Instacart Market Analysis

Yuheng Cai

UMID: 20698962

Motivation:

The dataset of this project is a relational set of csv. files describing Instarcart's customer order over time. We can use this dataset to do some interesting market analysis and even some forecasting that can assist in decision making.

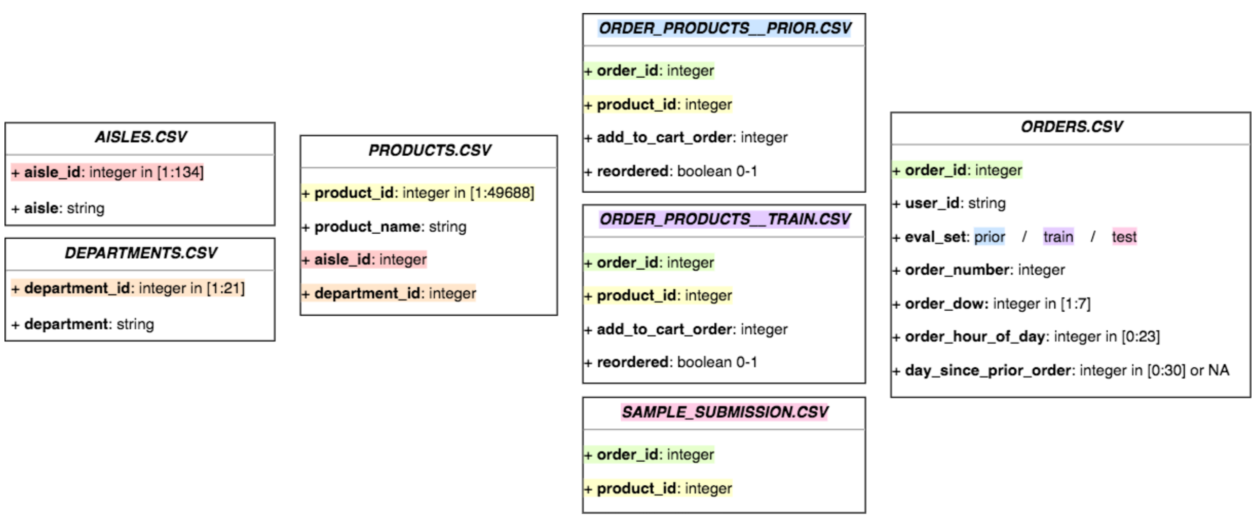
The four questions that I explored were as followed:

1. Which aisles of products do customers tend to buy more in weekend?
2. Which variables will affect the ratio of reordering and how?
3. How to divide customers into different groups?
4. How to predict customers' next order base on previous orders?

Data Source:

URL: https://www.kaggle.com/c/instacart-market-basketanalysis/data

The format of the dataset is csv and datatype of the data set is provided in Pic. 1. The two most import features among are 'eval_set' which divide orders into prior/tain/test sets and 'reordered' denoting whether the product is recorded.



Picture 1. Overview of Data Source

The biggest csv. File among all is order_products__train.csv which has 32434489 rows. There are three features relevant to time: order_dow, order_hour_of_day and day_since_prior_order.

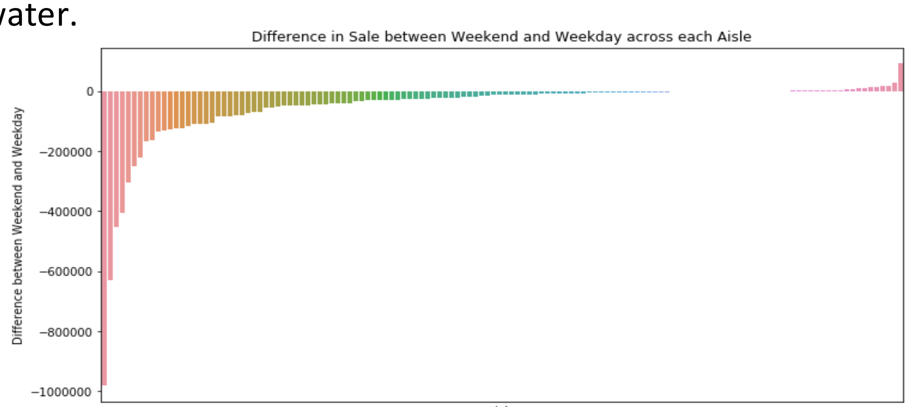
<u>Question 1: Which aisles of products do customers tend to buy more in weekend?</u>

The only column with missing values is 'days_since_last_order'. After checking the tag of order with missing value, I found these orders all come from 'prior' and it is possible that they are the earliest orders. Therefore, I simply replaced them with zero.

To prepare the data for analysis, first, I concatenated 'order_product_prior' and 'order_product_train' in the column direction. Then, I combine all the relational dataset except 'orders' by inner join then use the consequent dataset to right join with 'orders'.

Then, I separated the dataset into dataset of weekend and dataset of weekday. Second, I grouped these two data set respectively with 'aisle' and took mean of 'order_number'. Lastly, I combined two table and calculated the difference between weekend and weekday. The biggest challenge in this question was to prepare the dataset to calculate the difference between weekdays and weekend.

After sorting the result and made plot(Pic. 2) from it,we can tell from the result that most of the aisle sell less in weekend. The top 5 aisle sells more in weekend are specialty cheeses, baking ingredients, doughs gelatins bake mixes, hot dogs bacon sausage and ice cream ice. The top 5 aisle sells less in weekend are fresh fruits, fresh vegetables, yogurt, packaged vegetables fruits and water seltzer sparkling water.



Picture 2. Difference in Sale between Weekend and Weekdays across each Aisle

Table 1: Top 5 aisle Sells more in Weekend(right) and Weekdays(left)

aisle	dif
fresh fruits	-980392.5
fresh vegetables	-628924.8
yogurt	-453479.2
packaged vegetables fruits	-406561.6
water seltzer sparkling water	-305517.2

aisle	dif
specialty cheeses	15348.3
baking ingredients	17143.7
doughs gelatins bake mixes	18866.7
hot dogs bacon sausage	27494.9
ice cream ice	91982.4

Question2: Which variables will affect the ratio of reordering and how?

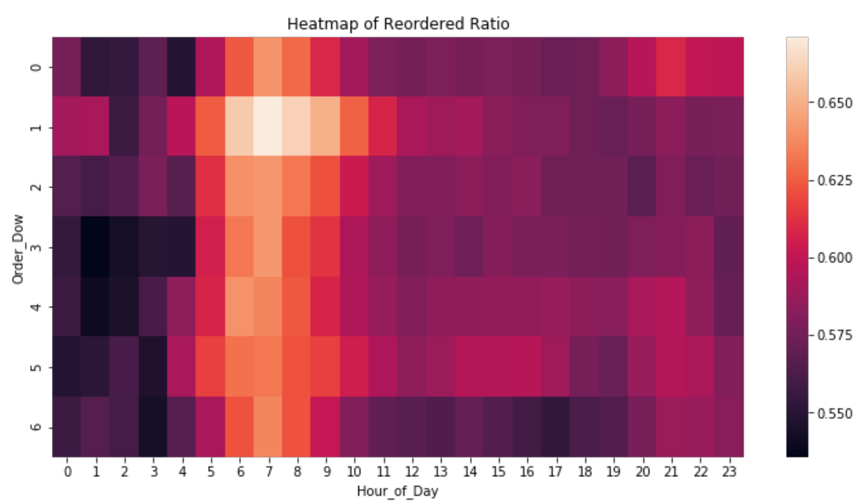
We do not need to deal with missing value because it has been resolved in question 1.

The variables we are interested in are 'order_dow', 'order_hour_of_day', 'add_to_cart_order', 'order_number', and 'day_since_last_order'. We will analyze their relationship with reordered ratio respectively.

(1)'order_dow' and 'order_hour_of_day':

Combing orders_df, order_products_prior_df and order_products_train_df, grouping the data by 'order_dow' and 'order_hour_of_day' to calculate the mean of 'reordered', then turning it to pivot table with 'mean of reordered' in the middle, we got the data set we need and plot a heatmap with it.

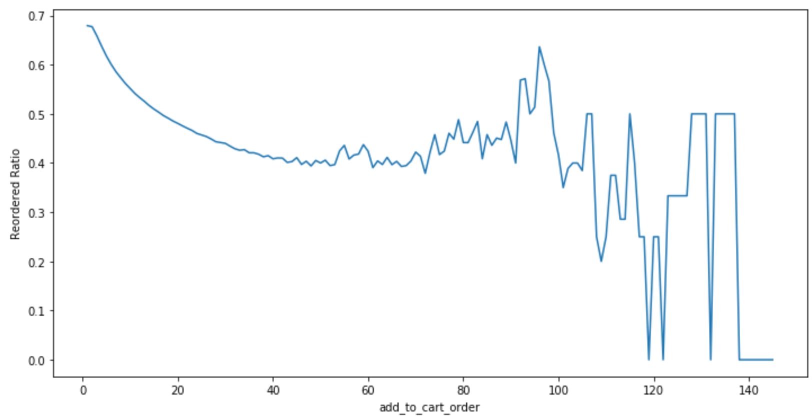
As you can see from Pic. 3, we know that there is a strong relationship between reorder ratio and time. Additionally, people tend to reorder on Tuesday, from 6 AM to 9 AM.



Picture 3. Heatmap of Reordered Ratio_1

(2)‘add_to_cart_order’:

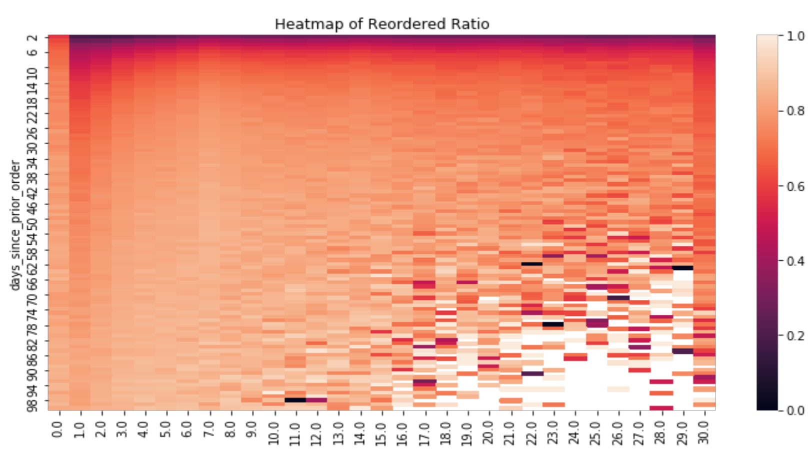
Combing ‘order_products_prior_df’ and ‘order_products_train_df’, grouping by ‘add_to_cart_order’ to take mean of ‘reordered’, we got the dataset we want. Then plot the two feature in Pic. 4. According to the plot below, we know there is also a strong relationship between add_to_cart_order and reordered ratio. The correlation is negative when the order is between 1 to 50, and start to fluctuate after that.



Picture 4. Plot of add_to_cart_order and reordered_ratio

(3)‘order_number’ and ‘day_since_last_order’:

Using the same dataset as in (1), I grouped it by 'order_number', 'days_since_prior_order' and calculate the mean of ‘reordered’ then turned it to pivot table. Finally, plot a heatmap base on it. We can know that, generally, higher the days_since_prior_order and order_number are, higher the reorder ratio will be.



Picture 5. Heatmap of Reordered Ratio_2

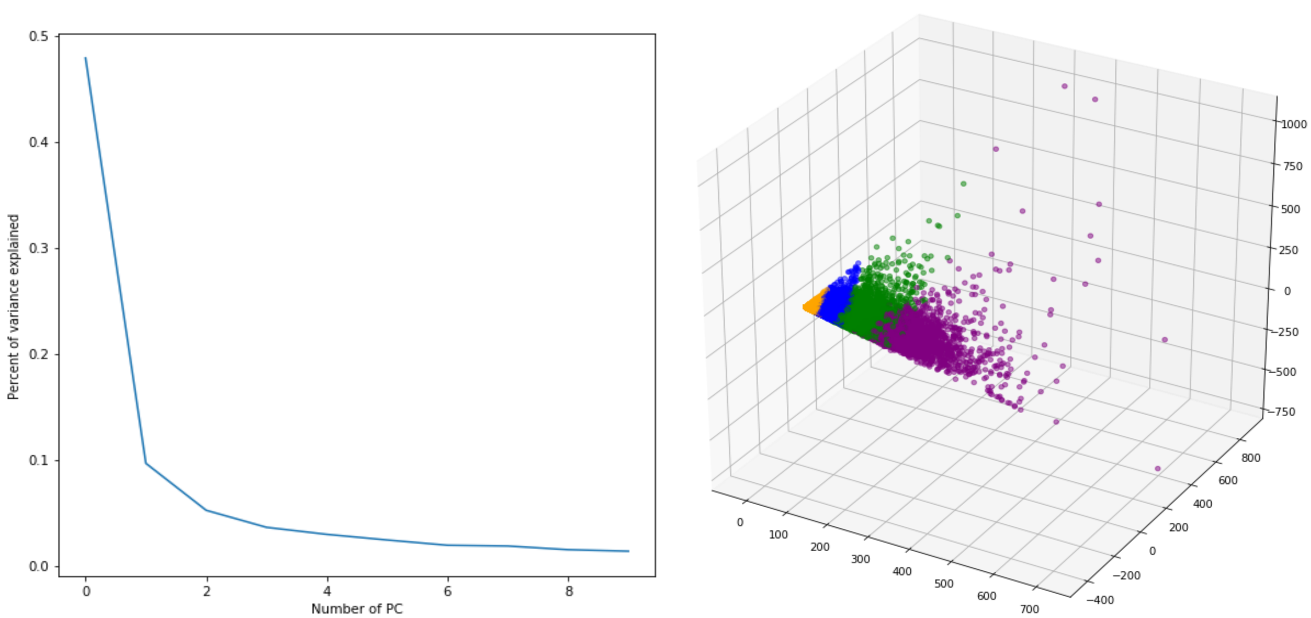
The challenge of this problem is to calculate the ‘reorder ratio’ from ‘reordered’.

<u>Question3: How to divide customers into different groups?</u>

We do not need to deal with missing value because it has been resolved in question 1.

The challenge of this problem is to determine which features are better for clustering. Because essentially, what make customers different from each other is what they buy, we focus on ‘aisle’ and ‘product’. Since there are too many products, we choose ‘aisle’ and implement PCA on it to make the features even fewer.

First of all, I took feature ‘user_id’ and ‘aisle’ to make a pivot table with count of orders in the middle. Then, I used PCA to reduce dimension of the dataset. After checking the scree plot, I decided to choose first 4 PCs because it is right to the elbow and cover enough information. Then, I implemented clustering base on these 4 PCs to get 4 customer segments.



Picture 6. Scree Plot(left), Visualization of Clustering with 3 PCs

For each cluster, we took its mean and sort ascendingly to get top 10 aisles. We found that all of the clusters have high demand of fresh fruits, fresh vegetable, package vegetables fruits, yogurt, etc. which are basic need for everyone. Cluster_3 features with baby formula which indicate that this cluster may be parents of small baby.

Cluster 1		**Cluster 3**	
fresh fruits	6.2501	fresh fruits	156.77620
fresh vegetables	5.6331	fresh vegetables	155.24617
packaged vegetables fruits	3.3805	packaged vegetables fruits	67.39355
yogurt	2.7794	yogurt	51.21861
water seltzer sparkling water	2.4288	packaged cheese	31.15338
packaged cheese	2.1992	milk	29.78329
milk	1.9399	soy lactosefree	19.66839
chips pretzels	1.9218	bread	18.21588
ice cream ice	1.4958	baby food formula	18.06795
soft drinks	1.4161	chips pretzels	17.12691
Cluster 2		**Cluster 4**	
fresh fruits	29.950	fresh fruits	77.140
fresh vegetables	29.662	fresh vegetables	68.742
packaged vegetables fruits	15.022	packaged vegetables fruits	34.798
yogurt	11.994	yogurt	30.187
packaged cheese	8.119	packaged cheese	18.031
milk	7.108	milk	17.097
water seltzer sparkling water	6.831	water seltzer sparkling water	13.059
chips pretzels	5.935	chips pretzels	11.826
soy lactosefree	5.435	soy lactosefree	11.810
refrigerated	4.799	bread	10.584

Question 4: How to predict customers' next order base on previous orders?

The orders.csv is the core dataset whose feature 'eval_set' divide the customers' orders into 'prior'/'train'/'test' orders. Each Customer who belongs to the training set have n-1 'prior' orders and 1 'training' order, while customer from testing set have n-1 'prior' orders and 1 'test' order.

I turned this problem into a binary classification problem, i.e., classified every user-product pair into either reordered(1) or not reordered(0). It is apparent that 'reordered' is Y for this classification problem, but how to choose X is quite tricky. There are only 6 meaningful features: add_to_cart_order, reordered, order_number, order_dow, order_hour_of_day, day_since_prior_order. For each user-product pair, in order to both include the meaningful features from this user-product pair's priors and the meaningful features of itself, here is the step I took:

(1) Get the original dataset by joining order_product_df, orders_df and products_df.
(2) Check missing value.(as previous question)
(3) Split the dataset into test and train according to users.
(4) After turning categorical variable into dummy variables, I used aggregation to shrink both continuous and categorical features from priors into one row for each user-product pair with 'max', 'min' or 'mean'.
(5) Merged the features from priors with features from the user-product pairs themselves.
(6) Built model by GradientBoostingClassifier and find the optimal learning rate 0.5 according to the metric Area Under the Curve(auc).
(7) Applied the model on the test data and got the prediction result. After some dataframe manipulation, I got the result with head as follow, which predicted what will each order consists of.

	order_id	products
0	34	39180 47029
1	182	9337 13629 39275
2	257	49235
3	313	12779 25890 45007
4	386	15872 21479 24852 38281 39180 40759 42265 4506...

Picture 7. Sample of Prediction Results