WHATS IN YOUR BASKET?

Abstract

Exploration of the market basket contents from an online grocery store

Raghu Bhaskar

Contents

Introduction

The work presented investigates the application of some popular product recommender algorithms on data collected from an online grocery store. The transaction details in the order include a user identification number, and an order identification number with the contents of the order plus the time and day of the week when order was placed. No dates are included in the data. For obvious reasons, no credit card data is included. The data was originally prepared for a Kaggle competition with the goal of predicting what the users' next item and whole order might include. The work presented here re-purposes the data to provide a recommendation to a user rather than a prediction of the order by a user. This is a different perspective than what the data was intended for. Overall, the ideal data set would have included a transaction date, and a user location to better characterize and exploit seasonal and local preferences.

The project work flow focuses on an order rather than a user to better provide recommendations that are slightly unique. This is a so-called "other customers also looked at" approach. The project also aims to target users to focus a promotional campaign on.

Business Problem

The business problem to be tackled is the selection of a product recommender for a hypothetical client who owns an online grocery store. Product recommenders are essential to increase revenue for online shopping portals. Recommenders direct and interact with customers to remind them to purchase items they typically purchase, incentivize purchases, or move overstocked products.

The term 'best' will be applied in two contexts. First, the 'best' means the recommender that most often correctly predicts the contents of a customer's basket. Second, the best recommender will offer the greatest number of opportunities to issue appropriate promotional offers for our hypothetical client.

Data

List of Files: Departments.csv(1KB), Aisles.csv(3KB), Order_products_Prior.csv(560MB), Orders.csv(1206MB), Products.csv(2MB)

Departments and Aisles data was only used in data exploration and visualization. Departments and aisles are organization aggregators and do not serve a real purpose in making recommendations, so they were not used in our analysis.

Products:

The file contains 49,687 products with associated aisles and department. Product names are awkward because they are often a combination of the product and the product description. For example, product ID 2034 is named "Kettle Cooked 40% Less Fat Original Potato Chips", and product ID 2508 is named "The Complete Cookie White Chocolate Macadamia".
Product names were one of the more challenging aspects of the project and required extensive Feature Engineering. Consider the products mentioned above, there is a need to distill the product names down to potato chips, and cookies respectively.

Order_Products_prior:

The Order_products_prior file contain four data columns: order_id (integer), product_id (integer), add_to_cart_order (integer), and reordered (binary). The _prior file holds 3.2M rows of data. Order_id is the record identifier for a sale. Product_id is the identifier for the product, add_to_cart_order details the sequence the products were added to the order_id, and reordered is a flag indicating whether the product was added to the customers cart in the past. This file relates the order, or cart contents, to the products.

Orders:

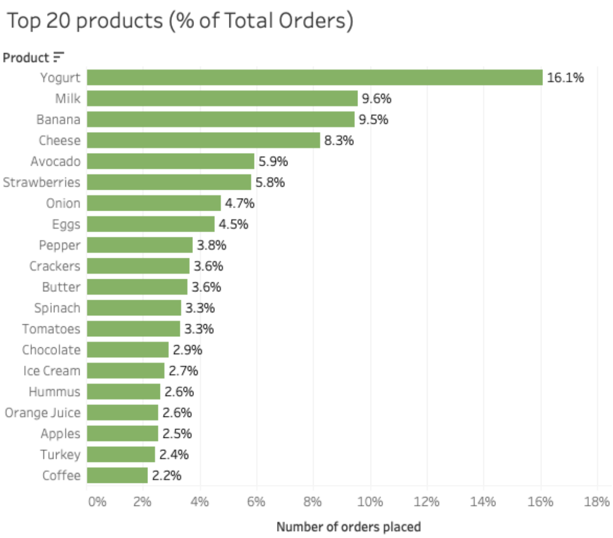
The orders file relates the order or cart contents to the customer. The file contains separate columns for order_id, user_id, eval_set(indicating whether the order falls in the training or test set , order_number (meaning the nth order of the customer), order_dow(indicating the day of the week when the order was placed, days_Since_prior_order(indicating the number of days since the last order

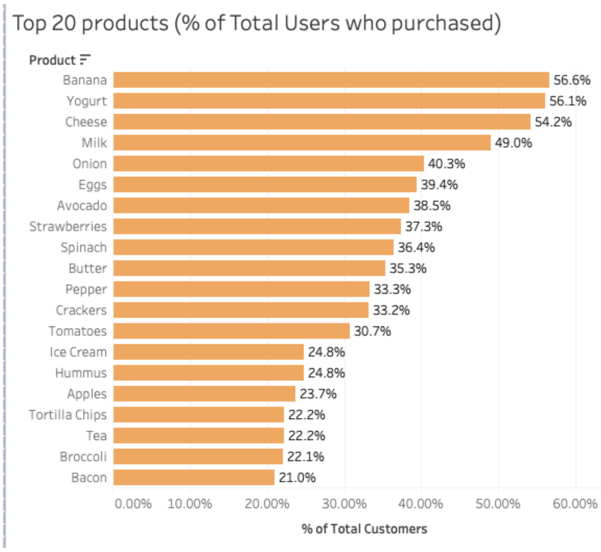
Data Analysis and Visualizations

Preliminary data analysis is an essential component of any data science driven project. We used tableau to perform the preliminary analysis and found some interesting results described below

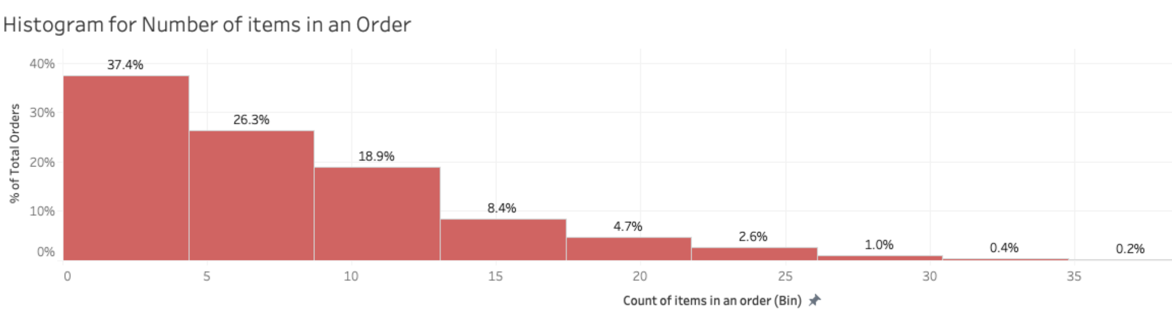
The bar chart on the left shows the ranking of product purchases by count for overall orders. The statistics do not account for product volume, just the presence of the product. Yogurt is, by a wide margin, the product that most commonly appears in an order. Milk and bananas are the second most common. The chart on the right ranks the proportion of users that had a specific product in their order. This is a slightly different perspective than the chart on the left. Ranking by order is a higher resolution statistic than by user. A user may include yogurt in one order, but not the next. In comparison, the product ranked order chart on the left measures each order.

Note that there is little difference in the top 5 items on both charts. Both top five include the same products. This commonality dilutes the predictive insights that can be drawn. 56% of all users bought bananas. Leveraging the presence of a banana in an order is no better than guessing. The real discriminating products are ranked quite low.

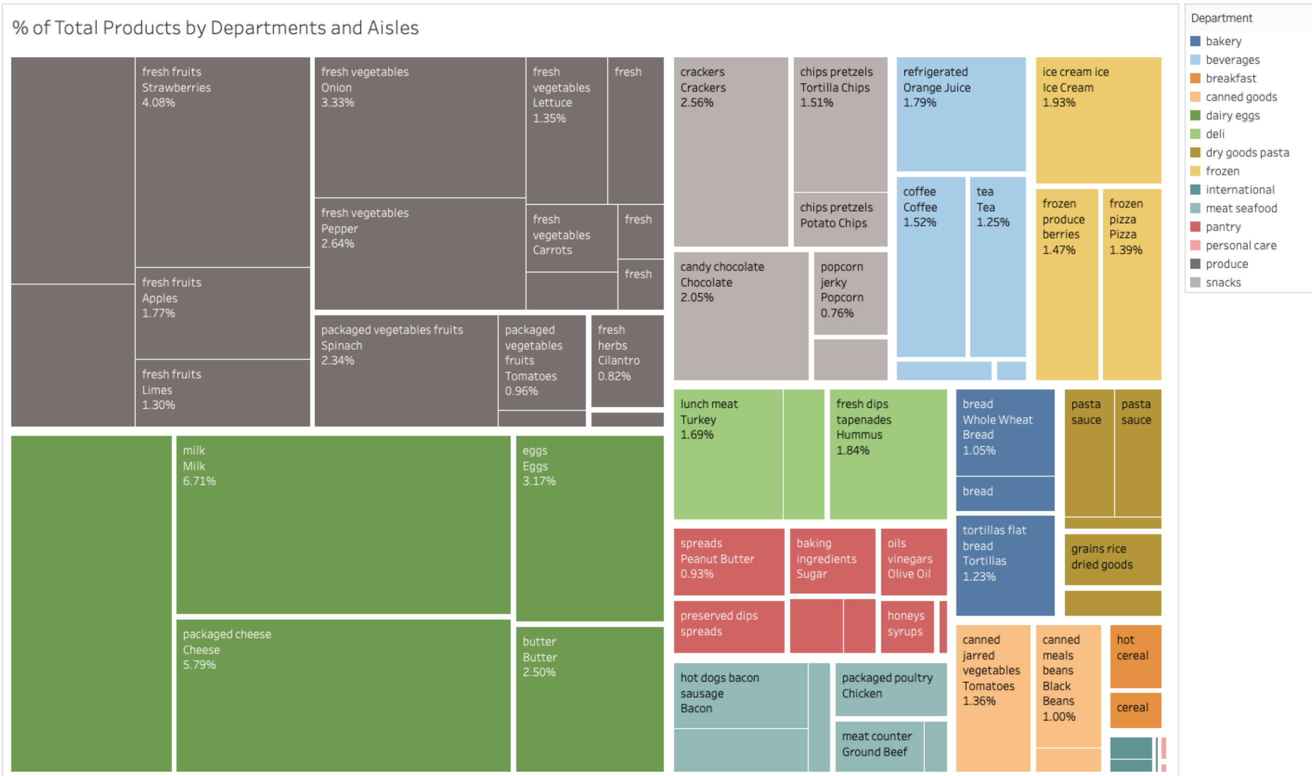




The histogram below shows the distribution of the number of items per order. Almost all the orders had less than 35 items and 37.4% of all the orders comprised of 5 or fewer items, whereas less than 5% of the orders had more than 20 items. The distribution for the orders is log-normal shaped.



The Tree map below shows the distribution of different types of products sold when sorted according to the departments and aisles. The area in dark grey representing the produce section and the area in green representing the dairy section cover the majority of the area of the tree map This means that more than 50% of the total products ordered were from Produce and Dairy section. On the other hand the minimum area of the tree map is occupied by the tiny strips on the bottom right showing that Least number of total products ordered were from personal care section.



Data Preparation

Due to the relatively large size of the dataset, we limited the analysis to 1000 random customers and their related orders (15,866 orders in total).

There were approximately 50,000 individual products in the dataset. Despite limiting the number of customers to 1000 and approximately 16,000 orders, we were concerned that many products within an order would not have exact matches with other orders. We decided that it would be prudent to select a list of items (some popular and some not so popular) from various department and group them. Product groupings (74 products in total) were restricted to a similar product names in the same aisle such that apples from produce were not grouped with apple jack cereal in the cereal aisle. For example, our grouped product "Apples" contained 72 different types of Apple product names (Fuji Apples, Organic Fuji Apples, Pink Lady Apples, etc.).

	aisle_id	department_id	product_id	product_name	department	aisle	product
39	24	4	3720	Golden Delicious Apples	produce	fresh fruits	Apples
40	24	4	4121	Organic Red Delicious Apples	produce	fresh fruits	Apples
41	24	4	4741	Bag of Gala Apples	produce	fresh fruits	Apples
42	24	4	26569	Braeburn Apples	produce	fresh fruits	Apples
43	24	4	16094	Fireside Apples	produce	fresh fruits	Apples
44	24	4	19828	Red Delicious Apples	produce	fresh fruits	Apples
45	24	4	17122	Honeycrisp Apples	produce	fresh fruits	Apples

The filtered datasets were joined together into a single data frame with columns user_id, order_id, product, and number_transactions (total of same product within a particular order).

	user_id	order_id	product	number_transactions
	All	All	All	All
1	118845	541292	Chocolate	5
2	118845	541292	Ice Cream	9
3	118845	670256	Ice Cream	7
4	118845	670256	Potato Chips	1
5	118845	956182	Coke	1
6	118845	956182	Peanut Butter	1
7	118845	956182	Tortilla Chips	1

Model Selection

Most of the analysis was done in R Studio using the recommenderlab package, which contains various recommendation algorithms that have been used in our analysis.

Applied Recommenders

IBCF: Item-based collaborative filtering (IBCF) is a recommendation method that looks for similar items. IBCF looks for the items the user has consumed, and It finds items that are similar to the consumed items and recommends new items accordingly.

UBCF: User Based Collaborative Filtering (UBCF) identifies users who are similar to the target user and estimate the desired rating based on the weighted average ratings from these similar users. UBCF uses the K-nearest neighbors' algorithm to find users similar to the target user and predicts the rating the target user will give to the items that k neighbors have ranked.

Association Rules: This is an algorithm that aims to observe frequently occurring patterns, correlations, and associations in transaction dataset. For each user, it identifies an item as highly correlated or associated to another item based on how frequently the two items appear in the same transaction. Based on this, the recommender recommends highly correlated items to that user.

Popular Items CF: Popular Item Collaborative Filtering (PICF) determines the popular items based on the ranking of the items provided by the users or the interaction that users have with the items. Then, it recommends the top popular products to the users.

Random: Random product recommendations were also included as a benchmark for the other algorithms.

Data Inputs

The recommenderlab accepts 2 types of input matrices for modeling:

1. **Real Rating Matrix** consisting of actual user ratings typically a rank ranging from 1 to 5.
2. **Binary Rating Matrix** consisting of NA's and 1's where 1's indicates the purchase of a product.

Real Rating Matrix:

We initially investigated the possibility of performing our analysis on a customer basis in which each customer's orders were consolidated and purchases of each product were summed.

	user_id	Apple Juice	Apples	Avocado	Bacon	Banana	berries	Black Beans	Broccoli	Butter	Cabbage	Carrots	Cheerios
41	8201	NA	1	NA	NA	5	NA	3	NA	NA	NA	NA	NA
42	8266	NA	5	1	NA	4	NA	1	NA	1	1	NA	NA
43	8295	NA	NA	1	NA	2	NA	NA	NA	NA	NA	NA	NA
44	8337	NA	1	NA	NA	2	2	NA	NA	NA	NA	NA	1
45	8746	NA	NA	17	NA	54	2	1	NA	18	NA	2	NA
46	8946	NA	2	NA	3	18	NA	NA	20	NA	NA	NA	NA
47	9021	NA	NA	NA	NA	NA	NA	NA	NA	1	NA	NA	NA
48	9990	NA	2	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
49	10254	NA	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	1
50	10523	NA	NA	NA	2	NA	NA	NA	1	NA	NA	NA	NA
51	10588	NA	NA	NA	NA	2	1	NA	2	NA	NA	NA	NA
52	10712	NA	1	10	NA	5	NA	NA	2	3	NA	1	NA
53	11333	NA	NA	NA	1	1	3	NA	5	5	NA	NA	NA

From the table above, one can see that there is a wide variation in product purchase totals within each order as well as large differences in purchase quantities for a particular product between customers. This is due to the fact that some items are typically purchased in multiples like different flavors of yogurt and some customers visit the site more often than other.

We made several attempts to scale these product totals into something that could be used as pseudo rating system that could be used to indicate interest in a particular product. We concluded that the higher purchase quantity of product does not necessarily put a higher weightage to a customer's preference for the same product.

A similar approach was taken while using individual orders from customers. Although item count at a single order level was considerably lower than at a customer level, there was not much relevant information gained by looking at the purchase quantities of a particular item.

	user_id	order_id	Apple Juice	Apples	Avocado	Bacon	Banana	berries	Black Beans	Broccoli	Butter	Cabbage	Carrots	Cheerios	Cheese	Chicken	Chocolate
1393	22037	[illegible]	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1394	22037	297849	NA	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1395	22037	330904	NA	NA	1	NA	1	NA	NA	NA	NA	NA	NA	NA	3	NA	NA
1396	22037	354323	NA	NA	1	NA	1	NA	NA	NA	NA	NA	NA	NA	3	NA	NA
1397	22037	451412	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	NA	1	NA	NA
1398	22037	546697	NA	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	4	NA	NA
1399	22037	573277	NA	NA	1	NA	1	NA	NA	NA	NA	NA	NA	NA	2	NA	NA
1400	22037	622420	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	1	NA	NA
1401	22037	656902	NA	NA	1	NA	1	NA	NA	NA	NA	NA	NA	NA	1	1	NA
1402	22037	665283	NA	1	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
1403	22037	680075	NA	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	1	NA

Binary Rating Matrix:

On researching for recommendation systems for transaction without rankings, we discovered that using a binary system is a common practice. So, we performed our analysis at a binary level. For every order, any item that was purchased as part of that order was indicated by a value of "1" independent of the quantity purchased.

	user_id	order_id	Apple Juice	Apples	Avocado	Bacon	Banana	berries	Black Beans	Broccoli	Butter	Cabbage	Carrots	Cheerios	Cheese	Chicken
1177	18711	2594163	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	1	1
1178	18711	2789030	NA	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	1	NA
1179	18711	2870074	NA	NA	1	NA	1	NA	NA	NA	NA	NA	NA	NA	1	NA
1180	18711	2893421	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	1	NA
1181	18711	2958307	NA	NA	NA	NA	1	NA	NA	1	NA	NA	1	NA	NA	NA
1182	18711	3209192	NA	NA	1	NA	1	NA	NA	1	NA	NA	1	NA	NA	NA
1183	18711	3328157	NA	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	1
1184	18951	219407	NA	NA	NA	NA	NA	NA	NA	NA	1	NA	NA	NA	1	NA
1185	18951	597316	NA	NA	NA	NA	1	NA	NA	NA	NA	NA	NA	NA	NA	NA

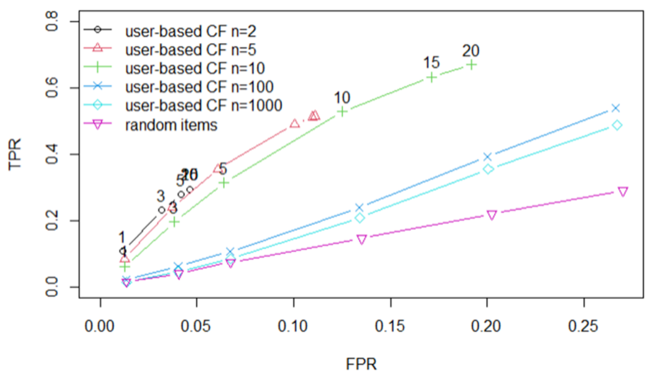
Model Evaluation and Validation

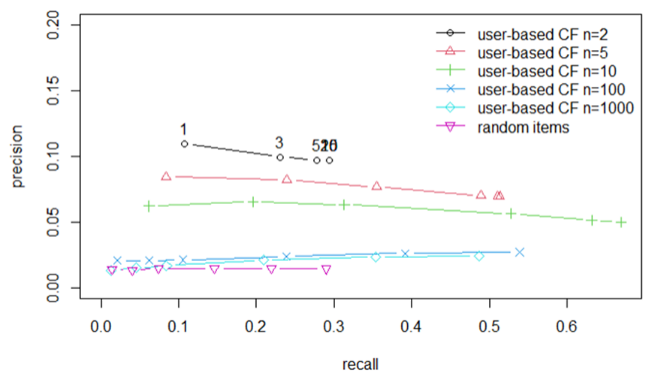
We evaluated several algorithms in the RecommenderLab Library including UBCF, IBCF, Associated Rules, Popular Items, and Random Items. For each of these algorithms we performed an optimization of the input settings prior to performing a final comparison of each of the processes.

We restricted our order dataset to orders with at least 5 items per order and performed a 5-fold (k=5) cross-validation in which the data was partitioned into an 80/20 training/testing split. For each order there was a single random item that was withheld for the evaluation to test the accuracy of the recommendation method.

UBCF Optimization:

For the User Based Collaborative Filtering (UBCF), the optimization process was performed using various values of nn, which is the number of nearest neighbors or similar orders that were used to suggest recommendations for other items. NN values of 2, 5, 10, 100, and 1000 were benchmarked against the Random Items Recommender. The results indicated that as the value of nn increased, the accuracy of the predictor approached that of a random item recommendation. With too many similar orders to compare, the recommendation becomes random. From this exercise, the optimal value for nn was determined to be 5.

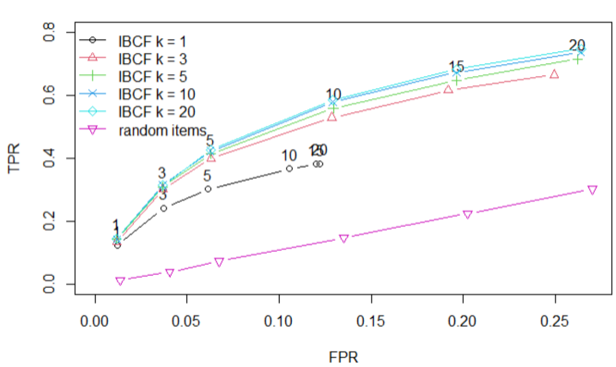


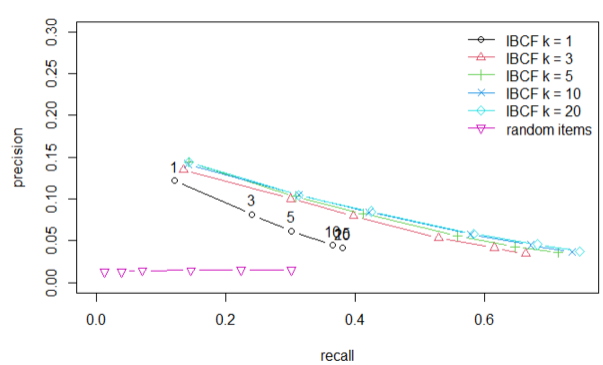


From the tables above, the ROC and the Precision-Recall curves are universally accepted techniques for measuring model performance at select threshold settings. The idea behind the ROC diagram is to plot the rate of False Positive Rate (1-specificity) outcomes versus True Positive Rate (Sensitivity) outcomes at probability of acceptance thresholds ranging from p=0.1 to p=0.9.

IBCF Optimization:

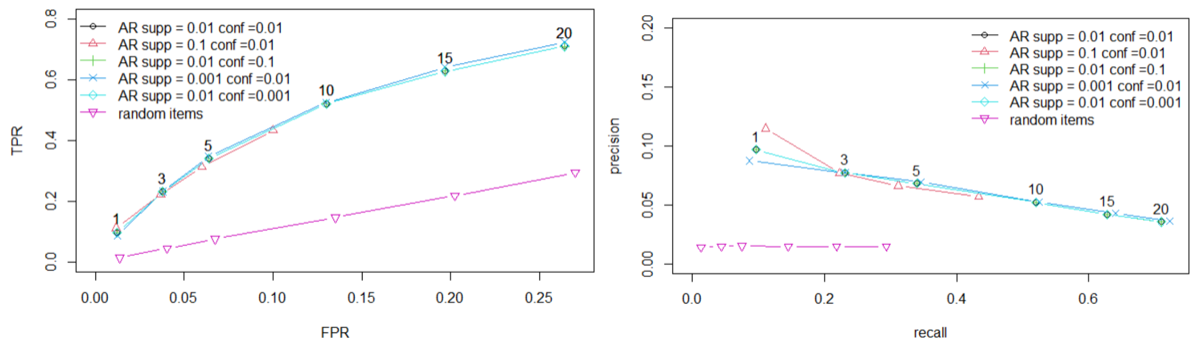
For the Items Based Collaborative Filtering (IBCF), the optimization process was performed using various values of k, which is the number of nearest similar items that were used to suggest recommendations for other items. From this exercise, the optimal value for k was determined to be 5 as there appeared to be no significant improvement in the ROC curves for k = 10 or 20.





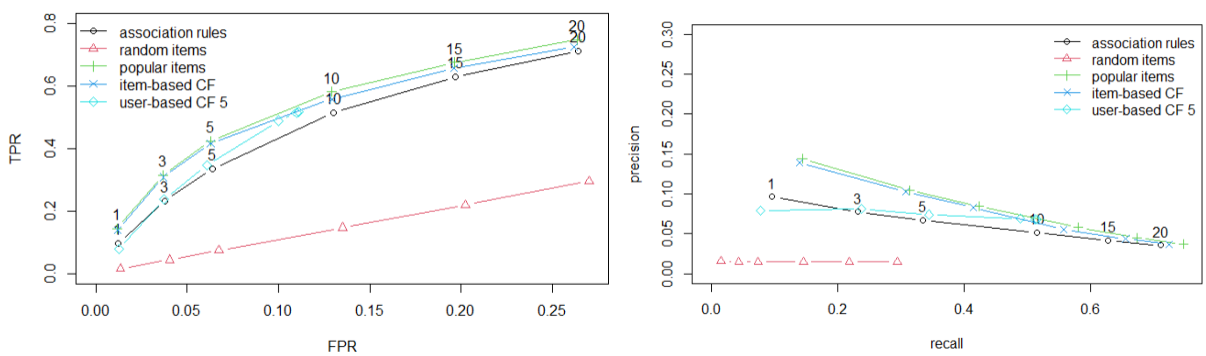
Associated Rules Optimization

For the Collaborative Filtering based on Associated Rules (AR), the optimization process was performed using various values for the minimum confidence (conf) and a support (supp) used as a criterion for Associated Rules. Values for confidence and support were adjusted by scale of magnitude in each direction from an original estimate of support = 0.01 and confidence = 0.01. From the analysis of the ROC curve, the accuracy of the recommender did not appear to be very sensitive to changes and the optimal values were kept at their original estimates.



Algorithm Comparison

The two figures show a comparison between the ROC and the Precision-Recall curves for the optimized recommender algorithms as well as a random items recommender. The performance curves show comparable accuracy with slightly better results in predictions in order of popular items, IBCF, UBCF and lastly Association Rules (AR).



All the recommendation algorithms performed significantly better than the random items-based recommender. The Association Rules recommender, which was the worst performing algorithm was also the slowest of all.

Random Orders Comparison

Five recommendations for each algorithm were generated for 3 Test Orders which consisted of 5 random products.
By definition, the Popular item algorithm recommended the same top 5 items for each of the test orders. The Associated Rules (AR) algorithm recommended items which were ranked in the top 10 most popular items. The UBCF and the IBCF algorithms recommended a variety of products that were not necessarily the most popular items. These recommendations appear to be more customized to the products in test orders than the recommendations made by the other systems.

	Source	Item 1	Pop Rank	Item 2	Pop Rank	Item 3	Pop Rank	Item 4	Pop Rank	Item 5	Pop Rank
	Test Order #1	**Olive Oil**	**38**	**Marinara**	**44**	**Apple Juice**	**53**	**Limes**	**26**	**Soy Sauce**	**64**
Recommendations	**UBCF**	Potatoes	56	berries	24	Tea	16	Ice Cream	13	Pepper	9
	IBCF	Pepper	9	Cilantro	28	Cucumbers	49	Hot Dogs	57	Popcorn	33
	Popular	Banana	3	Yogurt	1	Milk	2	Cheese	4	Avocado	5
	AR	Banana	3	Yogurt	1	Cheese	4	Avocado	5	Pepper	9
	Test Order #2	**Pickles**	**66**	**Cilantro**	**28**	**Pizza**	**21**	**Carrots**	**41**	**Cheerios**	**60**
Recommendations	**UBCF**	Pesto	61	Bacon	31	Cheese	4	Yogurt	1		
	IBCF	Limes	26	Pepper	9	Whole Wheat Bread	34	Black Beans	40	Potato Chips	30
	Popular	Banana	3	Yogurt	1	Milk	2	Cheese	4	Avocado	5
	AR	Milk	2	Onion	8	Banana	3	Cheese	4	Yogurt	1
	Test Order #3	**Lettuce**	**27**	**Spaghetti**	**47**	**Apple Juice**	**53**	**Carrots**	**41**	**Bacon**	**31**
Recommendations	**UBCF**	Broccoli	20	Marinara	44	Rice	35	Turkey	32		
	IBCF	Sausage	29	Orange Juice	23	Tomato	46	Marinara	44	Chicken	36
	Popular	Banana	3	Yogurt	1	Milk	2	Cheese	4	Avocado	5
	AR	Onion	8	Milk	2	Banana	3	Yogurt	1	Cheese	4

product	number_transactions	rank
Yogurt	6594	1
Milk	4140	2
Banana	3987	3
Cheese	3945	4
Avocado	2163	5
Eggs	2096	6
Strawberries	1866	7
Onion	1822	8
Pepper	1595	9
Spinach	1523	10
Crackers	1394	11
Butter	1293	12
Ice Cream	1279	13
Tomatoes	1227	14
Chocolate	1126	15
Tea	1090	16
Coffee	1044	17
Apples	970	18
Hummus	966	19
Broccoli	838	20
Pizza	755	21
Tortilla Chips	735	22
Orange Juice	728	23
berries	694	24
Tortillas	689	25
Limes	684	26
Lettuce	634	27
Cilantro	587	28
Sausage	570	29
Potato Chips	556	30

Comparison between UBCF and Popular Recommendations:

Product recommendations were made for each order in our sample data using the UBCF algorithm (nn = 5 & recommendations = 5). In using a nearest neighbor (nn) value of 5, the UBCF algorithm does not always return the specified number of recommendations. This can occur when the nearest neighbor orders have a small number of associated products. For the 13,120 orders, the UBCF made approximately 20,650 recommendations.

The table below to the left compares the rankings of the top UBCF recommended products to the popular ranking for each product. Although the recommender often selected popular items, it did not necessary select items in order of the popular ranking. For example, hummus was the 14th most recommended item although it is the 19th most popular item in our sample data.

A second analysis was performed to see what percentage of recommended products within a customer's order was subsequently purchased by that customer in his other orders. Product recommendations were made for each order in our sample data using the UBCF algorithm (nn = 5 & recommendations = 5). The table below to the right shows sample from that analysis. Overall, 60.3% of the UBCF recommended products for a particular order were eventually bought by that user. This compares well to the popular items recommendations which were purchased 67.0% of the time.

product	n	rec_rank	numb_trans	popl_rank
Banana	1252	1	3987	3
Milk	1210	2	4140	2
Yogurt	1186	3	6594	1
Cheese	1008	4	3945	4
Avocado	906	5	2163	5
Eggs	815	6	2096	6
Strawberries	764	7	1866	7
Onion	650	8	1822	8
Spinach	628	9	1523	10
Pepper	591	10	1595	9
Butter	569	11	1293	12
Crackers	561	12	1394	11
Tomatoes	500	13	1227	14
Hummus	402	14	966	19
Apples	396	15	970	18
Chocolate	393	16	1126	15
Broccoli	389	17	838	20
Coffee	352	18	1044	17
Ice Cream	338	19	1279	13
berries	325	20	694	24
Limes	308	21	684	26
Cilantro	297	22	587	28
Orange Juice	297	22	728	23
Tortillas	289	24	689	25
Lettuce	271	25	634	27
Tortilla Chips	268	26	735	22
Tea	258	27	1090	16
Pizza	256	28	755	21
Bacon	253	29	549	31
Whole Wheat Bread	223	30	480	34
Turkey	219	31	531	32
Rice	217	32	462	35

	order_id	user_id	products_in_current_order	number_predict	number_pred_bought_later	percent_bought	Prnt_TopProd_rec_bought
14	1006557	22358	3	2	2	100.0	100.00
15	1006649	118044	10	2	2	100.0	100.00
16	1007440	72748	5	2	1	50.0	50.00
17	1008814	50105	3	2	1	50.0	50.00
18	1009934	130102	4	1	0	0.0	100.00
19	1010053	127223	3	4	2	50.0	50.00
20	1010222	47599	4	5	4	80.0	100.00
21	1010277	182114	2	1	1	100.0	100.00
22	1010686	89296	3	2	2	100.0	50.00
23	1011308	53596	7	1	0	0.0	100.00
24	1012989	7186	3	1	1	100.0	100.00
25	1013067	188248	13	2	1	50.0	100.00
26	1013555	155042	9	4	3	75.0	100.00
27	1013918	111163	8	5	4	80.0	60.00
28	1013929	168288	9	3	3	100.0	66.67
29	1014494	172970	6	1	1	100.0	100.00
30	1014725	142304	4	2	1	50.0	100.00
31	101487	112822	5	2	1	50.0	50.00
32	1015329	52706	7	4	4	100.0	75.00
33	1015545	145660	2	2	0	0.0	100.00
34	1017158	43684	6	1	0	0.0	100.00
35	101718	60395	3	1	0	0.0	0.00
36	101959	111163	11	4	3	75.0	50.00
37	1019976	119871	13	5	5	100.0	100.00
38	1020750	69110	4	5	2	40.0	60.00
39	1020908	56106	8	5	5	100.0	60.00
40	1021153	127888	2	3	1	33.3	66.67

K-means Clustering

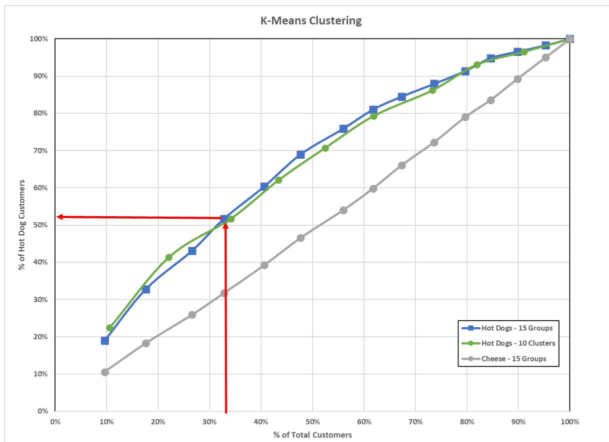
One of the most effective and widely used unsupervised machine learning algorithms is K-means clustering. K-means clustering groups similar data points together and discover underlying patterns and similarities. We grouped 1000 users into 15 clusters or groups based on their purchases. The table below shows the 15 user groups, and rows are ranked by item total sales, most popular items at the top and least popular items at the bottom. The percentage numbers in the cells show the proportion of total sales in each group.

Top Selling

Total Customers	Prod Rank	Product	C4	C1	C3	C10	C12	C13	C11	C15	C2	C5	C7	C14	C6	C8	C9
978		Users	10%	8%	9%	6%	8%	7%	8%	6%	6%	6%	6%	5%	5%	6%	5%
562	1	Banana	10%	7%	9%	7%	9%	9%	7%	5%	6%	5%	6%	4%	5%	6%	5%
548	2	Cheese	11%	8%	8%	6%	7%	7%	7%	6%	6%	6%	7%	5%	6%	6%	5%
517	3	Yogurt	10%	8%	9%	5%	8%	8%	7%	6%	5%	7%	6%	4%	4%	6%	5%
514	4	Milk	10%	9%	10%	6%	8%	6%	9%	4%	5%	6%	6%	5%	5%	6%	4%
449	5	Onion	11%	8%	9%	7%	8%	8%	7%	5%	5%	5%	6%	4%	5%	7%	5%
186	30	Rice	12%	8%	9%	8%	10%	6%	6%	5%	5%	5%	5%	4%	5%	6%	6%
184	31	Peanut.Butter	9%	10%	7%	8%	7%	7%	8%	4%	7%	4%	5%	6%	3%	5%	10%
183	32	Pizza	11%	11%	10%	9%	7%	6%	10%	1%	5%	4%	4%	3%	8%	6%	5%
182	33	Cilantro	12%	8%	9%	6%	8%	7%	9%	5%	4%	5%	4%	4%	6%	5%	7%
177	34	Bacon	12%	8%	7%	9%	10%	7%	6%	6%	2%	6%	5%	5%	5%	6%	5%
177	35	Black.Beans	15%	13%	5%	6%	8%	7%	6%	4%	3%	7%	7%	3%	5%	6%	6%
166	36	Salsa	13%	10%	5%	6%	8%	7%	8%	1%	6%	4%	10%	5%	5%	5%	7%
160	37	Sausage	13%	10%	9%	9%	8%	8%	7%	4%	4%	4%	6%	6%	5%	4%	4%
158	38	Carrots	9%	11%	12%	6%	8%	6%	6%	5%	4%	7%	7%	5%	6%	4%	4%
152	39	Chicken	6%	12%	5%	3%	7%	7%	8%	9%	7%	8%	7%	4%	7%	9%	3%
150	40	Potato.Chips	12%	7%	10%	3%	6%	8%	8%	6%	4%	6%	7%	5%	3%	10%	4%
146	41	Popcorn	8%	7%	11%	4%	5%	11%	10%	6%	9%	4%	6%	5%	5%	5%	3%
138	42	Tomato	9%	7%	7%	9%	10%	6%	9%	5%	5%	5%	7%	4%	5%	9%	4%
136	43	Spaghetti	12%	10%	4%	9%	5%	8%	9%	1%	5%	10%	7%	5%	5%	7%	4%
121	44	Ketchup	11%	6%	7%	11%	7%	7%	9%	2%	5%	7%	5%	7%	5%	5%	7%
113	45	Whole.Wheat	13%	5%	9%	7%	12%	5%	4%	5%	4%	4%	5%	3%	9%	9%	5%
108	46	Mushrooms	11%	5%	12%	6%	6%	6%	6%	6%	3%	8%	6%	6%	6%	4%	8%
104	47	Potatoes	10%	13%	9%	4%	8%	7%	8%	5%	3%	7%	6%	4%	4%	9%	6%
103	48	Oatmeal	13%	5%	10%	10%	6%	6%	7%	7%	3%	7%	11%	6%	2%	5%	5%
101	49	Mayonnaise	8%	13%	11%	9%	10%	5%	5%	6%	7%	3%	4%	4%	5%	6%	5%
100	50	Marinara	10%	9%	6%	9%	9%	10%	10%	4%	3%	7%	4%	5%	4%	7%	3%
93	51	Honey	12%	8%	12%	5%	11%	5%	3%	3%	3%	8%	11%	3%	4%	6%	5%
90	52	Ham	7%	8%	11%	9%	6%	6%	8%	4%	8%	11%	7%	6%	3%	3%	4%
88	53	Cabbage	9%	8%	16%	8%	6%	8%	8%	7%	2%	7%	6%	5%	6%	2%	3%
75	54	Cheerios	11%	11%	9%	5%	5%	7%	7%	5%	3%	7%	7%	4%	1%	11%	8%
66	55	Ground.Beef	8%	20%	5%	6%	17%	5%	12%	5%	5%	3%	5%	5%	2%	3%	3%
64	56	Pesto	11%	17%	9%	6%	9%	8%	6%	3%	3%	3%	9%	5%	2%	5%	3%
58	57	Hot.Dogs	19%	14%	10%	9%	9%	9%	7%	5%	3%	3%	3%	3%	2%	2%	2%
55	58	Apple.Juice	13%	16%	7%	5%	9%	7%	5%	4%	4%	2%	9%	4%	5%	9%	0%
55	59	White.Bread	16%	9%	7%	5%	4%	4%	15%	2%	5%	5%	4%	7%	4%	5%	7%
54	60	Raisins	7%	9%	7%	6%	9%	6%	9%	9%	4%	2%	11%	2%	7%	4%	7%
51	61	Cucumbers	6%	8%	8%	6%	8%	4%	8%	10%	14%	6%	4%	4%	10%	2%	4%
49	62	Soy.Sauce	8%	10%	10%	4%	6%	10%	10%	6%	8%	0%	2%	10%	2%	10%	2%
45	63	Pickles	16%	9%	9%	16%	7%	0%	0%	4%	0%	2%	13%	4%	4%	9%	7%
44	64	Noodles	14%	7%	7%	5%	5%	11%	7%	7%	7%	2%	7%	9%	5%	2%	7%
33	65	Coke	3%	6%	12%	6%	9%	3%	15%	3%	3%	3%	12%	6%	3%	6%	9%
30	66	Refried.Beans	17%	13%	0%	10%	7%	3%	7%	7%	0%	0%	0%	10%	7%	7%	13%
24	67	Steak	13%	25%	13%	4%	4%	4%	4%	0%	13%	13%	4%	0%	4%	0%	0%
17	68	Salmon	0%	6%	12%	0%	12%	0%	6%	6%	18%	0%	18%	0%	6%	0%	18%
17	69	Shav	18%	18%	12%	0%	6%	0%	6%	12%	6%	0%	12%	12%	0%	0%	0%
10	70	Kale	20%	0%	20%	20%	10%	0%	0%	20%	0%	0%	0%	10%	0%	0%	0%
9	71	Ghee	11%	22%	11%	0%	11%	0%	22%	0%	0%	11%	11%	0%	0%	0%	0%
8	72	Tampon	13%	13%	13%	13%	0%	13%	13%	0%	0%	13%	0%	0%	13%	0%	0%

The top 5 selling products varies marginally among the groups, and they are not the main drivers for differentiating the user groups. The red cells indicate where the proportion of total sales is more than 12% in a group. Blue cells indicate where the proportion of total sales are less than 4% in a group. The groups 4 and 1 on right hand side are heavily in favor of buying meat products such as hot dogs and steak. On the other hand, the group 8 and 9 are more in favor of buying vegetables and fruits. Less popular items are the main driver of differentiating these user groups.

K-means clustering provides an effective way of targeting individual groups for a product category. Based on the 15 user groups listed above, Instacart online marketing team probably will not send out meat product coupons to groups that have low percentage of total sales, like group 8 and 9. They will more likely send the coupons to groups that have high percentage of total sales such groups 4, 1, 3 and 10. For example, the graph below shows that marketing team can send hot dog coupons to 33% of total population who buys more than 50% of hot dogs. Customers who did not purchase hot dogs but were included in groups 4, 1, 3 and 10 have similar buying patterns as those who bought hot dogs. These people are more likely to buy hot dogs than customers who were grouped with other customers who purchase higher percentages of healthier items such as salmon and produce. This is much more efficient and cost-effective than sending the coupon to every user.



Conclusions and Recommendations

Ultimately, the project illustrates the importance of appropriate data. For one, the lack of location and timing data limited the specificity of the results. For another, diluting the products down to elemental products, we converted 1%, 2%, and whole milk down to milk for example, also limits the specificity of the results. A key finding in the results is the importance of obtuse products over ubiquitous products. Obtuse products distinguished orders, whereas ubiquitous products had no predictive power because nearly every order contained the product in question.

Our recommendation would be to use the UBCF recommender system: UBCF offered more sophisticated recommendations over the most Popular Items approach, or the Associated Rules approach. A challenge we had was developing reliable relationships between orders and products. We might have had more success using a bagging approach that aggregated all users' orders.

We strongly recommend using a binary recommender system in the absence of reliable rankings data. Where a good reliable ranking system is available ranking data is available it would be preferred over binary. Purchase quantities are not a substitute for a ranking system.

Really understand the data. For example, yogurt was a distracting product. It's sold in small volume containers and often bought in large numbers of small containers. Compare this to almost any produce that is sold by the pound. These products are recorded as a binary, either they were bought or not bought. Ostensibly, a user could have bought 100lbs of potatoes and the data records this as 1 potato, whereas the same user might buy 2 containers of blueberry yogurt, twice as many yogurts as potatoes. The counting of products biased our work and we resorted to a binary system to equalize and normalize the data.

What We Would Do Differently

Grouping items was essential (2% milk = milk), but we went too far and our recommender lost some resolution. A natural language search approach might have been helpful in deciphering products, or perhaps we could have made better use of the isle and department data. It's easy to see where graph theory is helpful.

The original dataset was engineered to provide information on predicting what the user would select next. This limited the predictive utility of the data. It would have been advantageous to have pricing, location, and date data. Making a good recommendation jointly depends on the season, the region, and the location of the user.

References

Alan Flores-Lopez, S. P. (n.d.). *What's for Dinner? Recommendations in Online.* California.

Brownlee, J. (2021, 01 13). *How to Use ROC Curves and Precision-Recall Curves for Classification in Python*. From Machine Learning Mastery: https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-classification-in-python/

Manasson, A. (2019, 12 12). *Why using CRISP-DM will make you a better Data Scientist*. From Case Study: Comparison of Los Angeles and New York Airbnb listings and trends using CRISP-DM: http://webcache.googleusercontent.com/search?q=cache:HtojZMfC7r0J:https://towardsdatascience.com/why-using-crisp-dm-will-make-you-a-better-data-scientist-66efe5b72686&hl=en&gl=ca&strip=1&vwsrc=0

Michael Hahsler, B. V. (2021, 2 26). *Lab for Developing and Testing Recommender Algorithms.* From https://cran.rstudio.com/web/packages/recommenderlab/recommenderlab.pdf

Peterson, M. (2021, 3 11). From Creating a Multifaceted Grocery Recommender System: http://webcache.googleusercontent.com/search?q=cache:Q0M7vUAI0YQJ:https://medium.com/codex/creating-a-multifaceted-grocery-recommender-system-c394208f5e0b&hl=en&gl=ca&strip=1&vwsrc=0

Usai, D. (2019, 3 13). *Clean a complex dataset for modelling with recommendation algorithms*. From towards data science: https://towardsdatascience.com/clean-a-complex-dataset-for-modelling-with-recommendation-algorithms-c977f7ba28b1

Usai, D. (2019, 3 25). *Market Basket Analysis with recommenderlab*. From towards data science: https://towardsdatascience.com/market-basket-analysis-with-recommenderlab-5e8bdc0de236

https://datecheckpro.com/2016/08/18/top-10-grocery-items-in-america/