Market Basket Analysis & Customer Segmentation

AUTHOR

Jungseo Lee (jungseo@umich.edu)

AFFILIATIONS

School of Information, University of Michigan

Future marketing campaigns would not be possible without understanding one's customers. It is important to keep members of the team informed about their customer's purchasing behavior before building their future strategies. A few good ways to do this are, first, finding out interesting frequent itemsets that are purchased together through market basket analysis, and second, dividing customers into a few groups customer segmentation.

01. Introduction

This project was inspired by my previous work experience at Nielsen. I was a part of the Intelligence Analytics team at the Seoul office. The team is mainly focusing on sales of the shelf assortment optimization software. The software analyzes the point-of-sales data from various in-store sales points including big supermarket chains, convenience stores, etc. From this point, I could experience how data is used in the real world and would like to know the backbone of the software so that I can understand how it concludes which shelf assortment is better than the other.

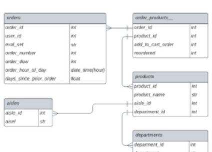
For building future sales or marketing strategies, it is really important to understand how customers are reacting to the current ones. In this project, I have gone through market basket analysis and customer segmentation analysis with Instacart order data to find out any interesting facts which will be beneficial for building future strategies as if I were a part of the data science team that was working with the marketing team at Instacart.

02. Data

Data is from the previous Kaggle competition held by Instacart 5 years ago.

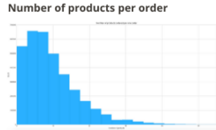
Data merged on 'orders' as the main data that I am focusing on is every order and the items that are purchased through each order.

Data Size
32,640,698



03. EDA

Number of products per order



Most of the order contains less than 10 products.

Number of orders by the hour of the day

The above plot shows the number of orders from the top 15 departments made by the hour of the day.

04. Basket analysis

- **Making unique product set**
- **Data prep to feed the Apriori algorithm**
- **Getting frequent itemset**
- **Finding out interesting combinations with calculating mutual information**

	order_id	user_id	order_hour_of_day	product_id	product_name	department
0	2539329	1	8	196	Soda	beverages
1	2539329	1	8	14084	Organic Unsweetened Vanilla Almond Milk	dairy eggs
2	2539329	1	8	12427	Original Beef Jerky	snacks
3	2539329	1	8	26088	Aged White Cheddar Popcorn	snacks
4	2539329	1	8	26405	XL Pick-A-Size Paper Towel Rolls	household

	product_name	unique_product
5	Soda, Pistachios, Original Beef Jerky, Bag of ...	{Soda, Original Beef Jerky, Bag of Organic Ban...
16	Soda, Original Beef Jerky, Pistachios, Organic...	{Soda, XL Pick-A-Size Paper Towel Rolls, Origi...
29	Soda, Original Beef Jerky, Pistachios, Organic...	{Original Beef Jerky, Soda, Pistachios, Organi...
512	Organic Low Fat Chocolate Milk, Organic Jasmin...	{Organic Blueberries, Mexican Coffee, Seedless...
624	Organic Baby Spinach, Michigan Organic Kale, B...	{Low Sodium Beef Broth, Organic Dried Porcini ...

	Apricot & Banana Stage 2 Baby Food	Free and Clear	#2 Coffee Filters	#4 Natural Brown Coffee Filters	& Baby Wipes	& Cheese Biscuit	& Cheese English Muffin	& Cheese Sandwiches	& Cheese Sauce	& Cheese Scramble	...	with Key-Lock Closures	with Magnesium	with Natural Pulp	with Oli O Mayonnais
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	[illegible]
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	[illegible]
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	[illegible]
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	[illegible]
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	[illegible]
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	[illegible]
9995	0	0	0	0	0	0	0	0	0	0	...	0	0	0	[illegible]
9996	0	0	0	0	0	0	0	0	0	0	...	0	0	0	[illegible]
9997	0	0	0	0	0	0	0	0	0	0	...	0	0	0	[illegible]
9998	0	0	0	0	0	0	0	0	0	0	...	0	0	0	[illegible]
9999	0	0	0	0	0	0	0	0	0	0	...	0	0	0	[illegible]

	support	itemsets
0	0.1304	(Bag of Organic Bananas)
1	0.0970	(Organic Strawberries)
2	0.0824	(Organic Baby Spinach)

min_support = 0.005

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	mutual information
2	(Clementines)	(Bag)	0.0221	0.0166	0.0142	0.642534	38.706864	0.013833	2.751030	0.076845
3	(Bag)	(Clementines)	0.0166	0.0221	0.0142	0.855422	38.706864	0.013833	6.763808	0.076845
4	(Milk)	(Organic)	0.0135	0.0233	0.0080	0.592593	25.433159	0.007685	2.397355	0.032559
5	(Organic)	(Milk)	0.0233	0.0135	0.0080	0.343348	25.433158	0.007685	1.502317	0.032559
6	(Milk)	(Vitamin D)	0.0135	0.0094	0.0094	0.696296	74.074074	0.009273	3.261732	NaN
7	(Vitamin D)	(Milk)	0.0094	0.0135	0.0094	1.000000	74.074074	0.009273	inf	NaN

metric = 'lift', min_threshold = 5

05. Customer segmentation

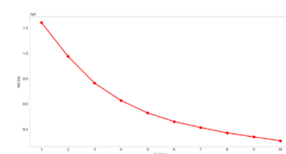
- **Data prep to feed the K-Means algorithm**
- **Dimension reduction**
- **Decide K-value**
- **Visualize the result and evaluate**

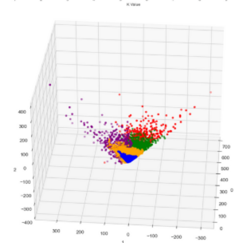
	order_id	user_id	aisle	avg_days_since_prior_order
0	2539329	1	soft drinks	18.466667
1	2539329	1	soy lactosefree	18.466667
2	2539329	1	popcorn jerky	18.466667
3	2539329	1	popcorn jerky	18.466667

```
new_df = pd.crosstab(df['user_id'], df['aisle'])
```

aisle	air fresheners candles	asian foods	baby accessories	baby bath body care	baby food formula	bakery desserts	baking ingredients	baking supplies decor	beauty	beers coolers	...	spreads	tea
user_id													
1	0	0	0	0	0	0	0	0	0	0	...	1	0
2	0	3	0	0	0	0	2	0	0	0	...	3	1
3	0	0	0	0	0	0	0	0	0	0	...	4	1
4	0	0	0	0	0	0	0	0	0	0	...	0	0
5	0	2	0	0	0	0	0	0	0	0	...	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
206205	0	0	1	0	0	0	0	0	0	0	...	0	0
206206	0	4	0	0	0	0	4	1	0	0	...	1	0
206207	0	0	0	0	1	0	0	0	0	0	...	3	4
206208	0	3	0	0	3	0	4	0	0	0	...	5	0
206209	0	1	0	0	0	0	0	0	0	0	...	0	0

206209 rows × 134 columns





06. Conclusion

The project was divided into two different subjects: 1) Basket analysis and 2) Customer segmentation.

For the basket analysis, as there were too many data points and the numbers of orders made by the hour of the day were quite different, data were categorized into 3 groups based on the hour of the day that the order was made. With the Apriori algorithm, I got nearly 400 item sets that supports are bigger than 0.005(0.5%) per category. With getting itemsets that are above the specific threshold lift that I set, I found a few interesting combinations of items from the frequent item sets. The found item sets might be highly dependent on each other. Example of these item sets is milk and vitamin D.

There are a total of 206,209 users in the data. What if I am a member of Instacart marketing team and need to plan out the marketing campaigns that will be beneficial to increase the sales from these customers? First of all, I need to have some information about the customers in a few different groups that are sharing similar characteristics in the group. To decide the number of clusters, I used the within-cluster sum of square metric and plot this in an elbow graph. This shows that 5 clusters might be useful to cluster these customers into groups. Next, I decided to use the number of orders that each customer made from each aisle as there are 134 aisles in the data instead of millions of unique products. However, even though the number of potential features reduced a lot from millions to about a hundred, it might be still a lot to throw them into a clustering algorithm. Therefore, I reduced 134 features into 3 with the SVD algorithm. With this, I successfully clustered all 206,209 customers into 5 different groups. As the silhouette score still has some gaps to improve, further analysis on this matter might be a good idea. I suggest trying different features like department instead of the aisle as it got smaller numbers of unique values. Or it would be a great idea that works with more customer behavior-related data like the average number of days from the previous order.

Online version:
https://www.canva.com/design/DAFUJnE4oI4/dMdwwu0sYxBl05rNwO8GFw/view?utm_content=DAFUJnE4oI4&utm_campaign=designshare&utm_medium=link&utm_source=publishsharelink

Market Basket Analysis and Customer Segmentation with Instacart Order Data

Jungseo Lee (jungseo@umich.edu)

Abstract

This paper presents an Apriori and K-Means clustering algorithm to find out insights into business situations. The project consists of two parts: (1) Market Basket Analysis and (2) Customer Segmentation Analysis. The first half of the project was for finding out interesting product combinations that are likely purchased together. During the second half, I successfully clustered all 206,209 customers into 5 different groups with the SVD and K-Means Clustering algorithm. This aims to divide customers into different groups for building future business strategies like targeted marketing campaigns.

Introduction

Understanding the business's current performance and the customers are critical to pursuing further growth. This project was inspired by my previous work experience at Nielsen. I was a part of the Intelligence Analytics team at the Seoul office. The team is mainly focusing on sales of the shelf assortment optimization software. The software analyzes the point-of-sales data from various in-store sales points including big supermarket chains, convenience stores, etc. From this point, I could get some idea of how data is used in the real world. This experience inspired me to have a better understanding of the backbone of the software which comes with the conclusion that one shelf assortment is better than the other.

Through this project, I would like to build a model that can mine some interesting insights from the data just like the software that I used at the previous workplace. I tried to tackle two main questions with this project: "What are the popular products that are selling together?" and "How should we segmentize our customers?". For building future business strategies, it is important to understand how customers react to the current ones. In this project, I was working with Instacart order data, and trying to think of solutions as if I were a part of the data science team that was working with the marketing team at Instacart. The project consists of two parts 1) Market Basket Analysis and 2) Customer Segmentation Analysis.

Methods

1. Data Source

   Data is coming from the previous Kaggle competition: "Which products will an Instacart consumer purchase again" held by Instacart in 2017. The data contains a sample of over 30 million product orders. The data is divided into 5 different files which can be relationally connected with shared columns. Below is the entity relationship diagram of the files.

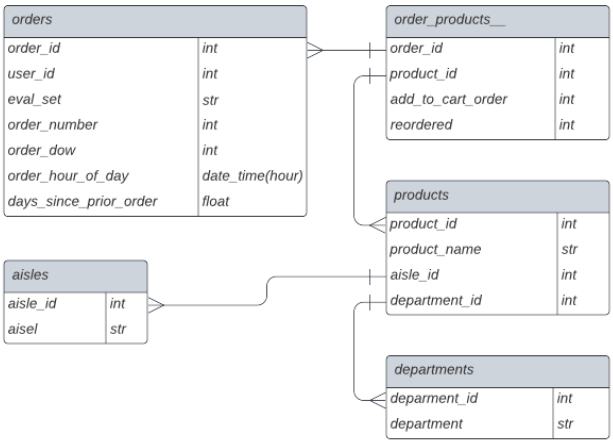


Figure 1. Entity Relationship Diagram

2. Exploratory Data Analysis (EDA)

EDA is a good starting point to do the data analysis. EDA helps not only to grasp some basic information about the data but also to find out unexpected, interesting insights. The base data set is 'orders'. It contains over 30 million rows. Each row represents each product that was ordered by a customer each time.

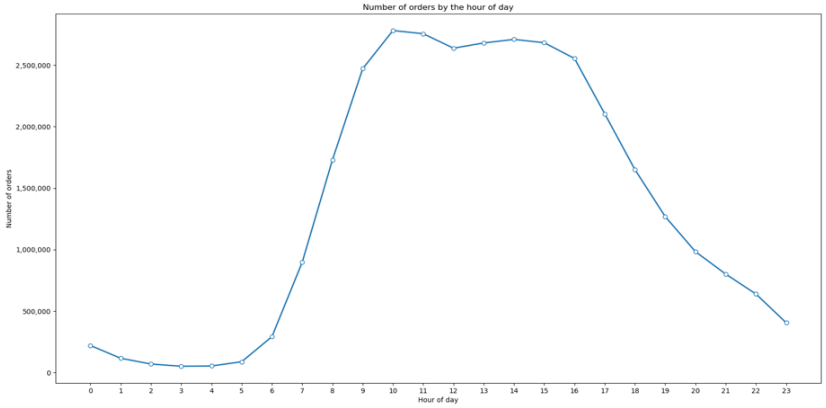


Figure 2. Number of Orders by the Hour of Day

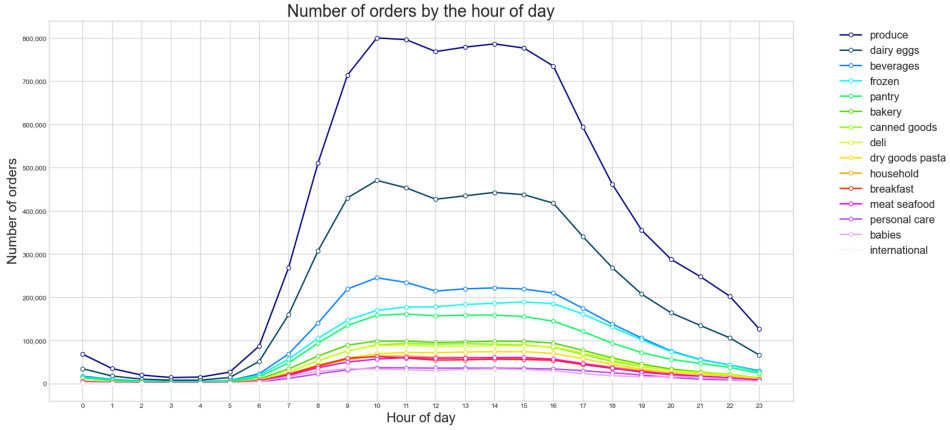


Figure 3. Number of Orders from Each Department by the Hour of Day

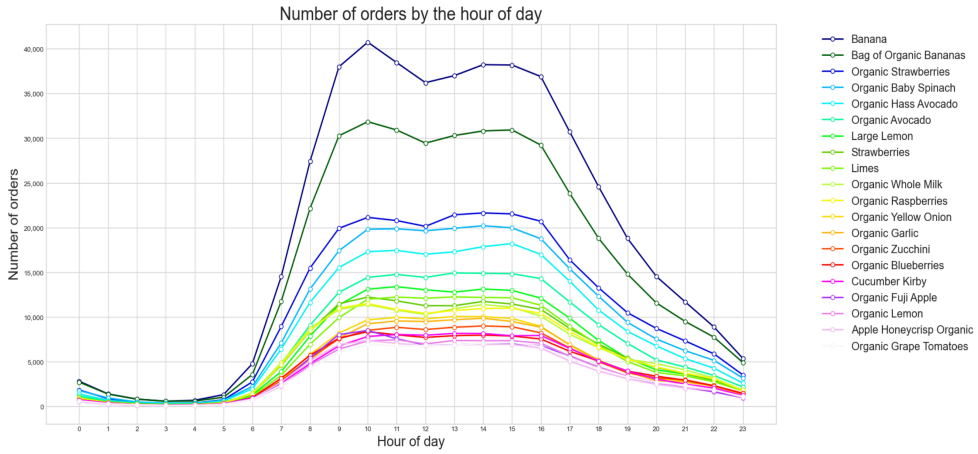


Figure 4. Number of Orders from Each product by the Hour of Day

Based on the plots that were produced during the EDA process, I found that most products are purchased during the daytime (8 a.m. to 4 p.m.) (Figure 2). Most orders are made with 2-8 products per order. "Product, dairy eggs, beverages, frozen, pantry" are the top 5 departments that were ordered the most (Figure 3). "Banana, Bag of Organic Bananas, Organic Strawberries, Organic Baby Spinach, Organic Hass Avocado" are the top 5 products that are ordered the most (Figure 4). There are a total of 3,421,083 orders, 49,677 unique products, and 206,209 users.

3. Data Preprocessing and Data Description for the market basket analysis

As there are two different goals, I needed to go through two different preprocessing works that were applicable to each algorithm. To find out the interesting combinations of the products through the market basket analysis,

1) Drop the data that gets any NaN value
2) Change the data type of each column
3) Concatenate all the product_ids in one column based on the order_id
4) Drop duplicates
5) Make a new column, unique_product, which is a set of the ordered products in an order
6) Make a database, a matrix with every available product in columns with an MLB function

The final data that was fed into the Apriori Algorithm has columns of all unique products and rows of users. The values in the matrix represent the number of times that the user purchased the product.

4. Data Preprocessing and Data Description for the customer segmentation

To segmentize the customers into groups, I needed to have data that has features regarding each customer. In the datasets we have right now, there are a lot of different levels of product categories including product, aisle, and department. As we have too many different products, it might not be useful to utilize the product information by each customer. I would like to build a dataset that shows the number of orders that each customer made from each aisle which might be relevant as it is one level higher category than the product.

1) Use Pandas crosstab to transform the data frame into a matrix that has aisles as columns and each user as a row. The values of the matrix are the number of times that each customer purchases a product from the aisle.

2) Reduce the dimension of the data to 3 components with the TruncatedSVD

5. Model

The Apriori algorithm played a key role to get the frequent items out of the data. The algorithm identifies the frequent individual items in the data and extends it to larger item sets that appear more often than the fixed threshold.

For the second half of the project, customer segmentation, Singular Vector Decomposition (SVD), and the K-Means clustering algorithm were the two main algorithms that derived the result.

Evaluation and Analysis

Even though the Apriori algorithm returns all the item combinations that happened more times than the threshold, it doesn't imply that all the item combinations are interesting. The below data frame (Table 1) is one of the result data frames that I got with a part of the data and an Apriori algorithm. For example, 'Bag of Organic Bananas' returns support of 0.1304. This means a 'Bag of Organic Bananas' appears around 13% of the time in a database. In other words, around 1 out of 8 orders contains the item. However, getting frequent item sets is not enough to come up with a conclusion to the question, "What are the interesting/insightful product combinations that customers are purchasing together?". There, I expanded the analysis by getting association rules and adding mutual information. Association rules are showing the probability of relationships between data items. The association rules package from mlxtend returns the item sets (antecedents and consequents, $X \rightarrow Y$) and the metric values (support, confidence, lift, leverage, and conviction). The association rules will return different results based on the evaluation metric and the threshold chosen.

	support	itemsets
0	0.1304	(Bag of Organic Bananas)
1	0.0970	(Organic Strawberries)
2	0.0824	(Organic Baby Spinach)
3	0.0518	(Organic Whole Milk)
4	0.0331	(Organic Zucchini)
...	...	...
421	0.0063	(Banana, Blueberries)
422	0.0060	(Limes, Organic Cilantro)
423	0.0052	(Bag of Organic Bananas, Organic Cilantro)
424	0.0063	(Banana, Organic Blackberries)
425	0.0051	(Organic Strawberries, Organic Blackberries)

Table 1. Frequent item sets above minimum support of 0.005

As I mentioned earlier, there are 5 different matrices that the association rules can return. Details about each metric can be found below.

1) Support: Typically, support is used to measure the abundance or frequency (often interpreted as significance or importance) of an item set in a database. We refer to an itemset as a "frequent itemset" if your support is larger than a specified minimum-support threshold.

- 'antecedent support': the proportion of transactions that contain the antecedent
- 'consequent support': the proportion of transactions that contain the consequent

- 'support': the proportion of transactions that contain the combined itemset (antecedents or consequents)

$$support(A \rightarrow C) = support(A \cup C), \text{ range: } [0,1]$$

2) Confidence: The probability of seeing the consequent in a transaction given that it also contains the antecedent.

$$confidence(A \rightarrow C) = \frac{support(A \rightarrow C)}{support(A)}, \text{ range: } [0,1]$$

3) Lift: Commonly used to measure how much more often the antecedent and consequent occur together than we would expect if they were statistically independent. (If antecedent and consequent are independent, the Lift score will be exactly 1.)

$$lift(A \rightarrow C) = \frac{confidence(A \rightarrow C)}{support(C)}, \text{ range: } [0,1]$$

4) Leverage: The difference between the observed frequency of antecedent and consequent appearing together and the frequency that would be expected if antecedent and consequent were independent. (0 indicates independence.)

$$leverage(A \rightarrow C) = support(A \rightarrow C) - support(A) \times support(C), \text{ range: } [-1,1]$$

5) Conviction: high conviction value means that the consequent is highly depending on the antecedent.

$$conviction(A \rightarrow C) = \frac{1 - support(C)}{1 - confidence(A \rightarrow C)}, \text{ range: } [0,\infty]$$

To measure how much two item sets, antecedent and consequent, occur together, I used minimum threshold 5 of a 'lift' as a metric in this project. This means the metric will represent how much these two item sets are dependent on each other. Add to this, I added a column 'mutual information' by calculating the mutual information values. Mutual information is a good metric to see how much information is obtained about one of the item sets by observing the other item set. By applying the customized function created to each row, I could come up with a final association rule matrix like below.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	mutual information
0	(Clementines)	(Bag)	0.0239	0.0158	0.0136	0.569038	36.015042	0.013222	2.283726	0.071074
1	(Bag)	(Clementines)	0.0158	0.0239	0.0136	0.860759	36.015042	0.013222	7.010173	0.071074
2	(Milk)	(Organic)	0.0135	0.0245	0.0073	0.540741	22.071051	0.006969	2.124073	0.027499
3	(Organic)	(Milk)	0.0245	0.0135	0.0073	0.297959	22.071051	0.006969	1.405189	0.027499
4	(Milk)	(Vitamin D)	0.0135	0.0080	0.0077	0.570370	71.296296	0.007592	2.308966	0.049977
5	(Vitamin D)	(Milk)	0.0080	0.0135	0.0077	0.962500	71.296296	0.007592	26.306667	0.049977
6	(Vitamin D)	(Organic)	0.0080	0.0245	0.0058	0.725000	29.591837	0.005604	3.547273	0.025363
7	(Organic)	(Vitamin D)	0.0245	0.0080	0.0058	0.236735	29.591837	0.005604	1.299679	0.025363
8	(Milk, Vitamin D)	(Organic)	0.0077	0.0245	0.0058	0.753247	30.744765	0.005611	3.953342	0.025937
9	(Milk, Organic)	(Vitamin D)	0.0073	0.0080	0.0058	0.794521	99.315068	0.005742	4.827733	0.039303

Table 2. Association rule matrix

From this, I found two item sets especially interesting. Those are Clementines – Bag and Vitamin D – Milk. The Clementines – Bag shows relatively high support which is 0.013 and high lift. This means not only the items are selling well separately, but also those two items are selling together in many cases. Another interesting product combination is Vitamin D and Milk. Even though the antecedent and consequent support are not extremely high, they show a high lift. This

shows the high dependency of two items on each other and the tendency of selling together. These two combinations also show high mutual information values. This infers that we can get quite an amount of information about the other variable by observing one of the items from the combination.

I got some insights about the products that are selling well together. If I were a part of the data science team working with a marketing team as I supposed when I started this project, understanding the customers might be the intriguing next step to take. Let's say I need to find a way to divide the users into a few groups for a targeted marketing campaign. The first thing to decide is 'how many groups that all customers divided into?". For this matter, I plotted an elbow graph that shows how the within-cluster sum of squares changes over the changes of the K-value.

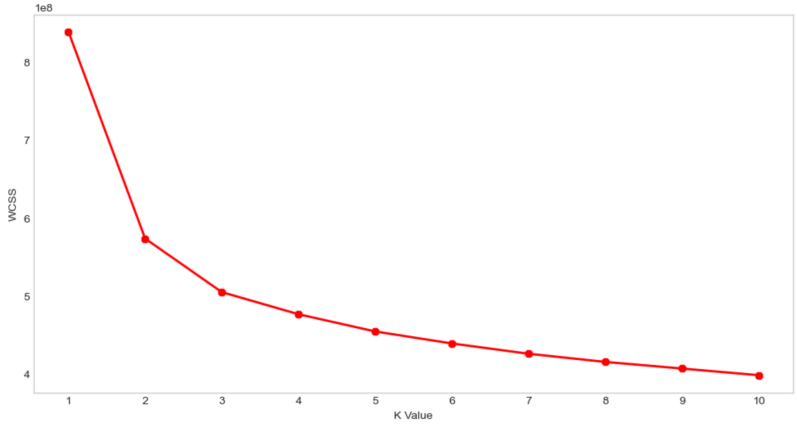


Figure 5. Elbow Graph with an Original Data

The graph starts to move almost parallel to the X-axis from the K-value of 3. This indicates that there is not much benefit to making more clusters as the decreasing proportion of the sum of squares is not as large as making fewer clusters. The optimal number of clusters is 3. Now we need to segmentize the customers into 3 groups. However, the total number of features that we have in the original data set is 134 as we transformed the dataset by the number of orders made by each aisle. This is a very sparse matrix that barely got values in most of the columns and rows. It is very unlikely that clustering algorithms like K-Means work well on high-dimensional data. Therefore, I went through the dimension reduction process first with a Truncated SVD package.

	0	1	2
user_id			
1	5.943374	4.407281	-1.318931
2	36.857383	37.677724	8.096282
3	21.907428	3.847017	-10.174675
4	1.911268	1.241777	-0.844316
5	9.906379	-0.859107	1.448649
...	...	...	...
206205	5.691563	3.246052	2.217840
206206	26.508920	1.932852	7.236908
206207	39.077295	8.984107	8.792154
206208	115.291945	20.022791	8.296676
206209	16.322382	8.365876	-4.201589

Table 3. Encoded SVD Results

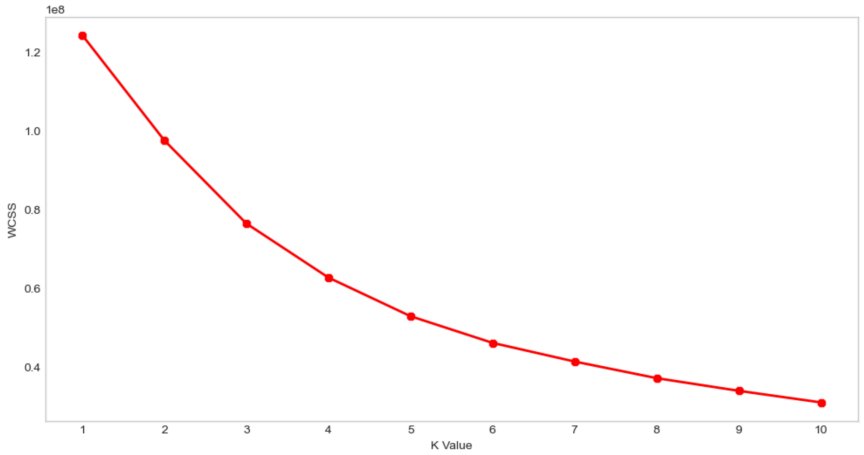


Figure 6. Elbow Graph with Reduced Dimensions

The data with reduced dimensionality returned 5 as an optimal number of K-values. A total of 206,209 users in the data were grouped into 5 different groups with the K-Means clustering algorithm.

	0	1	2	label
user_id				
1	5.943374	4.407281	-1.318931	0
2	36.857383	37.677724	8.096282	3
3	21.907428	3.847017	-10.174675	0
4	1.911268	1.241777	-0.844316	0
5	9.906379	-0.859107	1.448649	0
...	...	...	...	...
206205	5.691563	3.246052	2.217840	0
206206	26.508920	1.932852	7.236908	0
206207	39.077295	8.984107	8.792154	3
206208	115.291945	20.022791	8.296676	2
206209	16.322382	8.365876	-4.201589	0

Table 4. Customer Segmentation Results in Labels

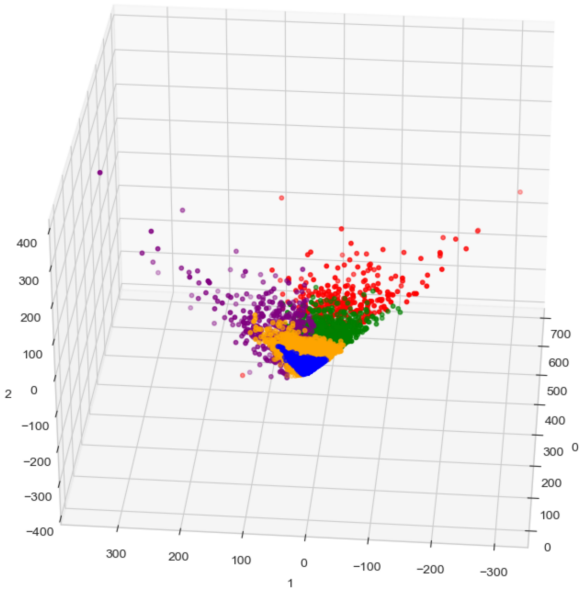


Figure 7. Customer Segmentation Visualization

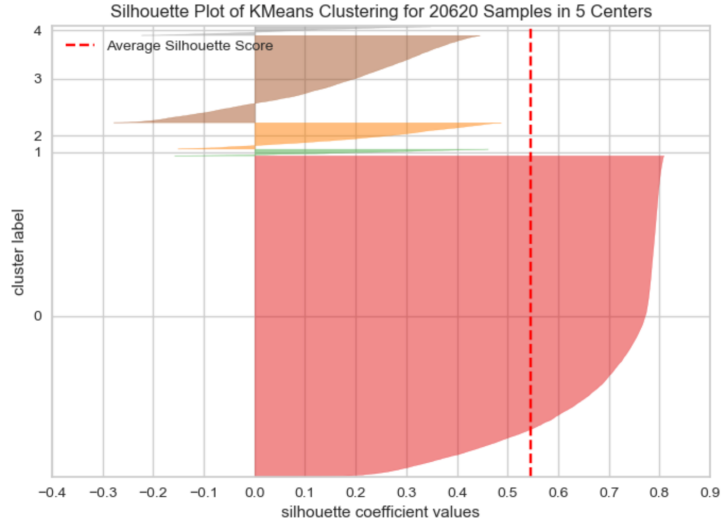


Figure 8. Silhouette Plot of K-Means Clustering

Based on the visualized silhouette score, I found that most of the data points clustered to label 0. The average silhouette score is around 0.54.

Discussion and Conclusion

The project was divided into two different subjects: 1) Basket analysis and 2) Customer segmentation.

For the basket analysis, as there were too many data points and the numbers of orders made by the hour of the day were quite different, data were categorized into 3 groups based on the hour of the day that the order was made. With the Apriori algorithm, I got nearly 400 item sets that have support values bigger than 0.005 (0.5%) per category. After getting item sets that are above the lift value I set, I found a few interesting combinations of items from the frequent item sets. The found item sets infer high dependency on each other. An example of these item sets is milk and vitamin D.

There are a total of 206,209 users in the data. I decided to use the number of orders that each customer made from each aisle as there are 134 aisles in the data instead of millions of unique products. However, even though the number of potential features is heavily reduced from millions to about a hundred, it is still a lot to be thrown into a clustering algorithm. Therefore, I reduced 134 features to 3 with the SVD algorithm. To decide the number of clusters, I used the within-cluster sum of square metric and plot this in an elbow graph. This shows that 5 clusters might be an optimal number to cluster these customers into groups. With this, I successfully clustered all 206,209 customers into 5 different groups.

This research can be continued in several directions to address its limitations.

1) The efficiency of the Apriori algorithm

   The Apriori algorithm, not surprisingly, suffers from a bad reputation for efficiency. Due to the structure of the algorithm, it needs to generate a large number of subsets and check if the subsets are occurring more often than the given threshold. As the data gets larger, the number of subsets the algorithm needs to generate grows in geometrical progression. This happened during this project as the initial data size that I tried to feed into the algorithm was huge. This continuously killed the kernel, and it prevented me from getting any results with the algorithm. There, I had no choice but need to cut down the data size. I trimmed down the data to 10,000 each by sampling based on the 'order_hour_of_day'. The algorithm must have high complexity issues both in time and space in genetics. To mitigate the impact of a large dataset, applying the Max-Miner algorithm may be promising. The algorithm tries to identify the maximal frequent item sets without generating all the potential frequent item sets and tries to skip things, 'jumps', in the search space.

2) Silhouette score of the clustered data

   As the silhouette score still has some gaps to be improved, further analysis on this matter might be a good idea. I suggest trying different features like department instead of the aisle as it gets fewer unique values. Or it would be a great idea that works with more customer behavior-related data like the average number of days from the previous order.

3) Interpretation of the customer segmentation result

   The dimension reduction helps the high-dimensionality data be clustered, but it makes interpretation hard in return. Because the features got heavily reduced to 3 components from more than one hundred, it is very hard to analyze what each feature means. There, it might be way harder to grasp the characteristics of each cluster like any demographical or the customers' behavioral characteristics and come up with a targeted solution.

Through this project, I could experience a holistic process of applying various algorithms to solve real business-like problems. I learned how preprocessing work is tedious but also critical which will bring immediate changes to the result. I have invested a lot of time to do the preprocessing work to make the data applicable to the machine learning algorithm. According to an IBM research publication, many researchers and practitioners focus on improving the quality of models while investing very limited efforts towards improving the data quality and reliability. The authors mentioned, "One of the crucial requirements before consuming datasets for any application is to understand the dataset at hand, and failure to do so can result in inaccurate analytics and unreliable decisions." This taught me how domain knowledge and groundwork are important. The project helped me to expand my interest in various data mining solutions and makes me expect to apply the lessons to future projects.

Reference

[1] Tan, Steinbach, Kumar. Introduction to Data Mining. Pearson New International Edition. Harlow: Pearson Education Ltd., 2014. (pp. 327-414).

[2] Michael Hahsler, http://michael.hahsler.net/research/association_rules/measures.html

[3] R. Agrawal, T. Imielinski, and A. Swami. Mining associations between sets of items in large databases. In Proc. of the ACM SIGMOD Int'l Conference on Management of Data, pages 207-216, Washington D.C., May 1993

[4] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data

[5] Apriori algorithm, Wikipedia, https://en.wikipedia.org/wiki/Apriori_algorithm