INSTACART MARKET BASKET ANALYSIS

6/19/2018

This document is a milestone report of the capstone project for Springboard's Career track course on Data Science. The report outlines the problem, exploratory data analysis and the initial findings.

Instacart Market Basket Analysis

INTRODUCTION

This is a milestone report for an on-going Capstone project for Springboard's Career track course on Data Science. It outlines the objective, identifies the client, findings from the initial exploratory data analysis are illustrated along with findings.

Instacart is an online grocery ordering and delivery app. After selecting the products through the Instacart app, personal shoppers review the order and do in-store shopping and delivery. Transactional data is used by Instacart to predict which products a user will buy again, try for the first time and add to the cart.

BUSINESS PROBLEM

3 million Instacart orders have been open sourced. Instacart challenged the data science community to use this anonymized data on customer orders over time to predict which previously purchased products will be in users next order. In other words, the goal of this project is to predict grocery re-orders given a user's purchase history (a set of orders and the products purchased within each order), which of their previously purchased products will the repurchase in their next order ? The mean F1 score will be used to determine the best

CLIENT

The client would be Instacart's business analysis team. This report would also serve those in retail, who are looking to enhance their recommender systems for on-line purchases based on the history of purchases made by their customers

DATA SET INFORMATION

The dataset for this project is a relation set of files describing the customers' orders over time. There are 200,000 customers. There are 3,000,000 orders. Each customer has between 4-100 orders.

The description of the files is as follows.

- **products.csv**: This is a csv file that associates a product_id with a product_name, aisle_id and department_id. Products are stocked in aisles and departments.

```
product_id,product_name,aisle_id,department_id
1,Chocolate Sandwich Cookies,61,19
2,All-Seasons Salt,104,13
3,Robust Golden Unsweetened Oolong Tea,94,7
…
```

- **aisles.csv**: This .csv file associates an id to an aisle. A

```
aisle_id,aisle
1,prepared soups salads
2,specialty cheeses
3,energy granola bars    ...
```

- **department.csv**: This is also a csv file and maps an id to the department.

```
department_id,department
1,frozen
2,other
3,bakery
...
```

- **order_products_prior.csv, order_products_training.csv:** order_products_prior.csv contains the previous order contents for all customers. An observation with 'reordered' column indicates that the customer has a previous order that contains the product. Some observations will have no 'reordered' items. 'add_to_cart_order' indicates the age-order of the ordered product. The file order_products_traing.csv is used for training the model.

```
order_id,product_id,add_to_cart_order,reordered
1,49302,1,1
1,11109,2,1
1,10246,3,0
...
```

- **orders.csv :** This file indicates to which set an order belongs. The field 'eval_set' is one of {test, train, prior}. The field 'order_dow' indicates the day of the week.

```
order_id,user_id,eval_set,order_number,order_dow,order_hour_of_day,days_since
_prior_order
2539329,1,prior,1,2,08,
2398795,1,prior,2,3,07,15.0
473747,1,prior,3,3,12,21.0    ...
```

EXTRACTION, TRANSLATION AND LOADING

The code for the initial analysis located on github :

- **https://github.com/krajeshj/InstacartMBA/blob/master/code/py/InstacartMBA_DataWrangling.ipynb**

The input directory has all the .csv files listed in the section DATA-SET INFORMATION.

Further, all the dataset files could be easily loaded into respective pandas data-frames using a call to pandas read_csv() function call.

- **products_df** : The csv file, products.csv is read into a data frame. There are 49688 observations (rows) by 4 features(columns) in the products_df. There are no missing values. The product with product_id 5 is 'Green Chile anytime Sauce' and is relational to aisle with id aisle_id 61 and department_id 19 found in the aisles_df and departments_df described in table 2 and table 3.

	product_id	product_name	aisle_id	department_id
0	1	Chocolate Sandwich Cookies	61	19
1	2	All-Seasons Salt	104	13
2	3	Robust Golden Unsweetened Oolong Tea	94	7
3	4	Smart Ones Classic Favorites Mini Rigatoni Wit...	38	1
4	5	Green Chile Anytime Sauce	5	13

Table 1. Format of departments_df data frame . The product with product_id 5 is 'Green Chile anytime Sauce' and is relational to aisle with id aisle_id 61 and department_id 19 found in the aisles_df and departments_df described in table 2 and table 3.

- **aisles_df**: The csv file, aisles.csv is read into a pandas data frame. There are 134 rows and all observations are non-null.

	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

Table 2: This is the data frame for aisles_df. Aisle_id 1 corresponds to 'prepared_soups salads' aisle in the online grocery-shopping app.

- **department_df** : The departments.csv file is read into a data-frame with 21 rows of observations. There were no non-null values

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

Table 3: This is the data frame in tabular form for the departments_df

	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

order_products_prior_df: There are 32434488 rows X 4 columns and there are no missing values.

	order_id	*product_id*	*add_to_cart_order*	*reordered*
0	*2*	*33120*	*1*	*1*
1	*2*	*28985*	*2*	*1*
2	*2*	*9327*	*3*	*0*
3	*2*	*45918*	*4*	*1*
4	*2*	*30035*	*5*	*0*

Table 4 depicts the prior order. Here all observations happened to be order_id = 2, there were 5 products in the cart. One of them had a product_id of 9327 that was added 3rd in the card and had not been ordered before.

- **order_products_train_df**: There are 1384616 orders in the training set. These are the most recent orders and could be used for training the model. Therefore these are the most recent orders. They have the same columns as order_products_prior_df.

	order_id	product_id	add_to_cart_order	reordered
0	1	49302	1	1
1	1	11109	2	1
2	1	10246	3	0
3	1	49683	4	0
4	1	43633	5	1

Table 5. order_products_train_df is a data frame of the most recent order and is used to create a model. This is similar to the order_products_prior_df

- **orders_df: This is the comprehensive data on all orders from which one can find the user, the time and day the order was place, if this forms the part of prior (or train / eval) order and the days since prior order.**

	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	19
1	2398795	1	prior	2	3	7	15
2	473747	1	prior	3	3	12	21
3	2254736	1	prior	4	4	7	29
4	431534	1	prior	5	4	15	28

Table 6 The orders_df has the complete information on all orders. The order_id has user_id, eval_set, order_number, order_dow(day of the week) , order_hour_of day and days_since_prior_order

MISSING VALUES

There are 3421083 unique orders in the orders_df data frame. There were missing values. There were 206209 observations with missing values. This represented 6.02% of the total observations. The missing values were from the column days_since_prior_order. The data frame was cleaned and converted into orders_clean_df. The missing values were imputed using mean, grouped by user.

In summary, there was only one data set that required cleaning. The orders_df is a history of the all the orders. It had many NaNs in the days since prior order. Upon closer look, it appeared that the very first order would not have known value for 'days since prior value'. In theory we could drop this entry. But in the academic interest of cleaning the data, for each user, the mean 'days_since_prior_order' was used for imputation of the value.

EXPLORATORY DATA ANALYSIS

The code for exploratory data analysis is on github :
https://github.com/krajeshj/InstacartMBA/blob/master/code/py/InstacartMBA_DataStoryTelling.ipynb

THERE ARE 3 DATA-FRAMES OF INTEREST TO EXPLORE ORDERS

Orders.csv: Each order describes which products a user bought. This file provides

- **order_id** - unique identification of an order
- **user_id** - Who ordered the product
- **eval_set** - prior /train / or test
- **order_dow** - day on which it was ordered
- **order hour_of_the_day**
- **days_since_prior_order**

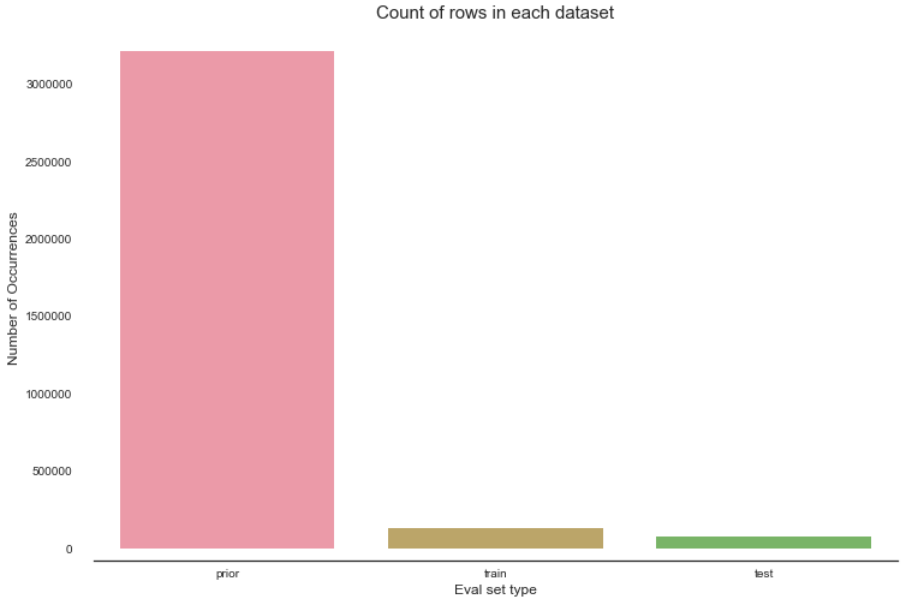
The prior eval set indicates a prior order. The he new orders would belong to either prior or test eval

order_products__*.csv: The order products file is an association between orders and the products that are in the order.

- order_id - unique id of the order
- product_id - product_id of the product ordered
- add_to_cart_order - the order in which the product was ordered e.g. Jam was ordered 1st, then bread, then eggs
- reordered - did this product appear in a prior order ?

CLASS OF EVALUATION SET in OBSERVATIONS IN ORDER_DF

Figure 1First all orders are classified as prior orders and last order. The last order is further divided into training set and test set. So prior order provides the history of a users ordering habit.



There are 1384617 observations in the order_df data set. Most orders are of eval_set t ype prior, i.e. historical orders data. The current order is divided into train and test data

NUMBER OF ITEMS IN AN ORDER

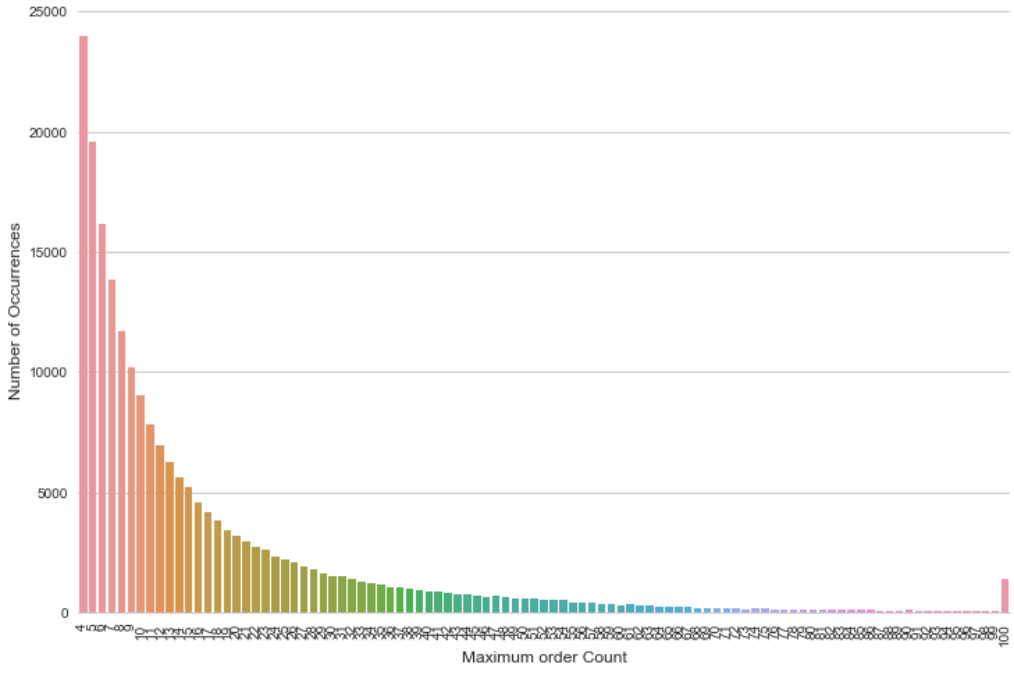


Figure 2 We see 23,986 orders had a max of 4 products on one end and about 47 orders had 99 items in them Each observation in orders_df data frame has order_number field. The order_number represents the order in which a user ordered products. The maximum of the order number per customer represents the customers ordering habits - how many products are ordered at the most by a user. We see ***23,986 orders had a max of 4 products*** on one end and about ***47 orders had 99 items*** in them

WHAT HOUR OF THE DAY DO MOST ORDERS ARRIVE

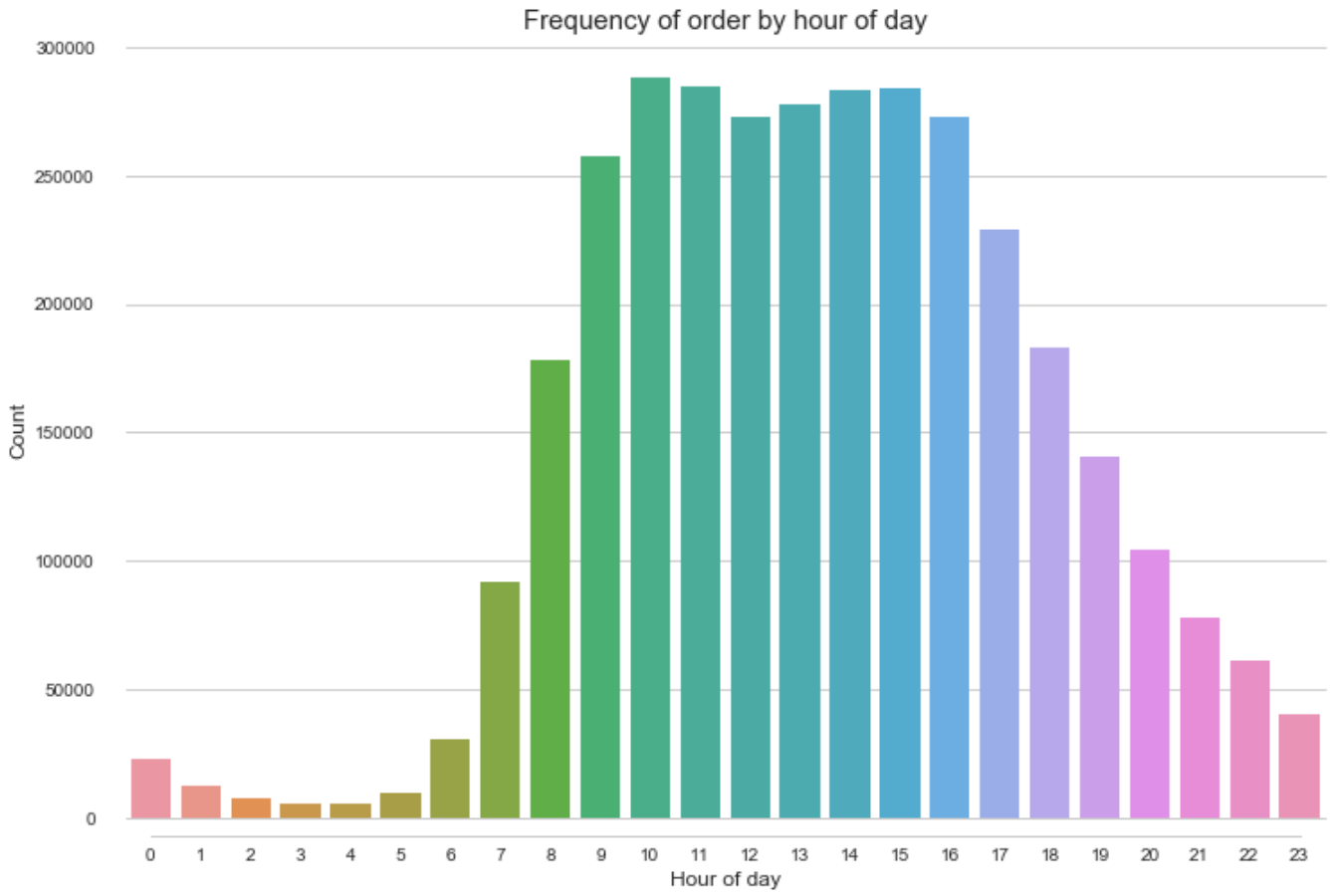


Figure 3. The peak ordering hours are from ***9.a.m. to 5 p.m.****. Most orders arrive at* ***10a.m.***

WHICH DAY OF THE WEEK DO MOST ORDERS ARRIVE

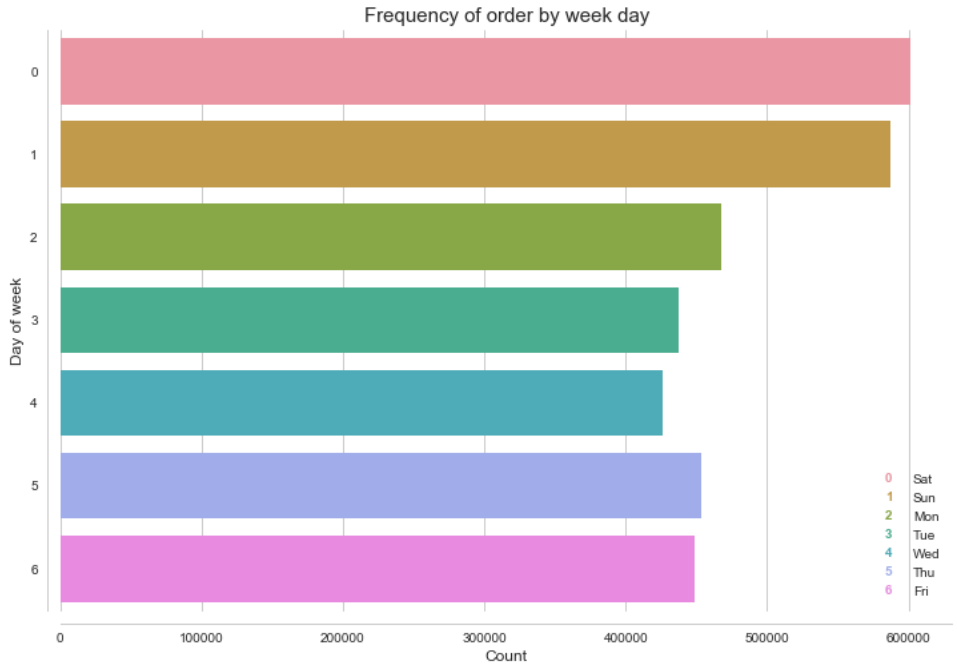


*Figure 4. Most orders arrive on **Saturday and Sunday***

WHEN DO MOST USERS ORDER ?

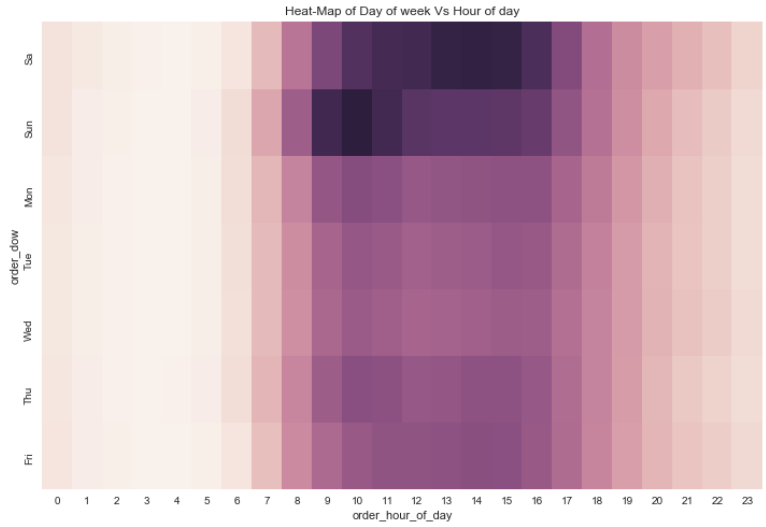


*Figure 5 : (above)Users order the most on **a Saturday afternoon or Sunday morning.***

HOW OFTEN DO USERS PLACE ORDER ?

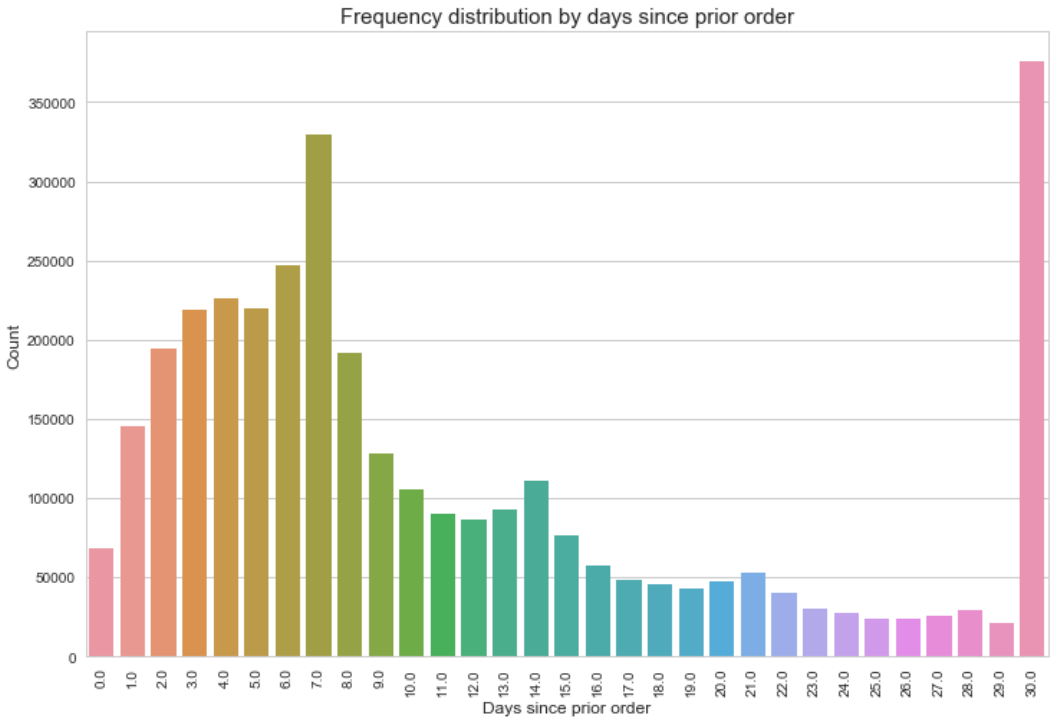


Figure 6: Most orders are ordered either ***every 30 days or every 6 or 7 days***

WHAT ARE THE TOP 16 ORDERS ?

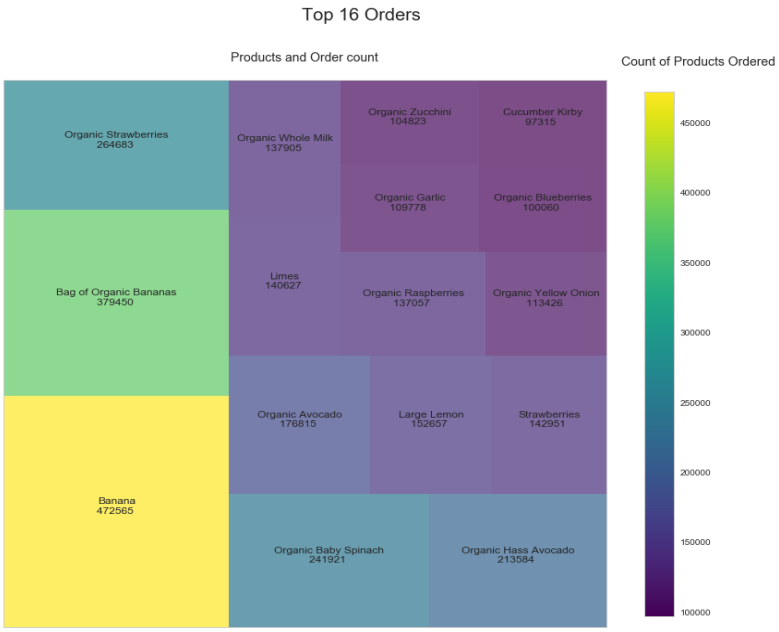


Figure 7: ***Bananas, Organic Bananas, Strawberries and Spinach*** *were top 3 items that were ordered.*

WHICH IS THE MOST RE-ORDERED ITEM ?

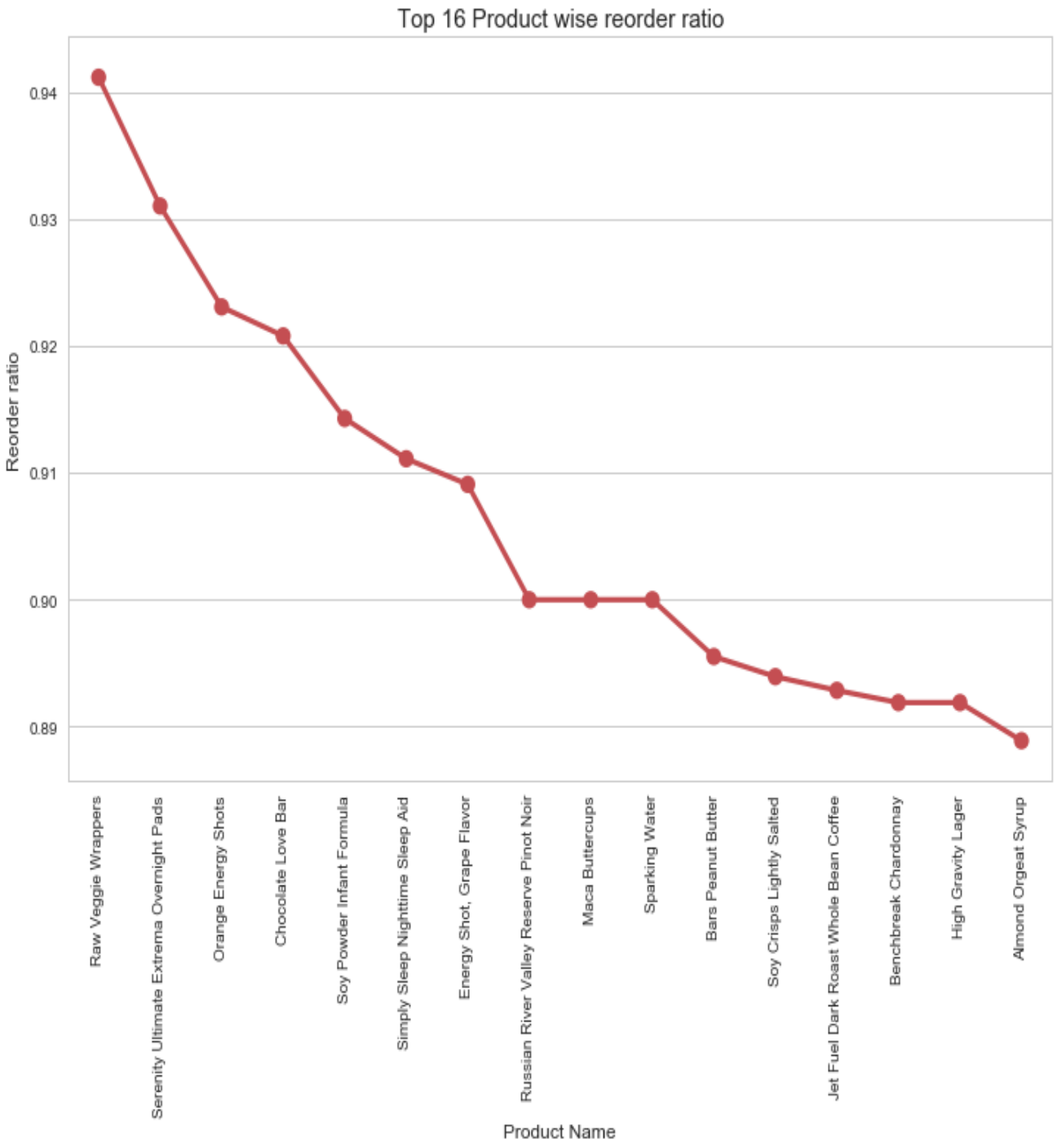


*Figure 8: The most re-ordered item is **Raw Veggie Wrappers***

WHICH DEPARTMENTS AND AISLES ARE MOST VISITED ?

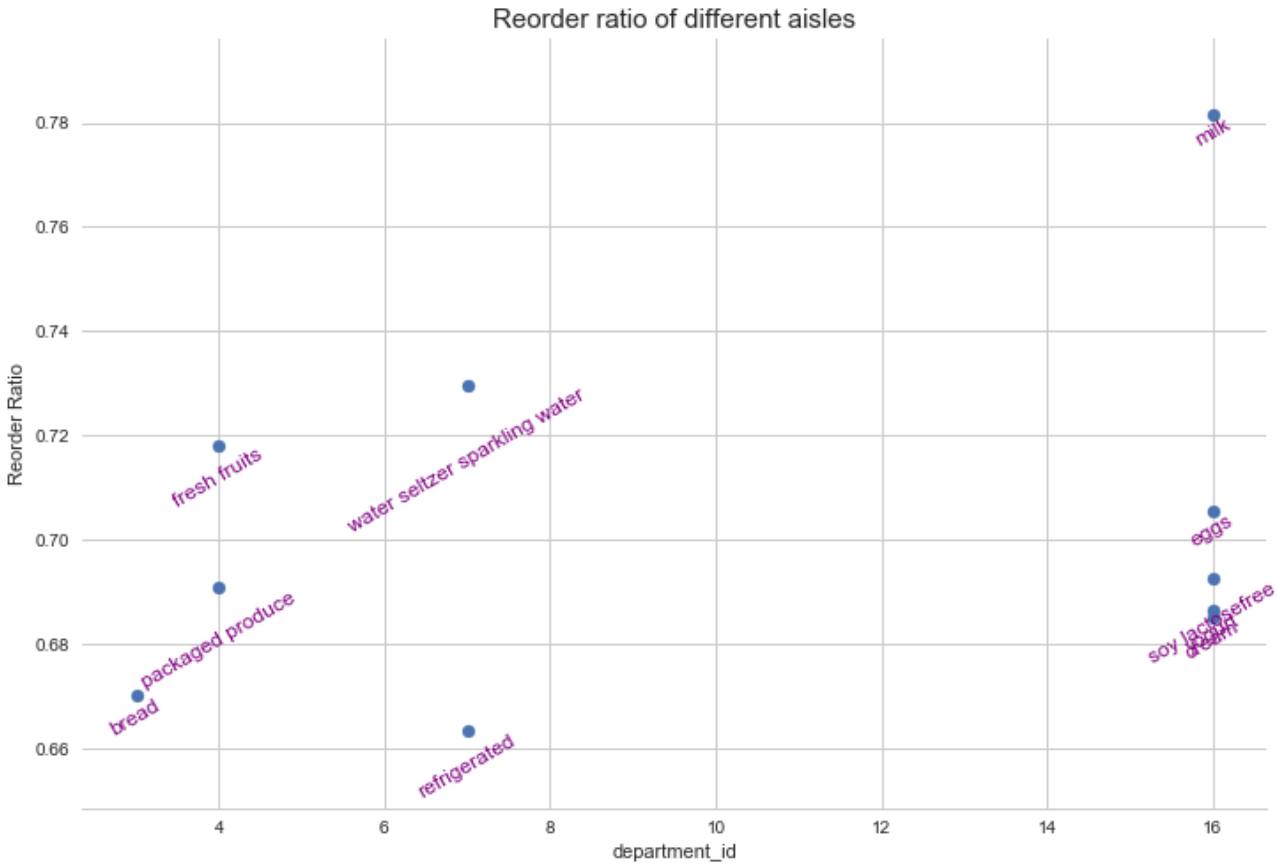


*Figure 9: **The aisles Milk, Water, Fresh Fruits** are visited the most for re-orders. Within a Department id 16 for example, one may recommend suggesting milk, eggs, and lactose free aisles for reorder.*

ADD-TO-CART ORDER – REORDER RATIO

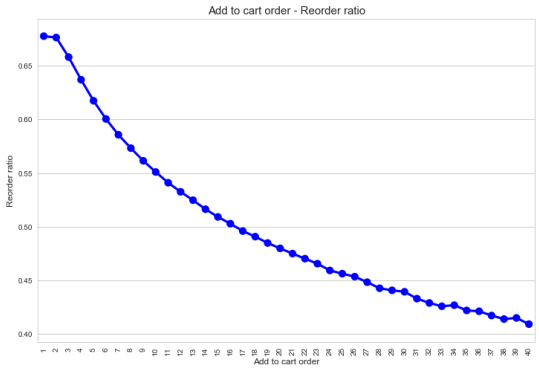


*Figure 10: (above) there is a **strong probability of a re-order** if the product was **added earlier in the order**. Customers tend to remember the most frequently used items and order them before browsing for new products.*

WHEN ARE RE-ORDERS LIKELY TO OCCUR?

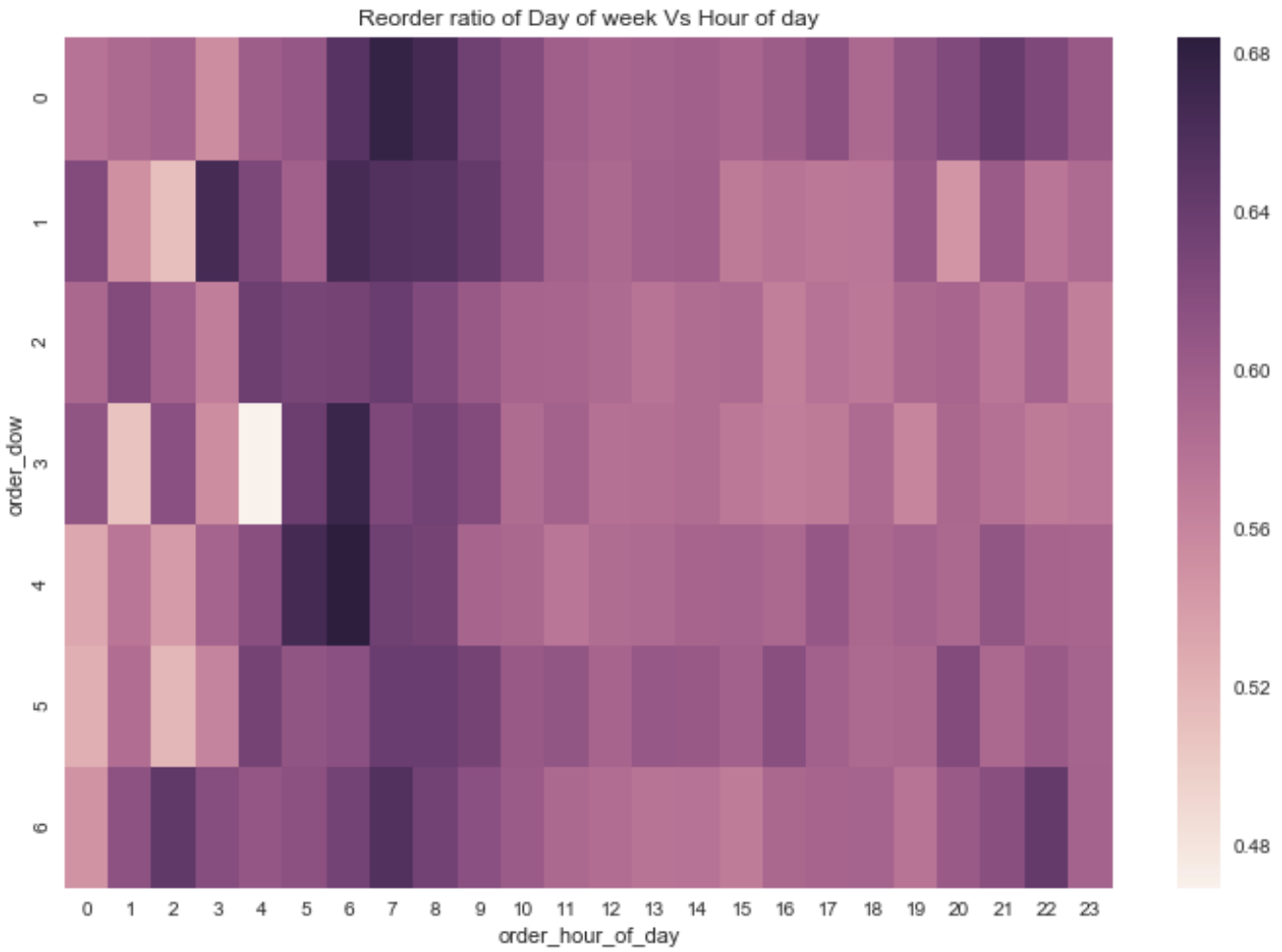


Figure 11Reorders mostly ***occur mid-week*** *and in early hours of the day* ***between 6 am and 8 am.***

SUMMARY:

- Peak orders Arrive at 10 am
- Sat and Sunday are the most popular days to order
- Saturday afternoon and Sunday Morning are busy days for ordering
- Most customers order every 7 days or every 30 days
- Bananas are the most ordered items
- Raw Veggie wrappers and Sanitary Pads are most reordered items
- Most popular aisle to re-order from are Milk, Water and Fresh Fruits
- For a product if the age order was low, it is likely to be re-ordered
- Most Reorders arrive in the Morning