Predicting a User's Next Instacart Order

Elliot Wilens
Metis Data Scientist

1

Objective and Problem Setup

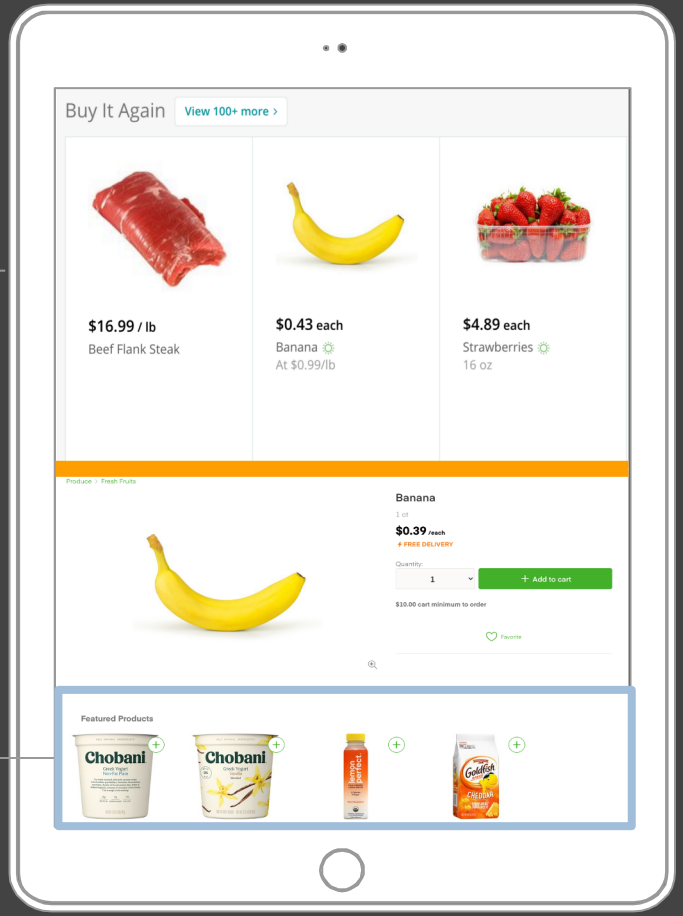
Objective: To predict all of the reorders in a user's next cart

But first... why?

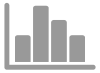
Use Cases:

1) Buy-it-again recommendations

2) Frequently bought with…



The Dataset



33,819,106

Total rows (1 per product per order)

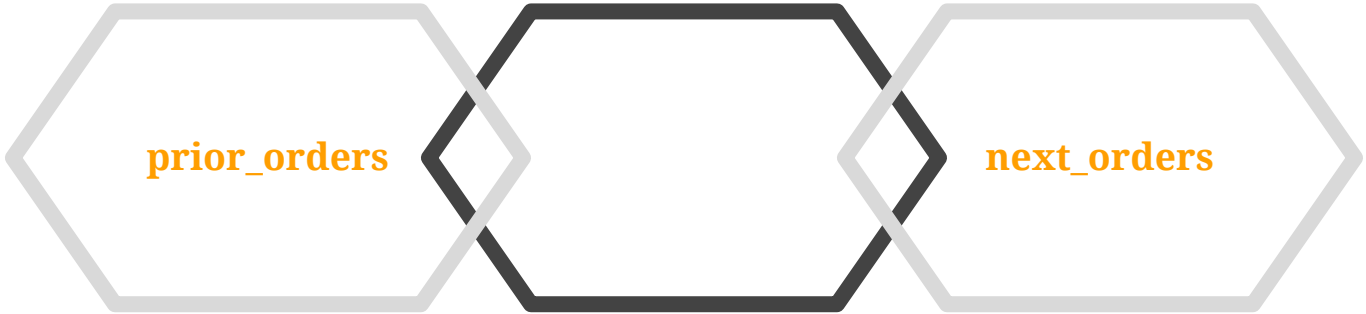
3,346,08

Total orders

3

337,418

Total users



Contains prior order details for all users

Contains order details for each user's 'next' order

THE DATASET

Contains prior order details for all users in next_orders

Contains order details for each user's 'next' order

df_features

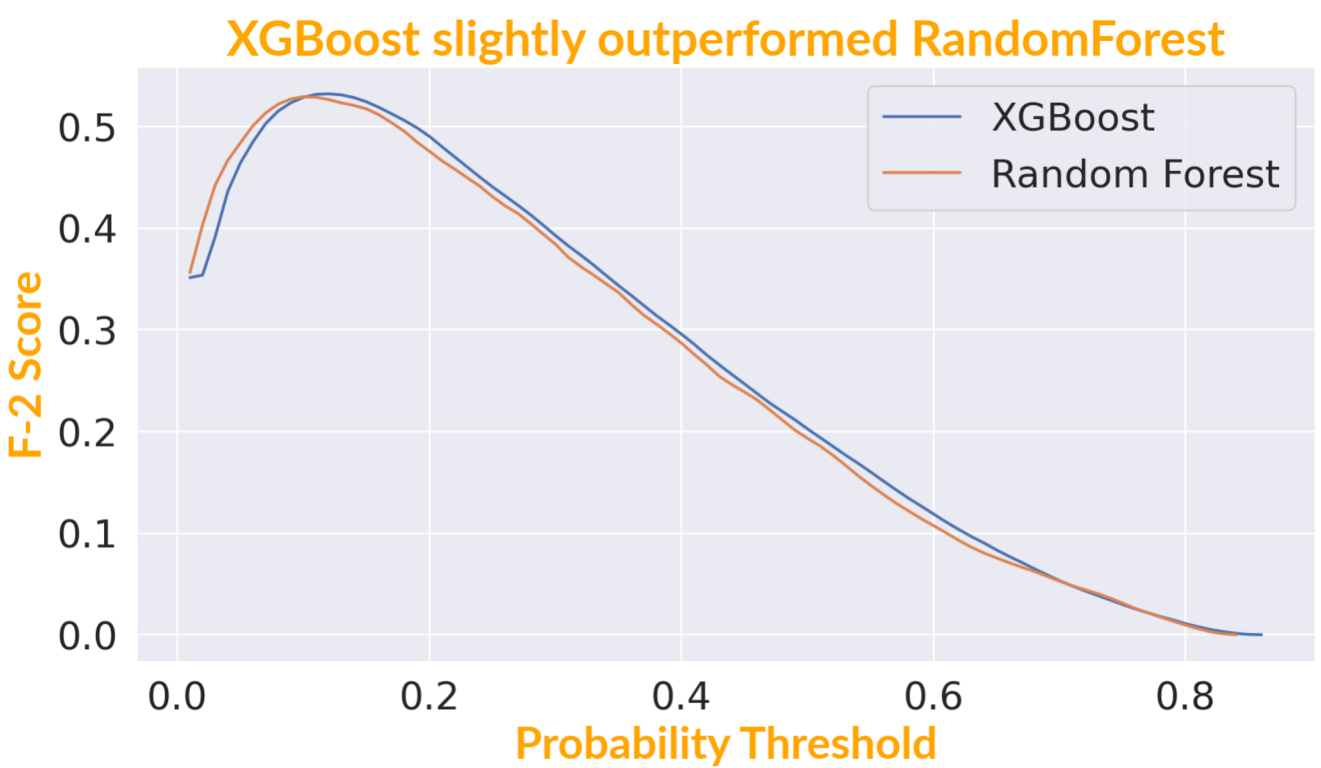
- Contains user & product statistics from prior_orders
- Contains next_order details
- Modeling done on these inputs

3

Model Selection & Results

Class imbalance apparent

- 10% of target are reorders



Used grid search to define the optimal XGBoost parameters:

Took over 28 hours using 16 vCPUs and just 25% of the dataset!

learning_rate	**0.009**
n_estimators	**400**
max_depth	**7**
colsample_bytree	**0.8**
min_child_weight	**9**

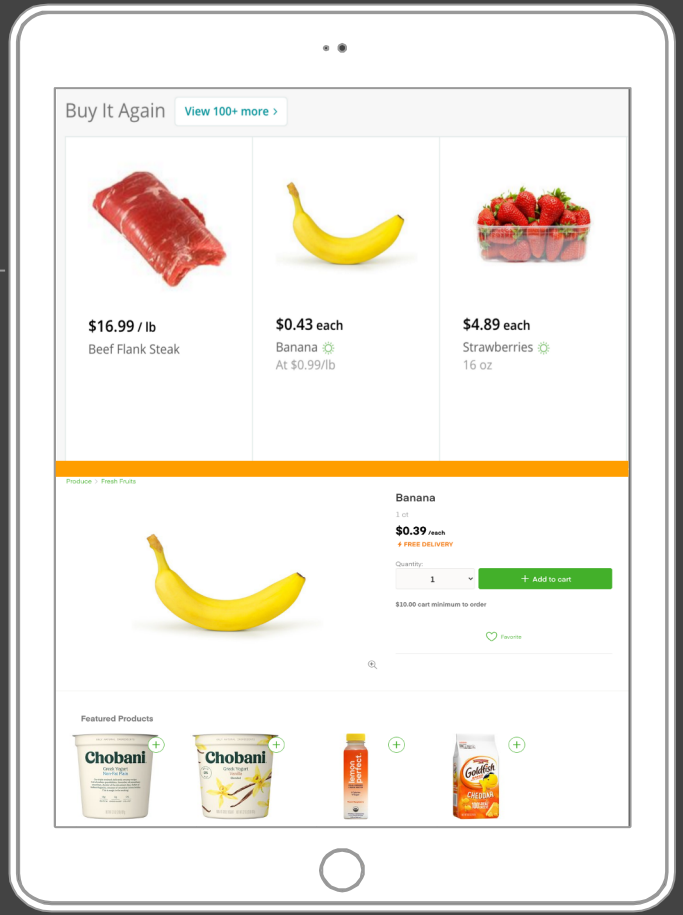
Scoring

2018 Kaggle competition crowned winner based on resulting F-1 scores

Do you remember our use case?

Use Cases:

1) Buy-it-again recommendations
2) Frequently bought with…



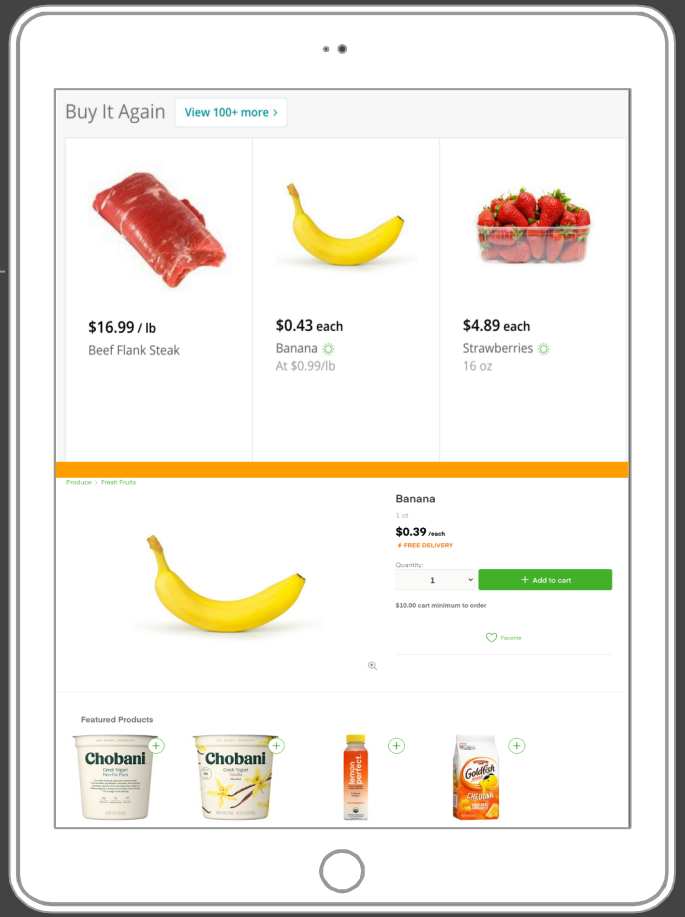
Use Cases:

1) Buy-it-again recommendations
2) Frequently bought with…

But how do these features help Instacart?

1. User ease of use

1. **Increase product conversion rates**



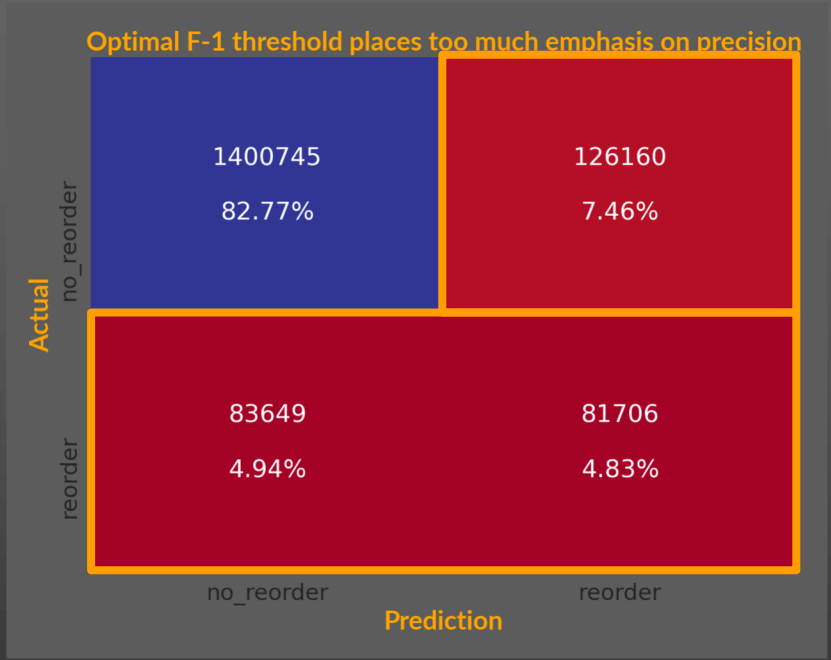
What's Instacart's risk of incorrectly classifying an input as positive (reorder)?

Not Much.

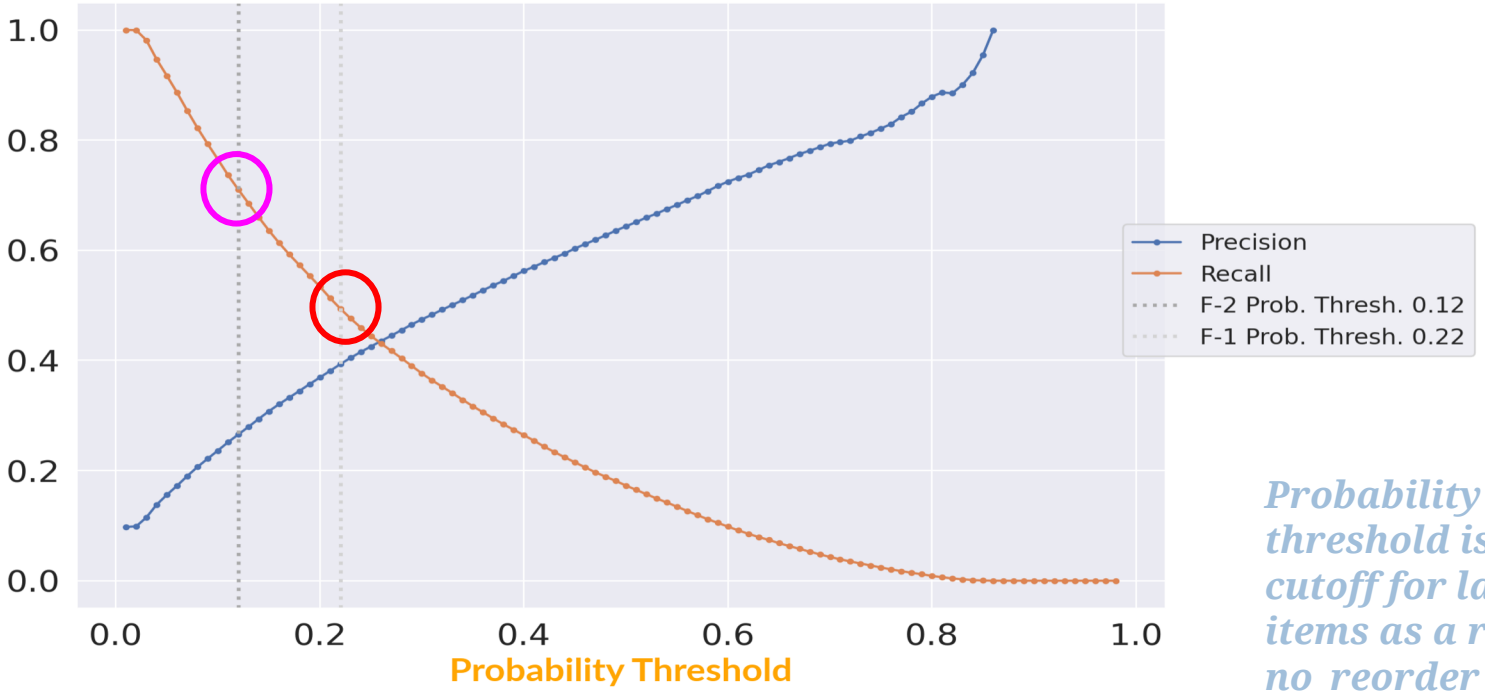
In fact, we may be better off including items that the user is *less likely to buy* based on their prior orders. This will help Instacart to **increase conversion**!

Therefore, we ought to prioritize **recall**!

Recall = True Positives / Actual Positives



Choosing a threshold with an ideal recall/precision balance using F-2 Scores



Probability threshold is the cutoff for labeling items as a reorder/ no_reorder

MODEL RESULTS

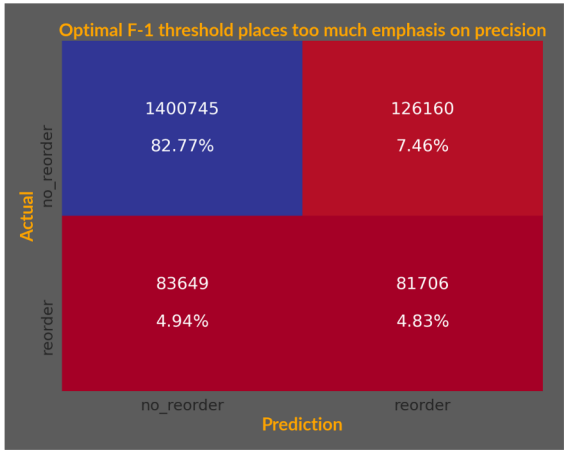
Adjusted F-2 Score

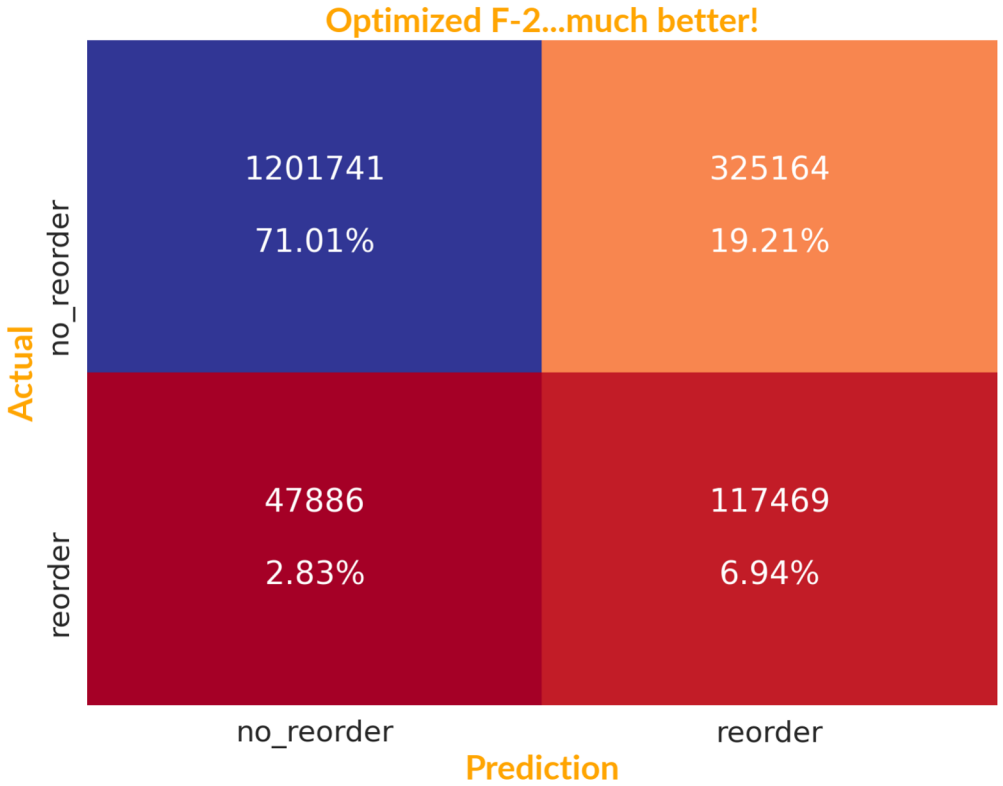
0.53

Probability Threshold

0.12

Previous threshold (0.22)





THANK YOU!

Any questions?

You can find me...

On LinkedIn: www.linkedin.com/in/elliotwilens/

On GitHub: www.github.com/edubu2

wilensel@gmail.com

THANK YOU!

APPENDIX

Algorithms

- XGBoost
- RandomForest

Techniques

- Grid Search & K-Fold Cross Validation
- Precision/Recall Curves
- F-1 & F-beta (F-2) scores

Python Libraries

- Scikit-learn, StatsModels
- multiprocessing
- pickle
- pandas/numpy

Tools

- PostgreSQL
- Tableau
- Google Cloud Deep Learning VMs
- Jupyter Notebook

Some key features (32 total):

User Features	Product Features	User/Product Features
avg_cart_size	percent_reorders	order_streak
days_since_prior_order	qty_sold	last_five_buys
avg_time_between_orders	qty_reordered	In_last_cart (0/1)