a better understanding of the method performances because customers orders are likely to be impacted by the given recommendations generated by testing methods.

Future approach

Currently, we have introduced 3 methods and every method has its advantages and disadvantages. Similar approaches such as the boosting method could be used to merge the recommendations, where we have each method as a weak learner and find out optimized weight for each after a number of iterations.

One good use case of our methods is recommending similar products when the recommended product is out of stock, we could also use associated rules to find a similar product for replacement. We could also use the dynamic recommendation method to generate an infinite number of recommendations upon customers request. For example, a customer can keep scrolling or swiping to see more associated products. There are customer clusters with poor accuracy scores, especially group 3, who have lowest prediction rates compared to the rest of groups for every method. Further investigations are needed to be conducted. Due to high computation overhead, we only use K-means clustering algorithm to classify customers into groups. In the future, we can try other different clustering methods which may improve model accuracy.

Collaboration

Data preprocessing: all
Models research: all
Models implementation: Zhiheng Dong, Xiaolong Wei
Final report: all

Reference:

[1] URL: https://www.kaggle.com/c/instacart-market-basket-analysis/overview
[2] Yao Xie, “Computational Data analysis” (class lecture, ISYE 6740, Georgia Institute of Technology, Atlanta, May 1st, 2021)
[3] "Collaborative recommendations using item-to-item similarity mappings"

Appendix A:

Topic Modeling inference

K-means algorithm: [2]

input: m data points denoted $\{x^1, x^2, \ldots, x^m\} \in R^n$
output: k cluster centers denoted $\{\mu^1, \mu^2, \ldots, \mu^k\} \in R^n$,
assignment $r^{ij} \in \{1,0\}$ which assign each data point x^i to one cluster μ^j, denoted $\{r^{11}, \ldots, r^{mk}\}$

(Note that we will first initialize j cluster center, $\{\mu^1, \mu^2, \ldots, \mu^j\}$, randomly)

DO Step 1: assign each data point x^i to one cluster μ^j, which is subject to:
$r^{ij} = 1$, if $j = argmin^k ||x^i - \mu^k||^2$ (Note that j gives the minimum value of $||x^i - \mu^k||^2$)
$r^{ij} = 0$, Otherwise
(Note that each assignment r^{ij} will assign each data point x^i to its closest cluster μ^j, having r^{ij} = 1)

DO Step 2: adjust cluster centers μ, which is subject to: $\mu^j = \frac{\sum_{i=1}^{m} r^{ij} x^i}{\sum_{i=1}^{m} r^{ij}}$ for each cluster.

(Note that centroid adjustments will minimize the distortion function $J = \sum_{i=1}^{m} \sum_{j=1}^{k} r^{ij} ||x^i - \mu^j||^2$)

WHILE old centroids μ' is not the same as the new centroids μ after adjustments
Repeat Step 1 and Step 2

RETURN k cluster centers: $\{\mu^1, \mu^2, \ldots, \mu^k\}$,
m data point assignments: $\{r^{11}, \ldots, r^{mk}\}$

*** To compute $\mu^j = \frac{\sum_{i=1}^{m} r^{ij} x^i}{\sum_{i=1}^{m} r^{ij}}$, we take the partial derivative of $J = \sum_{i=1}^{m} \sum_{j=1}^{k} r^{ij} ||x^i - \mu^j||^2$ with respect to μ^j, with r^{ij} fixed for each cluster.

$\frac{\partial J}{\partial \mu^j} = 2 \sum_{i=1}^{m} r^{ij}(x^i - \mu^j)$.

$2 \sum_{i=1}^{m} r^{ij}(x^i - \mu^j) = 0$ ===> $\sum_{i=1}^{m} r^{ij} x^i - r^{ij} \mu^j = 0$ ===> $\sum_{i=1}^{m} r^{ij} \mu^j = \sum_{i=1}^{m} r^{ij} x^i$ ===> $\mu^j = \frac{\sum_{i=1}^{m} r^{ij} x^i}{\sum_{i=1}^{m} r^{ij}}$

Table 1: Result of Top items for each group

Group 0 :
[0.60000] Premium Solid White Albacore Tuna in Water
[0.50000] Distilled Drinking Water
[0.42857] Sourdough Round Loaf
[0.33333] Sun Dried Tomatoes in Oil
[0.31481] Variety Diet Tea
[0.28571] Diet
[0.25806] Spearmint Gum
[0.25000] Sweet Mint Gum
[0.25000] Corn Snack, Hot Chili Pepper & Lime, Takis
[0.25000] Shrimp & Angel Hair Pasta

Group 1 :
[0.60000] Double Strength Energy Drink
[0.50000] Beef Jerky
[0.33633] Coffee Mate French Vanilla Creamer Packets
[0.31316] Sport Bottle with Flip Cap Natural Spring Water
[0.30000] Chunky Tomato Garlic & Onion Pasta Sauce
[0.25275] Straightshot Probiotic Oat Drink
[0.25000] Twice Baked Potatoes
[0.24242] Original Coffee
[0.23529] Eureka Lemon & Marionberries Ice Cream
[0.23529] Tonics Cleansing Probiotic Tonic With Apple Cider Vinegar Cinnamon

Group 2 :
[0.68750] Organic Chocolate 1% Milk with DHA Omega-3
[0.51163] Diet Dr Pepper Bottles
[0.46667] Mango
[0.42857] Bottled Water
[0.40000] Plain Bagels
[0.40000] Original Beef Jerky
[0.40000] Original Hawaiian Sweet Rolls
[0.38622] Natural Spring Water
[0.37500] Fat Free Vanilla Yogurt
[0.33333] Diet Cherry Coke

Group 3 :
[0.37500] Dry Ice
[0.34625] Organic Raspberry Black Tea
[0.21429] Medium Thick and Chunky Fiesta Salsa
[0.21242] Blend of 12 Aged Rums

[0.20880] Sparkling Wild Berry Calorie Reducing Drink
[0.20000] Genmai Miso Aged and Fermented Soy and Brown Rice
[0.18227] Red Label Scotch
[0.18170] Fresh Farmed Tilapia Fillet
[0.17781] Water Mineral
[0.17509] Chunky Beef Meaty Ground Dog Food

Group 4 :
[0.55556] Smoked Maple Ham
[0.27273] Dry Ice
[0.25000] Healthy Growth for Puppies Dry Dog Food
[0.21053] Napa Valley Cabernet Sauvignon
[0.20000] Palmiers, Petite
[0.17391] Enfagrow Older Toddler Vanilla Milk Drink Powder
[0.17241] Healthy Indulgence Turkey & Duck Cat Food
[0.15686] Premium Lemon Juice
[0.15385] Triple Sec Liqueur
[0.15000] Dark Chocolate, Fine Extra

Group 5 :
[0.50000] Single Malt Scotch Whiskey
[0.50000] Unflavored Oral Electrolyte Solution
[0.42105] Plain Cultured Goat Milk Kefir
[0.40581] Roasted Salted Pistachios
[0.40000] Organic Whole Chocolate Milk
[0.39048] Cold-pressed, Deliciously Hydrating Watermelon Water
[0.37500] 100% Lactose Free Organic 2% Reduced Fat Milk
[0.37500] Coconut Water Kefir
[0.37500] Super Cleanse Tablets
[0.35088] Organic Homogenized Whole Milk

Group 6 :
[0.80000] Uncured Pastrami
[0.50000] Artesian Sparkling Water
[0.50000] Irish Whiskey Ireland
[0.40000] Fat Free Cream Cheese
[0.40000] Honey BBQ Glazed Chicken Wings
[0.40000] Gum, Sugarfree, Spearmint
[0.35812] Soda
[0.32143] Chocolate Chip Mini Muffins
[0.30000] Chardonnay Sonoma
[0.28571] Fridge Pack Cola

Group 7 :
[0.50000] Original Club Crackers
[0.37500] Seeded Honey Wheat Organic Bread
[0.26848] Laxative Tablets
[0.25000] Reposado
[0.25000] 100% Pure Coconut Water
[0.25000] Organic Professor Fizz Zero Calorie Soda
[0.25000] Raspberry Truffle Nutrition Bar
[0.25000] Just Green Unsweetened Tea
[0.22291] SmartBlend Lamb & Rice Formula Adult Dog Food
[0.22222] William Fevre Chablis Champs Royaux France

Group 8 :
[0.22222] Cranberry Grape 100% Juice Blend
[0.15789] Premium Dog Food Turkey & Chicken Formula
[0.15789] Chicken Formula Dog Food
[0.15385] Stone Go To IPA
[0.15385] Swaddlers Diapers Jumbo Pack Size Newborn
[0.14894] Diet Dr Pepper Bottles
[0.14286] Pinot Gris
[0.13462] Uncrustables Reduced Sugar Peanut Butter & Grape Soft Bread Sandwiches
[0.13384] Top Seed Bread
[0.13333] Sushi Spicy Tuna Roll Ready To Eat

Group 9 :
[0.66667] French Roast Whole Bean Coffee
[0.60000] Toasted Coconut Chips
[0.50000] Sliced American Cheese
[0.46667] Fat Free Skim Milk
[0.40000] Soft & Smooth Made with Whole Grain White Twin Pack Bread
[0.40000] Carb Balance Flour Tortillas
[0.33333] Uncrustables Peanut Butter & Grape Jelly Sandwich
[0.33333] Double Strength Energy Drink
[0.33333] Alkalized Water
[0.31821] Sparkling Water

Group 10 :
[0.52632] Unsweetened Almond Milk
[0.50000] Organic Whipping Cream
[0.42857] Milk Chocolate Minatures
[0.31579] Vanilla Soy Milk

[0.28571] Real Aged Cheddar Macaroni & Cheese
[0.28571] Chardonnay California
[0.28571] Life Water Zero Calorie, Variety Pack
[0.26087] Gluten Free Plain Bagels
[0.25000] Organic Breakfast Blend Coffee
[0.23684] Original Soy Milk

Group 11 :
[0.60000] Traditional Caramel Filled Amsterdam Waffle
[0.50000] Chocolate Chip Gelato
[0.43750] Pure Mint With Herbal Accent Sugar Free Gum
[0.41176] Natural Chicken & Apple Breakfast Sausage Patty
[0.40000] Omeprazole Acid Reducer Tablets
[0.38739] Spring Water
[0.37500] Ruby Red Grapefruit Juice
[0.33333] Salted Sweet Cream Butter Quarters
[0.33333] Anti Viral Upright Facial Tissue
[0.32692] Diet Coke Soda

Table 2: Dynamic Pairwise Recommandations using 3 Random Cart Items

GroupID	Non-Regularized	Regularized
Average	0.11234	0.06894
Group [0]	0.12079	0.087
Group [1]	0.11448	0.05623
Group [2]	0.11667	0.07333
Group [3]	0.05818	0.03489
Group [4]	0.15741	0.0808
Group [5]	0.13909	0.07102
Group [6]	0.14415	0.05655
Group [7]	0.21284	0.15881
Group [8]	0.19923	0.10831
Group [9]	0.08028	0.01042
Group [10]	0.15074	0.11879
Group [11]	0.11259	0.04911

Table 3: Pairwise Recommandations using 3 Random Cart Items

GroupID	Non-Regularized	Regularized
Average	0.13042	0.09612
Group [0]	0.13939	0.11391
Group [1]	0.17900	0.07548
Group [2]	0.34167	0.20833
Group [3]	0.08947	0.06038
Group [4]	0.15604	0.10620
Group [5]	0.26641	0.16894
Group [6]	0.28022	0.12917
Group [7]	0.28596	0.25898
Group [8]	0.17297	0.12638
Group [9]	0.18750	0.11562
Group [10]	0.28139	0.23958
Group [11]	0.18495	0.09558

Table 4: 5 Frequent Ordered Items & Top 5 Associated Pairwise Items

GroupID	Non-Regularized	Regularized
Average	0.28293	0.27059
Group [0]	0.28616	0.27610
Group [1]	0.33667	0.31267
Group [2]	0.42286	0.42286
Group [3]	0.24632	0.23144
Group [4]	0.30279	0.29168
Group [5]	0.39038	0.39020
Group [6]	0.38585	0.36461
Group [7]	0.42114	0.42092
Group [8]	0.33483	0.32269
Group [9]	0.32852	0.33328
Group [10]	0.44878	0.43443
Group [11]	0.30126	0.29967