Capstone Project 2

Consolidated Report

Recommend Products To Instacart Customers

- *Oindrila Sen*

Introduction:

Online Shopping is a boon to many people. No more makeup, no more worries about the bad hair day and no more looking for a parking spot - The entire Retail World is just a click away. Receiving all your groceries at your doorstep on the same day is Fun. **Instacart** is the name who offers this service.

Problem Statement:

In this project, I want to explore customer shopping behavior by analyzing their previous purchases and build a shopping Recommendation Engine that could give them a tailored shopping experience to drive engagement. In this Dataset, we do not have any Feedback/Rating for any of the items brought by the customer. That means, we have Implicit Data and Recommendation Engine has to be built only focussing on user's purchase history.

Data Source:

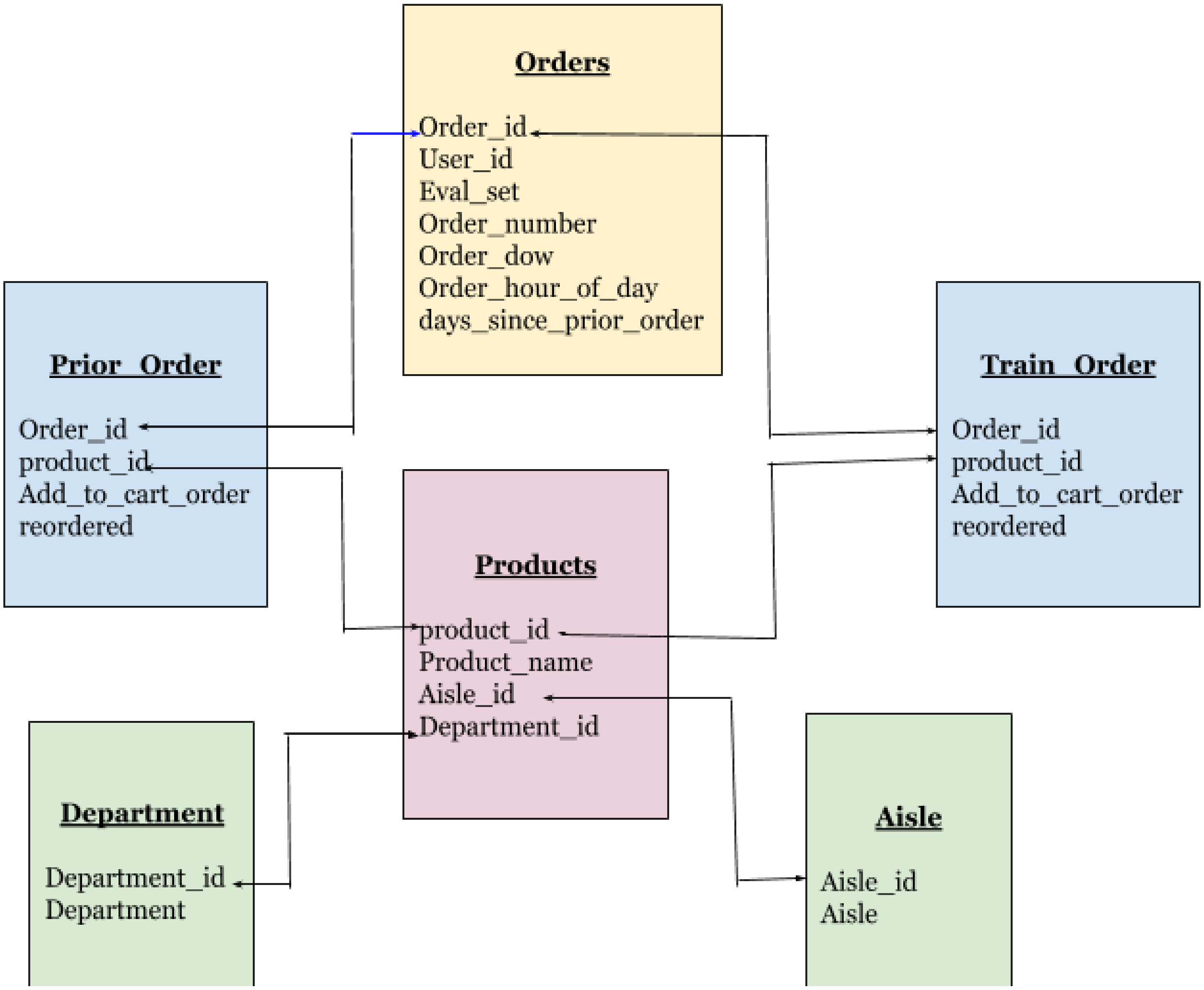
For this project, I am using the Dataset from “The Instacart Online Grocery Shopping Dataset 2017”, Accessed from https://www.instacart.com/datasets/grocery-shopping-2017 on 07/10/2019.

This anonymized dataset contains a sample of over 3 million grocery orders from more than 200,000 Instacart users. For each user, the dataset provides between 4 and 100 of their orders, with the sequence of products purchased in each order. It also provides the week and hour of day the order was placed, and a relative measure of time between the orders.

Data Wrangling:

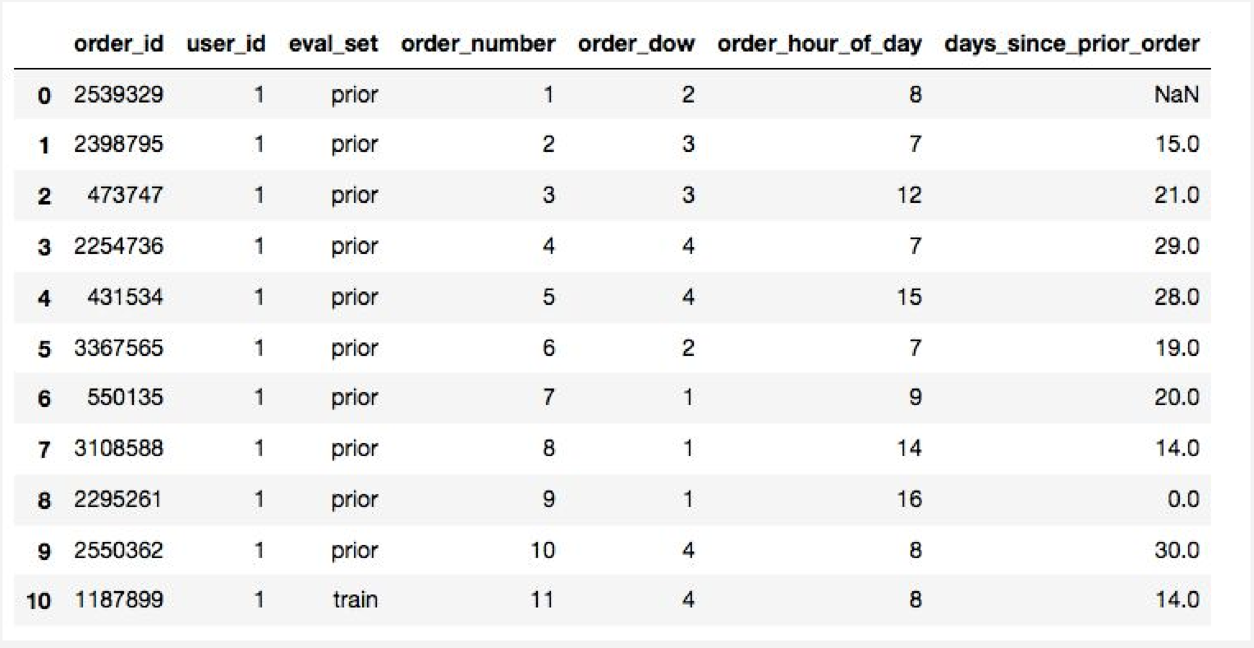
Before starting to talk about the different Data Wrangling method, let’s take a look at the Data Model in Detail:

Data Model:



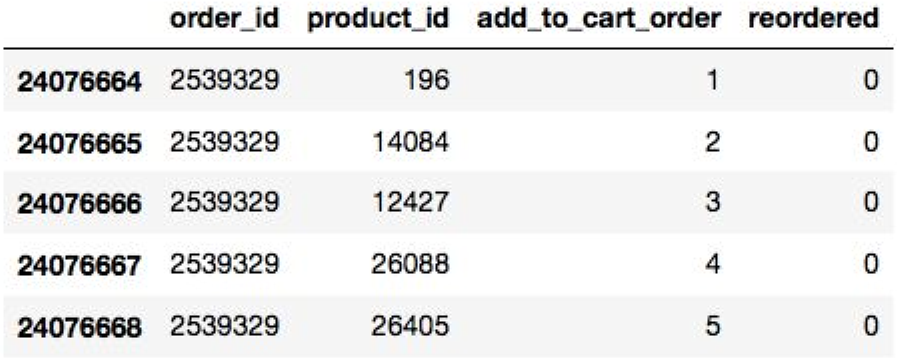
Now, Let's checkout the data:

1. Orders:



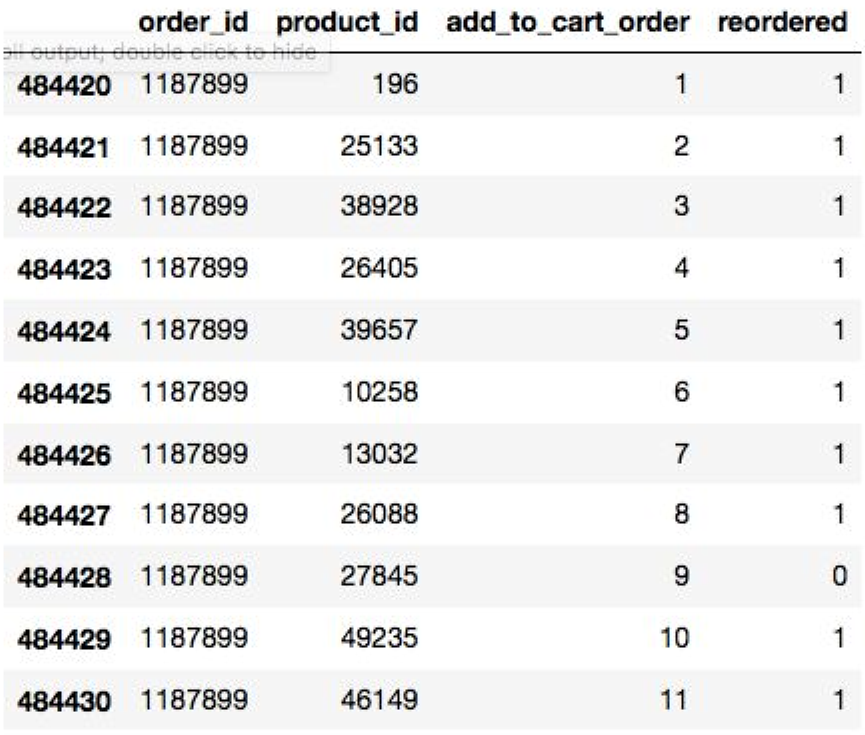
	order_id	user_id	eval_set	order_number	order_dow	order_hour_of_day	days_since_prior_order
0	2539329	1	prior	1	2	8	NaN
1	2398795	1	prior	2	3	7	15.0
2	473747	1	prior	3	3	12	21.0
3	2254736	1	prior	4	4	7	29.0
4	431534	1	prior	5	4	15	28.0
5	3367565	1	prior	6	2	7	19.0
6	550135	1	prior	7	1	9	20.0
7	3108588	1	prior	8	1	14	14.0
8	2295261	1	prior	9	1	16	0.0
9	2550362	1	prior	10	4	8	30.0
10	1187899	1	train	11	4	8	14.0

2. Prior Order



	order_id	product_id	add_to_cart_order	reordered
24076664	2539329	196	1	0
24076665	2539329	14084	2	0
24076666	2539329	12427	3	0
24076667	2539329	26088	4	0
24076668	2539329	26405	5	0

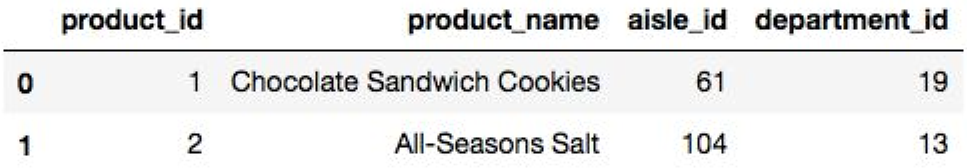
3. Train Order



ll output; double click to hide

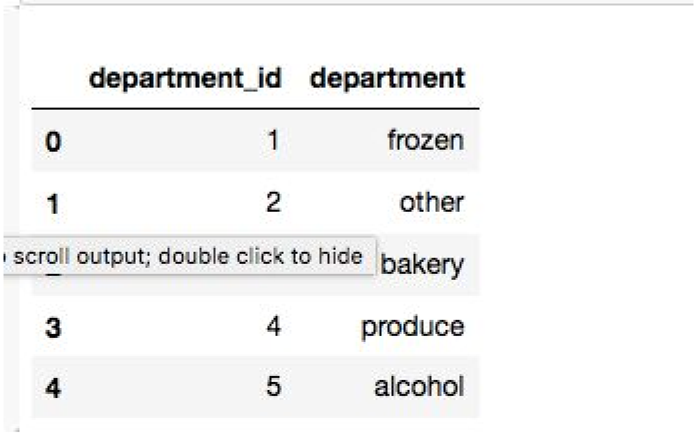
	order_id	product_id	add_to_cart_order	reordered
484420	1187899	196	1	1
484421	1187899	25133	2	1
484422	1187899	38928	3	1
484423	1187899	26405	4	1
484424	1187899	39657	5	1
484425	1187899	10258	6	1
484426	1187899	13032	7	1
484427	1187899	26088	8	1
484428	1187899	27845	9	0
484429	1187899	49235	10	1
484430	1187899	46149	11	1

4. Products



	product_id	product_name	aisle_id	department_id
0	1	Chocolate Sandwich Cookies	61	19
1	2	All-Seasons Salt	104	13

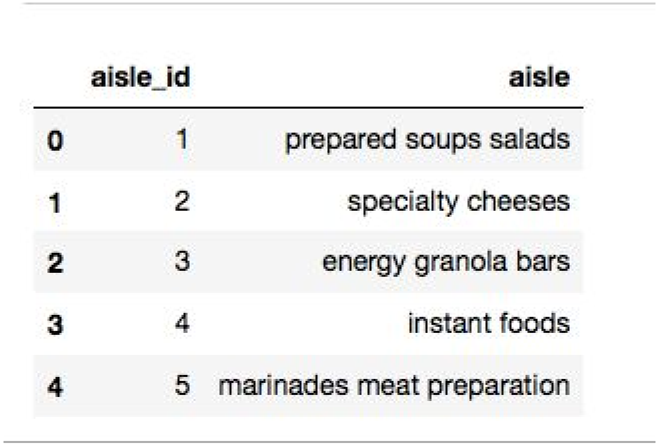
2. Department



	department_id	department
0	1	frozen
1	2	other
		bakery
3	4	produce
4	5	alcohol

scroll output; double click to hide

3. Aisle



	aisle_id	aisle
0	1	prepared soups salads
1	2	specialty cheeses
2	3	energy granola bars
3	4	instant foods
4	5	marinades meat preparation

After loading all the data, here is a snapshot:

Total Aisles: 134

Total Departments: 21

Total Products Count: 49688

Total Users/Customers: 206209

Total Orders: 3421083

Apparently, it is a huge Dataset.

Prepare Dataset:

1. Combine aisles, departments and products
2. Combine Prior Orders and Products
3. Combine Train Orders and Products¶¶
4. Combine Prior orders and Order Details
5. Combine Train Orders and Order Details
6. Merge The Train and Prior Dataset

Data Cleaning:

1. Convert NAN values to Zero¶
2. Change the below Column's DataTypes

- ❖ Change Eval_set to Category
- ❖ Change Department name Data type to Category
- ❖ Change Aisle name to Category

- ❖ Change Reordered column to Boolean

3. Sort the Data

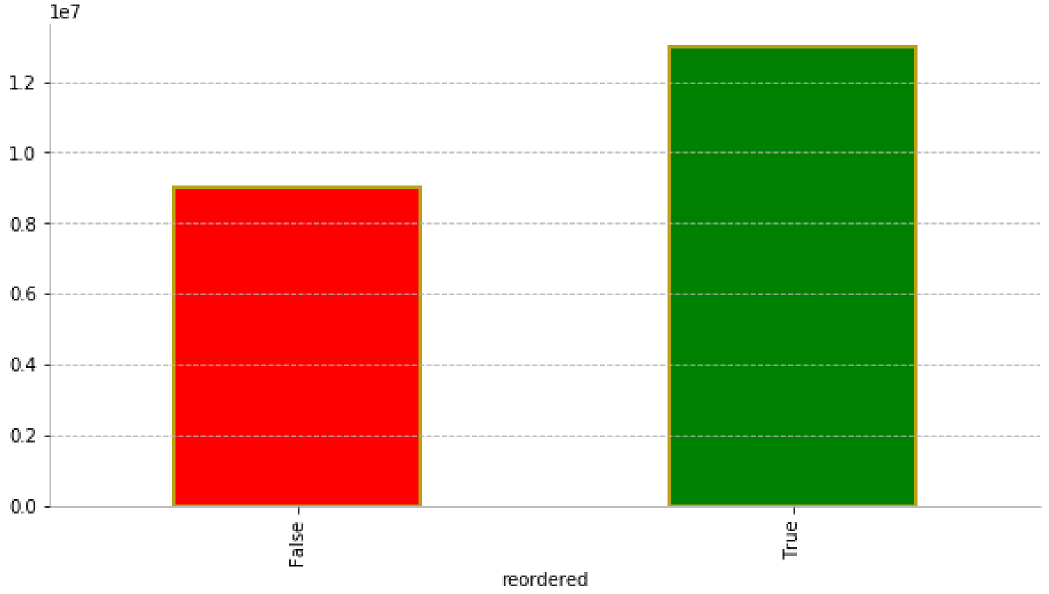
Feature Extraction:

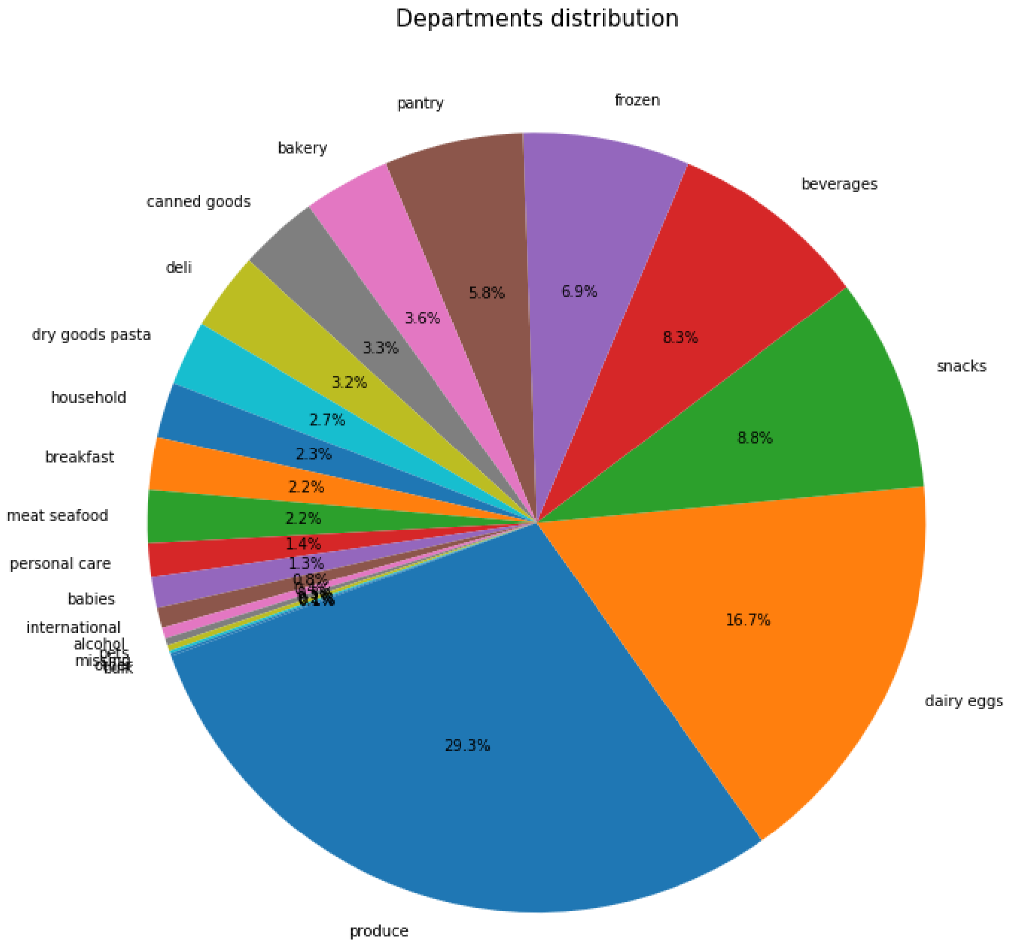
The below new features are extracted from the existing data and added as a new column.

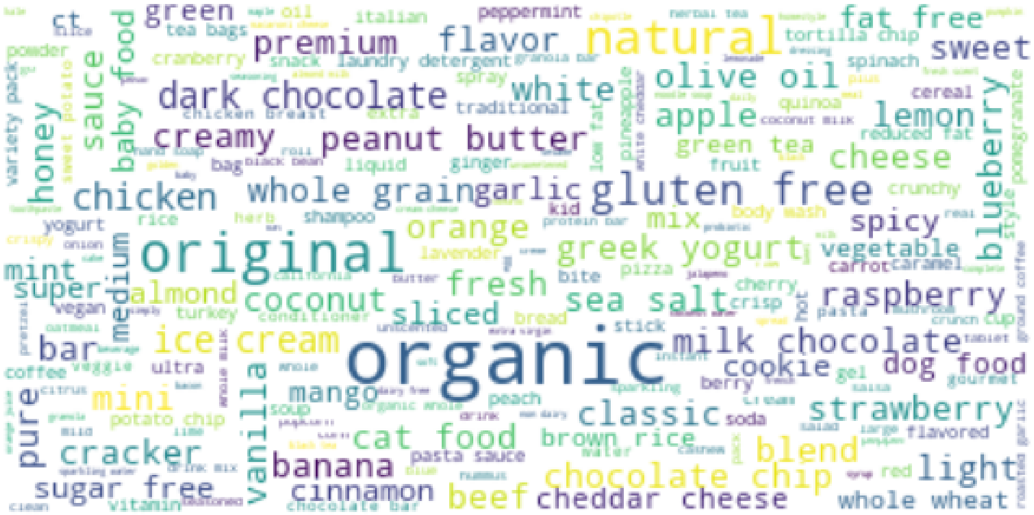
1. **Size of the Order:** How many Items are added per order?
2. **Total Orders per Customer**: How many transactions are done by the Customer?
3. **Loyal Customer:** Potential Customers are those customers who have ordered more than 32 times (The Average Order Counts)
4. **Order Span:** For how long the Customer is doing transactions?
5. **Customer Department Count**: How many times a customer bought a product from a particular department group?
6. **Customer Item Count**: How many times a customer bought a product?
7. **Weekend Customer:** Does the customer prefer to buy on weekends or a week day?
8. **Shopping Hour:**Does the customer prefers to shop at morning or evening or at mid night?
9. **Is_Organic:** The products with Organic In the Name are the Organic Products
10. **Prefers Organic**: Does the Customer prefers organic products?

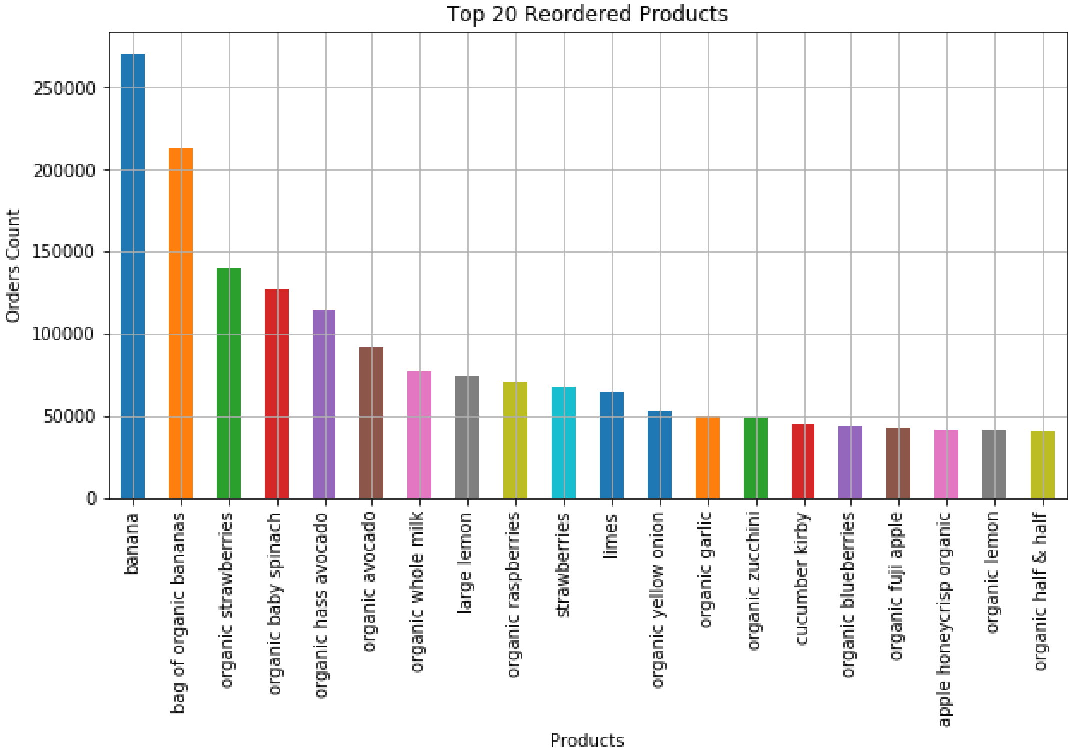
Exploratory Data Analysis:

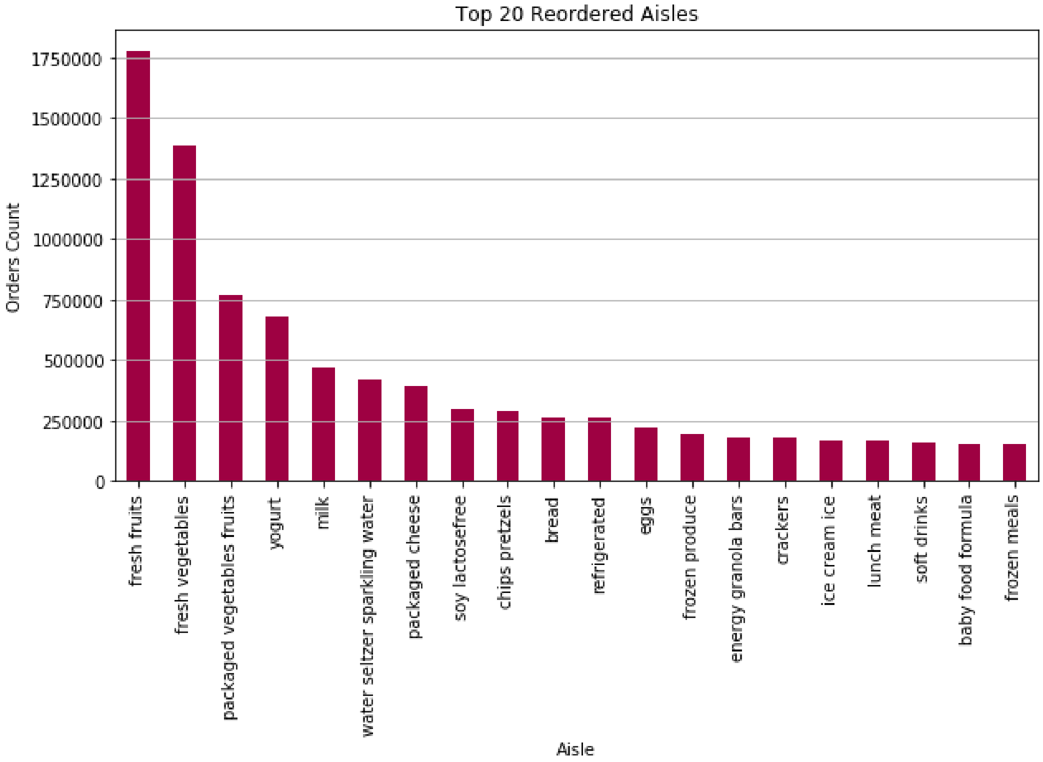
Now, let’s take a look at the visual aides:

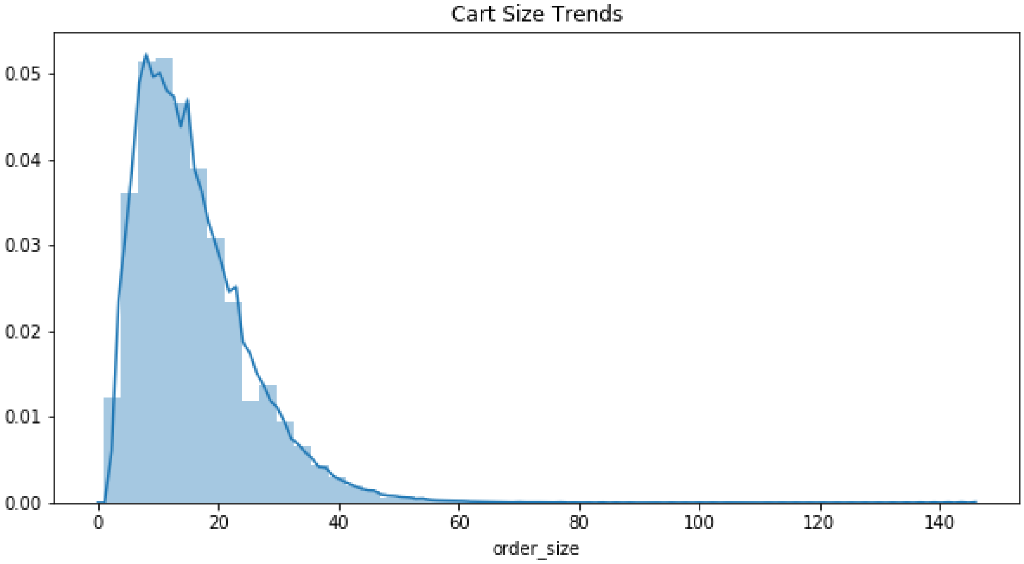

1e7
1.2
1.0
0.8
0.6
0.4
0.2
0.0
False
True
reordered

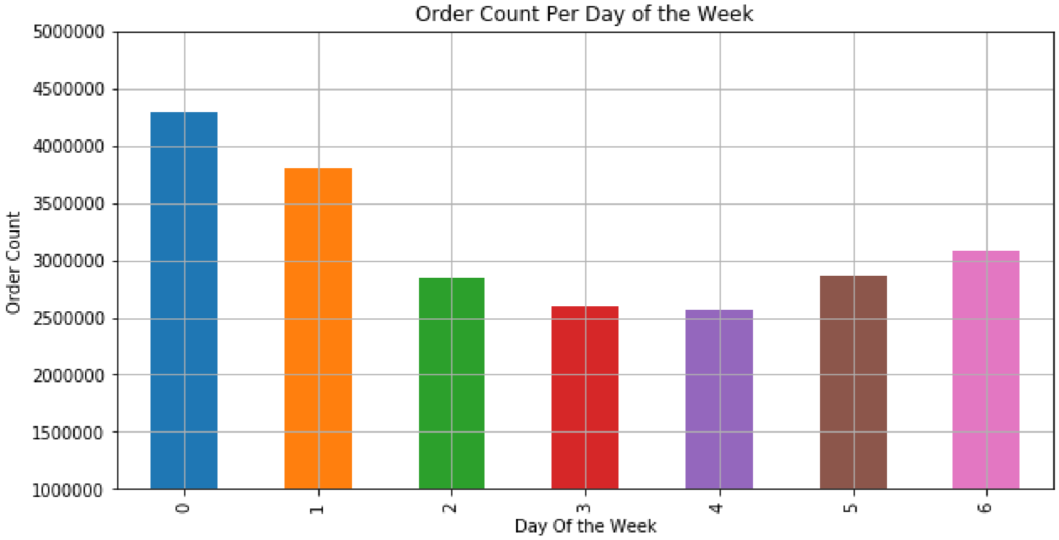

Departments distribution
pantry
frozen
bakery
beverages
canned goods
deli
dry goods pasta
snacks
household
breakfast
meat seafood
personal care
babies
international
alcohol
pets
missing
other
bulk
dairy eggs
produce
5.8%
6.9%
3.6%
8.3%
3.3%
3.2%
2.7%
8.8%
2.3%
2.2%
2.2%
1.4%
1.3%
0.8%
0.1%
16.7%
29.3%

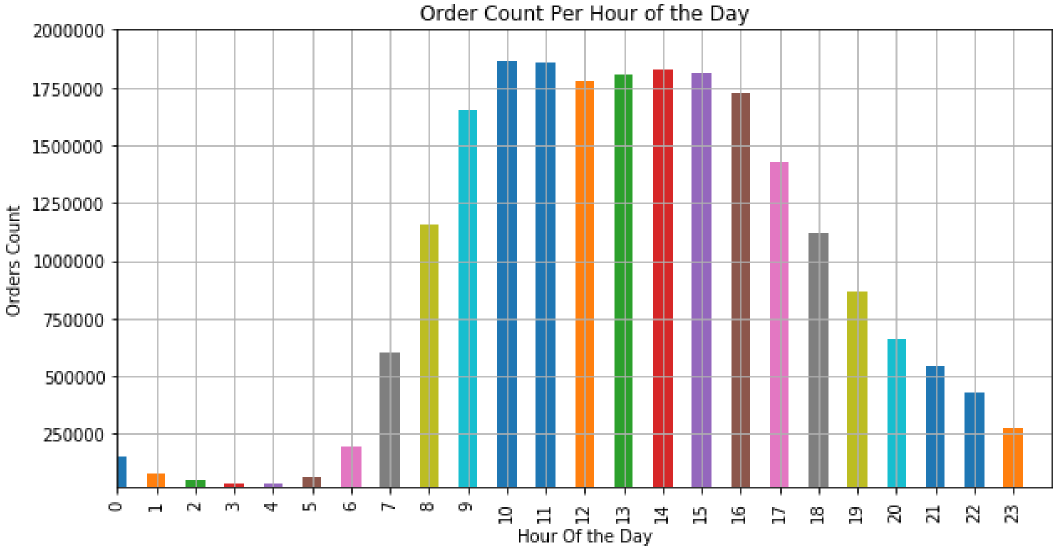


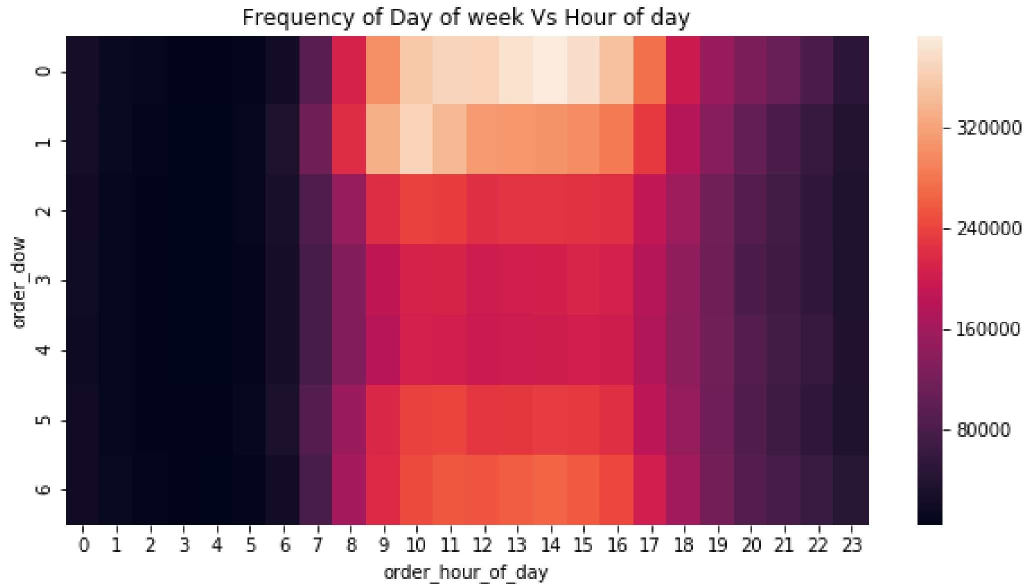

Top 20 Reordered Products

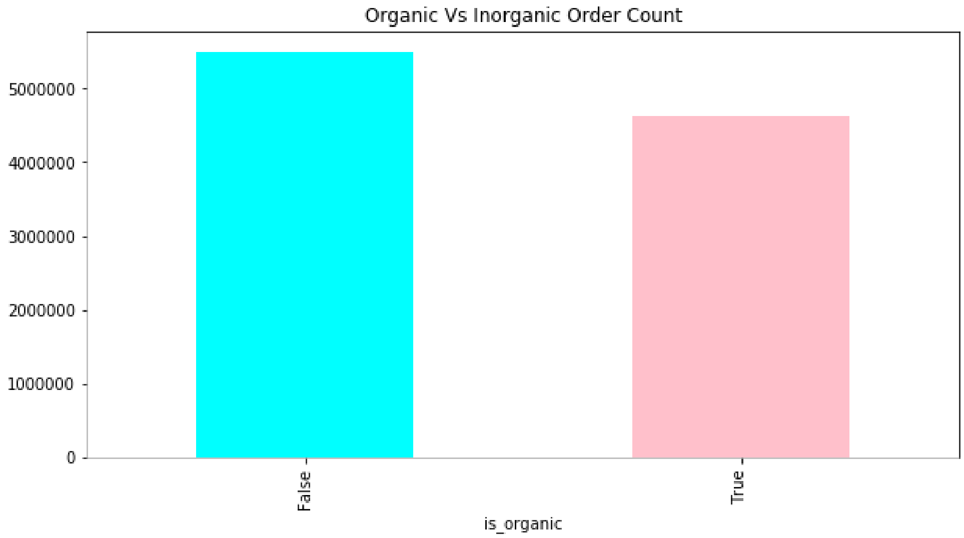

Top 20 Reordered Aisles
Orders Count
1750000
1500000
1250000
1000000
750000
500000
250000
0
fresh fruits
fresh vegetables
packaged vegetables fruits
yogurt
milk
water seltzer sparkling water
packaged cheese
soy lactosefree
chips pretzels
bread
refrigerated
eggs
frozen produce
energy granola bars
crackers
ice cream ice
lunch meat
soft drinks
baby food formula
frozen meals
Aisle

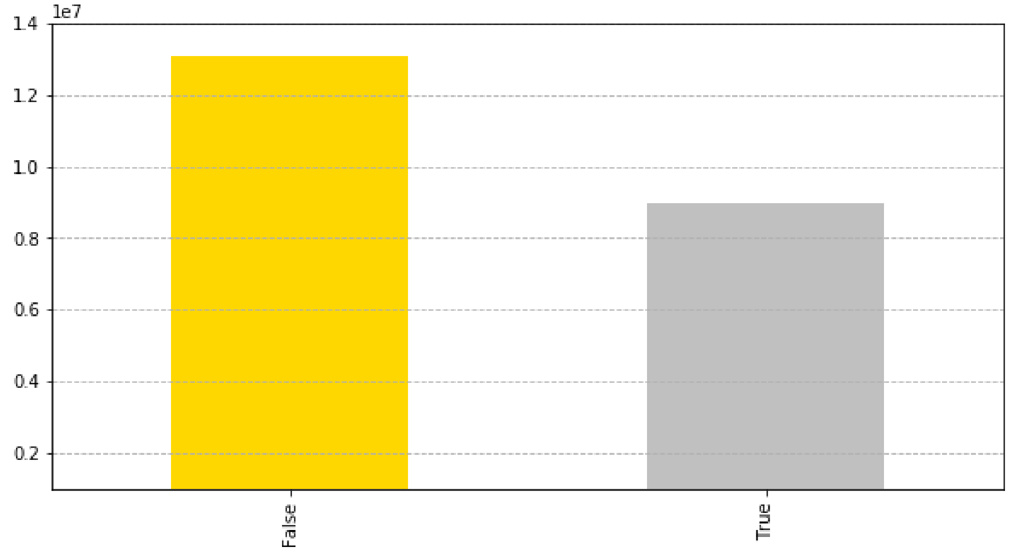

Cart Size Trends
0.05
0.04
0.03
0.02
0.01
0.00
0
20
40
60
80
100
120
140
order_size


Order Count Per Day of the Week
Order Count
5000000
4500000
4000000
3500000
3000000
2500000
2000000
1500000
1000000
0
1
2
3
4
5
6
Day Of the Week


Order Count Per Hour of the Day
Orders Count
2000000
1750000
1500000
1250000
1000000
750000
500000
250000
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
Hour Of the Day


Frequency of Day of week Vs Hour of day
order_dow
0
1
2
3
4
5
6
order_hour_of_day
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
320000
240000
160000
80000


Organic Vs Inorganic Order Count
5000000
4000000
3000000
2000000
1000000
0
False
True
is_organic



Recommendation Model:

Basic Recommendations:

1. Trending Products at Instacart

Produce Items are the most popular products at Instacart. Here, we are recommending the most bought items in general.

2. Recommendations from appropriate Departments

Recommending the most popular products from the Department in which the customer is browsing.

3. Recommendations of reorder products

There are some products that the customer buys often or regularly. Recommending the products that the customers do reorder.

4. Recommendations of Unique Products:

There are some products which the customers do not buy regularly or even not aware that they are being sold at Instacart. This is the Opposite of Popular Products.

Recommendation Using Machine Learning Algorithms:

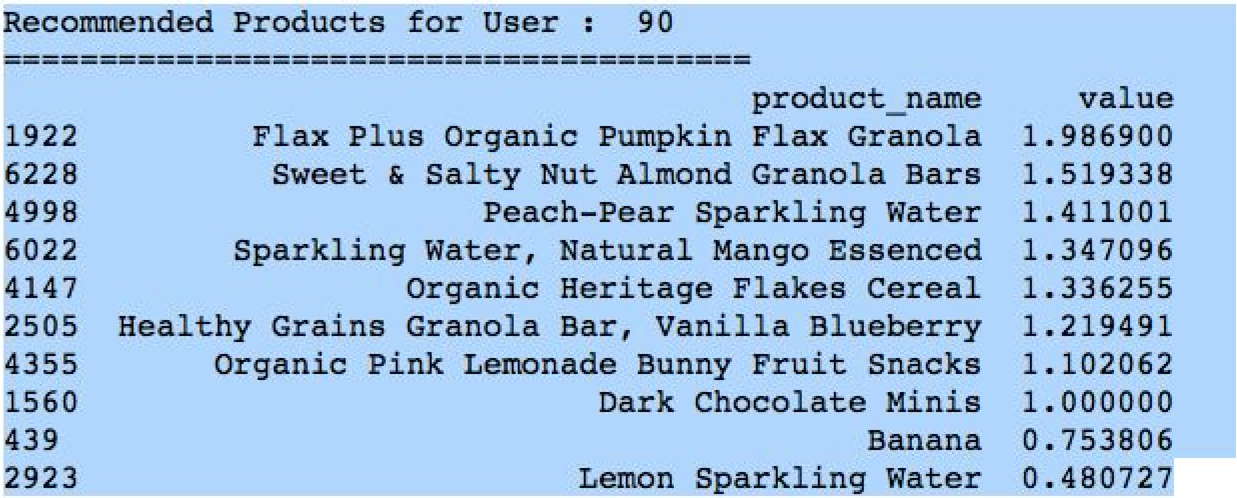
Cosine Similarity Model:

Cosine similarity is a measure of similarity between two non-zero vectors by calculating the cosine of the angle between them. In simple words, we can use the Cosine Similarity algorithm to find similarity between two things. In this scenario, we can find similar **Users** or similar **Products**.

In this project, I have found similar users. The cosine of a 0 degree angle is 1. Therefore, if the value is closer to 1, the more similar the users are. Finally I have recommended items based on similar profiles that have similar purchase history as our target customer.

For Example:

Recommended Products for Customer# 90

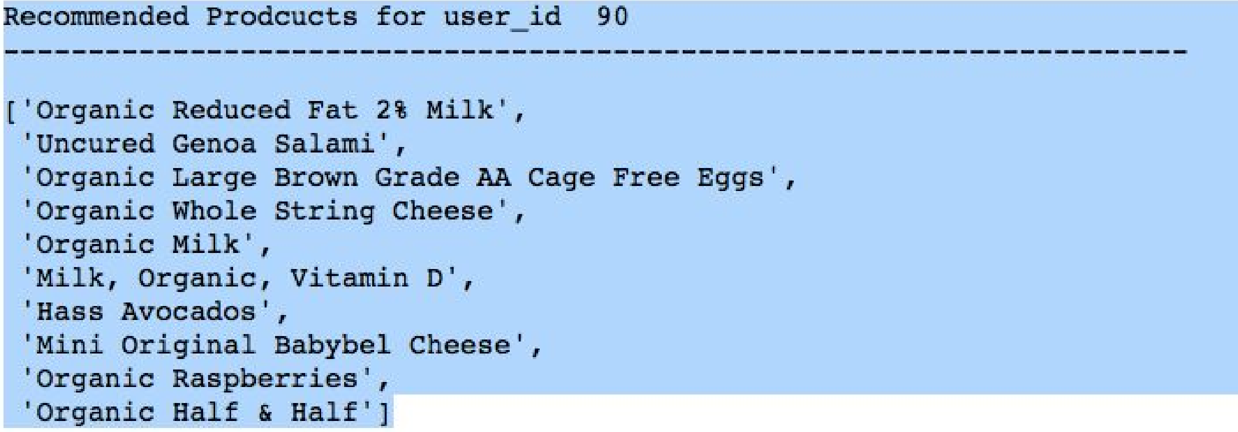


```
Recommended Products for User :  90
=======================================
                                product_name     value
1922         Flax Plus Organic Pumpkin Flax Granola  1.986900
6228          Sweet & Salty Nut Almond Granola Bars  1.519338
4998                     Peach-Pear Sparkling Water  1.411001
6022         Sparkling Water, Natural Mango Essenced  1.347096
4147                 Organic Heritage Flakes Cereal  1.336255
2505   Healthy Grains Granola Bar, Vanilla Blueberry  1.219491
4355        Organic Pink Lemonade Bunny Fruit Snacks  1.102062
1560                           Dark Chocolate Minis  1.000000
439                                          Banana  0.753806
2923                          Lemon Sparkling Water  0.480727
```

Alternating Least Square Model(ALS):

Alternating Least Squares (ALS) is a model I have used to fit our data and find similarities. ALS uses Matrix Factorization method for recommendations. The idea is to take a large matrix and factor it into some smaller representation of the original matrix.

For the same Customer# 90, below are the recommendations from ALS Model:



```
Recommended Prodcucts for user_id  90
----------------------------------------------------------------------

['Organic Reduced Fat 2% Milk',
 'Uncured Genoa Salami',
 'Organic Large Brown Grade AA Cage Free Eggs',
 'Organic Whole String Cheese',
 'Organic Milk',
 'Milk, Organic, Vitamin D',
 'Hass Avocados',
 'Mini Original Babybel Cheese',
 'Organic Raspberries',
 'Organic Half & Half']
```

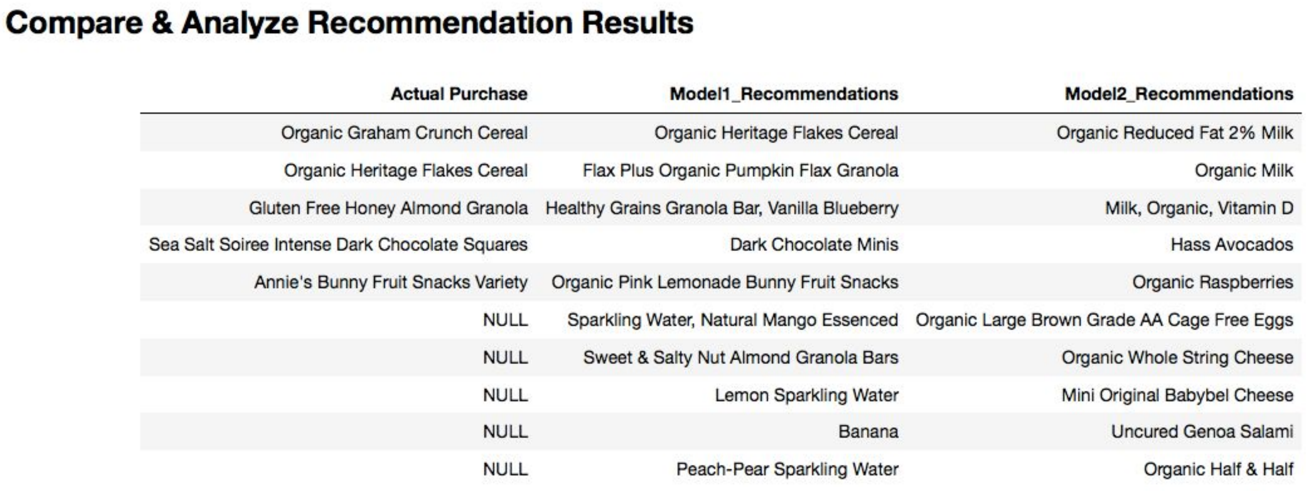
Customer Classification using KMeans, KMode and KPrototype:

My idea was to classify the customers based on their shopping behavior and then recommend products based on their cluster. But it turned out that the current dataset is not appropriate for this method. In this Dataset, most of the customer buy some food products or Beverages or similar products. It is difficult to distinguish Customers based on the items bought. Thus Recommendations turns out difficult using this method.

But I have used and learnt PCA, t-SNE, Elbow Method and Shilhoute Method to get an idea of the data or the cluster size.

Analysis and Conclusion:

The objective of this project is to understand the behavior of different customer, the shopping trend and then finally recommend products to customers based on the Purchase History. I have applied different recommendation algorithm and have grouped Customers based on their behavior.Let's take a look at a particular customer and see how the recommendations are:



Compare & Analyze Recommendation Results

Actual Purchase	Model1_Recommendations	Model2_Recommendations
Organic Graham Crunch Cereal	Organic Heritage Flakes Cereal	Organic Reduced Fat 2% Milk
Organic Heritage Flakes Cereal	Flax Plus Organic Pumpkin Flax Granola	Organic Milk
Gluten Free Honey Almond Granola	Healthy Grains Granola Bar, Vanilla Blueberry	Milk, Organic, Vitamin D
Sea Salt Soiree Intense Dark Chocolate Squares	Dark Chocolate Minis	Hass Avocados
Annie's Bunny Fruit Snacks Variety	Organic Pink Lemonade Bunny Fruit Snacks	Organic Raspberries
NULL	Sparkling Water, Natural Mango Essenced	Organic Large Brown Grade AA Cage Free Eggs
NULL	Sweet & Salty Nut Almond Granola Bars	Organic Whole String Cheese
NULL	Lemon Sparkling Water	Mini Original Babybel Cheese
NULL	Banana	Uncured Genoa Salami
NULL	Peach-Pear Sparkling Water	Organic Half & Half

In the above table, we can see the Actual Purchase of Customer# 90 and the Recommendations from Model1(Cosine Similarity) and Model2(Alternating Least Squares).

The Customer seems to buy Cereals and Granola and some Chocolates and fruit snacks.

Model1 has recommended some other variety of cereals and granolas and even some chocolate and fruit snacks which seems quite appropriate.

Interestingly, **Model2** has recommended different varieties of Milk which goes perfectly with the Cereals and Granolas and some other breakfast items like eggs, salami and fruits.

Thus, in my opinion both the recommendations are appropriate. Now, let's see if we can evaluate the performance of both the Models with some Metrics.

Evaluation Metrics:

Evaluating implicit feedback based recommendations is always tricky. Below is the approach that I have followed here:

- Used the high volume dataset (products that users reordered more than once)
- Got the top 20 recommendations using both the models
- Calculated Recall by using # of user actions (products bought) that were captured by the top 20 recommendations.
- Calculated these for all the users and average them.

As per the above metrics, it seems like the **ALS Model** did a little better than the **Cosine Similarity Model**. The score of Cosine Similarity Model is **0.122** and that of ALS Model is **0.220**.