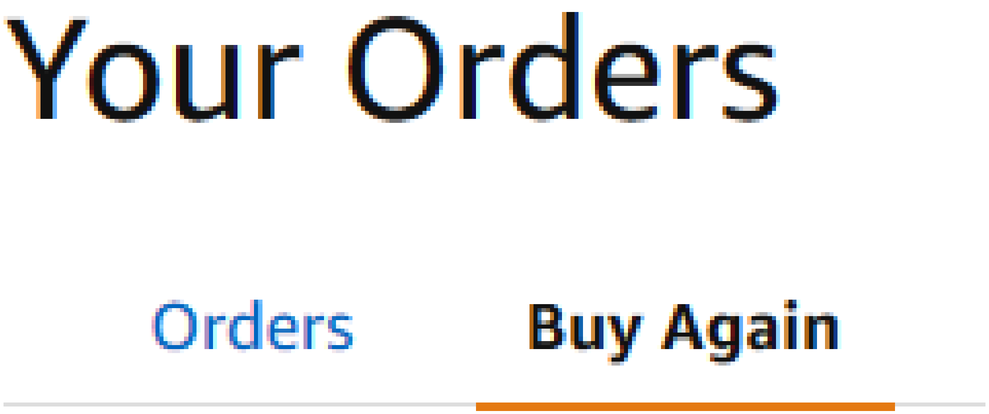


"Buy Again"
Market Basket Analysis

Springboard Data Science Career Track - Capstone One

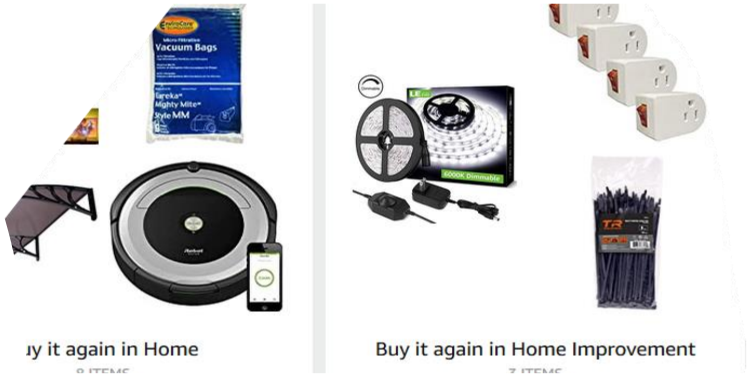
Rose Zdybel

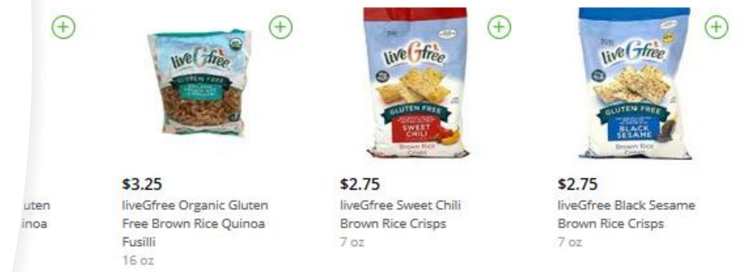
The problem

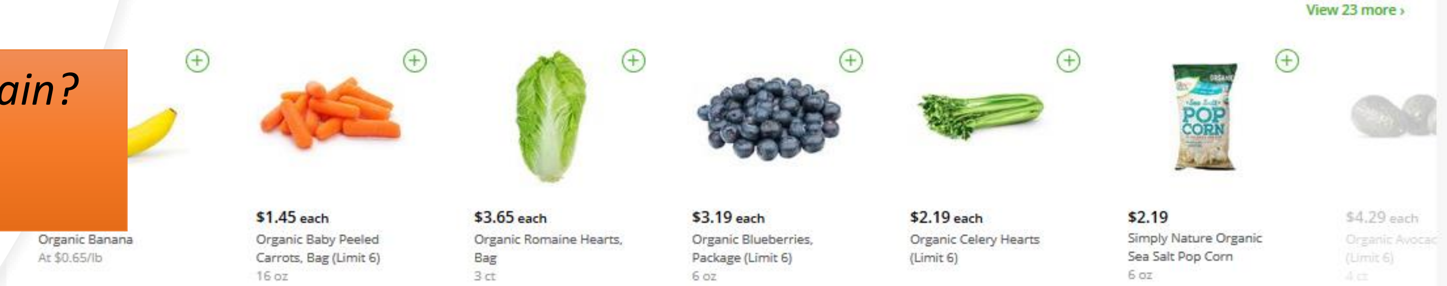
“Buy again” recommendations are not always that accurate and timely, even for frequently purchased items.

How well can we predict what items a user will buy again?

What factors have predictive capability?



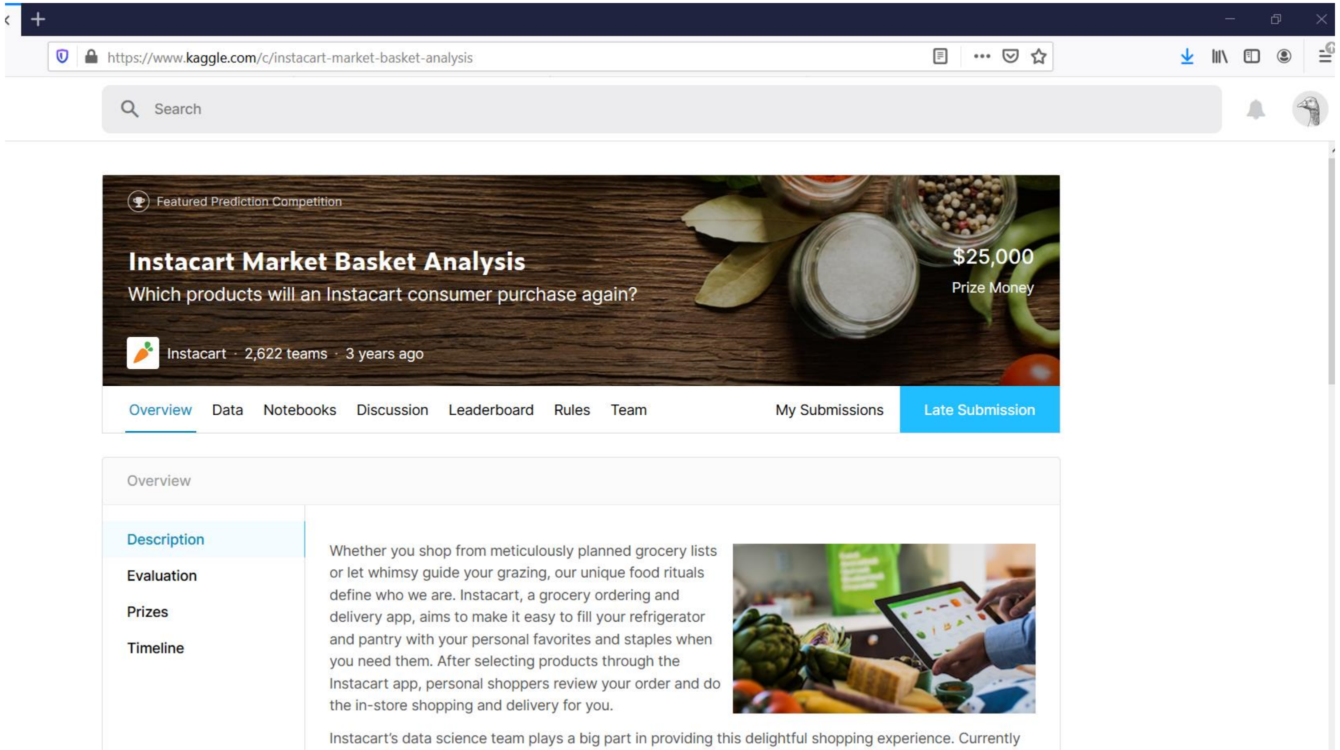




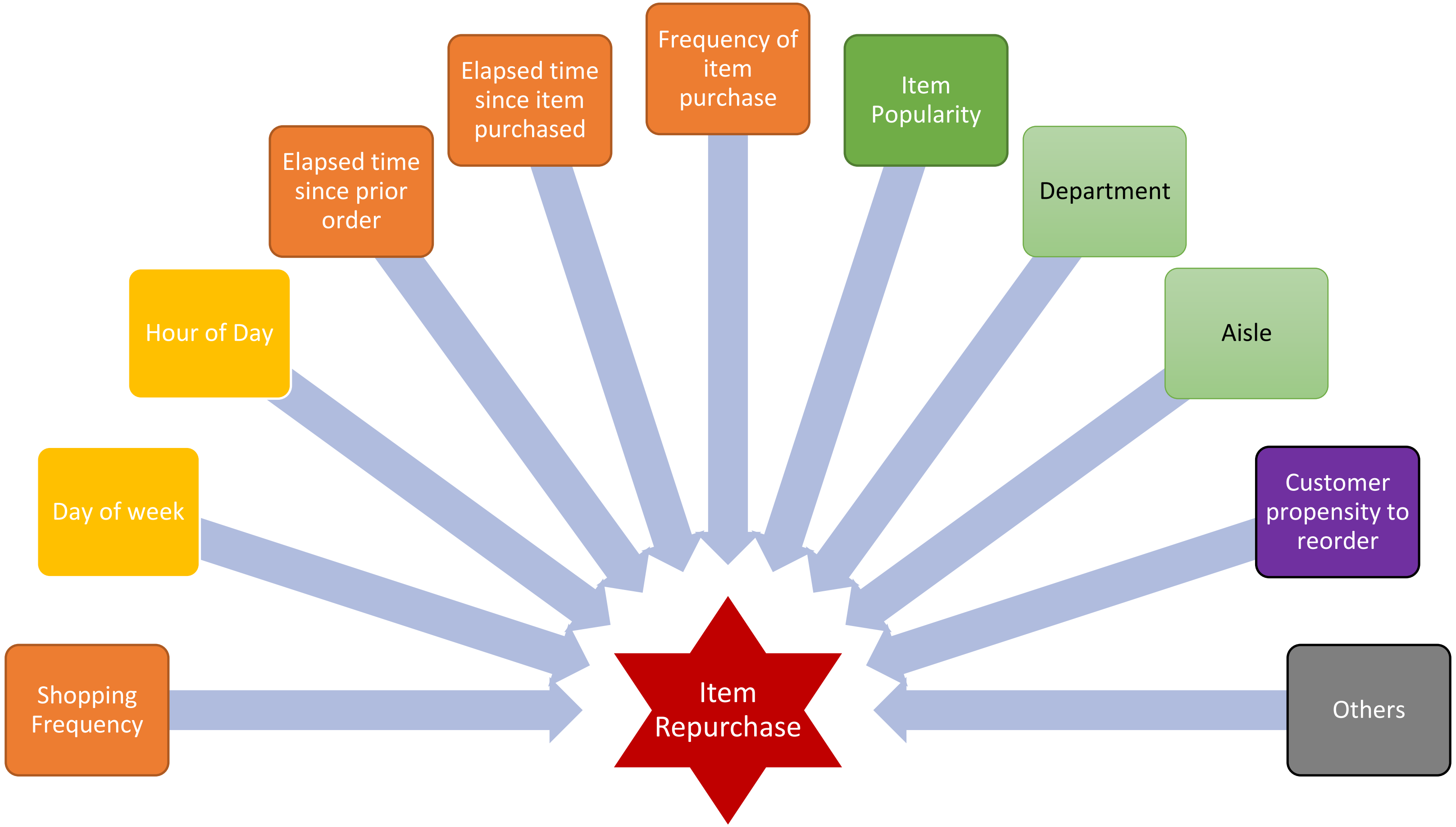
Who cares?

Retailers are interested in improving their recommendation systems

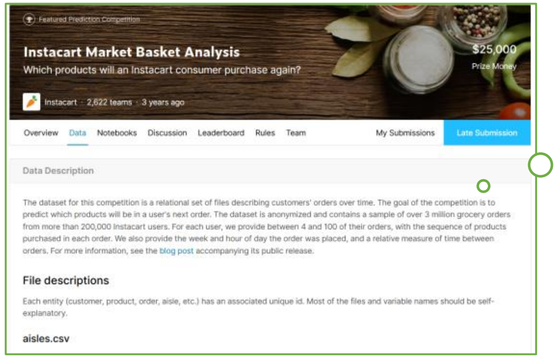
- Improve customer experience
- Keep customers coming back
- Increase sales



What factors might affect item repurchase?



Data Information

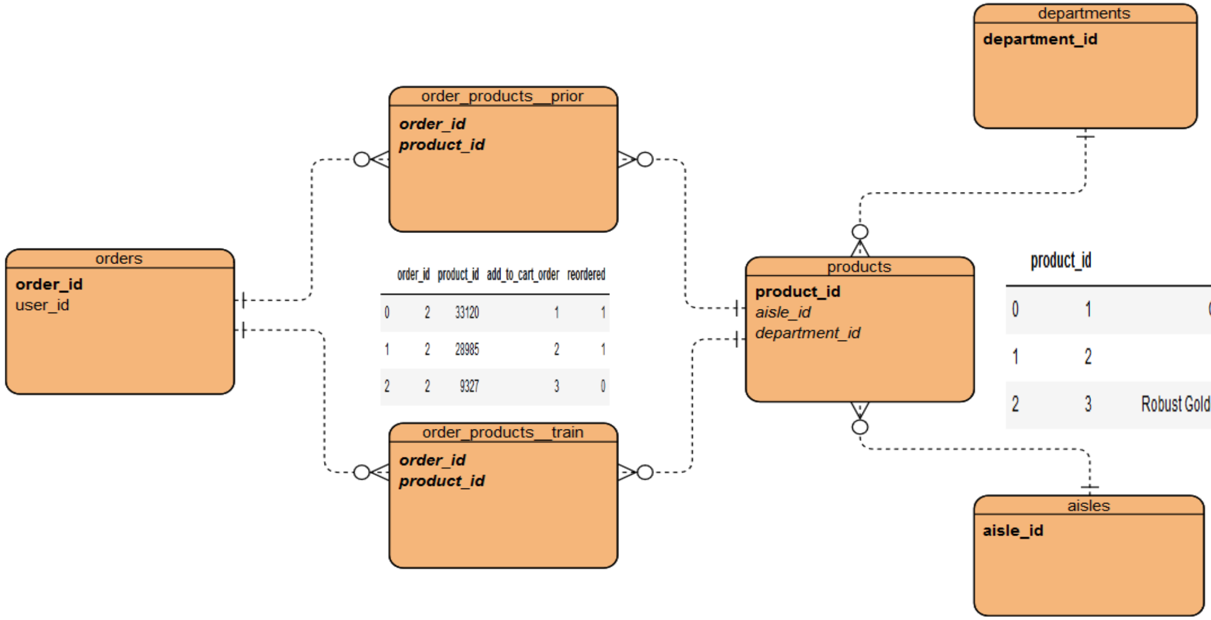


Data Source

	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



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

	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

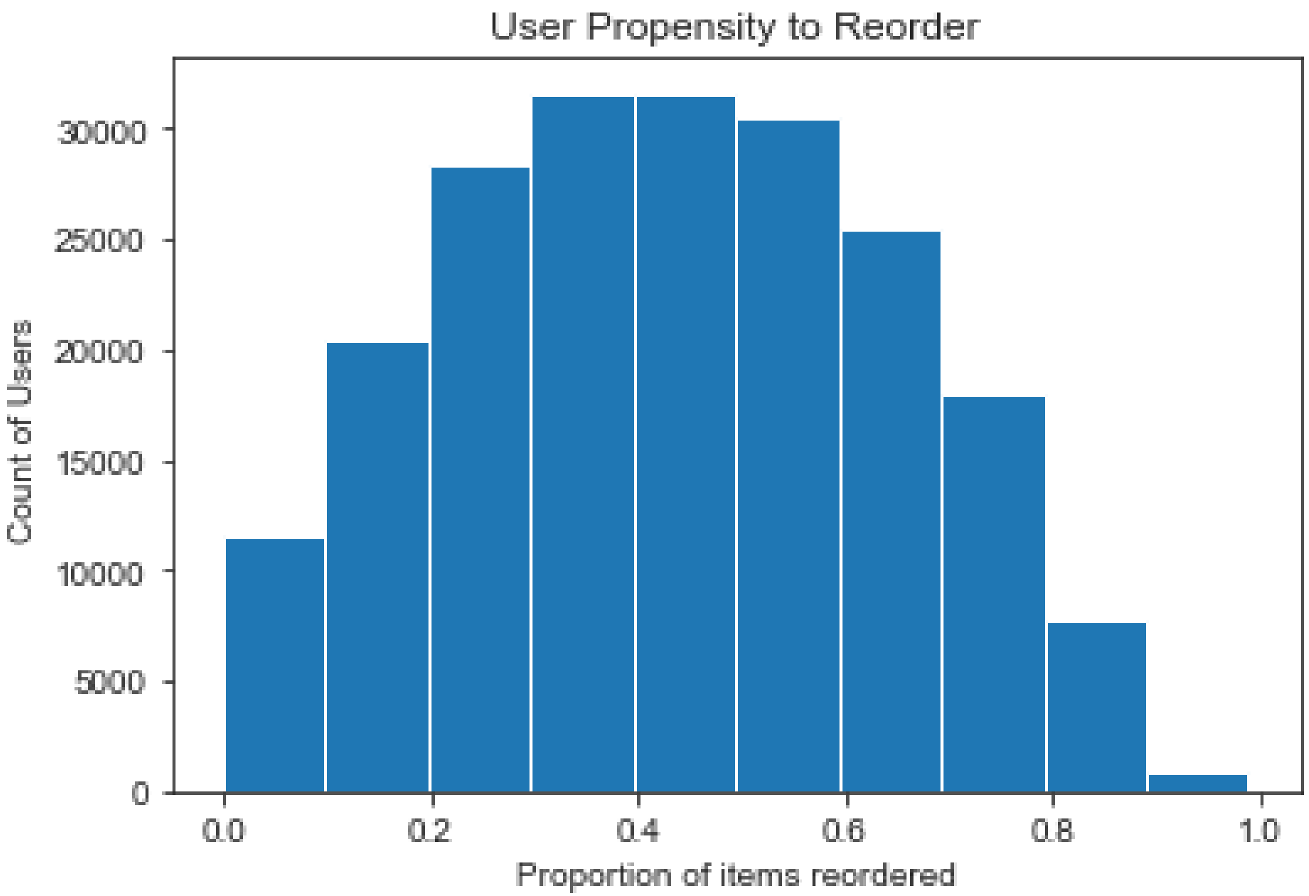
- by Dept

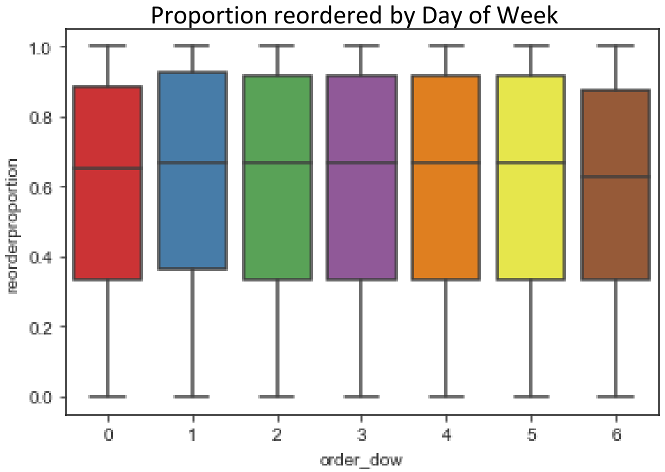
Most Reordered

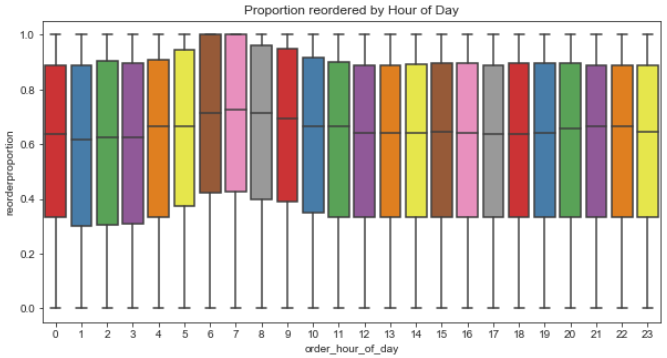
- By Product
- By Department
- By aisle

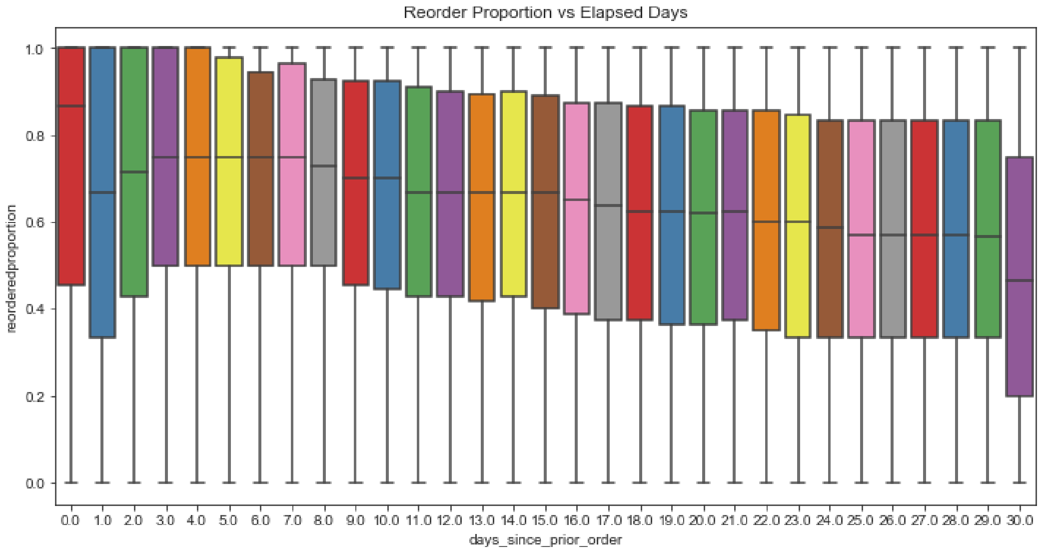
* Details are in this presentation

User Propensity to Reorder



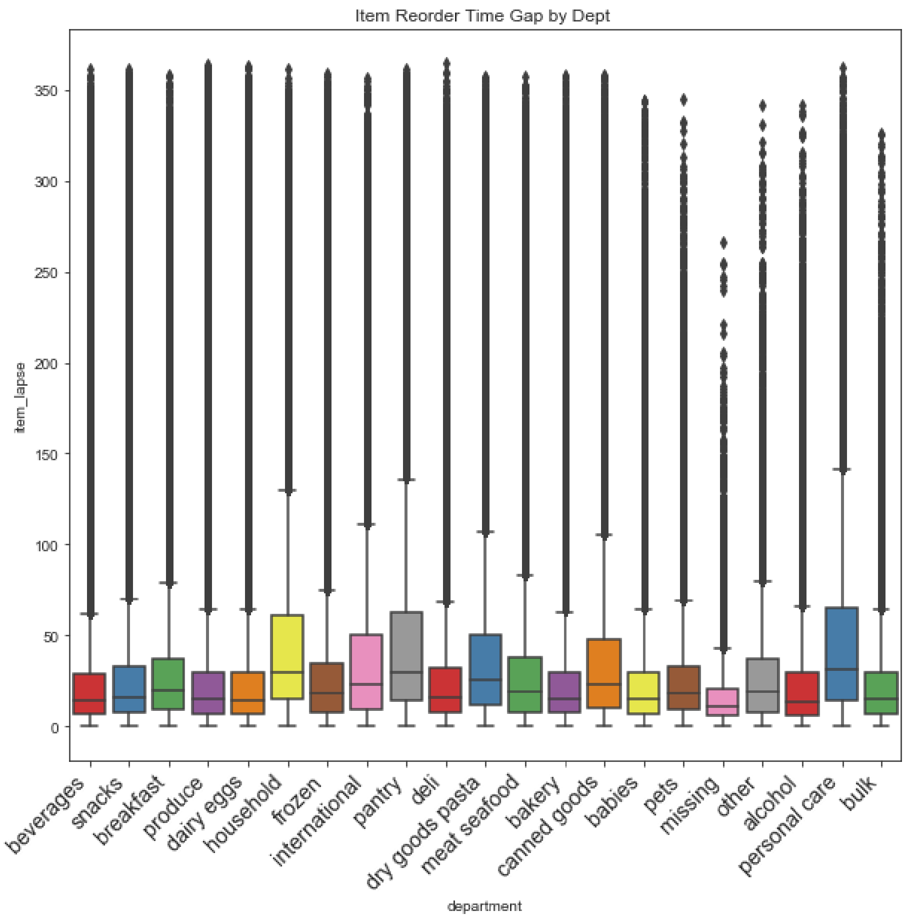




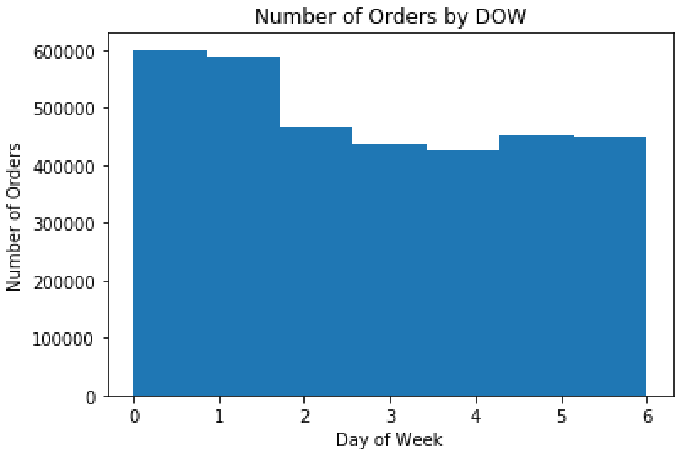


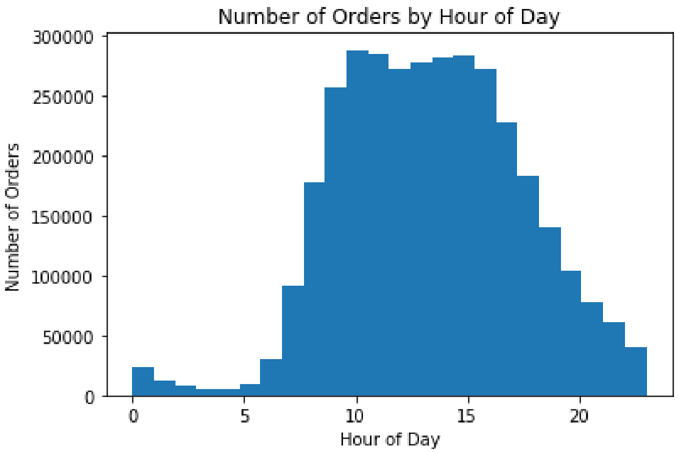
Proportion of order items repurchased: Time-based variations

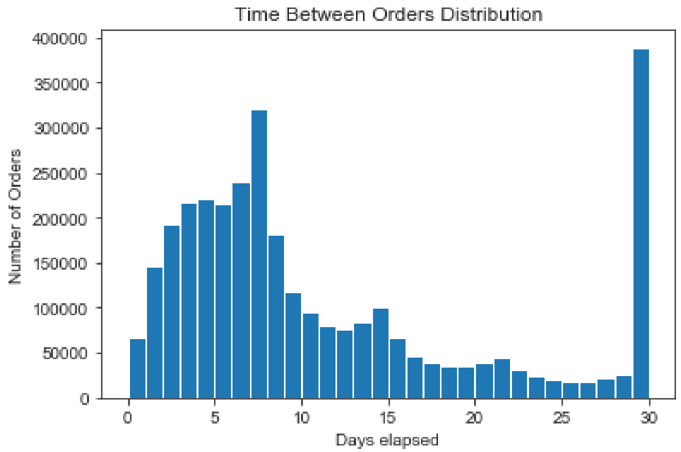
Product repurchase: Time gap variations by department



How often and when do users shop?

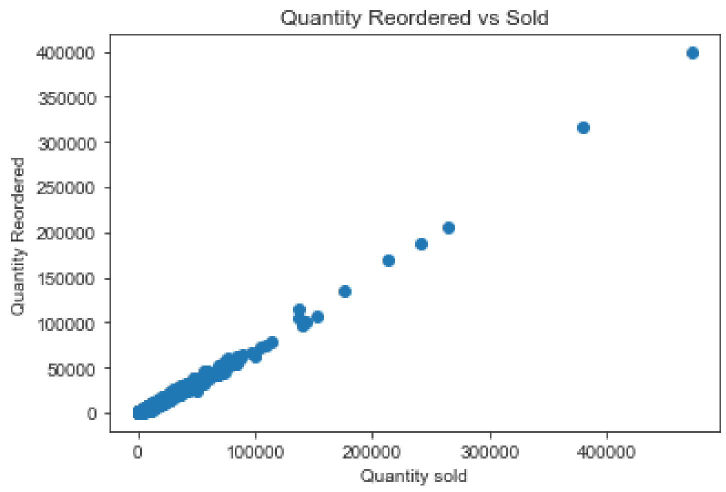


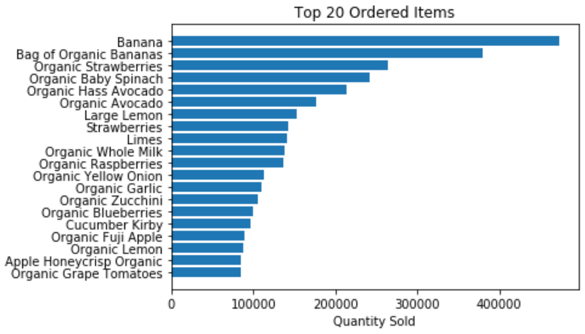


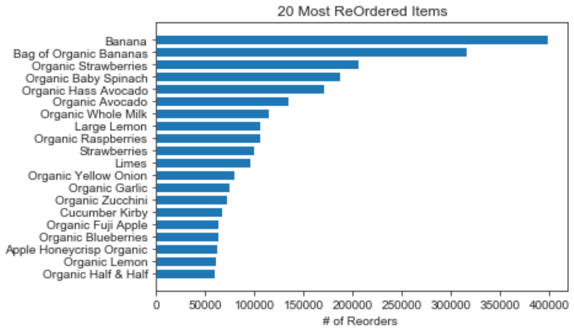


Most Popular:
Products, Departments, Aisles

- Quantity Sold
- Quantity Reordered



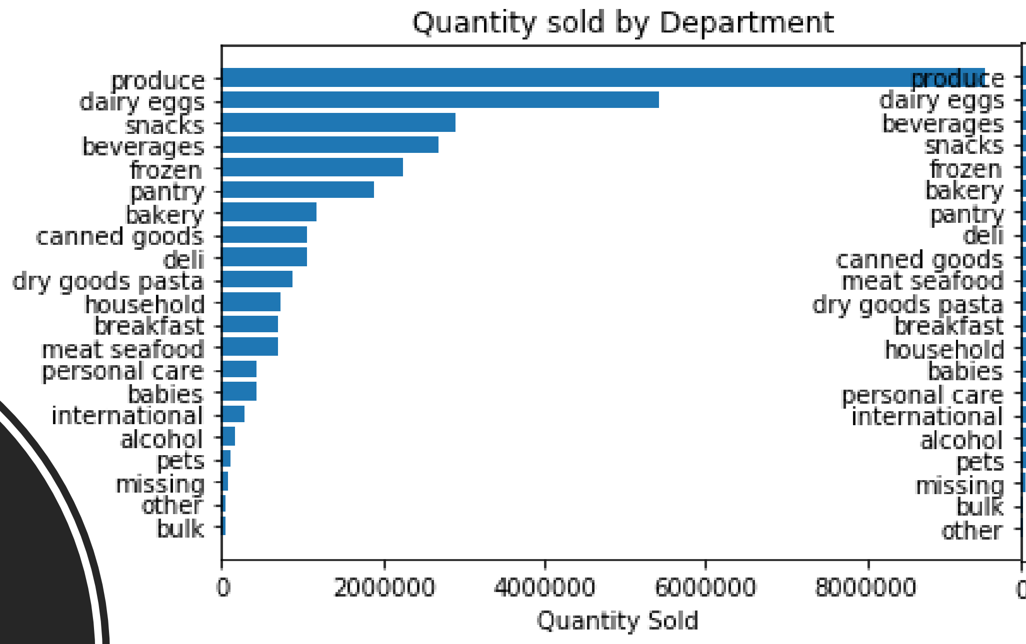


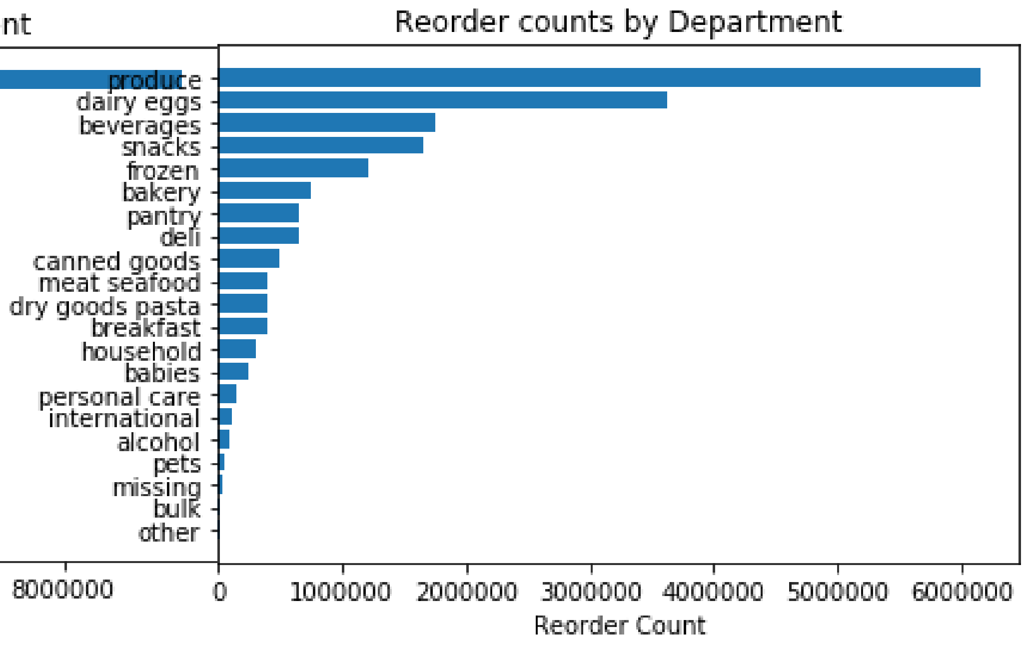


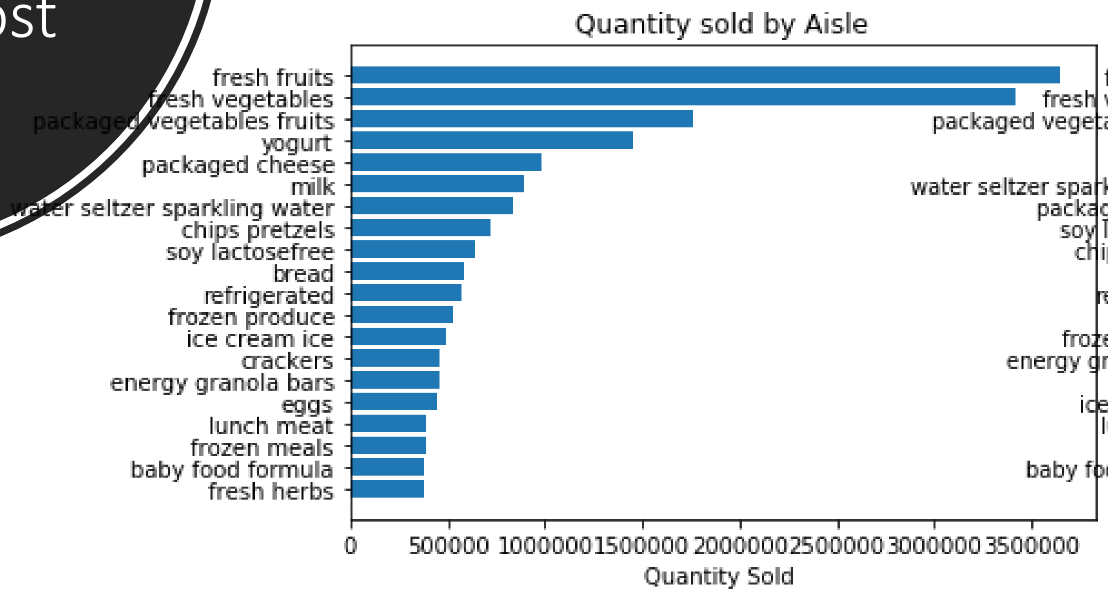
Top Products:
Sold, reordered, correlation

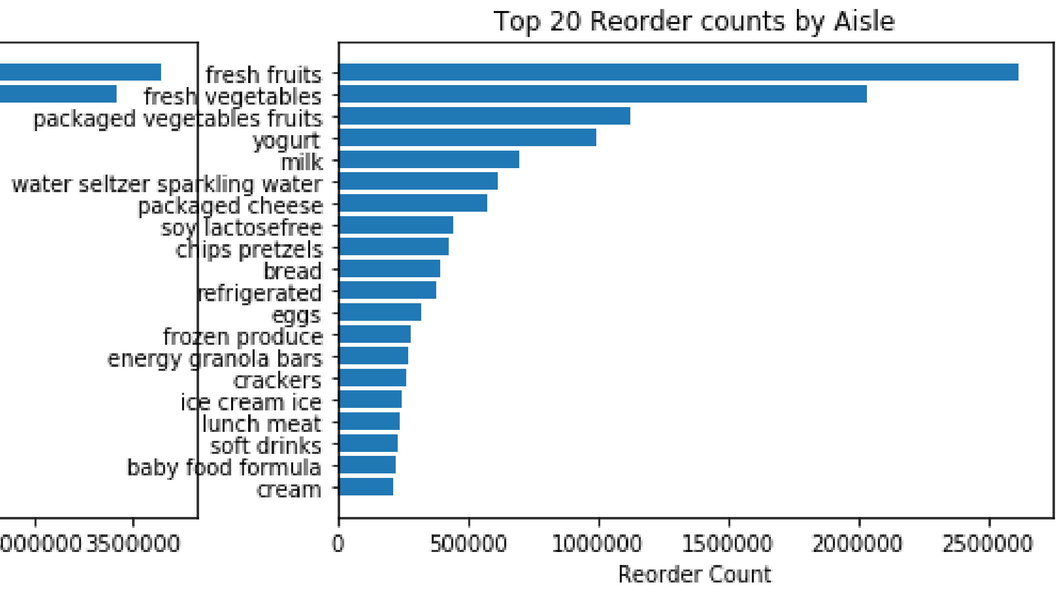
Aisles & Departments

No surprise: Most popular have the most reorders









modeling

Modeling process steps

Split Data: Train/Test/Submission

- 80/20 split
  - Since relatively large dataset

Model comparison/selection

- Models (default hyper parameters)
  - Random Forest
  - GradientBoostClassifier
  - Light Gradient Boost
- Metric: ROC-AUC

Cross Validation (CV) for hyperparameter tuning

- 5-fold cv (on train data)
- Random Grid Search
- ROC-AUC metric

Train model with optimal parameters

- Train model
- Tune discrimination threshold

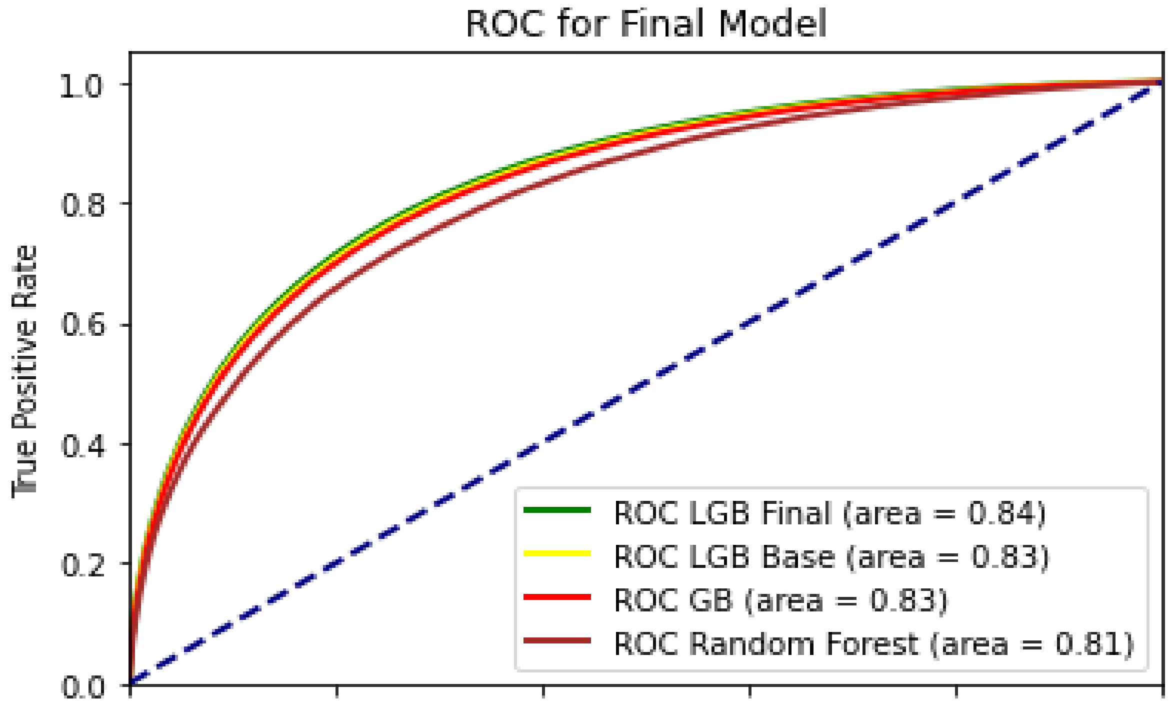
Predictive Capability

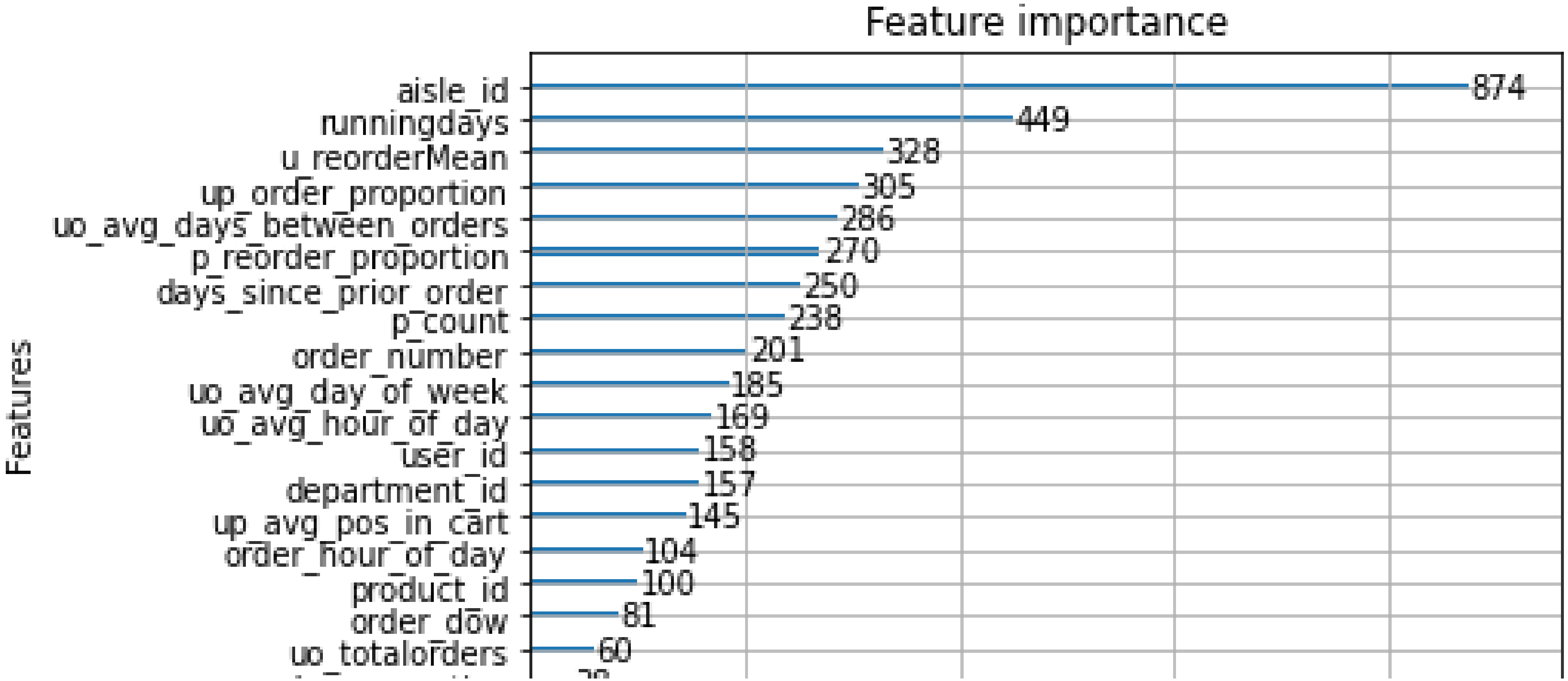
- On hold-out ‘test’ data
- On Kaggle competition submission data

Model selection and results

- Light Gradient Boost (LBG 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 (LBG 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

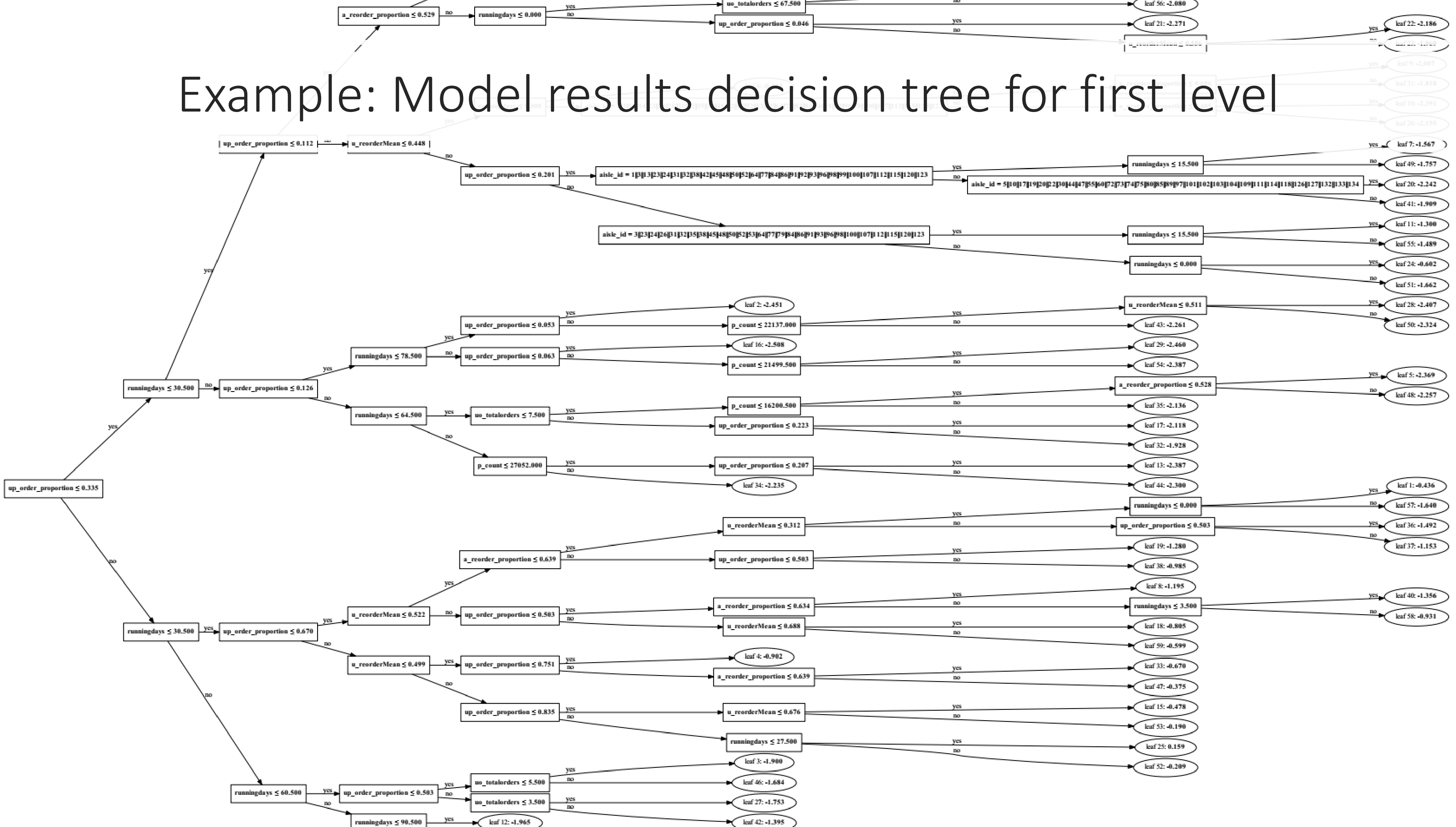




Model Results: Feature Importance

- 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

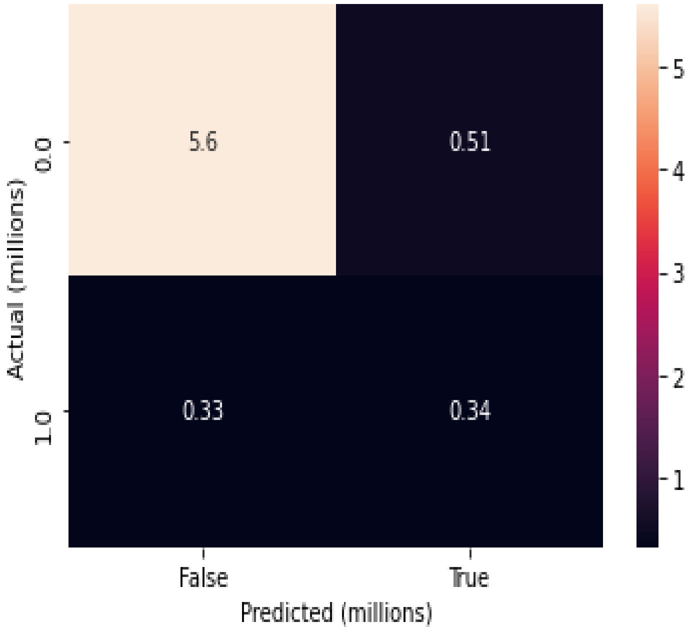
- **Test hold-out Data**
  - f1-score = 0.44
  - This is relatively close to the train data score, reflecting the splitting of the relatively large dataset.

Train Data Results

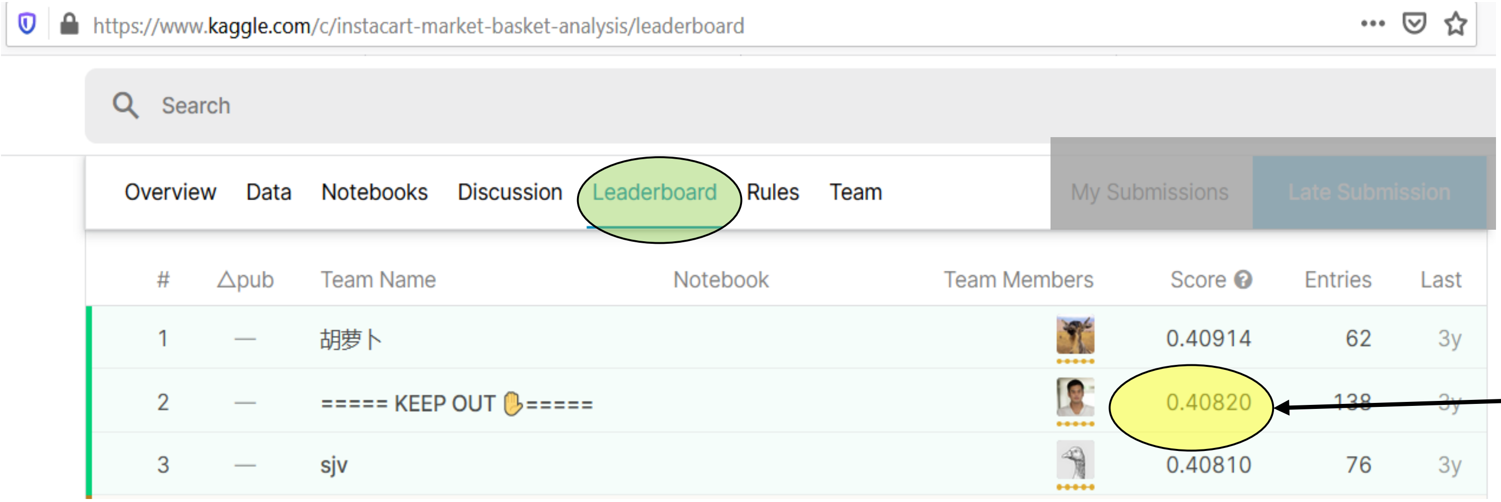
class	precision	recall	f1-score	support
0	0.94	0.92	0.93	6,106,860
1	0.40	0.51	0.45	662,213

* 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



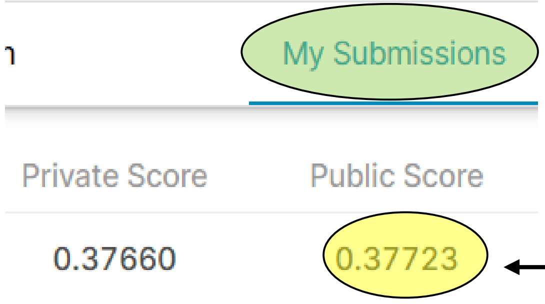
Leaderboard score

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

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



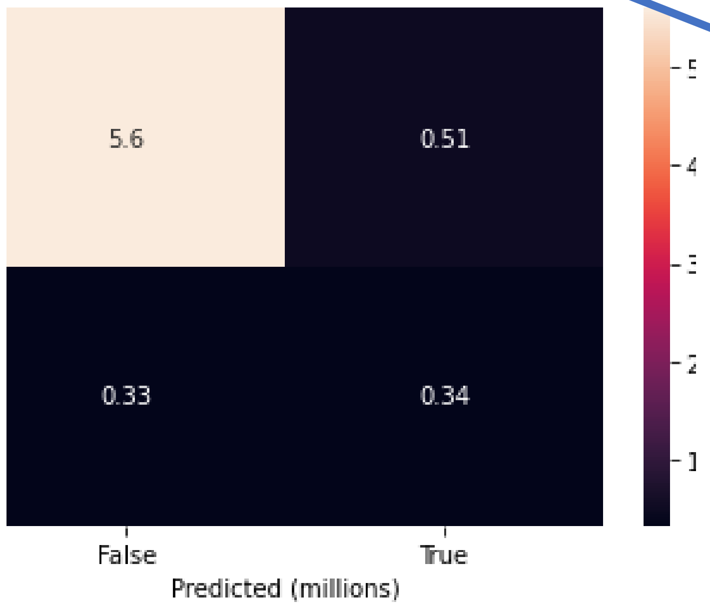
Submission score

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	suppo
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