Instacart Shopping Data

In Depth Analysis Report

Sava Dashev

Machine learning

We created three models using kmeans. The variables in the first model (Model 1) are only the aisles in the data. The second model (Model 2) adds up time data and average number of items in each aisle per user. The third model (Model 2a)variables are variables of first model and max number of items.

All three model use normalized averages. Some users have more orders in the data and other have less. We divide the number of items in each aisle by the total number of orders each user has. This way, the number of orders does not skew data.

Model 1

To optimize the number of clusters we used the inertia elbow method. The graph has an elbow when we have only two cluster. The two clusters differ by how many items on average are placed from each aisle. The first ten items in both are almost identical and in the same order. The number of items cluster 0 order is

Model 2

Model two graph suggests 4 cluster according to the elbow method. The first ten items in all clusters are almost identical. What is different, is how many items on average are in each order. The clusters differ by how many items on average per aisle each order has. There are about 5.5, 12, 19.8 and 8.3 items in basked for clusters 0 – 3. This observation prompted Model 2a. Is the most important variable the number of items?

Model 2a

Adding only max number of items in basked to Model 1 gives us another model for which the elbow method suggest only 3 or 4 clusters. Both divisions of customers in clusters are similar to what we see in Model 1.

We therefore need all variables in Model 2 to divide customers in reasonable groups.

In each of the clusters there are aisles from which the contribution to the basked is almost 0. We considered threshold of 0.005 items on average per basket. The Model 2 suggest that 6 of the aisles are almost never shopped from: 'baby accessories', 'beauty', 'eye ear care', 'frozen juice', 'kitchen supplies' and 'specialty wines champagnes'. Looking at the data as whole, there are 14 aisles that have less than 0.005 items in basked. However, placing the customers in groups, only 6 appear to be lower than that threshold for all clusters.

The clusters most ordered products are the same. The quantity of each and the ratio of number of ordered products in clusters vary. When we have two clusters, cluster 0 38% of all items are the top ten categories, and cluster 1 = 52% are top 10.

We use Model 2 for more detailed observations. The top 10 of each cluster are from the same group. However, the order of the items is different. The baskets also differ in the total number of items per basket. The smallest basket is in Cluster 0: about 5.5 items, the largest number is cluster 2: 19 items. Cluster 1 is about 12, and Cluster 3 is 8.3.

Looking at the common elements in first 15, 12 of them are the same. Only 3 items differ in 4 clusters. However, the ratio and quantities ordered differ.

The number in the table above shows the average number of items from each aisle ordered in single order. For example, fresh vegetables 2.12 means that on average, each user in that group orders more than 2 items from this aisle in each order.

For each cluster, the average amount differs significantly. We can compare the amounts using ratios. The biggest ratio is usually cluster 0 to cluster 2. It is about 3.5:1 for most products.

The exception is sparkling water, where most of customers orders do not vary that much.

| **Cluster** | **0** | **1** | **2** | **3** |
| --- | --- | --- | --- | --- |
| **Aisle** |  |  |  |  |
| **frozen produce** | 0.086489 | 0.208658 | 0.333845 | 0.133110 |
| **soy lactose free** | 0.107396 | 0.230188 | 0.335294 | 0.169346 |
| **water seltzer sparkling water** | 0.204346 | 0.270603 | 0.377591 | 0.237742 |
| **chips pretzels** | 0.125387 | 0.266254 | 0.481361 | 0.177670 |
| **fresh fruits** | 0.551040 | 1.252642 | 1.997348 | 0.970997 |
| **packaged vegetables fruits** | 0.271392 | 0.662812 | 1.053878 | 0.457796 |
| **bread** | 0.093507 | 0.213827 | 0.356345 | 0.144944 |
| **packaged cheese** | 0.144934 | 0.374205 | 0.672385 | 0.226145 |
| **milk** | 0.148617 | 0.276552 | 0.440531 | 0.237531 |
| **refrigerated** | 0.096877 | 0.200341 | 0.333638 | 0.152171 |
| **fresh vegetables** | 0.468563 | 1.393585 | 2.119698 | 0.854190 |
| **yogurt** | 0.187258 | 0.514296 | 0.908825 | 0.348712 |

The difference is clear, when we choose one of the products as a base, and the other quantities in each cluster are scaled. I used fresh fruits and fresh vegetables as base. The fresh fruits table is shown below:

| **Cluster** | **0** | **1** | **2** | **3** |
| --- | --- | --- | --- | --- |
| **aisle** |  |  |  |  |
| **frozen produce** | 0.184582 | 0.149727 | 0.157497 | 0.155832 |
| **soy lactose free** | 0.229203 | 0.165177 | 0.158180 | 0.198253 |
| **water seltzer sparkling water** | 0.436112 | 0.194177 | 0.178134 | 0.278324 |
| **chips pretzels** | 0.267599 | 0.191057 | 0.227089 | 0.207999 |
| **fresh fruits** | 1.176020 | 0.898863 | 0.942279 | 1.136746 |
| **packaged vegetables fruits** | 0.579200 | 0.475616 | 0.497183 | 0.535942 |
| **bread** | 0.199561 | 0.153437 | 0.168111 | 0.169686 |
| **packaged cheese** | 0.309316 | 0.268520 | 0.317208 | 0.264749 |
| **milk** | 0.317176 | 0.198446 | 0.207827 | 0.278077 |
| **refrigerated** | 0.206754 | 0.143759 | 0.157399 | 0.178147 |
| **fresh vegetables** | 1.000000 | 1.000000 | 1.000000 | 1.000000 |
| **yogurt** | 0.399644 | 0.369045 | 0.428752 | 0.408237 |

For all clusters, the bottom 60 aisles contribute less than 0.02 to 0.05 items per aisle in the basket for each customer.

Other models

We used DBScan with the same data. The model did not perform well. We used eps up to 0.24, at which point the model crushed. Using smaller eps, with leaf size 30, give 12 – 16 clusters. Most of users were classified as noise. For that reason, we abandoned the model.

The above model runs on aisles. We tried similar approach using individual items. The number of items is more than 48,000 and the number of users is 206,000. We used 5% of all users. The model crushed even when the scaled to 5% of the users. For that reason we stopped using it.