**Insta Market Basket Analysis**

**Big Data and Intelligence Analytics**

**INFO 7245 - SPRING 2018**

PROFESSOR:

**Nicholas Brown**

TEAM MEMBERS:

**Sameer Suman (**[www.linkedin.com/in/sameersuman/](http://www.linkedin.com/in/sameersuman/)**)**

Table of Contents

[Problem Statement 3](#_Toc505376356)

[Main Approach 3](#_Toc505376357)

[Data 4](#_Toc505376358)

[aisles.csv 4](#_Toc505376359)

[departments.csv 4](#_Toc505376360)

[order\_products\_\_\*.csv 5](#_Toc505376361)

[orders.csv 5](#_Toc505376362)

[products.csv 5](#_Toc505376363)

[Part 1: Data wrangling and exploratory data analysis 5](#_Toc505376364)

[Data cleaning 5](#_Toc505376365)

[Classification 15](#_Toc505376366)

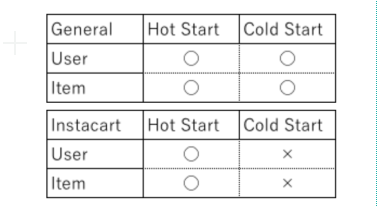
# **Problem Statement**

Recently [Instacart Market Basket Analysis competition](https://www.kaggle.com/c/instacart-market-basket-analysis) challenged Kagglers to predict which grocery products an Instacart consumer will purchase again and when. Imagine, for example, having milk ready to be added to your cart right when you run out, or knowing that it's time to stock up again on your favorite ice cream.

This focus on understanding temporal behavior patterns makes the problem fairly different from standard item recommendation, where user needs and preferences are often assumed to be relatively constant across short windows of time. Whereas Netflix might be fine assuming you want to watch another movie similar to the one you just watched, it's less clear that you'll want to reorder a fresh batch of almond butter or toilet paper if you bought them yesterday.

 The goal of this competition was to predict grocery reorders: given a user’s purchase history (a set of orders, and the products purchased within each order), which of their previously purchased products will they repurchase in their next order?

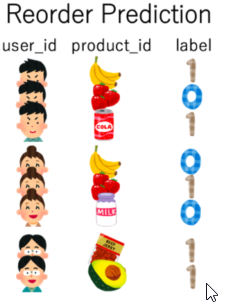
The problem is a little different from the general recommendation problem, where we often face a cold start issue of making predictions for new users and new items that we’ve never seen before. For example, a movie site may need to recommend new movies and make recommendations for new users.



The sequential and time-based nature of the problem also makes it interesting: how do we take the time since a user last purchased an item into account? Do users have specific purchase patterns, and do they buy different kinds of items at different times of the day? And the competition’s F1 evaluation metric makes sure our models have both high precision and high recall.

## Main Approach

I used XGBoost to predict: **Predicting reorders** - which previously purchased products will be in the next order? This model depends on both the user and product.



.

## **Data**

The dataset for this competition is a relational set of files describing customers' orders over time. The goal of the competition is to predict which products will be in a user's next order. The dataset is anonymized and contains a sample of over 3 million grocery orders from more than 200,000 Instacart users. For each user, we are provided between 4 and 100 of their orders, with the sequence of products purchased in each order. We also provide the week and hour of day the order was placed, and a relative measure of time between orders. For more information, see the [blog post](https://tech.instacart.com/3-million-instacart-orders-open-sourced-d40d29ead6f2) accompanying its public release.

Each entity (customer, product, order, aisle, etc.) has an associated unique id. Most of the files and variable names should be self-explanatory.

### **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. You may predict an explicit 'None' value for orders with no reordered items. See the evaluation page for full details.

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. You are predicting reordered items only for the test set orders. '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

# **Part 1: Data wrangling and exploratory data analysis**

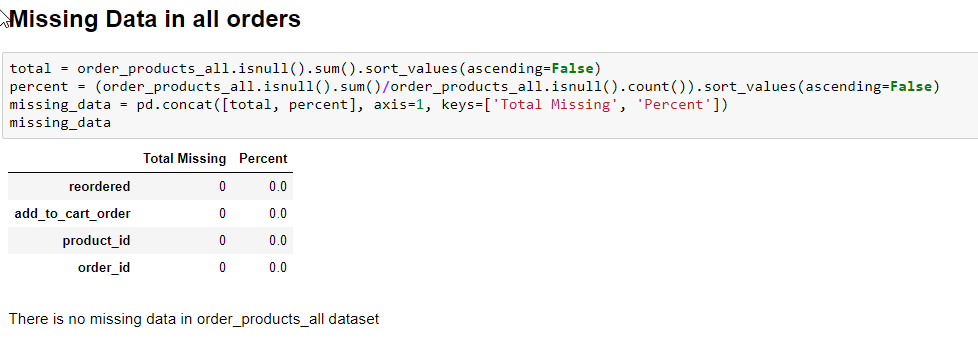
In this section, we will perform the data the following operations:

* Data cleaning
* Exploratory data analysis in Python

## **Data cleaning**

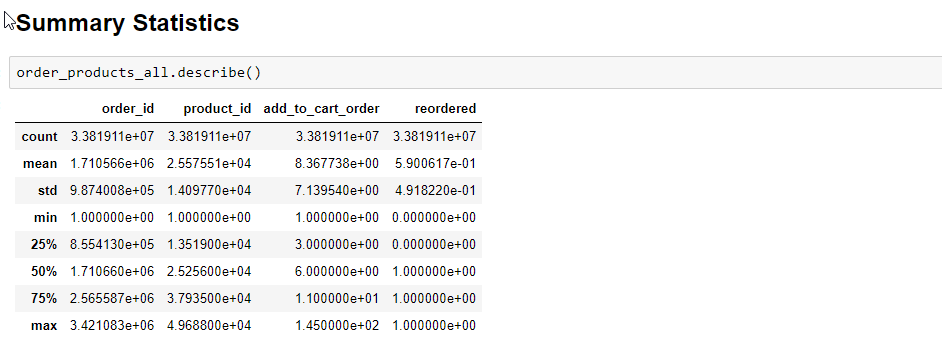
This section comprises of handling missing data. We get the concatenated .CSV file from the previous part, on which we perform data cleaning steps.

I first analyzed all the columns having empty values as shown below:



Above report suggested that there are no missing records or columns with missing values. Data is absolutely clean and that there is no leak.

Secondly, I analyzed the data for any inappropriate data.



Data provided to us has appropriate values.

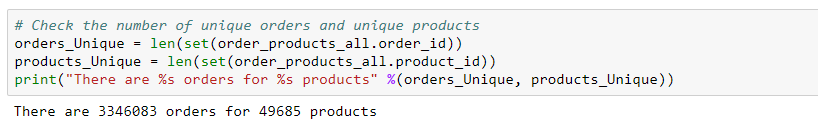
Data Distribution



I am trying to analyze data distribution by answering following questions:

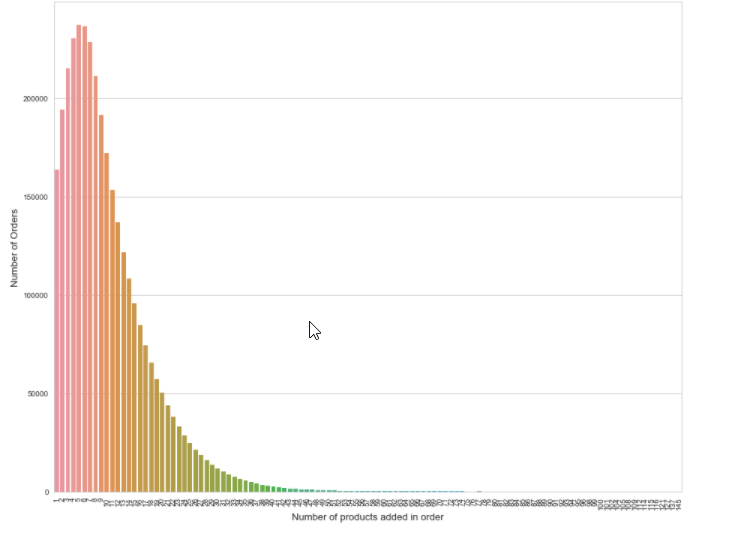
1. How many are unique customer of Instacart
2. How many number or product purchase average per customer?
3. What is the reorder frequency?
4. Which products are frequently reordered?
5. What’s the most popular ordering hours and day
6. Can we segment by customers by product order i.e identifying aisles and favourite department of customers?

Here are snapshot of the codes alongwith description

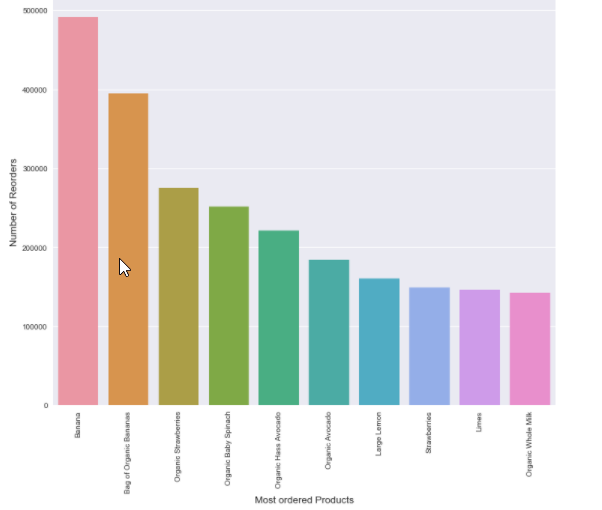


This is a basic distinct query on order id and product\_id all the list of orders\_products data frame.

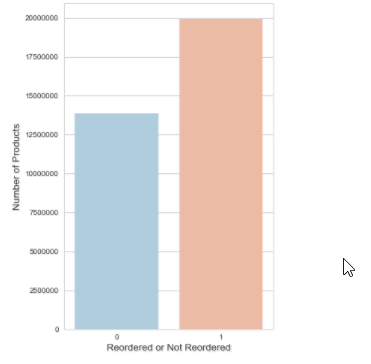
2. I analysed number of products that people usually order



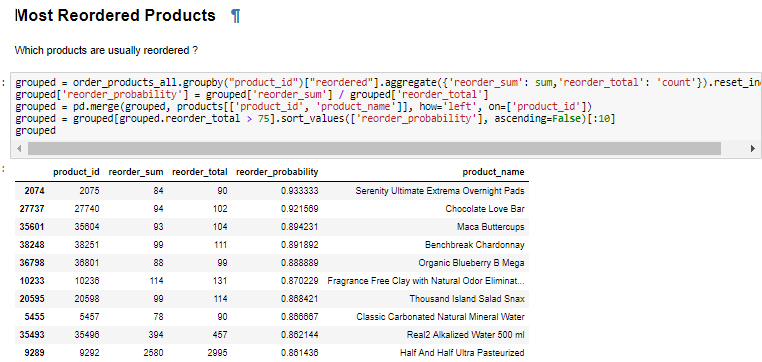
Most ordered product



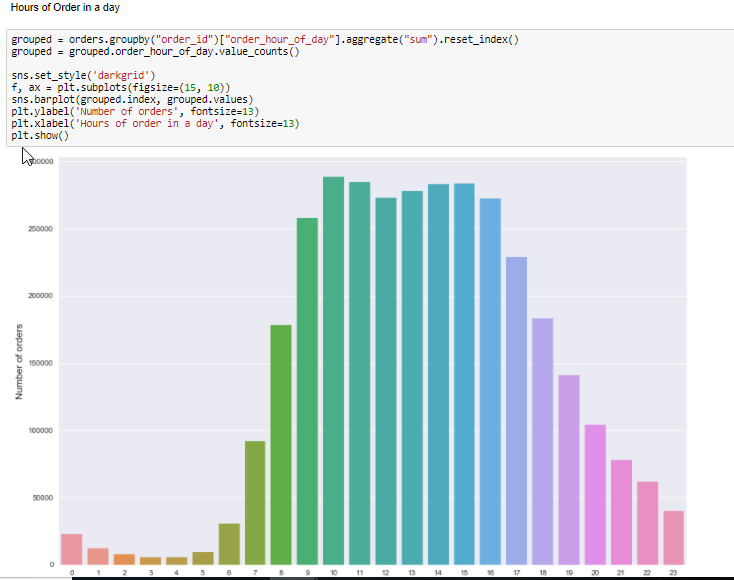
Reorder frequency



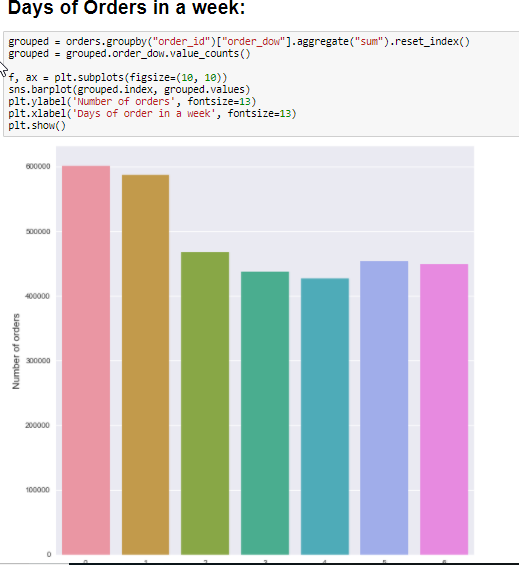
Most reordered Product



Hours of the day when people purchased the most

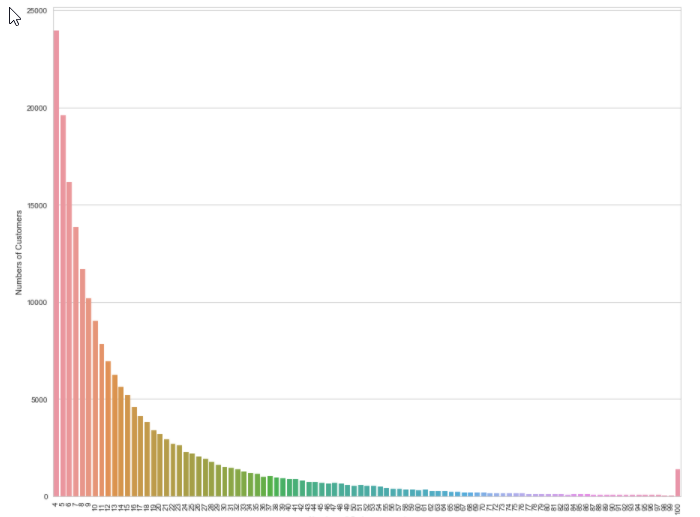


Days of the week when people purchased the most



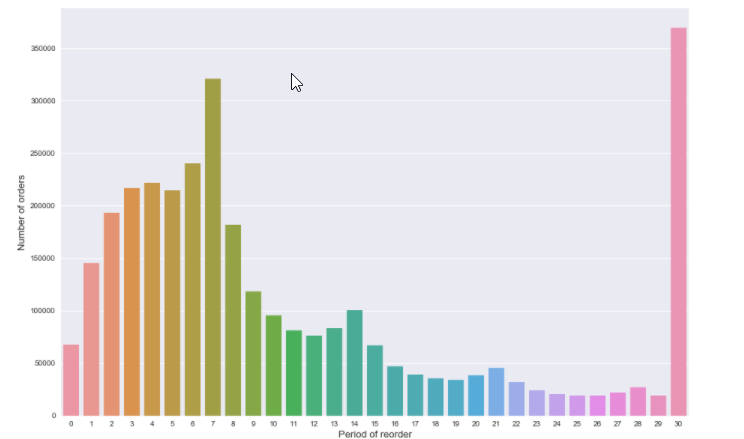
Orders made by each customer

Most customers ordered 4-100 products. The dataset is anonymized and contains a sample of over 3 million grocery orders from more than 200,000 Instacart users. For each user, we are provided between 4 and 100 of their orders, with the sequence of products purchased in each order. We also provide the week and hour of day the order was placed, and a relative measure of time between orders. For more information, see the [blog post](https://tech.instacart.com/3-million-instacart-orders-open-sourced-d40d29ead6f2) accompanying its public release.



**I also analysed** most important department and most important aisles

Period of reorder ranges from 7 days to 30 days

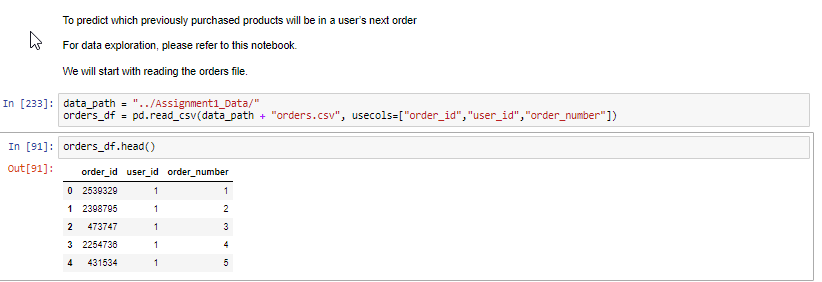


Orders data is split among prior, train and test data

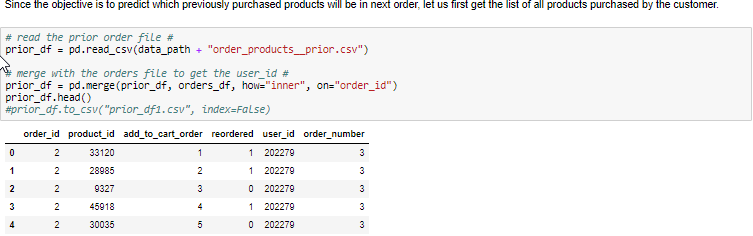
Part B- Data Analysis using XgBoost

I will break down my machine learning implementation in steps

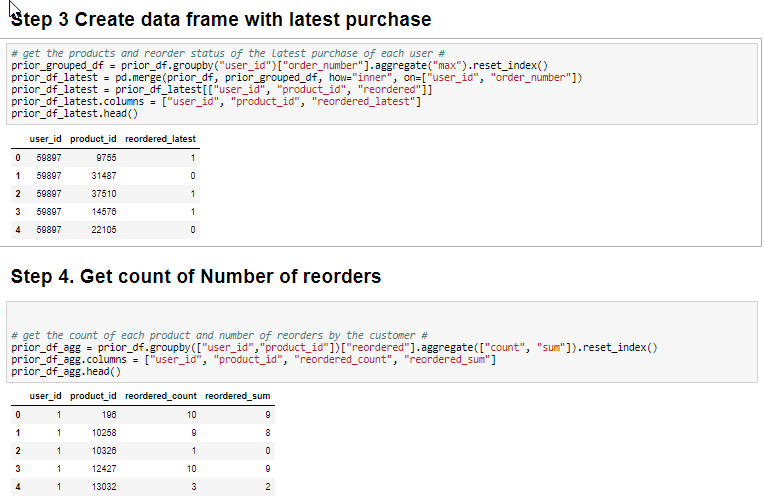
Step 1: Read the Order file

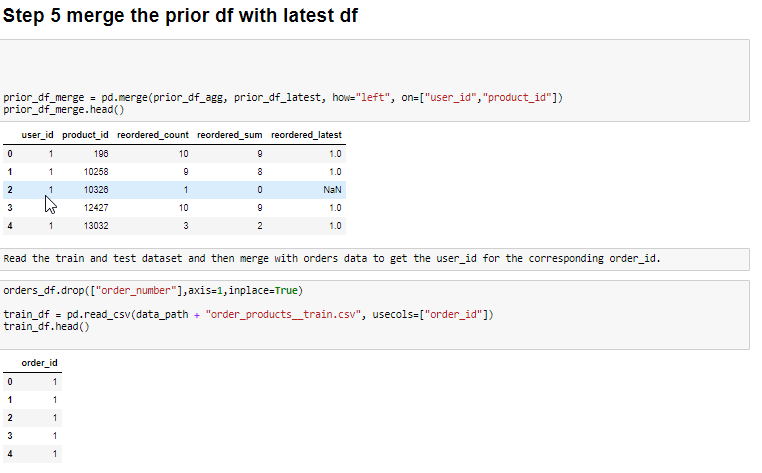


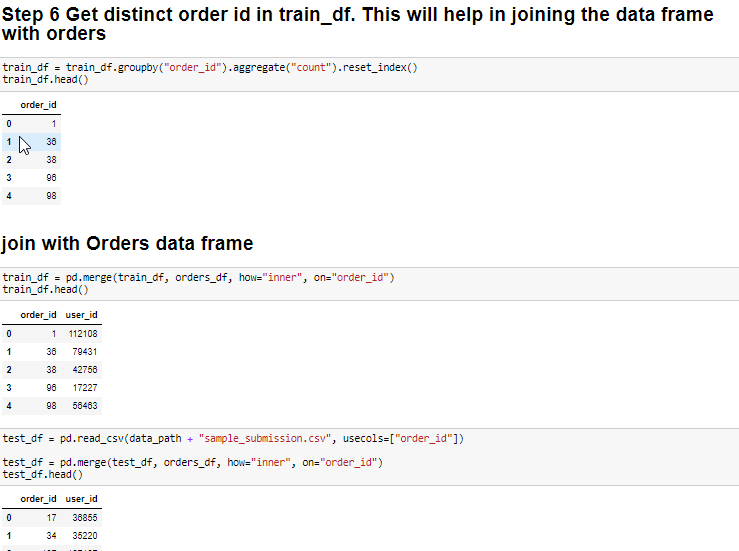
2. Read the prior file. Do an inner join on Orders table on order id

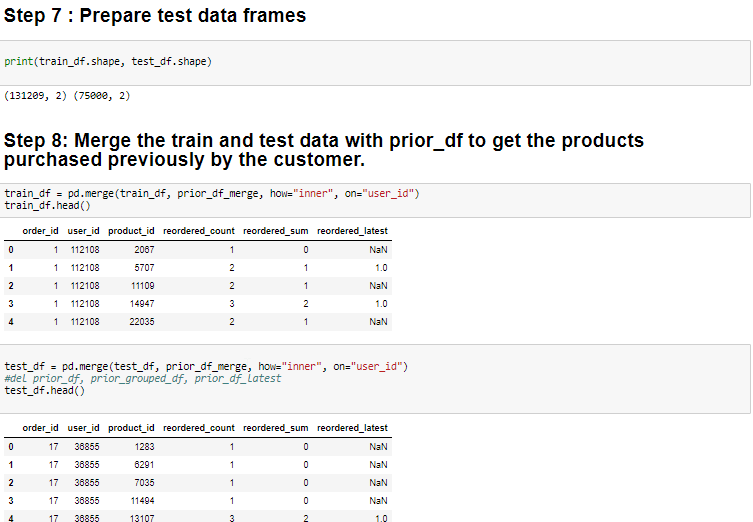


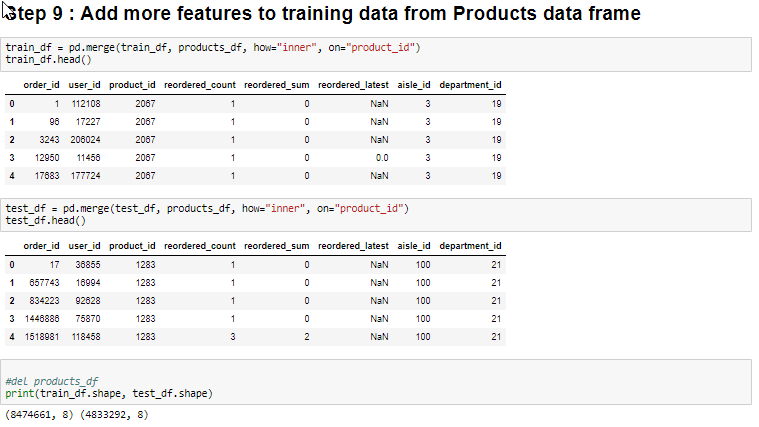
3.

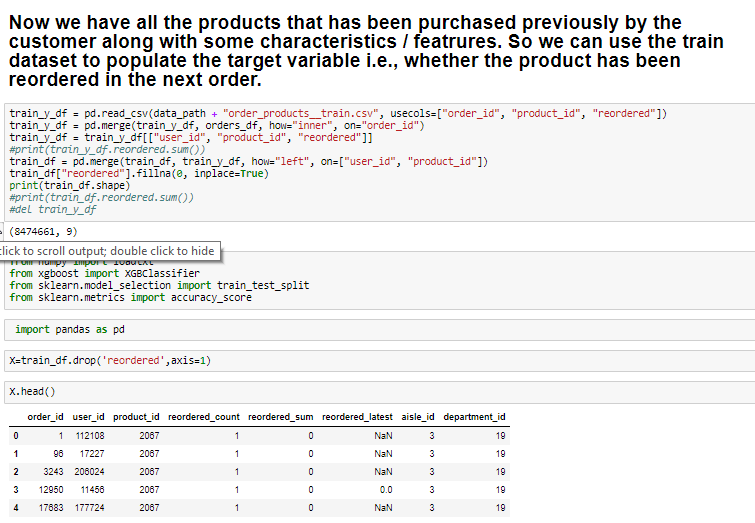


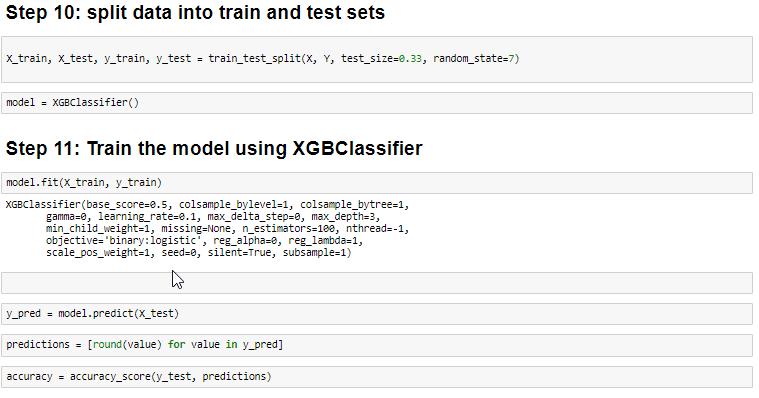














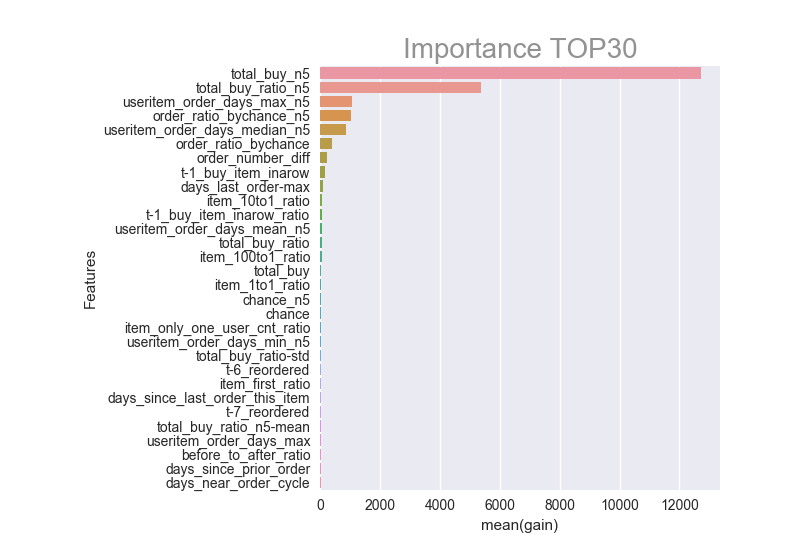


Conclusion

I applied XGBoostClassifier to predict if the product against order will be repeated or nont. My accuracy is 90.43%. I also used test data provided on the Kaggle to identify those products.

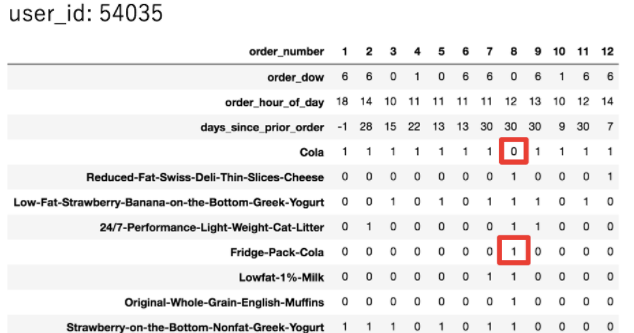
### Which features were the most useful?

For the reorder prediction model, we can see that the most important feature



Let’s think about the reordering problem. Common sense tells us that an item purchased many times in the past has a high probability of being reordered. However, there may be a pattern for when the item is not reordered. We can try to figure out this pattern and understand when a user doesn’t repurchase an item.

For example, consider the following user.



This user pretty much always orders Cola. But at order number 8, the user didn’t. Why not? Probably because the user bought Fridge Pack Cola instead. So I created features to capture this kind of behavior.

**I**mportant Finding for Reorders - #2

**Days\_since\_last\_order\_this\_item(User A, Item B)** is a feature I created that measures the number of days that have passed since User A last ordered Item B. This also played an important role in identifying the results.

References:

[www.kaggle.com](http://www.kaggle.com)