Instacart Shopping Data

Data wrangling

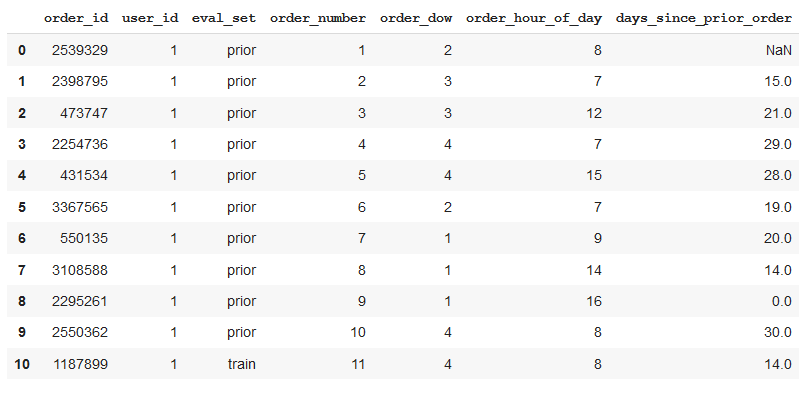
Sava Dashev

For my Capstone One Project I use old Kaggle competitions dataset. It is from 2017 competition. This is real data collected over time. It is anonymized to protect the identities of customers and retailers. The data does not contain actual prices for products.

Data

The dataset is distributed in 5 separate csv files. The main file is orders.csv. This file contains data about individual orders:

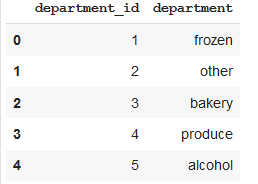
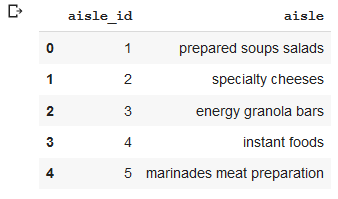
* the customer identifier - user\_id,
* what day and time was the order made - order\_dow and order\_hour\_of\_day,
* how many days passed since prior order was made - days\_sinse\_prior\_order,
* data is divided into three tiers - coded in eval\_set,
* and of course each order has randomized identification - order\_id.

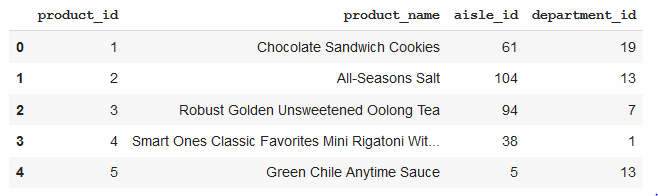


The other four files include:

1. Description of products

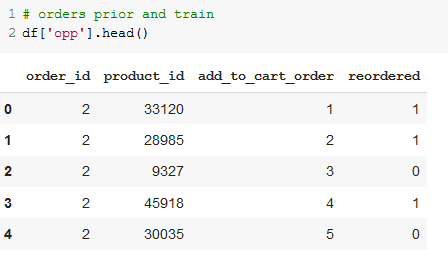
* Products file contains names and id-s for products and id-s for corresponding aisles and departments
* Aisles file contains aisle id and name;
* Department file contains department id and name.

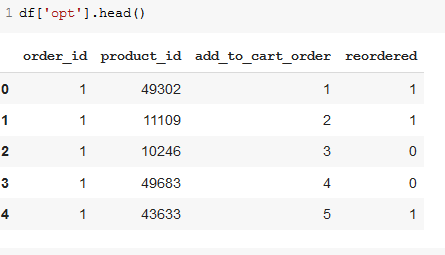




1. Description of ordered items.

* The data is distributed into two files orders prior and orders train
* Each of the two files contain identical data. Each row has variable linking it with orders table - order\_id, and products table - product\_id, also two other pieces of information - in which order was the product ordered (1, 2, 3,...) and also was that product reorder.

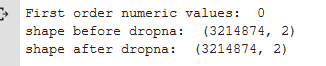




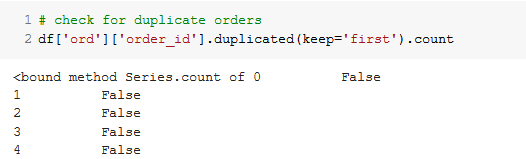
Checking data

The data was relatively clean. We needed to check data in each file for completeness. We read each file into separate dataframe. We checked the head of each dataframe and the info for the dataframe.

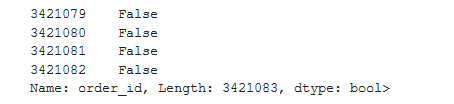
For the orders dataframe we checked all numerical data. All variables have the same number of entries with the exception of days\_since\_prior\_order. When new user logs in, his/her first order has NaN as value.



Next, we checked if there are duplicate orders for prior, test and train sets. All order id-s are unique, so each order belongs to exactly one set.



and the end



Combining data

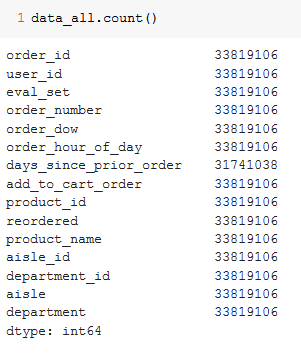
We combined data for EDA and further processing.

First, we combined prior and train dataframes into one dataframe.

We merged combined data and orders dataframe into data\_all frame.

We merged together products, aisles and departments dataframes.

As a last step for combining data, we merged data\_all dataframe and the products dataframe.



We found that individual rows can be identified by user, order number and add to card order. We set these to be index for the dataframe. We sort the dataframe.

The test orders, which have no ordered products, was separated into different data frame.

Finishing

We checked that last version of data\_all. We counted all entries for each variable to be sure that and save it as a csv file for further processing. Also, we saved test orders data into small csv file for EDA.

