

Your Orders

Orders

Buy Again



“Buy Again” Market Basket Analysis

Springboard Data Science Career Track - Capstone One

The problem

“Buy again” recommendations are not always that accurate and timely, even for frequently purchased items.

How well can we predict what items a user will buy again?
What factors have predictive capability?



Buy it again in Home



Buy it again in Home Improvement



\$3.25
liveGfree Organic Gluten Free Brown Rice Quinoa Fusilli
16 oz



\$2.75
liveGfree Sweet Chili Brown Rice Crisps
7 oz



\$2.75
liveGfree Black Sesame Brown Rice Crisps
7 oz



Organic Banana
At \$0.65/lb



\$1.45 each
Organic Baby Peeled Carrots, Bag (Limit 6)
16 oz



\$3.65 each
Organic Romaine Hearts, Package (Limit 6)
3 ct



\$3.19 each
Organic Blueberries, Package (Limit 6)
6 oz



\$2.19 each
Organic Celery Hearts (Limit 6)



\$2.19
Simply Nature Organic Sea Salt Pop Corn
6 oz

[View 23 more >](#)

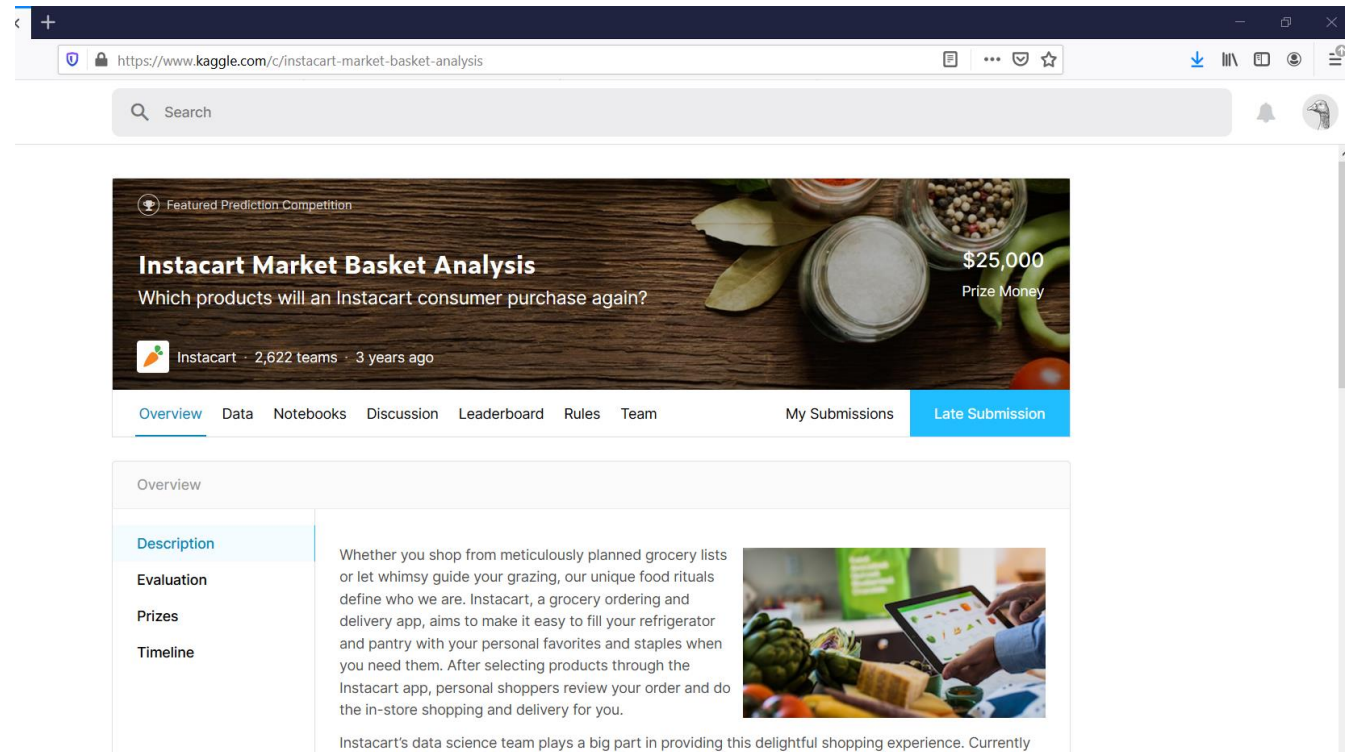


\$4.29 each
Organic Avocados (Limit 6)
4 ct

Who cares?

Retailers are interested in improving their recommendation systems

- Improve customer experience
- Keep customers coming back
- Increase sales



The screenshot shows the Kaggle website interface for the 'Instacart Market Basket Analysis' competition. The browser address bar displays 'https://www.kaggle.com/c/instacart-market-basket-analysis'. The page features a dark header with the competition title 'Instacart Market Basket Analysis' and the question 'Which products will an Instacart consumer purchase again?'. A prize money of '\$25,000' is highlighted. Below the header, a navigation bar includes links for 'Overview', 'Data', 'Notebooks', 'Discussion', 'Leaderboard', 'Rules', 'Team', 'My Submissions', and 'Late Submission'. The 'Overview' section is active, showing a 'Description' tab. The description text explains that the competition is about predicting products in a grocery basket based on a list of items. An image of a person using a tablet with various fruits and vegetables is shown. The text concludes by stating that Instacart's data science team plays a big part in providing this delightful shopping experience.

Featured Prediction Competition

Instacart Market Basket Analysis

Which products will an Instacart consumer purchase again?

\$25,000
Prize Money

Instacart · 2,622 teams · 3 years ago

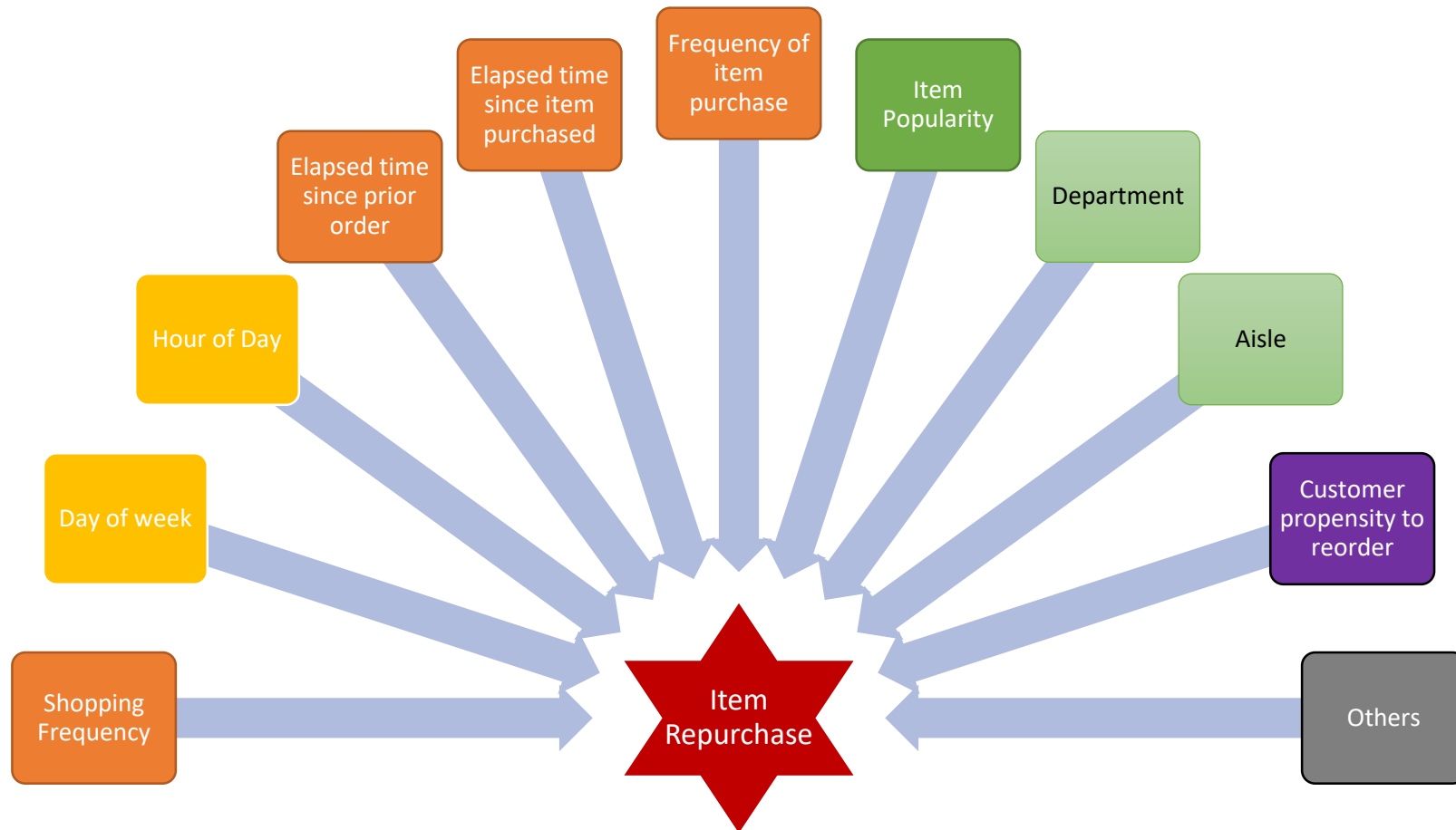
[Overview](#) [Data](#) [Notebooks](#) [Discussion](#) [Leaderboard](#) [Rules](#) [Team](#) [My Submissions](#) [Late Submission](#)

Overview

Description	Whether you shop from meticulously planned grocery lists or let whimsy guide your grazing, our unique food rituals define who we are. Instacart, a grocery ordering and delivery app, aims to make it easy to fill your refrigerator and pantry with your personal favorites and staples when you need them. After selecting products through the Instacart app, personal shoppers review your order and do the in-store shopping and delivery for you.
Evaluation	
Prizes	
Timeline	

Instacart's data science team plays a big part in providing this delightful shopping experience. Currently

What factors might affect item repurchase?



Data Information

Featured Prediction Competition

Instacart Market Basket Analysis

Which products will an Instacart consumer purchase again?

Instacart

2,822 teams

3 years ago

Overview

Data

Notebooks

Discussion

Leaderboard

Rules

Team

My Submissions

Like Submission

Data Description

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 provide 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](#) accompanying its public release.

File descriptions

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

Data Source

	name	size
0	aisles	134
1	departments	21
2	orders	3421083
3	order_products__prior	32434489
4	order_products__train	1384617
5	products	49688

	department_id	department
0	1	frozen
1	2	other
2	3	bakery

order_id	user_id	eval_set	order_number	order_dow	order_hour_of_day	days_since_prior_order	
0	2539329	1	prior	1	2	8	NaN
1	2398795	1	prior	2	3	7	15.0
2	473747	1	prior	3	3	12	21.0

```
graph LR
    orders[orders] --> op_prior[order_products__prior]
    orders --> op_train[order_products__train]
    op_prior --> products[products]
    op_train --> products
    products --> departments[departments]
    products --> aisles[aisles]
```

The diagram illustrates the relationships between five data tables: orders, order_products__prior, order_products__train, products, departments, and aisles. The orders table is linked to both order_products__prior and order_products__train. Both of these tables are linked to the products table. The products table is linked to both departments and aisles.

	product_id	product_name	aisle_id	department_id
0	1	Chocolate Sandwich Cookies	61	19
1	2	All-Seasons Salt	104	13
2	3	Robust Golden Unsweetened Oolong Tea	94	7

	aisle_id	aisle
0	1	prepared soups salads
1	2	specialty cheeses
2	3	energy granola bars

Model- focused Data Exploration

Customer

- Propensity to Reorder *

Frequency and time effects on Reordering

- Reorder Proportion
 - By day of week
 - By hour of day
 - By elapsed time

Item reorder frequency

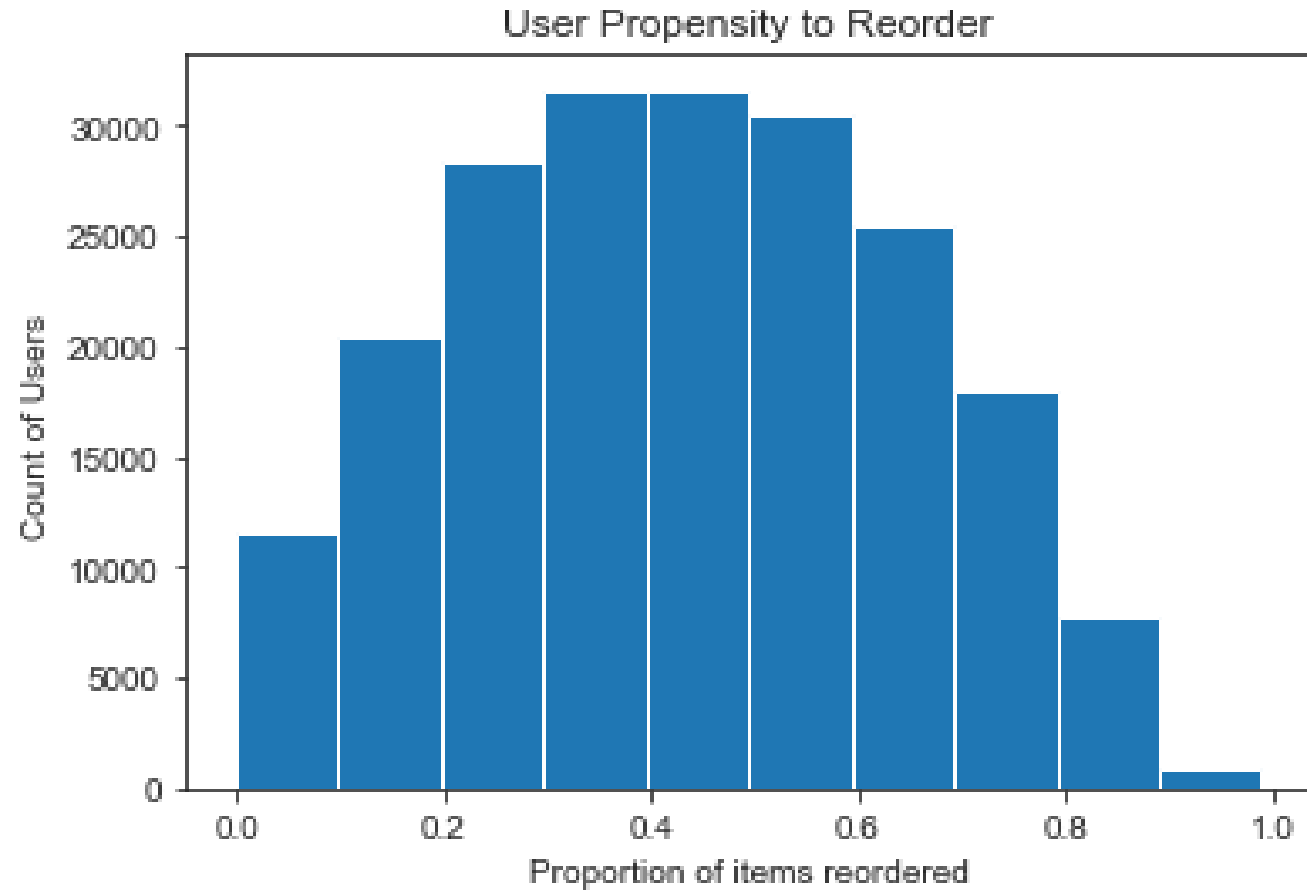
- by Dept

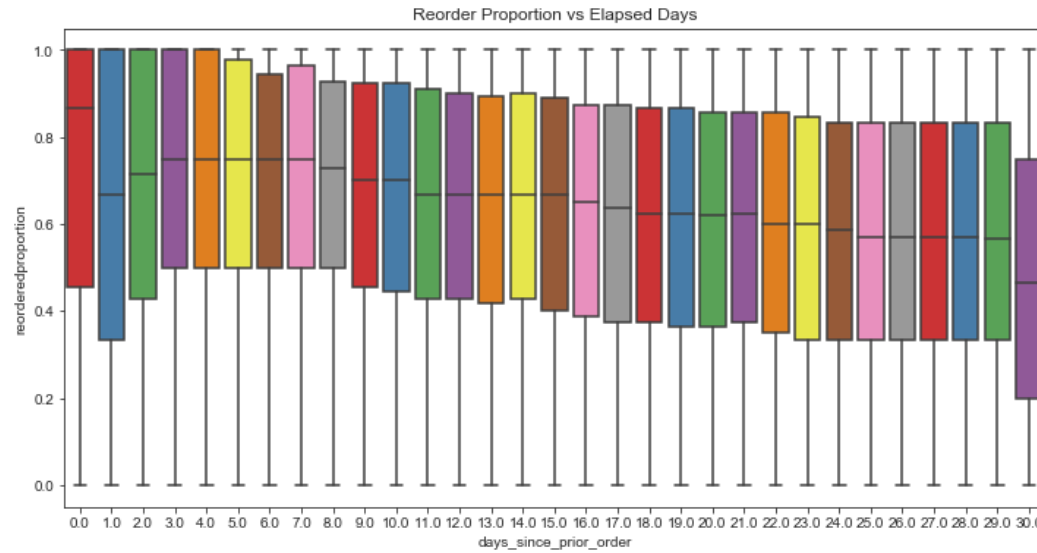
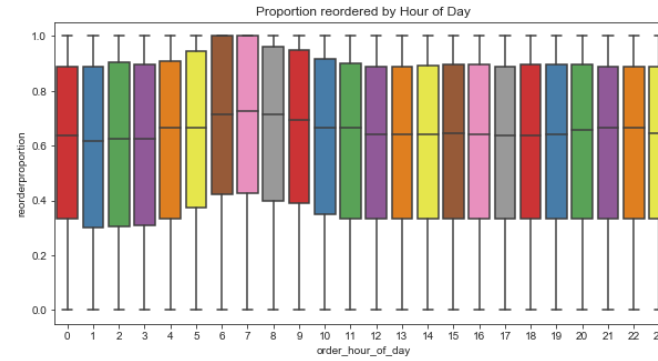
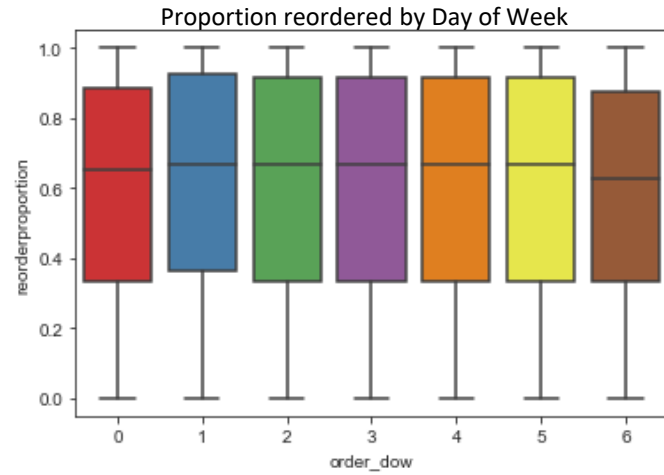
Most Reordered

- By Product
- By Department
- By aisle

* Details are in this presentation

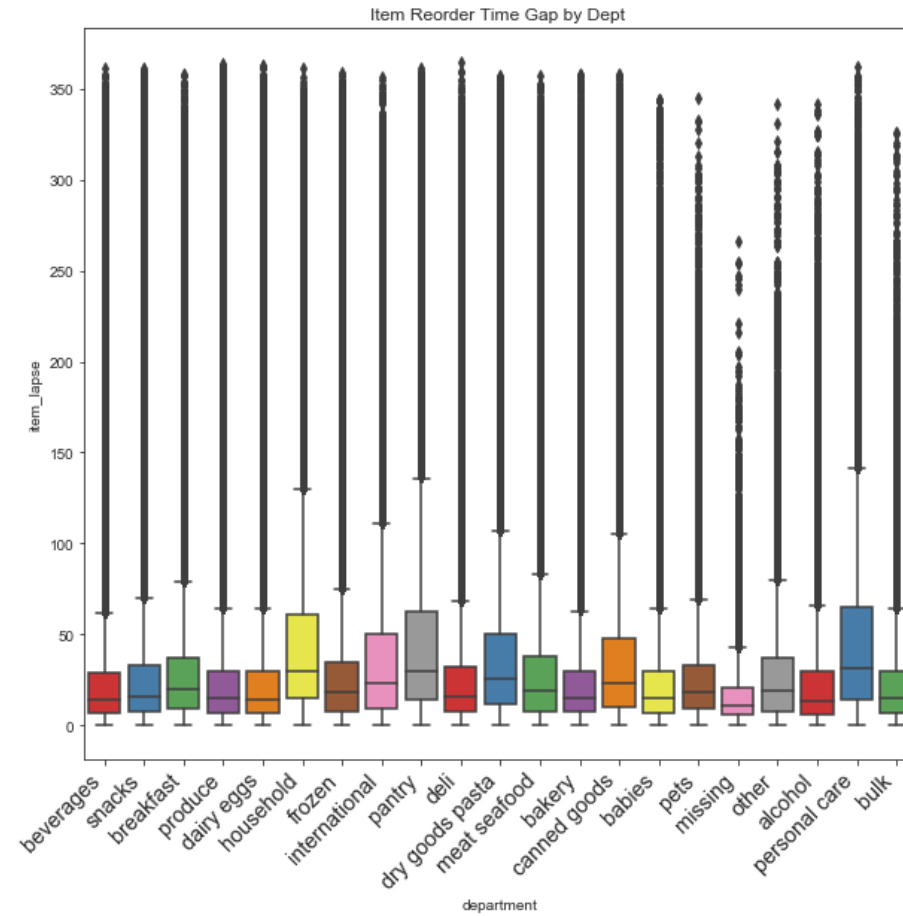
User Propensity to Reorder



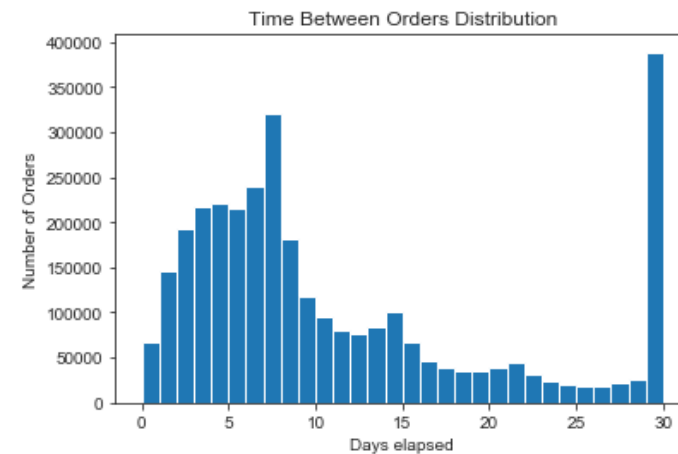
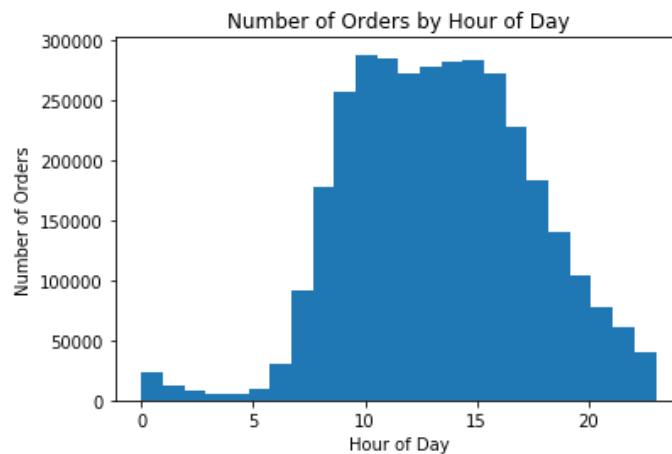
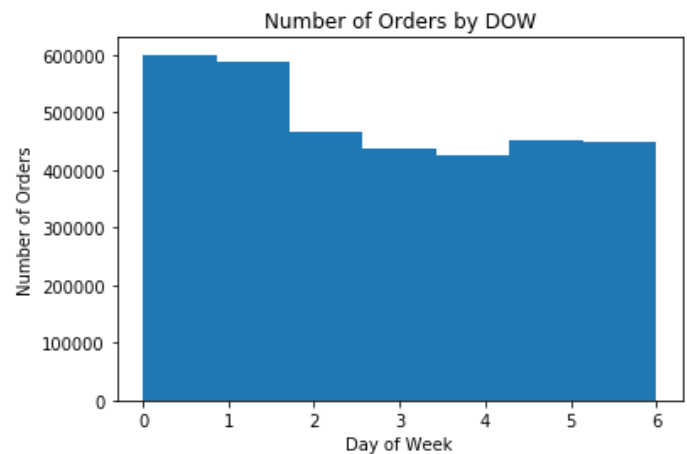


Proportion of
order items
repurchased:
Time-based
variations

Product repurchase: Time gap variations by department

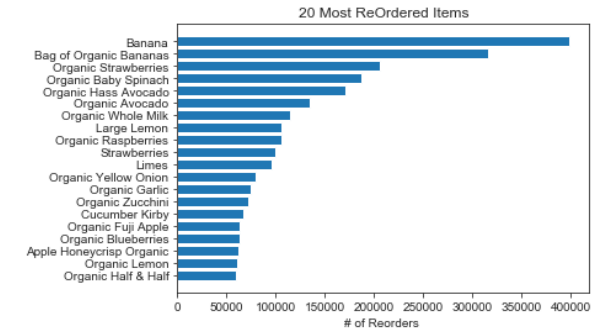
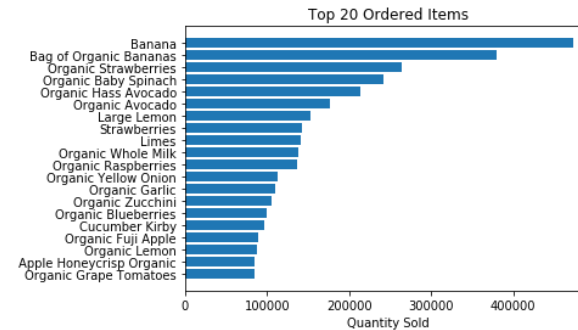
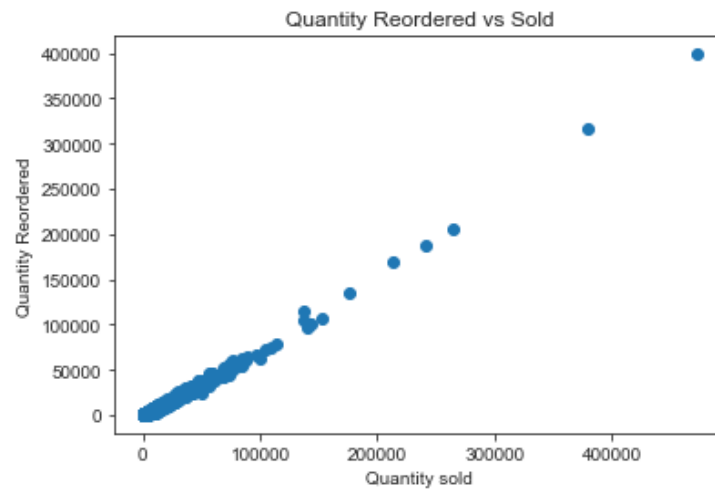


How often and when do users shop?



Most Popular: Products, Departments, Aisles

- Quantity Sold
- Quantity Reordered

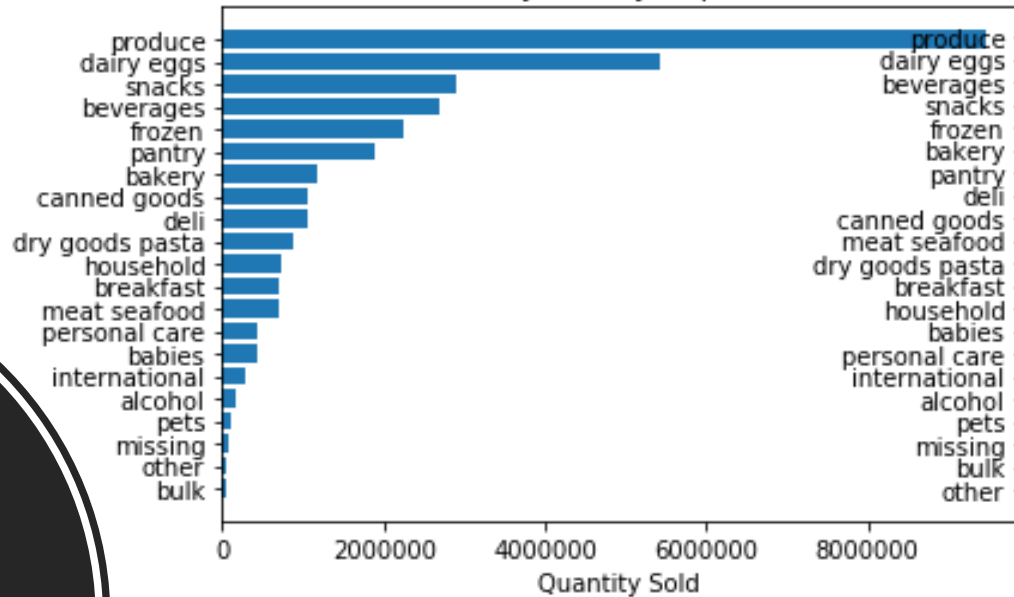


Top Products: Sold, reordered, correlation

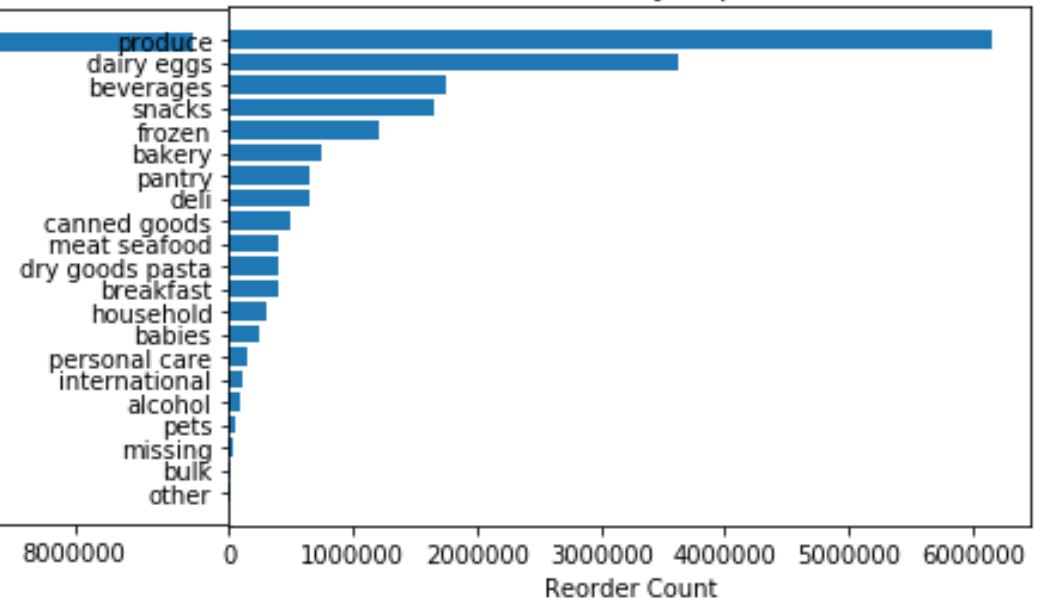
Aisles & Departments

No surprise:
Most popular
have the most
reorders

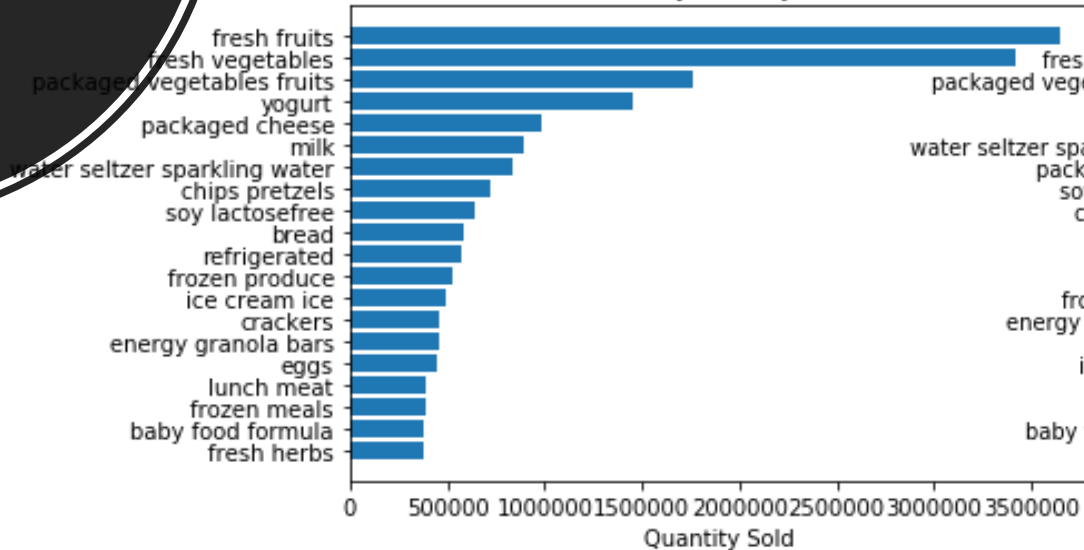
Quantity sold by Department



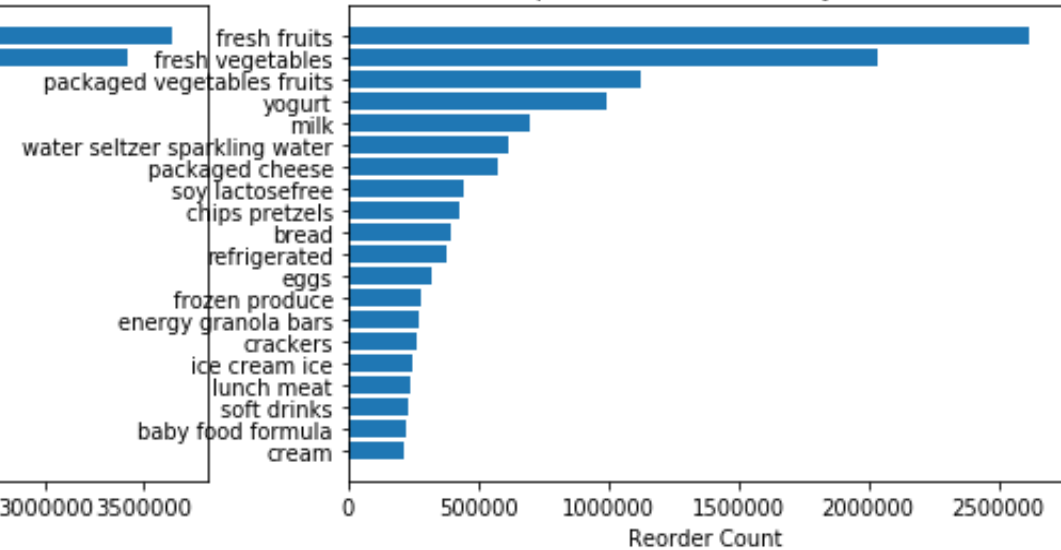
Reorder counts by Department



Quantity sold by Aisle

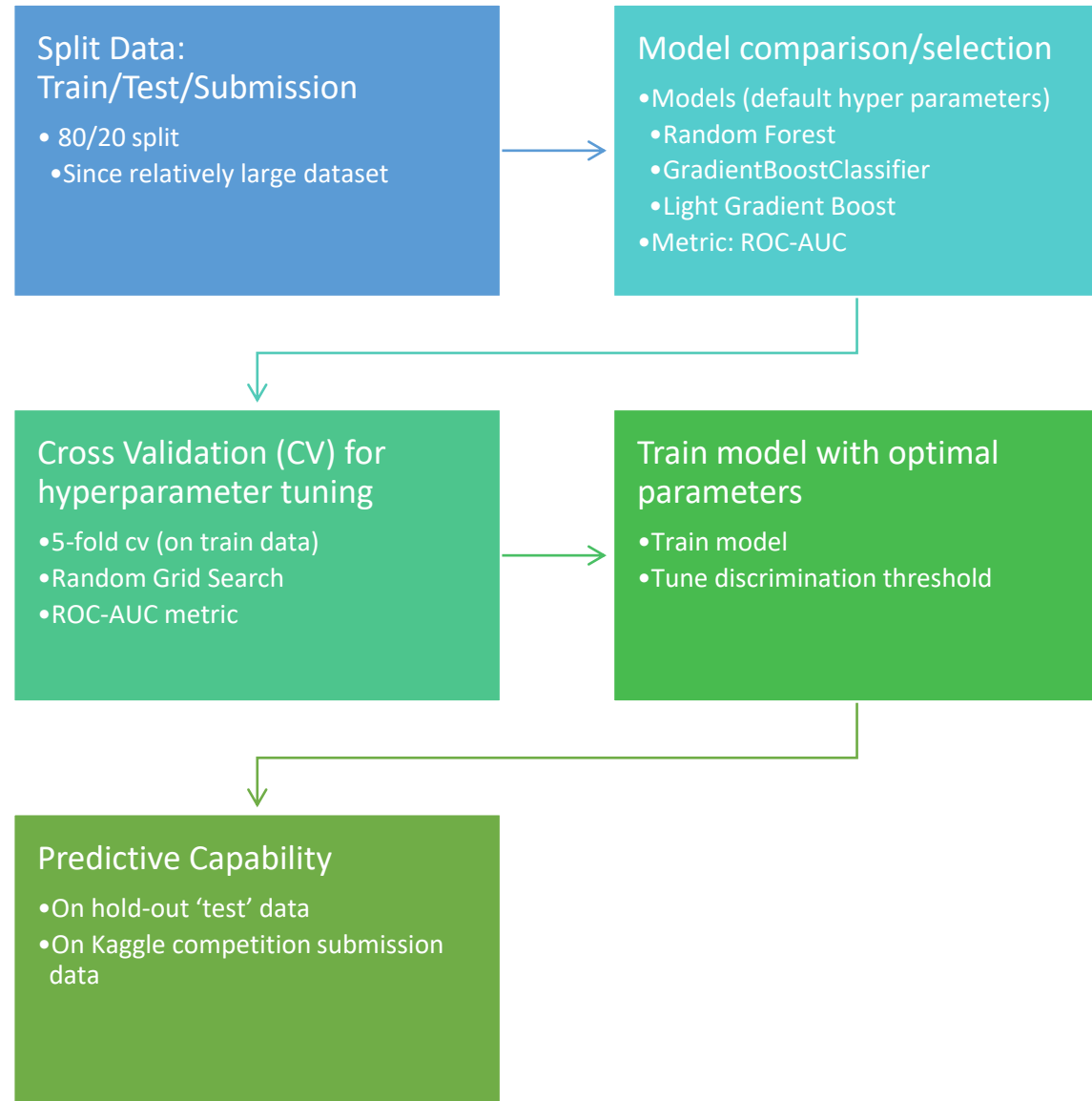


Top 20 Reorder counts by Aisle



modeling

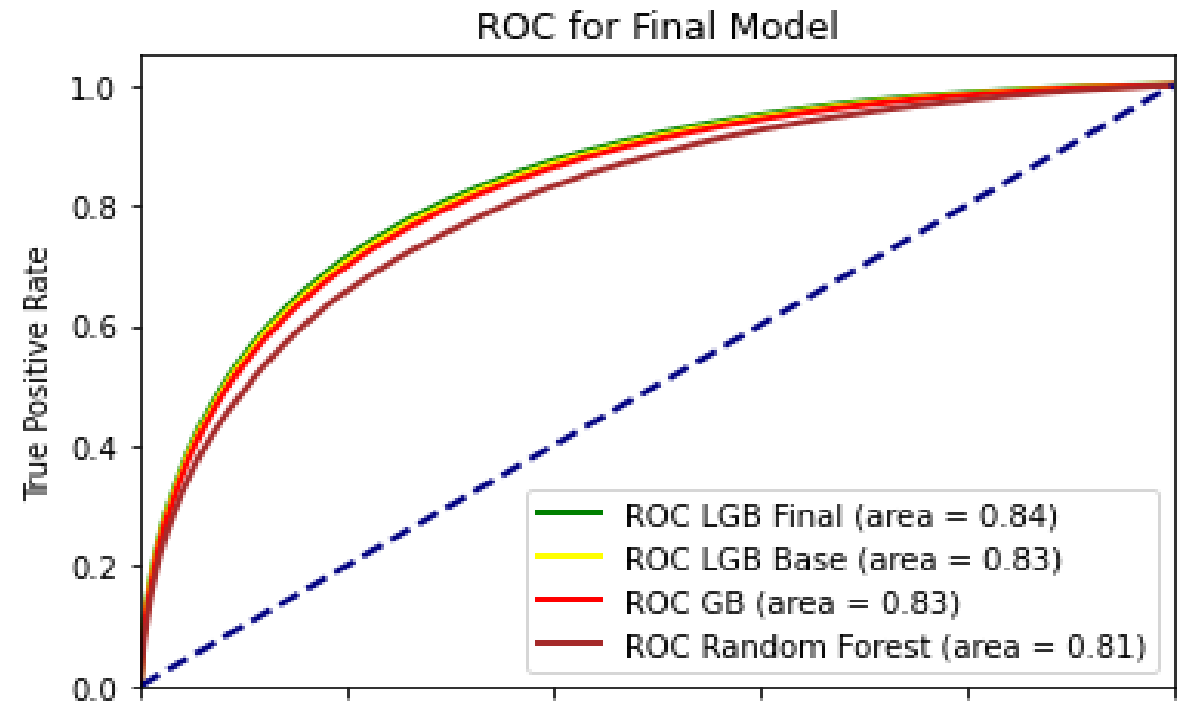
Modeling process steps

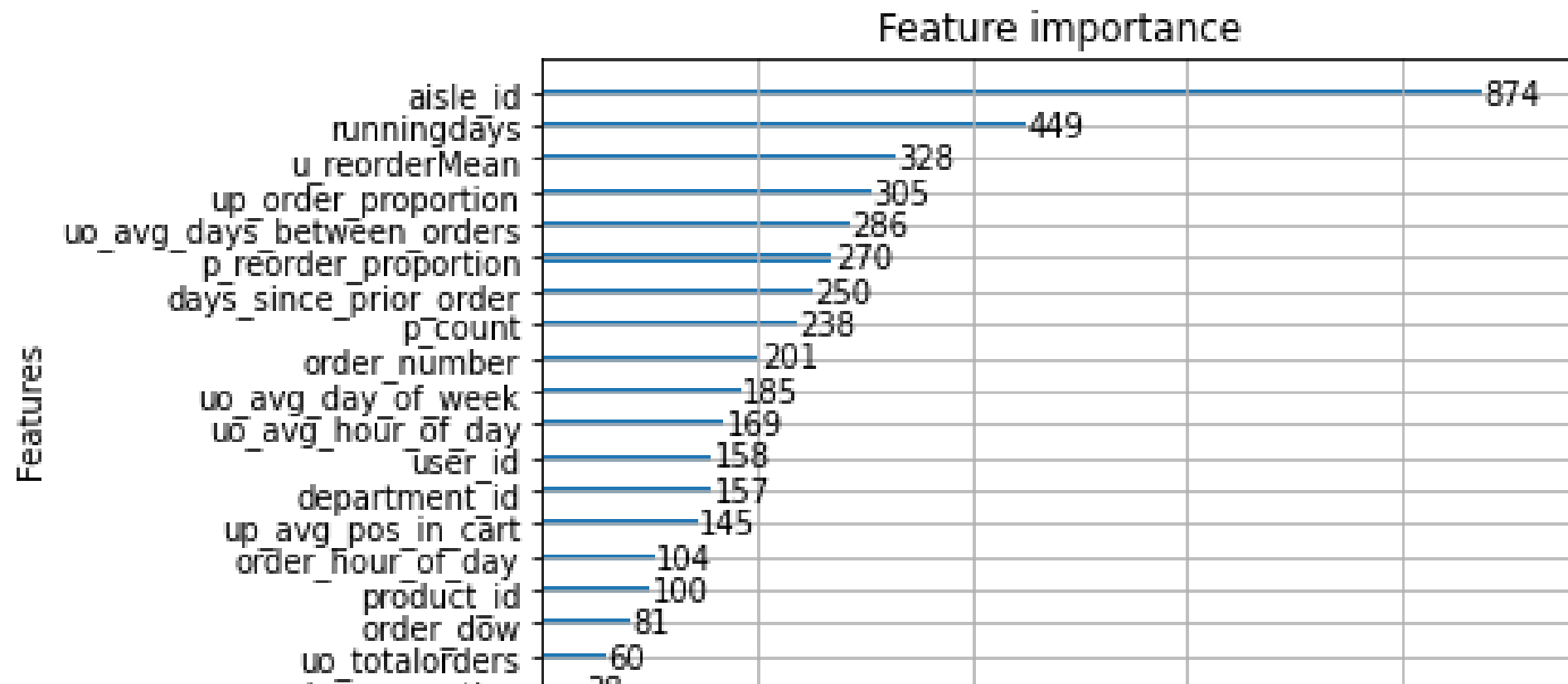


Model selection and results

- Light Gradient Boost (LGB Base) and Gradient Boost Classifier (GB) have similar ROC-AUC
- Train time is much better for LGB than for GB:
 - LGB: < 1 minute
 - GB: 30 minutes
- Light Gradient Boost (LGB Base) gave best metric / train time in comparison and improved slightly after hyperparameter tuning (LGB Final)

Model	ROC-AUC	Train time (mm:ss)
Random Forest	0.81	02:00
GradientBoostClassifier (GB)	0.83	30:00
Light Gradient Boost (LGB)	0.83	00:55

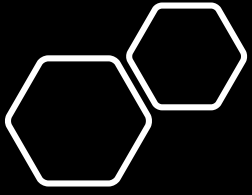




Model Results: Feature Importance

- Most discriminating features:
 - Aisle_id: Aisle id of product
 - Runningdays : days since user last ordered product
 - u_reorderMean : proportion of user's items that are repurchased items (aggregate over all orders)
 - up_order_proportion : proportion of orders a user purchases product

Example: Model results decision tree for first level



Model Metrics

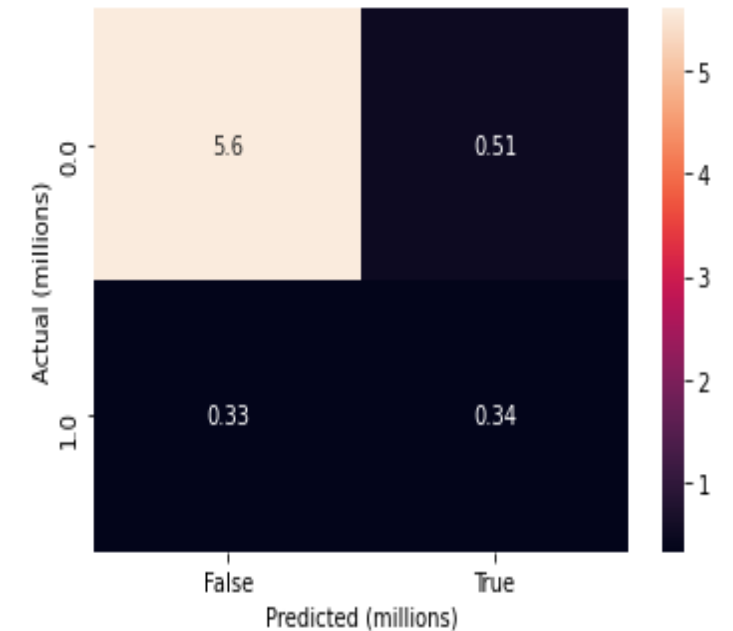
- **Test hold-out Data**
 - f1-score = 0.44
 - This is relatively close to the train data score, reflecting the splitting of the relatively large dataset.

Train Data Results

class	precision	recall	f1-score	support
0	0.94	0.92	0.93	6,106,860
1	0.40	0.51	0.45	662,213

* While the precision and recall, and ultimately the f1-score are not that great, we will discuss with respect to Kaggle competition on next slide

	Predicted: NO	Predicted: YES
Actual: NO	5,601,014	505,846
Actual: YES	326,745	335,468



Kaggle submission results

https://www.kaggle.com/c/instacart-market-basket-analysis/leaderboard

Search

Overview Data Notebooks Discussion **Leaderboard** Rules Team

My Submissions Late Submission

#	△pub	Team Name	Notebook	Team Members	Score ?	Entries	Last
1	—	胡萝卜			0.40914	62	3y
2	—	===== KEEP OUT 🐼 =====			0.40820	138	3y
3	—	sjv			0.40810	76	3y

Leaderboard score

Leaderboard vs submission score

Kaggle submission score not that unreasonable compared to leaderboard score, when we consider we have developed a basic model, which can be further improved

Submission score

class	precision	recall	f1-score	support
0	0.94	0.92	0.93	6,106,860
1	0.40	0.51	0.45	662,213

Training score

1

My Submissions

Private Score	Public Score
0.37660	0.37723

Training score vs submission score:

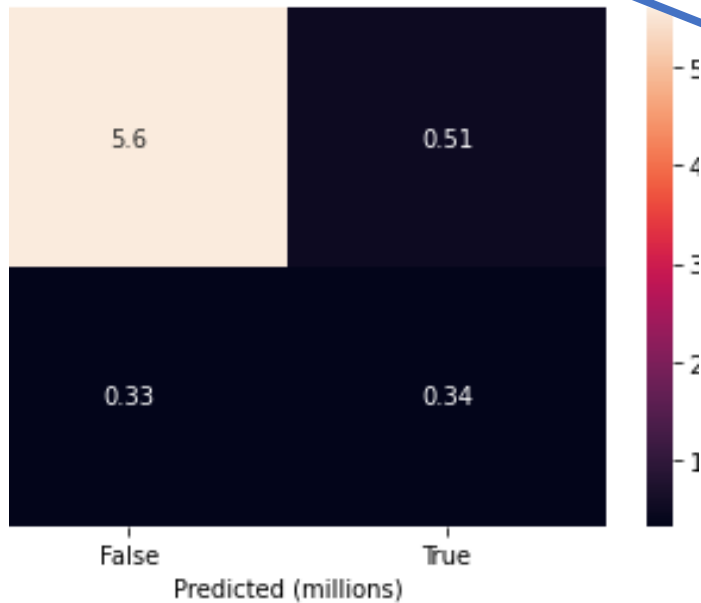
Submission score differs from test score more than we would expect.

➤ Possible explanations:

- differences in generating input cross product
- 'none' prediction option discussed on Kaggle community board, along with a competitor-supplied 'kernel' to address

➤ Difference not explored here, and left for future consideration

	precision	recall	f1-score	support
0	0.94	0.92	0.93	6,106
1	0.40	0.51	0.45	662,2



Prediction Capabilities

- F1 score = 0.45 on the test data
 - Recall = 0.51
 - we are predicting only about half the reordered items as reordered
 - Precision = 0.4
 - we are predicting more non reordered items as reordered than we are correctly predicting reordered items (in part due to imbalanced data)
- Apparently, the not so great prediction capability is one of the reasons there is a competition for improvement

Conclusion

- We developed a basic model for the 'buy again' problem
- Used a Light Gradient boosting algorithm
- Prediction capabilities are not that unreasonable when compared to the competition results
- Prediction capabilities for this problem, in general, have room for improvement
- Future work considerations
 - Additional feature engineering
 - Refine current features
 - For example, for reorder proportions, consider only orders after which the item was first purchased, rather than all orders
 - Inclusion of additional features
 - Implementation of the 'none' prediction feature