**Instacart Market Basket Analysis**





**Moran Almagor & Shira Frank**

**Table of Contents**

1. **Introduction……………………………………………………………………………3**
2. **Methodology……………………………………………………………………………4**

2.1 Exploratory data analysis………………………………………..……4

2.2 Clear outcome variable definition………………………………..…..4

2.3 Variable engineering………………………………..…………………4

2.4 Missing values ………………………………………….……..………7

2.5 Outliers determination and treatment………………………………..7

# 

# **1.Introduction**

Whether we shop from meticulously planned grocery lists or let whimsy guide our grazing, our unique food rituals define who we are. Instacart, a grocery ordering and delivery app, aims to make it easy to fill our refrigerator and pantry with our personal favorites and staples when we need them.

Retail is one of the domains that collects huge amount of transaction data everyday. They need to understand their customer’s purchasing patterns, find insights and behaviors in order to take better business decisions[1], in order to know which orders to make, discounts, sales, and predict what will be the amount of revenues in the near future. Those are valuable also to many E-Commerce websites, such as book sales on amazon.com, DVD rental service on netflix.com etc.  
The prediction based on personalisation algorithm[2]. These algorithms model consumer shopping behavior and are used to automatically identify items that are new to the individual consumer, but are likely of interests to them.

The main goal in our project, is to predict a list of suggestion products to each user according to the predictions of the models that we will use.

The prediction[3] will be according to various parameters that are included in the Instacart data. Examples of the parameters are day of the week and time the orders were placed, in what order items were placed in the shopping cart, and which virtual departments and categories the purchased products belong to.

Previous works had been done by using models like "basket-sensitive random walks"[3], "SVD approximations[4]", "Support Vector Machine", "Logistic Regression". The AUC is roughly around 0.64 – 0.68.   
Those are the most common models to predict customer’s purchasing patterns, because they are supervised machine learning algorithm which can be used for both classification or regression problems.

## **2. Methodology**

**2.1 Exploratory data analysis**

The source of the data had been taken from a competition that had been held on august, 2017 in kaggle website. The dataset of "Instacart" contains a sample of over 3 million grocery orders from more than 200,000 "Instacart" users, 134 categories, 22 departments.  
The dataset is anonymized and does not contain any customer information, such as: geographic area of the users, real ID number, and the exact dates of purchases due to user privacy.

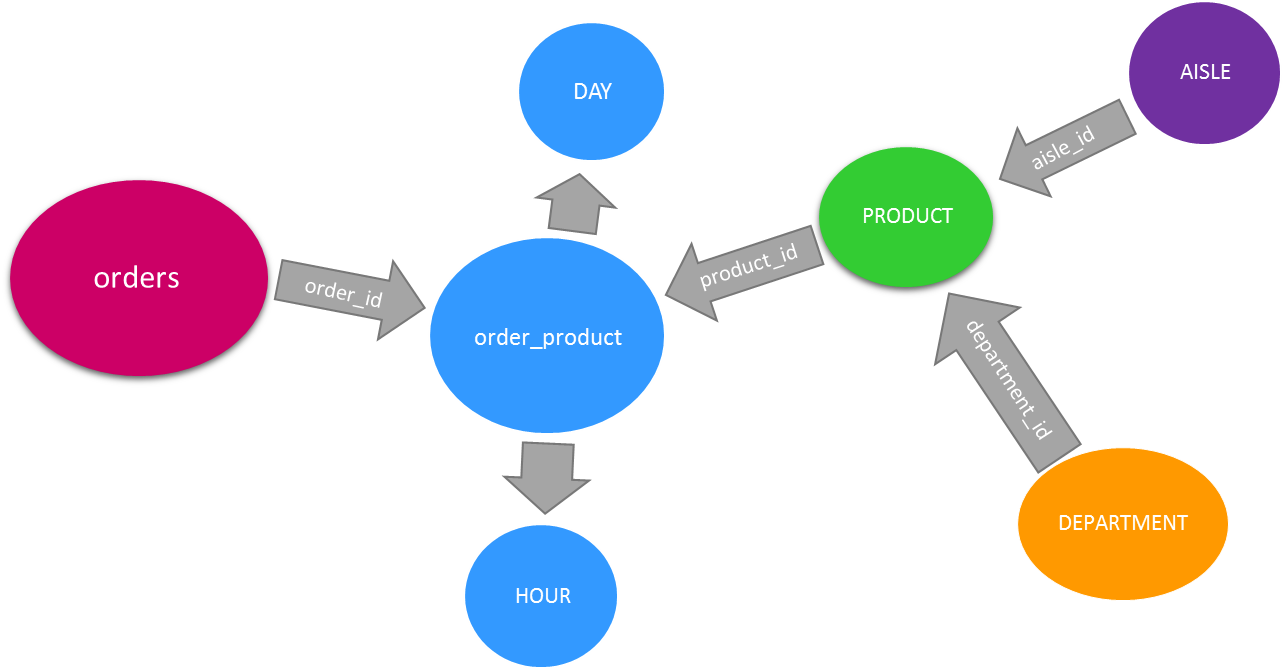
The data is available in 7 csv files :  
orders, departments, aisles ( categories ), products, order\_products\_\_train, order\_products\_\_prior, sample\_submission.

We will focused on 5 tables, that contains the relevant information; the first file contains products information like product id, product name, aisle id (where the product is placed in the store) and the department id. The second file contains orders and it links each order id with the customer who ordered the same. It also contains the day of the week, hour of day and sequence in which the order was placed. The third file contains the products purchased in each order with the sequence in which the product was added to the cart and whether the customer reordered it or not. The fourth and fifth files are the metadata for aisles and departments respectively, containing unique ids and names for each. For each user, there are multiple orders and each order has multiple products[1].  
We crossed the information from the different tables to one big table that contains the most important columns, "flat file" – user\_id, order\_id, order\_number, order\_dow (day of week), order\_hour\_of\_day, day\_since\_prior\_order, reordered, department\_id.

Due to the big size of the dataset that we got, above 30 millions of records, we couldn't succeed to upload all the data into python from SQL server, since of reasons of limited memory, we decided to cut the dataset.

We decided to cut the data by classifying the top 30 products that were sold. Any customers who did not buy from this products list had been delete, thus we were left with approximately 4.3 million records, instead of 30 million, and the amount of memory required for the project has been decreased significantly, which allows convenient and fast work.

# Entity Relationship Diagram (ERD):



The subjects in the project are exclusion criteria.

**2.2 Clear outcome variable definition**

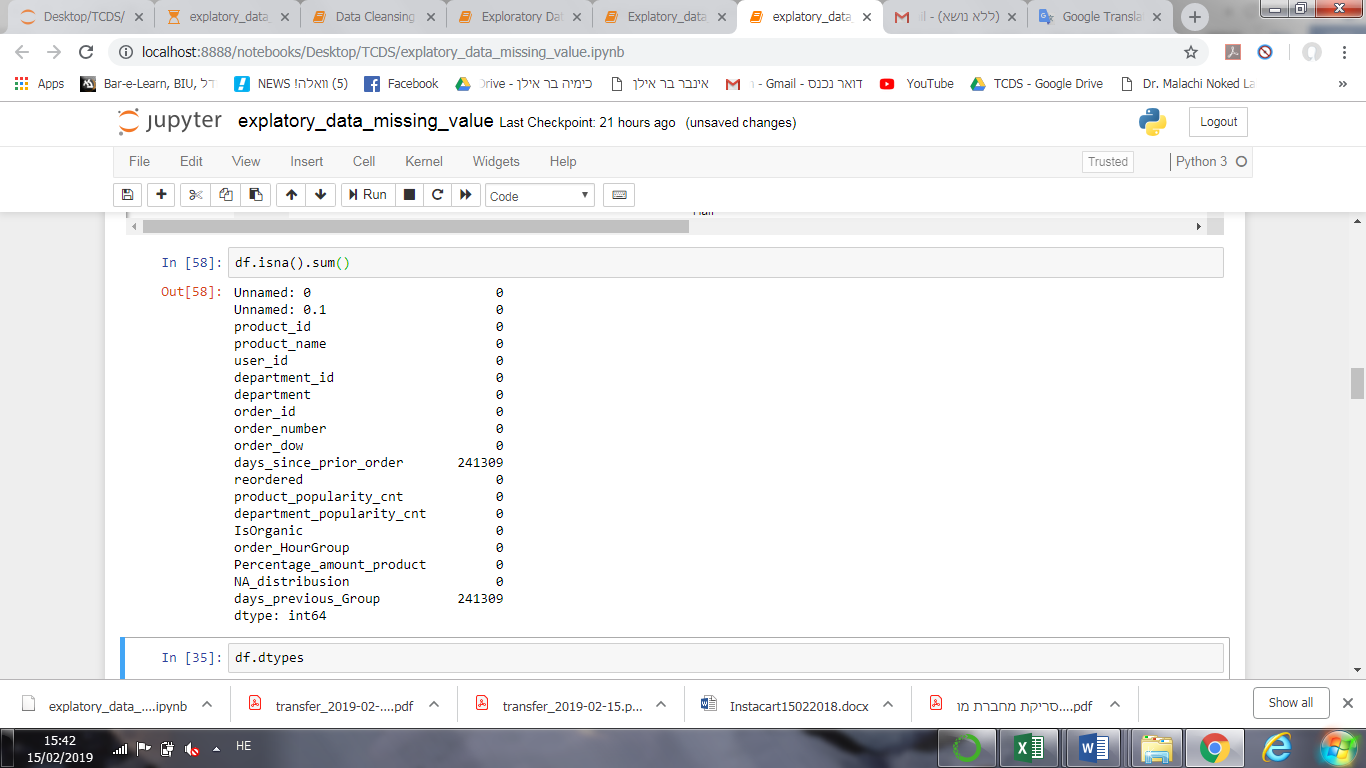
The outcome will be a list of suggestion products to each user according to the predictions of the models that we will use. The amount of the suggestion products will be determined by analyzing and processing the dataset. It's will be between 2-5 products per user, according to their purchases.

**2.3 Variable engineering**

In order to enrich our data, we added six more columns as follows:

|  |  |  |
| --- | --- | --- |
| Serial number | Column name | Description |
| 1 | product\_popularity\_cnt | Describes in percentages the popularity of product\_id. |
| 2 | department\_popularity\_cnt | Describes in percentages the popularity of department\_id.. |
| 3 | IsOrganic | Describes if the product is organic or not, by boolean variable (no = 0/ yes = 1). |
| 4 | order\_HourGroup | Describes the HourGroup which the order had been made ,after dividing the hours into 3 categories according to the number of orders made per hour. |
| 5 | Percentage\_amount\_product | Describe in percentages the amount of products that user\_id purchased per order divided to the maximum products purchased per order (i.e 16 products) |
| 6 | days\_previous\_Group | Describe the column "days\_since\_prior\_order", by using the function "HGroups\_days", which represent how much time passed from the last order, by dividing the time that pass between one order to another per user\_id into 4 categories (1 = first week, etc.). |

**2.4 Missing values**

In order to identify the missing values, we use the command ".isna().sum()", and we noted that all the missing values, are in the column 'days\_since\_prior\_order', and therefore also in the column 'days\_previous\_Group', because it is derived from the same original column that contains the missing values, as mentioned above.  
It can be concluded that for those customers, who have not previously purchased in 'Instacart', there are missing values in the columns 'days\_since\_prior\_order' and 'days\_previous\_Group'. Because the value "0"' did not appear in those column, instead it is wrote "NaN"/"None". The meaning are that those missing values are for customers that this is their first order.

Thus, we wanted to know the distributions of those purchases, for customers who bought frequently, and for those who bought at the first time.   
For this goal, we divided the dataset to two groups. The first contains all the customers who bought frequently (they have values in the columns 'days\_since\_prior\_order' and 'days\_previous\_Group'), and the second contains all the customers who bought at the first time.

We divided the dataset by adding new columns " NA\_distribusion", that contains "0" for missing values in the column 'days\_since\_prior\_order', and "1" for the others. According to this, we could divided the dataset to two groups.  
The distribution's plots show almost the same values of amount of purchases, and very similar means. The mean's gap between the two groups is 3.2%, therefore, we decided to delete the rows that contains the missing values, because it will not effect on the results, on the calculations.

After deleting the chosen rows, we delete the columns " NA\_distribusion", because it's not relevant anymore.

**2.5 Outliers determination and treatment**

In order to identify outliers we used unsupervised cluster analysis - DBSCAN that calculates the average of selected variables (i.e 'order\_dow','days\_since\_prior\_order','product\_popularity\_cnt' and 'Percentage\_amount\_product' with extreme values and without them in order to measure their degree of influence on the dataset.

References:

[1] R. Priya, “RETAIL DATA ANALYTICS USING GRAPH,” 2018.

[2] B. Dias, “Grocery Shopping Recommendations Based on Basket-Sensitive Random Walk,” no. January, 2009.

[3] A. Flores-lopez, S. Perry, and P. Bhargava, “What ’ s for Dinner ? Recommendations in Online Grocery Shopping,” vol. xx, no. xx, pp. 1–6, 2017.

[4] N. Sano, N. Machino, K. Yada, and T. Suzuki, “Recommendation system for grocery store considering data sparsity,” *Procedia Comput. Sci.*, vol. 60, no. 1, pp. 1406–1413, 2015.