Capstone Report

Problem Statement:

Predict products mixes that will be included in the next purchase order by users. That includes products a user will buy again, try for the first time, or add to their cart during next session.

Benefits

Main audiences: Instacart e-commerce Team

By knowing the user's purchase habits and what user would like to purchase in the next order, Instacart can customize product pages to individuals by showing relevant products, which enables better conversion.

Secondary audiences include:

- **Instacart users**: Users will save time on browsing the needed products, and may find new products of their interests faster, which enhance user experiences.
- **Partner retailers:** Instacart can aggregate prediction of product purchases by store and send that to partner retailers for inventory planning purchases.

Dataset Description

Obtained from Kaggle, data are in form of 6 datasets

- Aisles: product subcategory (size: (134, 2)). Examples: instant food, energy granola bars, etc.
- Department: product department category (size: (21, 2)). Examples: bakery, produce, etc.
- Products: name and id (size: (49688, 4)
- orders: orders id, user_id, order number, order day of week, hours of day, days since prior order (size: (3.4M x 7))
- order_products__prior: product details among order in the “prior” eval set under orders dataset
- order_products__train: product details among order in the “train” eval set under orders dataset
- eval set “test” is the target variable of the problems

Cleaning and Wrangling

- the data has been cleaned by data provider, no missing records, and data are in the right data types

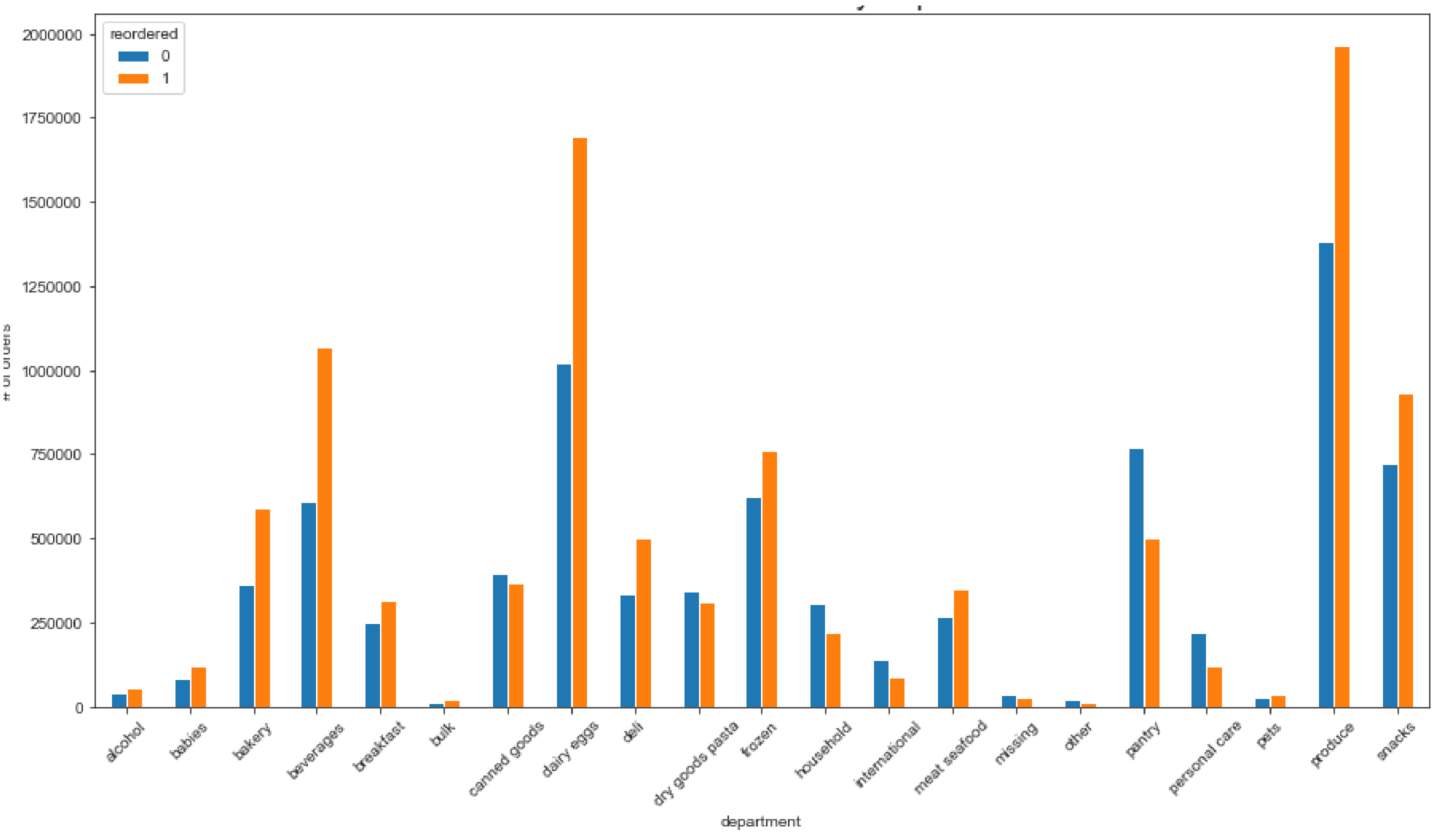
Exploratory Analysis

The goal is to discover features that may have predictive power to target variables (i.e. product id by user) through visualizing their historical pattern.

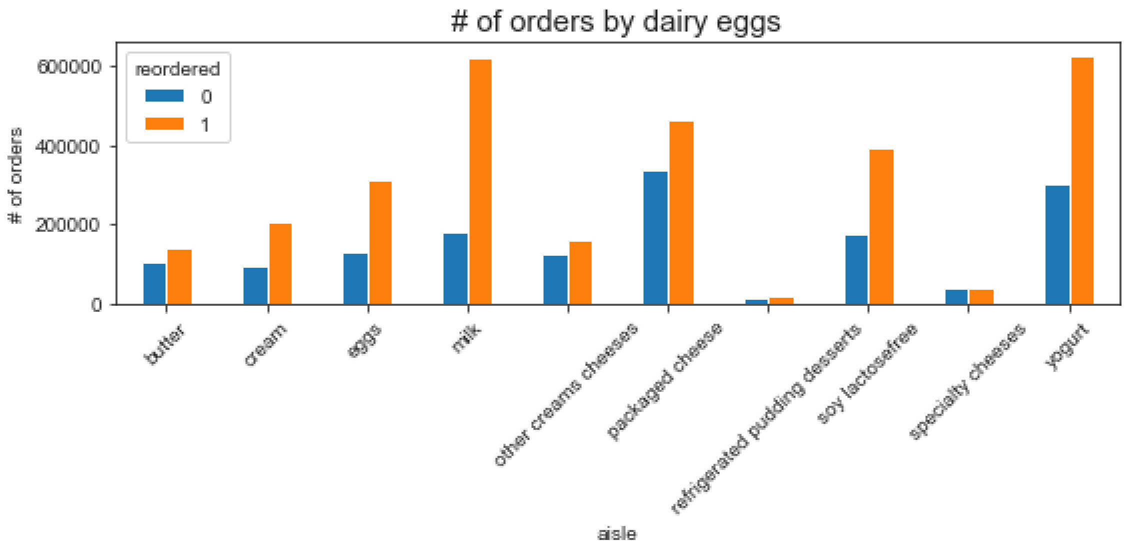
Areas of focus:

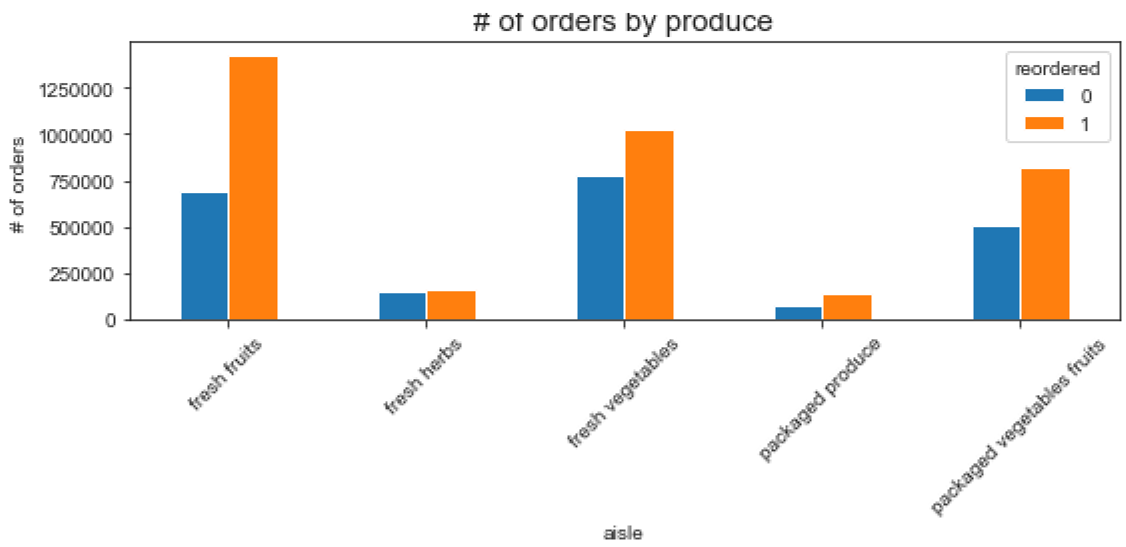
- What types of products have a higher reorder frequency?
- Do users buy the same type of product over time (i.e. does product mix by user stay relatively constant?)
- Would the day of the week (e.g. Monday vs Friday) influence the number/ type of product being purchased?
- When do users tend to buy from instacart (day of week and hour of the day)?
- How many days are between each order?
- What's user attrition rate?

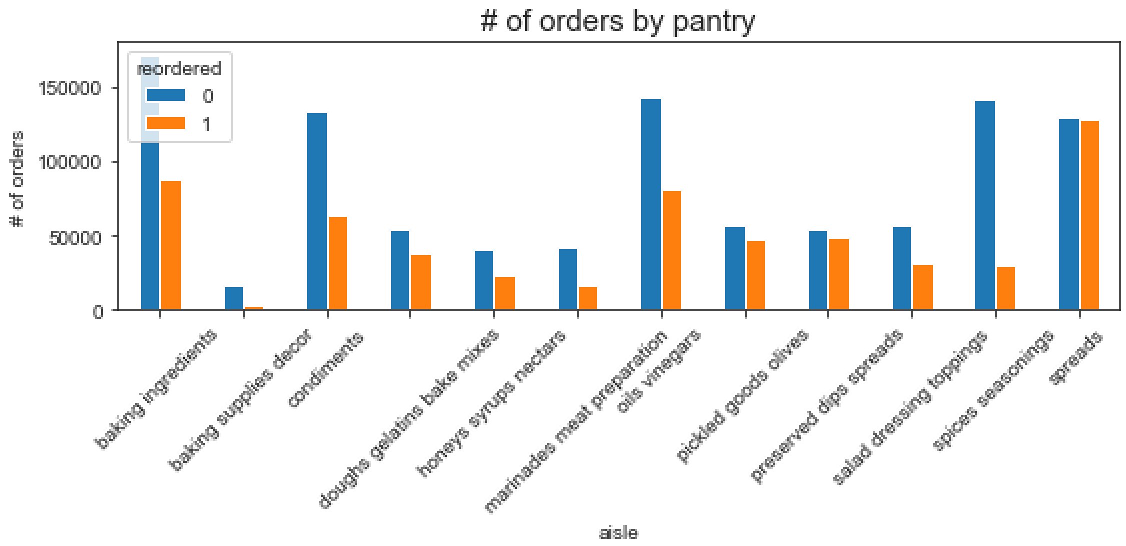
Order/Reorder Frequency by Department:



- At the department level, beverages, dairy, and produce appear to have the highest re-purchasing rates. Pantry does not seem to be popular items for re-purchasing.

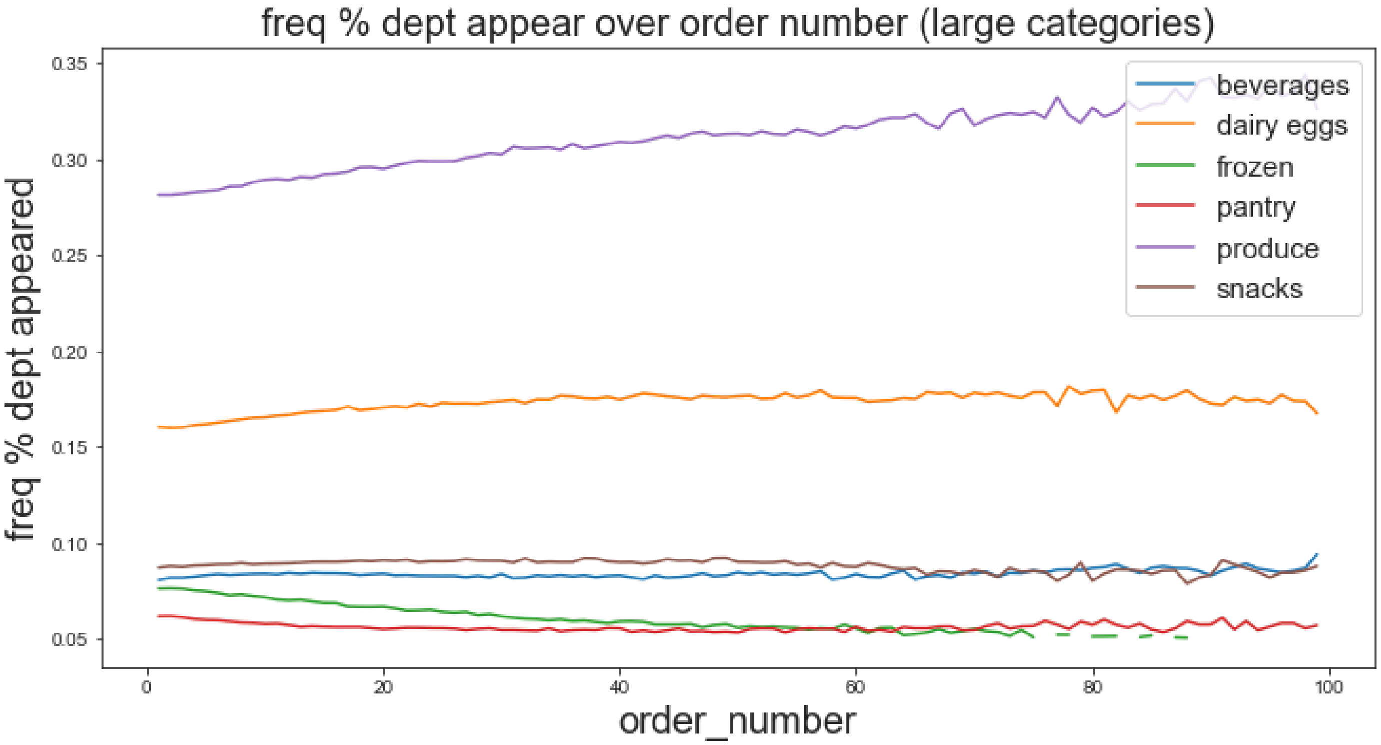






- Aisles within each department vary on re-ordering rates

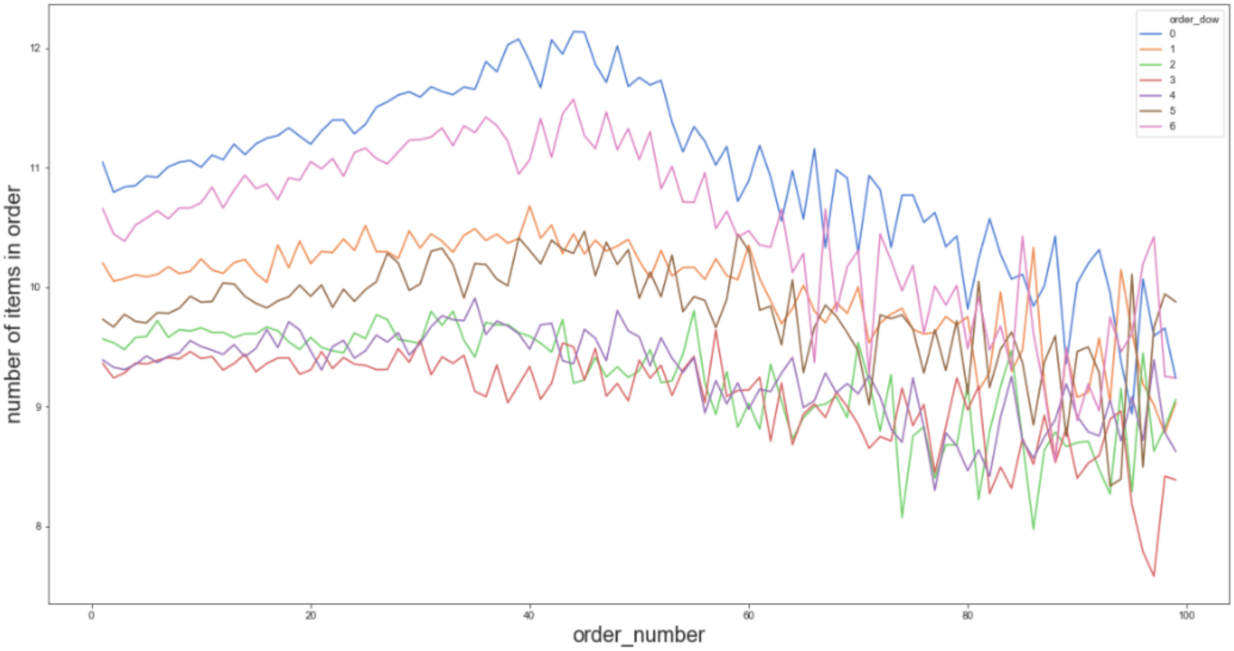
Order Mix % by Department



- Department mix appear to be stable over time

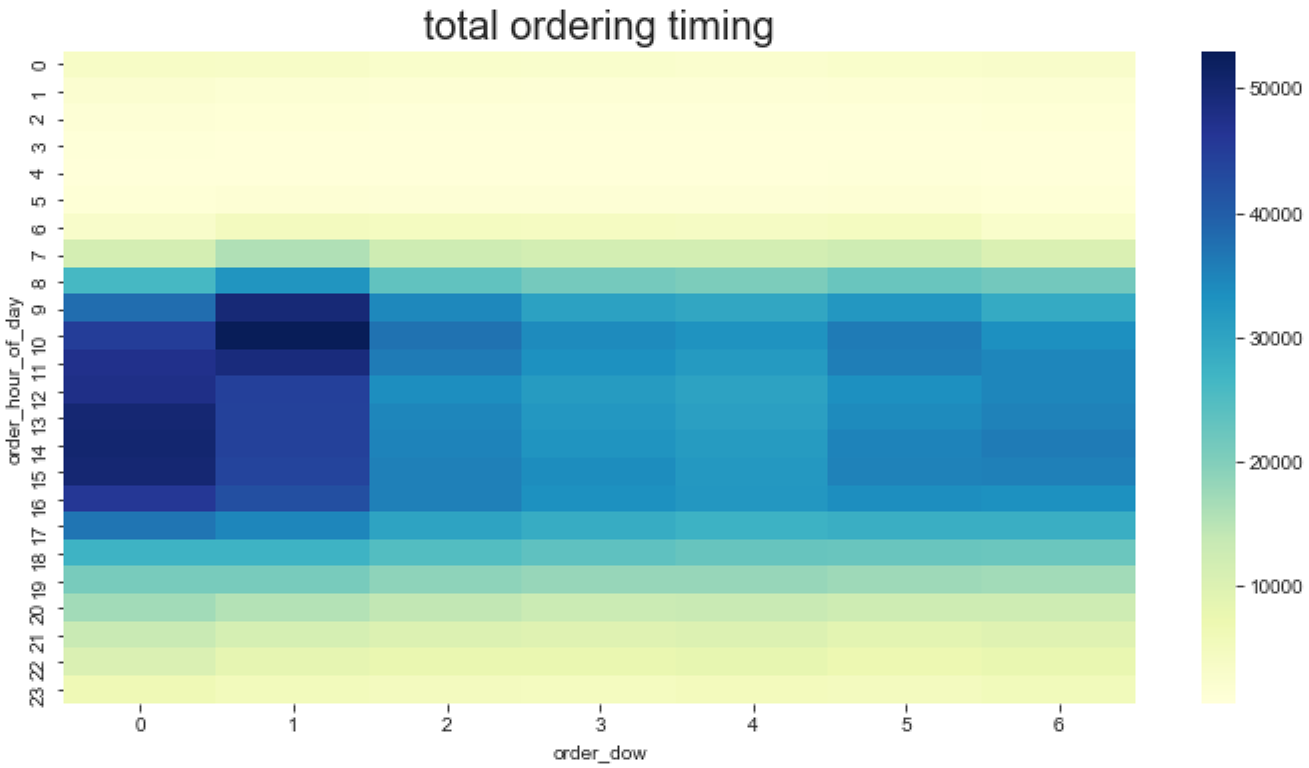
Day of Week

- Users tend to buy more items on Day 0 (Sunday) and Day 6 (Saturday). But overall orders are between 9-11 items



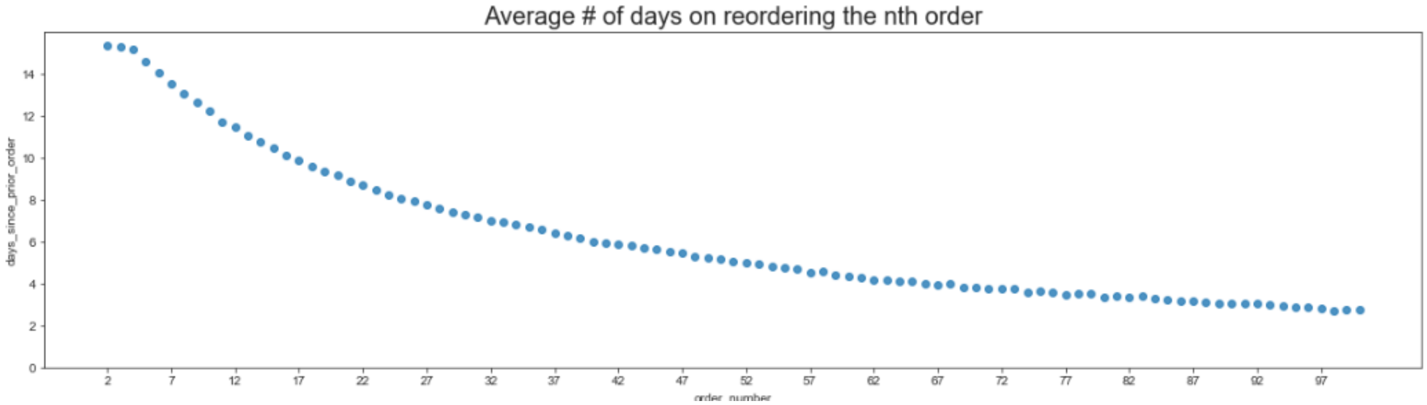
Hour of Day

- Generally, most orders are placed between 8am to 6pm



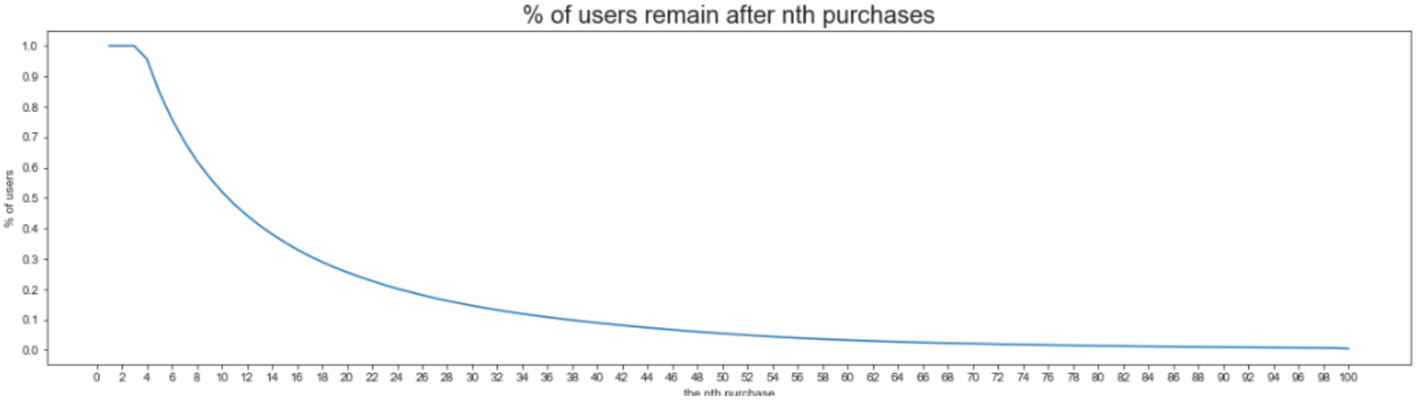
Order Lags

- For the first 3 orders, the day separation is about 15 days, and gradually decrease as the user re-order more often



User attrition

- After the 4th purchase, user’s attrition rates are at ~5% per additional one re-purchase. By the 10th purchase, only 50% of users remain using the services. About 10% of the users making 38th purchase or more



Summary

- Product categories have different reorder rates, among all beverages (water), dairy eggs (milk, yogurt) and produce (fruits) are the most popular items for re-ordering. Pantry items are the least likely to be re-ordered
- Product categories mix does not vary significantly over time. Users will gradually trade-off one item favor the other, but the reorder items appear to be similar to the previous order that user made
- Users tend to purchase on day 0 and 1, from 9am to 4pm
- Some products are being purchased on different days
- Most users who signed up will use the service 3-4 times, on average 15 days between each other, then users' attrition starts. Perhaps that's an opportunity to improve user loyalty. 10% of the users become very loyal to the service and purchase almost every week.
- In each order, around 9-10 items are added to carts.

Next steps will be to convert the above observation into features to predict the likelihood of a product to be re-ordered by the user.

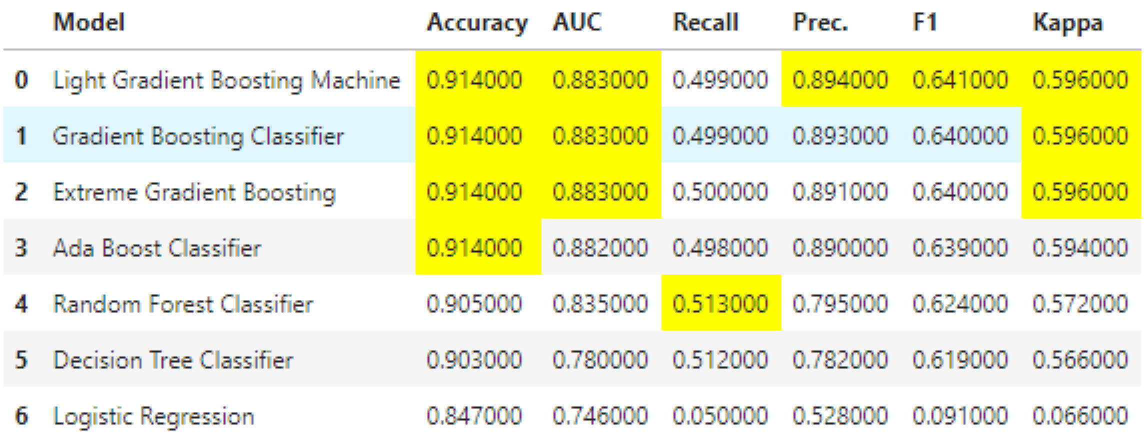
The following features are created for machine learning:

- product_appear: % of order which a particular product appears in overall user order history
- buy_cnt: the absolute number of time a product was ordered by a user in the last three orders
- reordered: number of times a product was ordered by a user
- lag: average days difference when a user purchased a product under certain departments
- last_purchase: was the product just ordered by a user in the latest order

Machine Learning

Used PyCaret to compare multiples models.

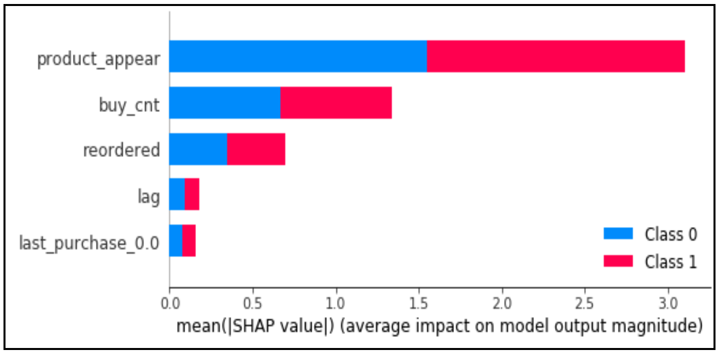
Training Results:



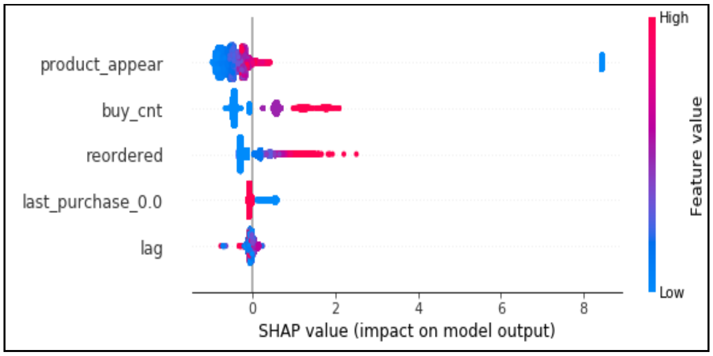
	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa
0	Light Gradient Boosting Machine	0.914000	0.883000	0.499000	0.894000	0.641000	0.596000
1	Gradient Boosting Classifier	0.914000	0.883000	0.499000	0.893000	0.640000	0.596000
2	Extreme Gradient Boosting	0.914000	0.883000	0.500000	0.891000	0.640000	0.596000
3	Ada Boost Classifier	0.914000	0.882000	0.498000	0.890000	0.639000	0.594000
4	Random Forest Classifier	0.905000	0.835000	0.513000	0.795000	0.624000	0.572000
5	Decision Tree Classifier	0.903000	0.780000	0.512000	0.782000	0.619000	0.566000
6	Logistic Regression	0.847000	0.746000	0.050000	0.528000	0.091000	0.066000

Narrow down to Light Gradient Boosting Machine (Lightbgm) and Gradient Boosting Classifier

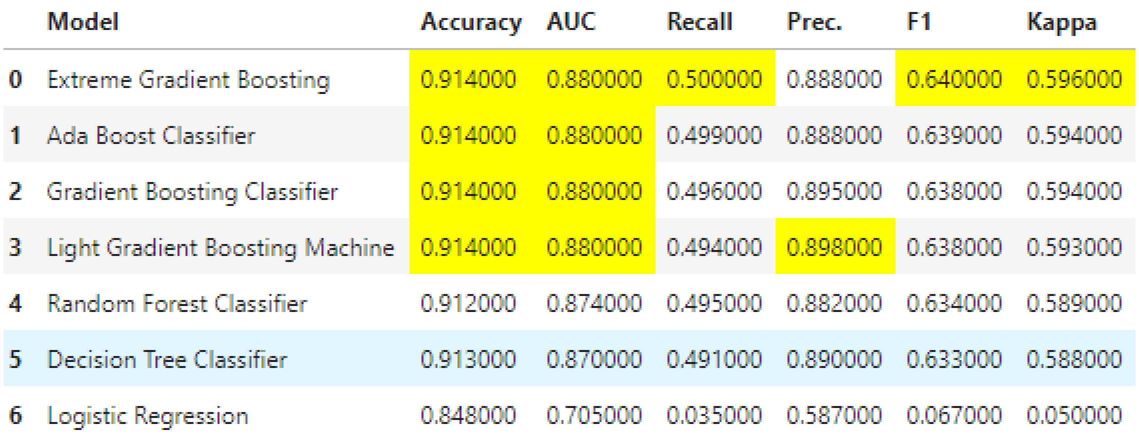
Lightgbm



Gradient Boosting



Lag features appear to have small influence on predicting outcome in both model. This prompt the question on whether model should keep this feature. Re-access performance by dropping 'lag' feature:



	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa
0	Extreme Gradient Boosting	0.914000	0.880000	0.500000	0.888000	0.640000	0.596000
1	Ada Boost Classifier	0.914000	0.880000	0.499000	0.888000	0.639000	0.594000
2	Gradient Boosting Classifier	0.914000	0.880000	0.496000	0.895000	0.638000	0.594000
3	Light Gradient Boosting Machine	0.914000	0.880000	0.494000	0.898000	0.638000	0.593000
4	Random Forest Classifier	0.912000	0.874000	0.495000	0.882000	0.634000	0.589000
5	Decision Tree Classifier	0.913000	0.870000	0.491000	0.890000	0.633000	0.588000
6	Logistic Regression	0.848000	0.705000	0.035000	0.587000	0.067000	0.050000

Overall F1 performance stays roughly the same, but Extreme Gradient Boosting (Xgboost) becomes the best model. Tree-based model such as Random Forecast improved in F1 score.

Submit predictions Lightgam, Gradient Boosting based on the original model and Random Forecast, and Xgboost with ‘lag’ removed.

Kaggle Submission Results

Models	Naive	Lightbgm	Gradient Boosting	Random Forest	Xgboost
F1	0.31180	0.36354	0.36244	0.36025	0.36049

Recommendations/next steps

- Light Gradient Boosting has the best performance, and recommend as the winning model.
- To maximize F1 score, we were trying to balance between Recall and Precision. From the training dataset, most models only have Recall rates of ~0.5 (i.e. half of the purchased items won't be captured by the model).
- With the smaller size of the test dataset, we have to lower the probability threshold of classifying 0 or 1 to mitigate the penalty from low recall. The final model has a probability threshold set ~ 0.2.
- Further improvement on the F1 score would include creating a two-step model to engineer on maximizing F1 score.
- Obtaining new features such as product prices or promotion timing by retailers would be helpful.