Problem

Instacart, a grocery ordering and delivery app, aims to make it easy to fill your refrigerator and pantry with your personal favorites and staples when you need them. After selecting products through the Instacart app, personal shoppers review your order and do the in- store shopping and delivery for you.

Instacart uses customers' transactional data to develop models which recommend products that users will buy again based on their previous purchases. The goal of the project is to predict which previously purchased products will be in a user's next order. By providing an optimal and accurate recommendations of products to purchase, Instacart can enhance customers overall shopping experience, browsing experience, increase revenue from sales, and increase overall customer satisfaction.

Data

The data can be obtained from kaggle, https://www.kaggle.com/c/instacart-market-basket-analysis/data. The dataset is a relational set of files describing Instacart customers' orders over time. The dataset is anonymized and contains a sample of over 3 million grocery orders from more than 200,000 Instacart users. For each user, Instacart provides between 4 and 100 of their orders, with the sequence of products purchased in each order. Instacart also provide the week and hour of day the order was placed, and a relative measure of time between orders.
The data comprises of six different csv files and most of them are self-explanatory. Snapshot of the csv files are included below.

aisles.csv:

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

departments.csv:

```
department_id,department
1,frozen
 2,other
```

3,bakery
…

order_products__*.csv:
These files specify which products were purchased in each order.
order_products__prior.csv contains previous order contents for all customers.
'reordered' indicates that the customer has a previous order that contains the product.
Note that some orders will have no reordered items.
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 tells to which set (prior, train, test) an order belongs. 'order_dow' is the day of 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
...

products.csv:
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
...

Since the dataset was obtained from Kaggle, the data is already pretty clean. The only missing values are inside the days_since_prior_order column of orders.csv. There are some “NaN” values in that column which most likely represent the first order of a particular user. Since it’s the first order, there is no prior order so days_since_prior_order is “NaN”.

```
print(orders.info())
orders.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3421083 entries, 0 to 3421082
Data columns (total 7 columns):
order_id                int64
user_id                 int64
eval_set                object
order_number            int64
order_dow               int64
order_hour_of_day       int64
days_since_prior_order  float64
dtypes: float64(1), int64(5), object(1)
memory usage: 182.7+ MB
None
```

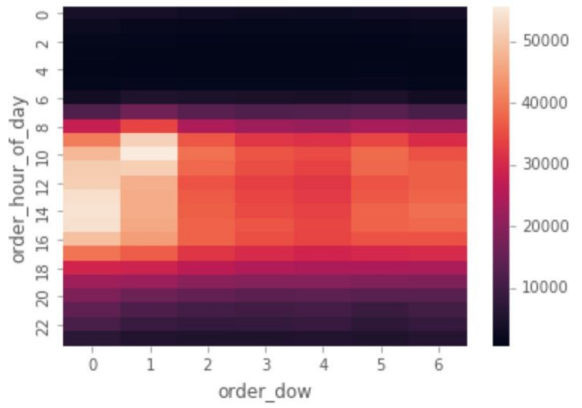
	order_id	user_id	eval_set	order_number	order_dow	order_hour_of_day	days_since_prior_order
0	2539329	1	prior	1	2	8	NaN
1	2398795	1	prior	2	3	7	15.0
2	473747	1	prior	3	3	12	21.0
3	2254736	1	prior	4	4	7	29.0
4	431534	1	prior	5	4	15	28.0

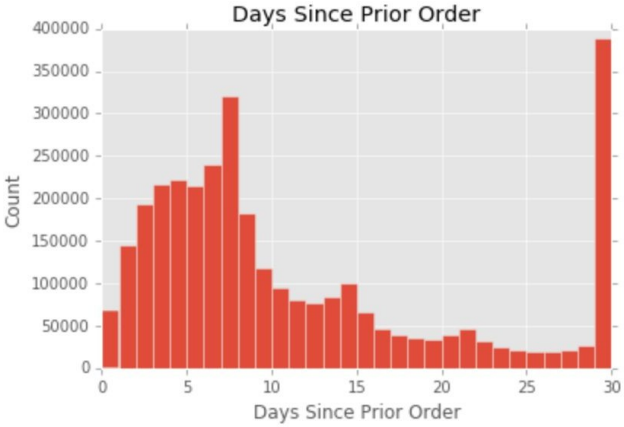
The “NaN” comprises about 6% of total number of rows in the orders data. When doing the EDA involving the “days_since_prior_order” column, these “NaN” data are excluded. In the modeling part of the project, these “NaN” can be replaced by “-1” and the model should be able to learn what is the meaning of “-1”.

Initial Findings

Analysis of Orders Data:
From the plot below, we can see that Day 0 and Day 1 have the most number of orders. It is unclear what days Day 0 and Day 1 represent. The peak hours are between 9AM and 5PM but from this plot we cannot really see clearly the corresponding day and time combination.

Use heatmap (below) instead. From the heatmap, the peak day and hours combination occurs on day 0 and day 1 between 9AM and 5PM. From the Days Since Prior Order plot, we can see that a lot of customers put another order after a week or a month which makes sense because people tend to reorder after a week or a month.





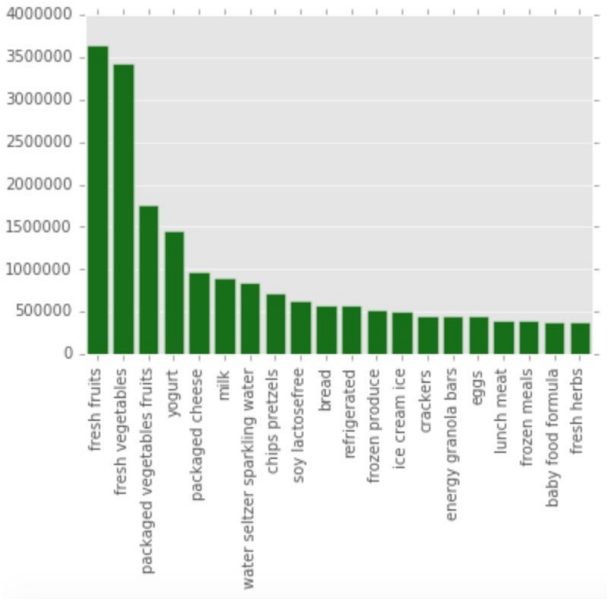
Analysis of Products Data:

Product	Count
Banana	472565
Bag of Organic Bananas	379450
Organic Strawberries	264683
Organic Baby Spinach	241921
Organic Hass Avocado	213584
Organic Avocado	176815
Large Lemon	152657
Strawberries	142951
Limes	140627
Organic Whole Milk	137905
Organic Raspberries	137057
Organic Yellow Onion	113426
Organic Garlic	109778
Organic Zucchini	104823
Organic Blueberries	100060
Cucumber Kirby	97315
Organic Fuji Apple	89632
Organic Lemon	87746
Apple Honeycrisp Organic	85020
Organic Grape Tomatoes	84255

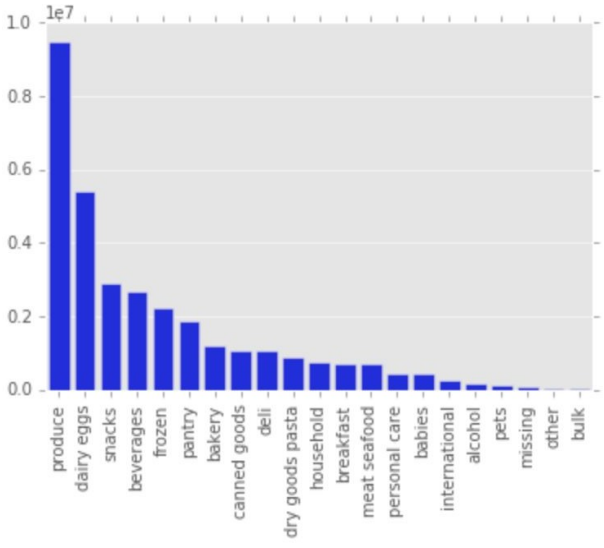
Name: product_name, dtype: int64

From the table above, the top 20 products are mostly fruits and vegetables (except Whole Milk). The bar plot of top aisles below confirms that the top products and aisles are those of fruits and vegetables. From the bar plot of top departments below, produce department dominates orders which is consistent with fruits and vegetables as the top products.

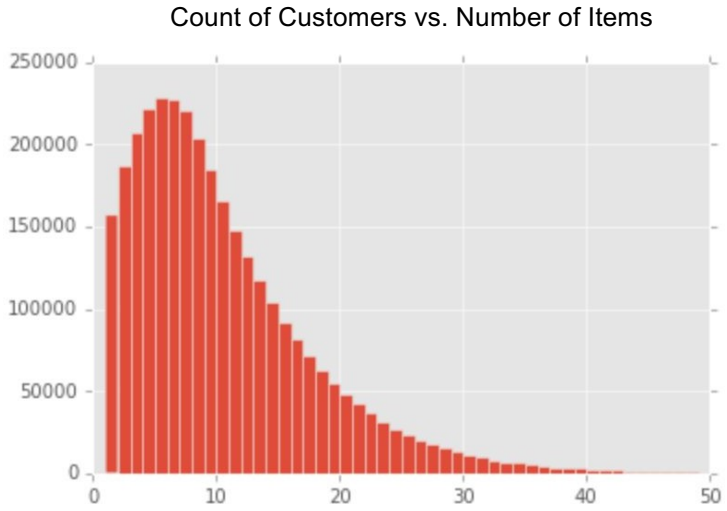
Top 20 Aisles



Top 20 Departments



The bar plot below shows the count of customers (y-axis) that buy how many items (x-axis). A lot of customers bought 4 to 7 products per order.



The bar plot below shows top 20 products that has the highest reorder rate (~90%).