Instacart Market Basket Analysis

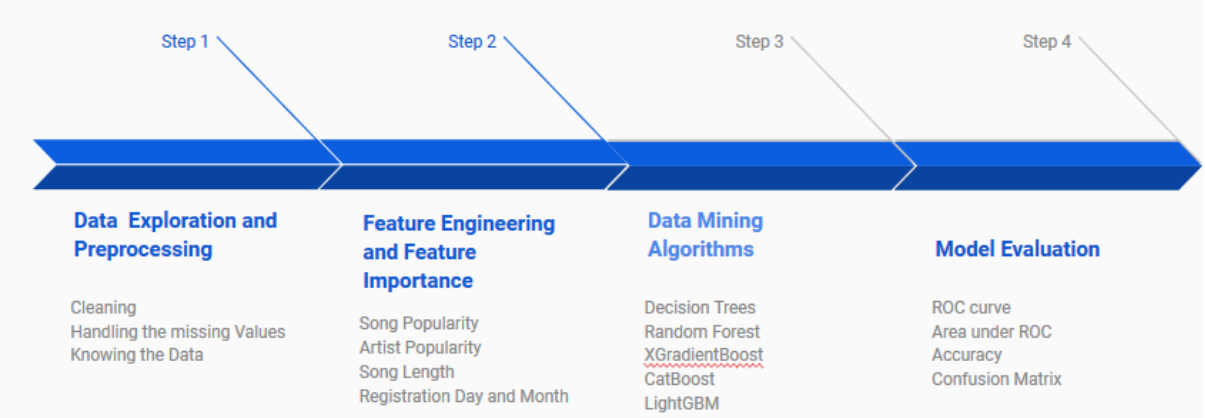
To predict which products will an Instacart consumer purchase again

Group 14

Problem Definition

In this project we had a relational set of files describing customers' orders over time. The goal of the project is to predict which products will be in a user's next order. The dataset contains a sample of over 3 million grocery orders from more than 200,000 Instacart users. For each user, dataset provide between 4 and 100 of their orders, with the sequence of products purchased in each order. Dataset also provide the week and hour of day the order was placed, and a relative measure of time between orders.

This is a binary classification problem. Due to limited time and knowledge, we first implement some basic models to establish a threshold area under ROC curve. The models implemented are XG Boost, and other algorithms such as Random Forest and Logistic.



Knowing the Data

Each entity (customer, product, order, aisle, etc.) has an associated unique id.

Orders file gives a list of all orders we have in the dataset, 1 row per order. The orders.csv doesn't tell us about which products were ordered. This is contained in the order_products.csv.

order_id	user_id	eval_set	order_number	order_dow	order_hour_of_day	days_since_prior_order
2539329	1	prior	1	2	8	NA
2398795	1	prior	2	3	7	15
473747	1	prior	3	3	12	21

Order_products_train file gives us information about which products (product_id) were ordered. It also contains information of the order (add_to_cart_order) in which the products were put into the cart and information of whether this product is a re-order(1) or not(0). Still we don't know what these products are. This information is in the products.csv.

order_id	product_id	add_to_cart_order	reordered
1	49302	1	1
1	11109	2	1
1	10246	3	0

Products file contains the names of the products with their corresponding product_id. Furthermore the aisle and deparment are included.

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

Order_products_prior file is structurally the same as the other_products_train.csv.

order_id	product_id	add_to_cart_order	reordered
2	33120	1	1
2	28985	2	1
2	9327	3	0

Aisles file contains the different aisles.

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

Department file contains the different departments. –

department_id	department
1	frozen
2	other
3	bakery

Data Exploration and Feature Engineering

With a little more data exploration, we first convert character variables to factor.

In Figure 1, we have a clear effect of hour of day on order volume. Most orders are between 8.00-18.00.

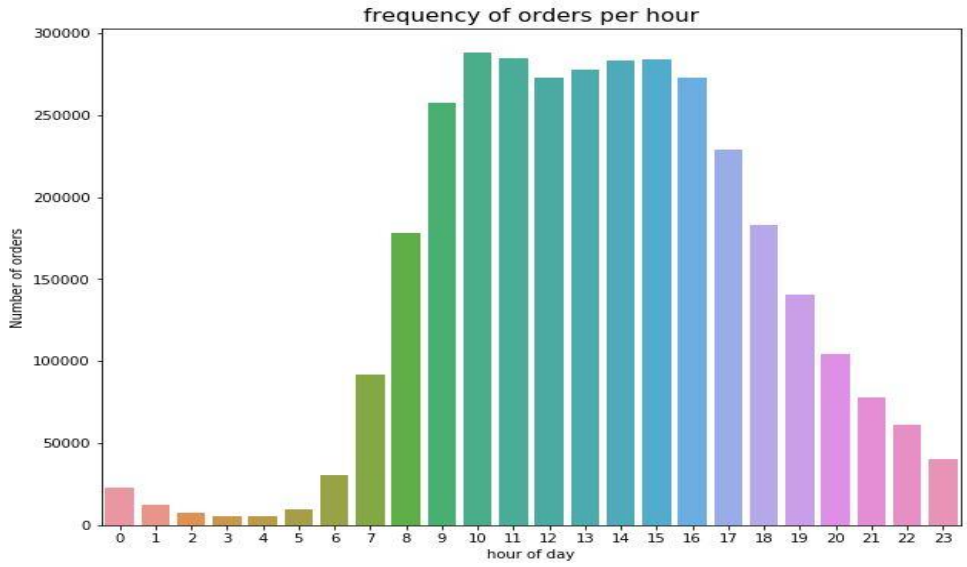


Figure 1

In Figure 2, we have a clear effect of day of the week. Most orders are on days 0 and 1. Unfortunately there is no info regarding which values represent which day, but one would assume that this is the weekend.

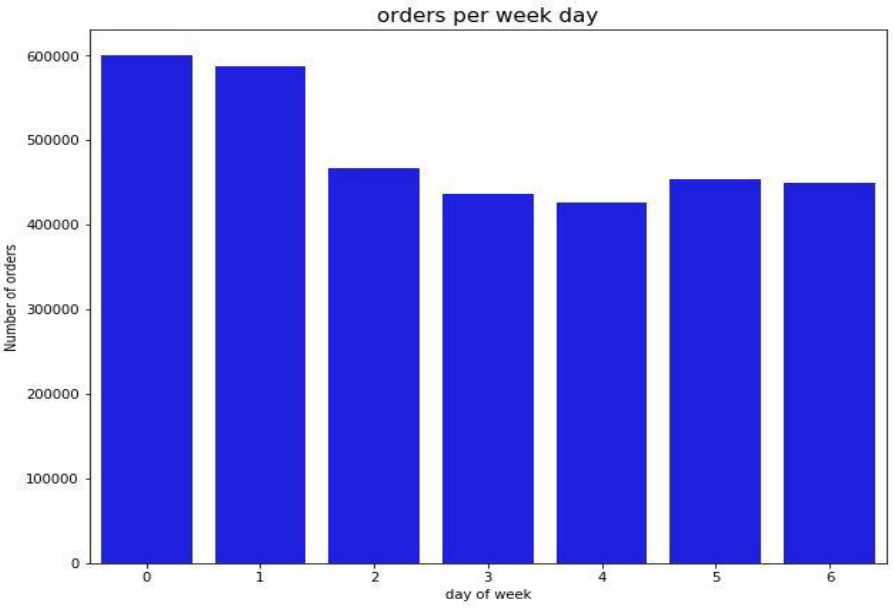


Figure 2

In Figure 3, people seem to order more often after exactly one week.

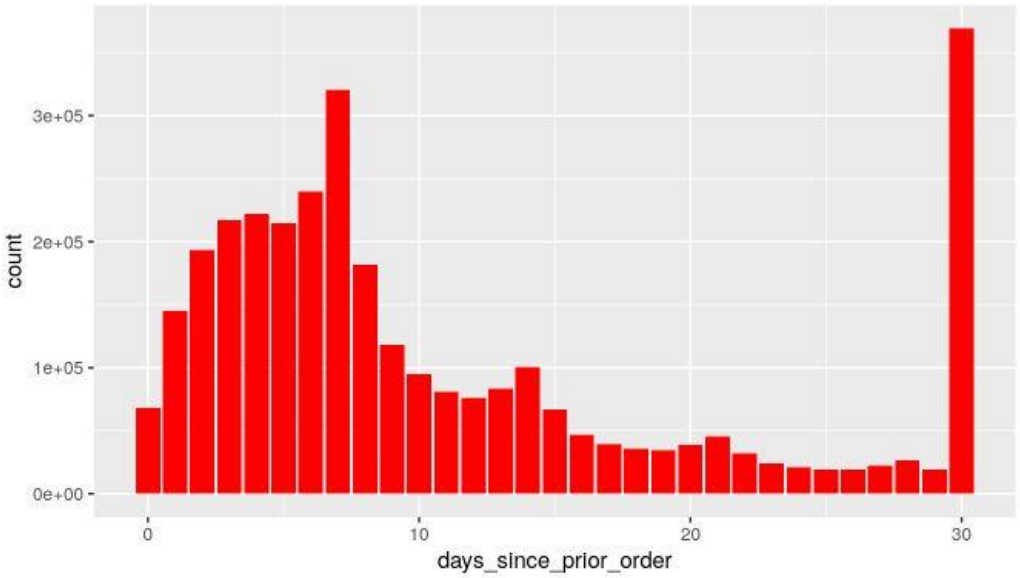


Figure 3

From Figure 4, we can clearly estimate that 59% of the ordered items are reorders.

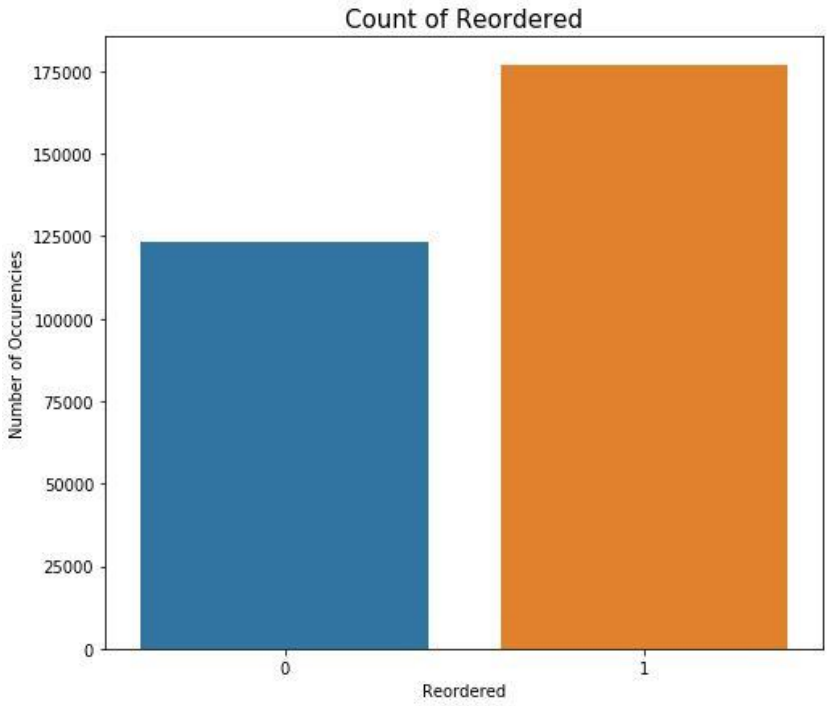
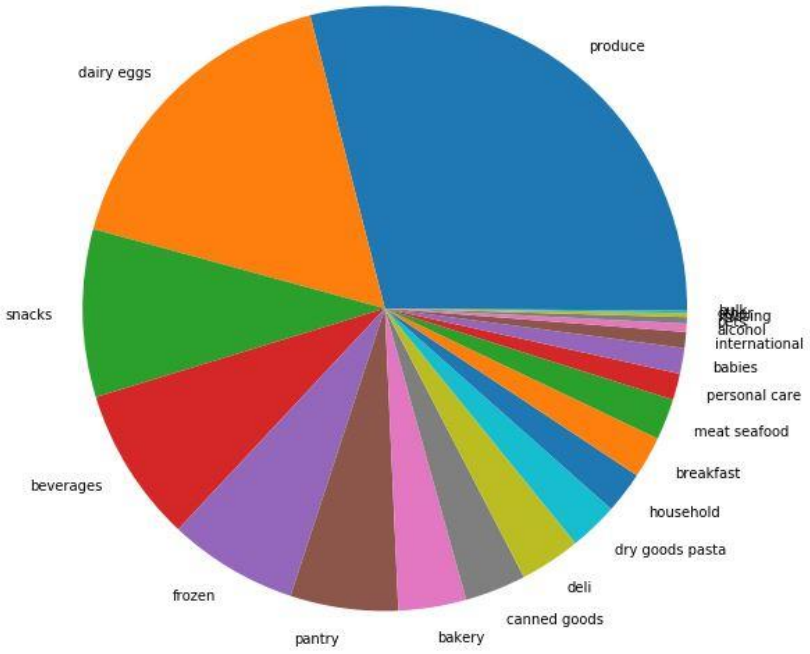


Figure 4

Department distribution –



Finally, we calculated the mean on the basis of the product name and its reordering status.

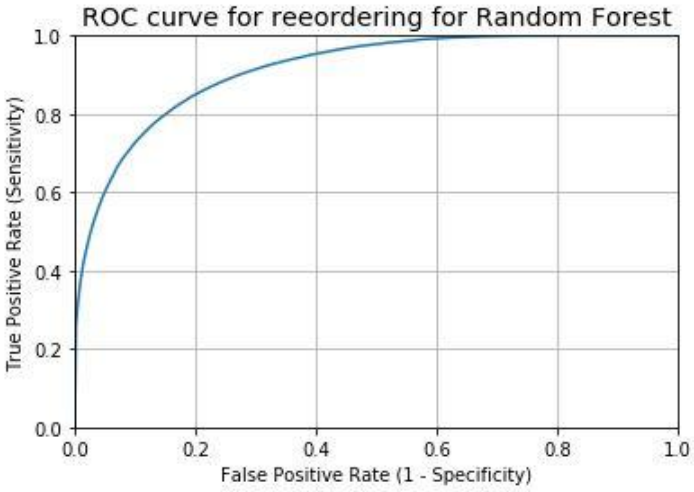
We also calculated the mean and frequency on the basis of order id and its reordering status.

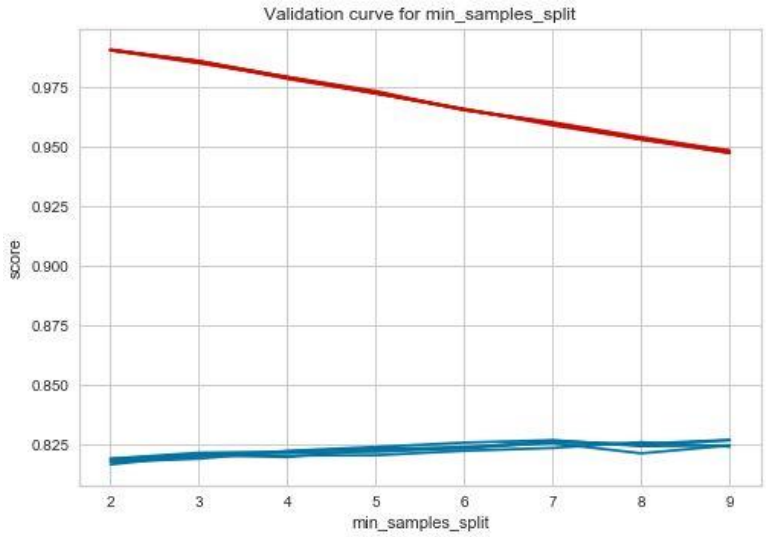
Data Mining Algorithms

Here we used three machine learning algorithms (Logistic, Random Forest, XG Boost) for training our model. XG Boost considered as one of the most powerful and commonly used machine learning techniques.

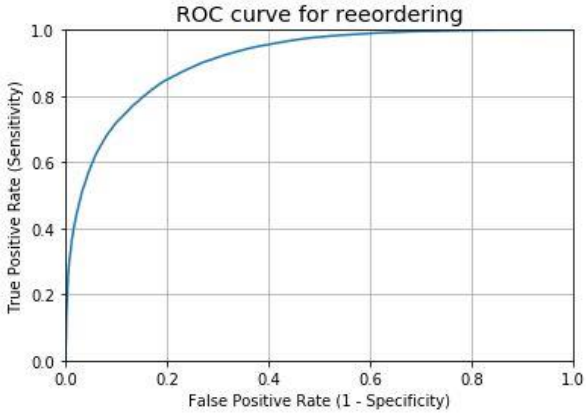
We divided the merged train data here in two parts training data and test data by using model_selection with test_size = 0.3.

1. Random Forest – Here we replaced the NULL value with value 0. Then we created some features and then changed the data type of object columns to categorical variable.
   Here we had to train our dataset on 0.01% of our dataset because the size of our dataset is in millions and these algorithms have a high time complexity.



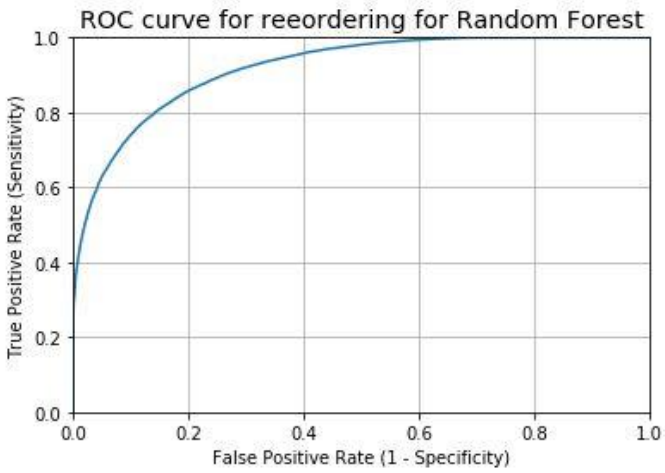


2. Logistic –
   We did hyper parameter tuning using L1 and L2 regularisation.
   In L1 regularisation, we firstly use SAGA solver but accuracy is not that good so at the end we use Linear Solver and bias approx. 10000.
   In L2 regularisation, we use LBFGS Solver.



3. XG Boost –
   After hyperparameter tuning we got these parameters –

Parameters	Values
metric	binary_logloss
boosting	Gbtree
learning_rate	0.1
verbose	1
num_leaves	250
bagging_freq	1
feature_fraction	0.9
feature_fraction_seed	1
max_bin	128
max_depth	10
num_rounds	10
application	binary



A final summary comparing the performance of various models is tabulated below.

Sr. No	Model	AUC	Accuracy
1.	Logistic	0.910	0.832
2.	Random Forest	0.913	0.830
3.	XG Boost	0.919	0.834

XG Boost classifier gave us the best performance. So, we chose to interpret the XG Boost model.