Predicting a User's Next Instacart Order

Elliot Wilens
Metis Data Scientist

Objective and Problem Setup

Algorithms

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

Tools

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

Python Libraries

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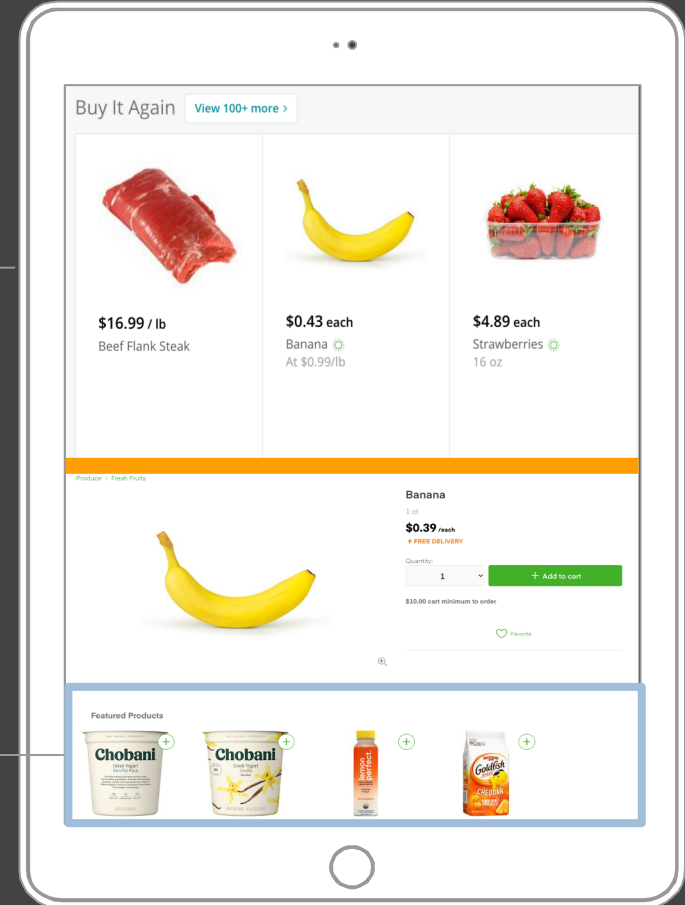- ☐ multiprocessing
- ☐ pickle
- ☐ pandas/numpy

"

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...

2

Dataset & Feature Engineering
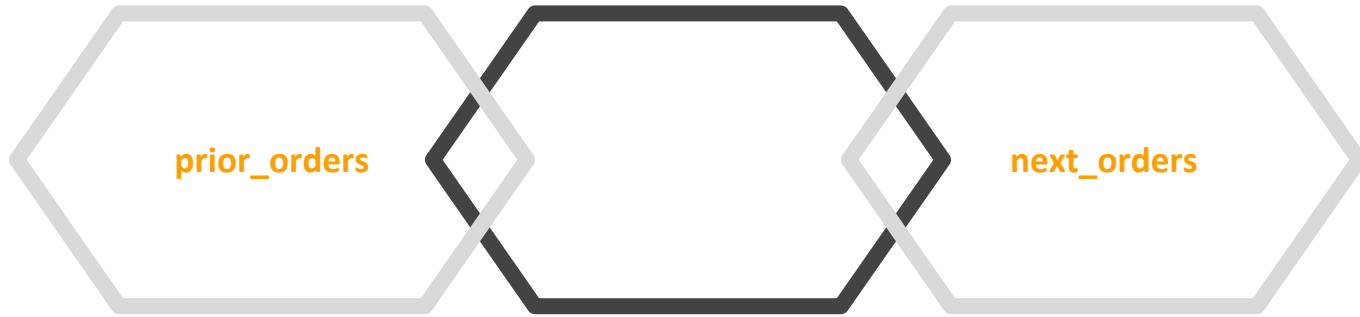
33,819,106

Total rows (1 per product per order)
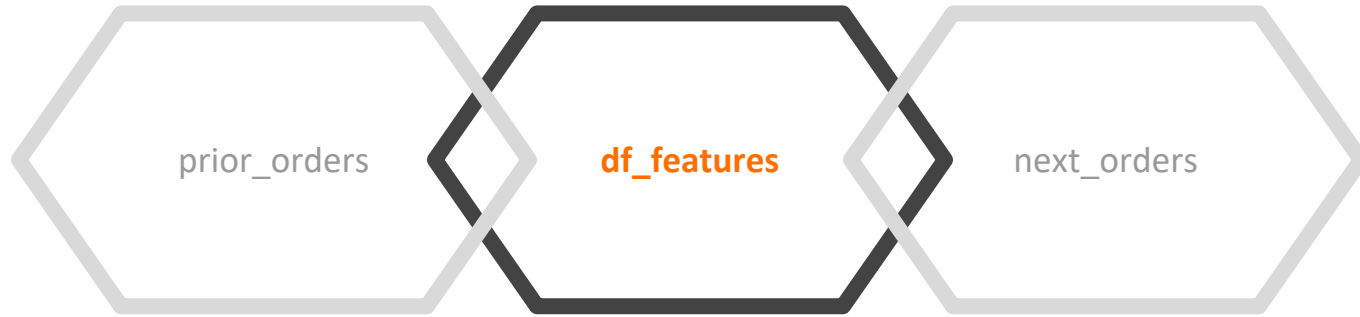
3,346,083

Total orders

337,418

Total users

prior_orders

next_orders

Contains prior order
details for all users

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

prior_orders

df_features

next_orders

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
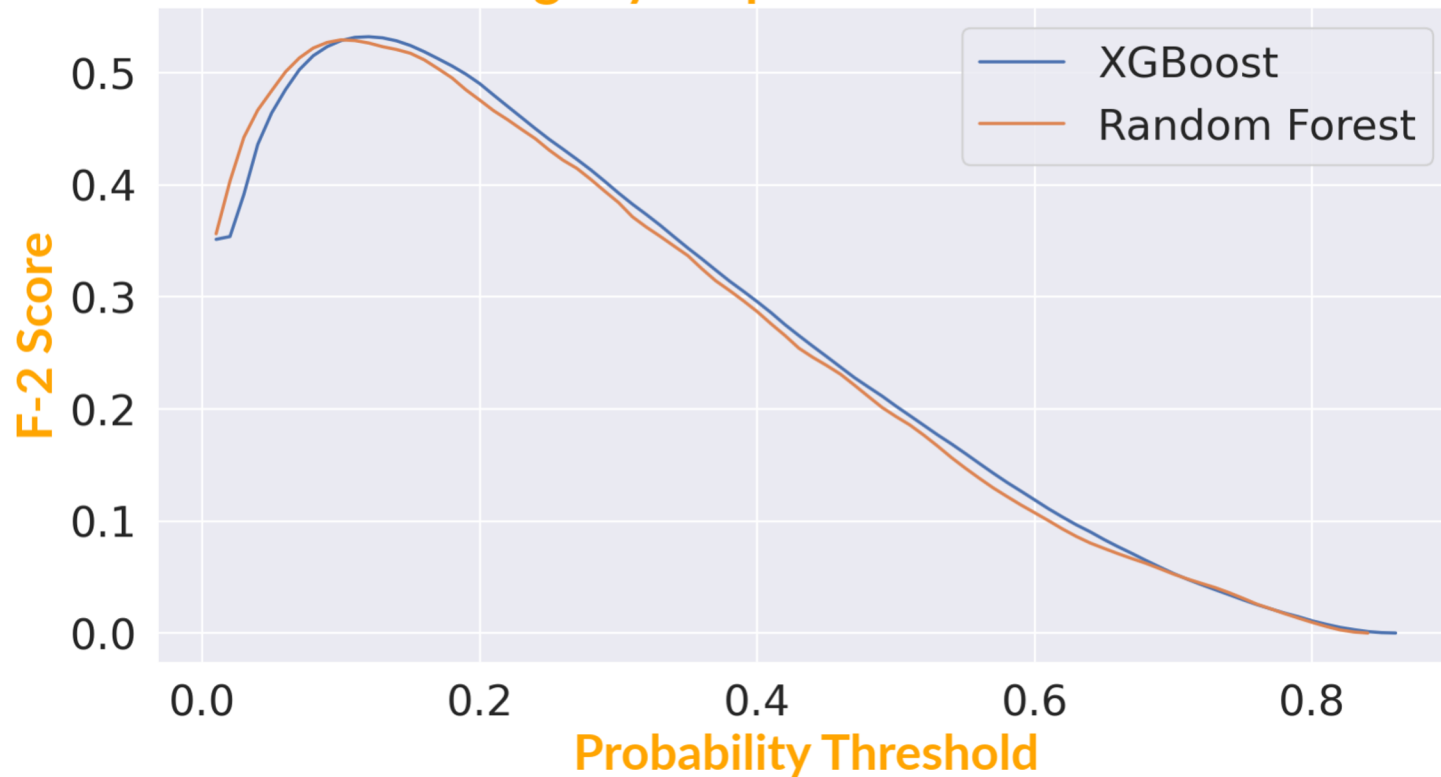- Modeling done on these inputs

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)

Model Selection & Results

XGBoost slightly outperformed RandomForest

Used grid search to define the optimal XGBoost parameters:

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

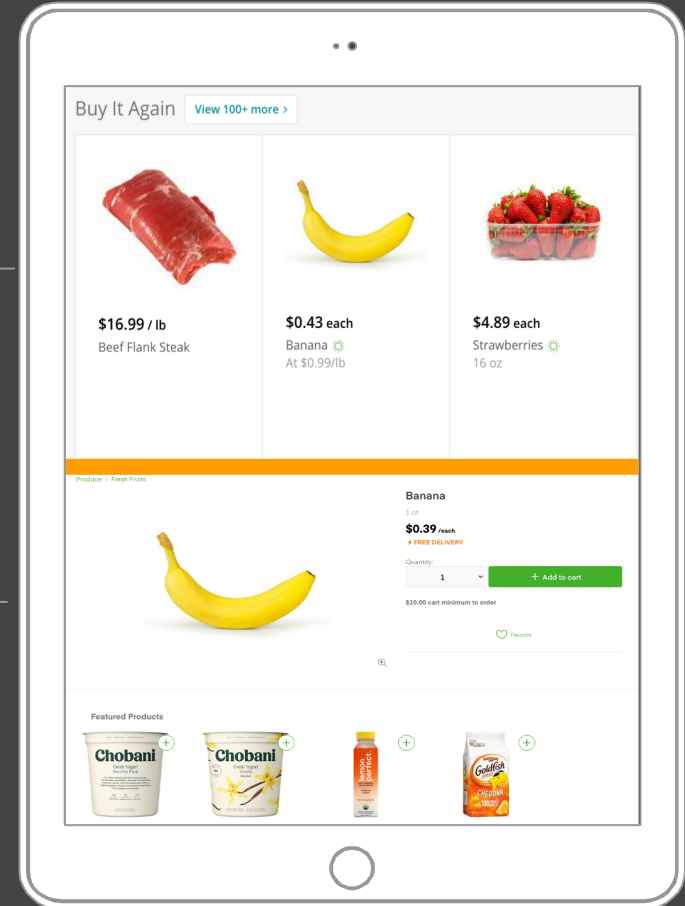|	
learning_rate	**0.009**
n_estimators	**400**
max_depth	**7**
colsample_bytree	**0.8**
min_child_weight	**9**

Scoring

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

Do you remember our use case?

Use Cases:

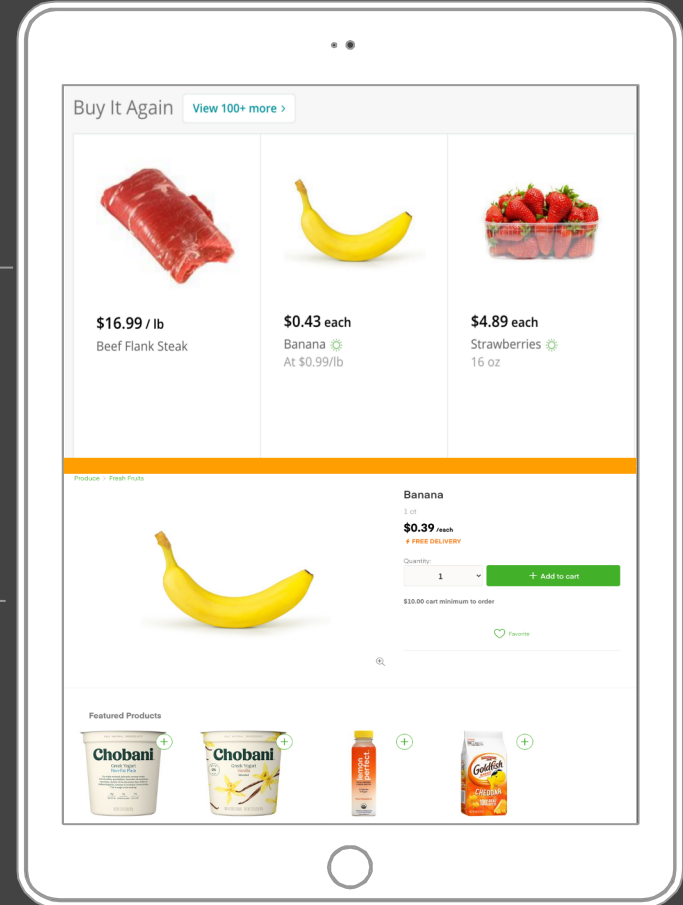1) Buy-it-again recommendations

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

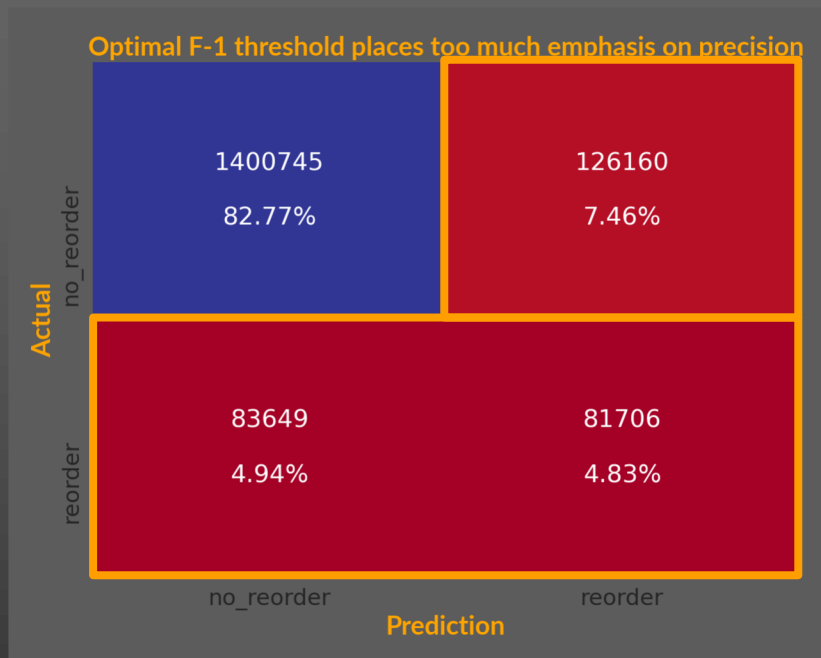1. **Increase product conversion rates**

What's Instacart's risk of incorrectly classifying an input as positive?
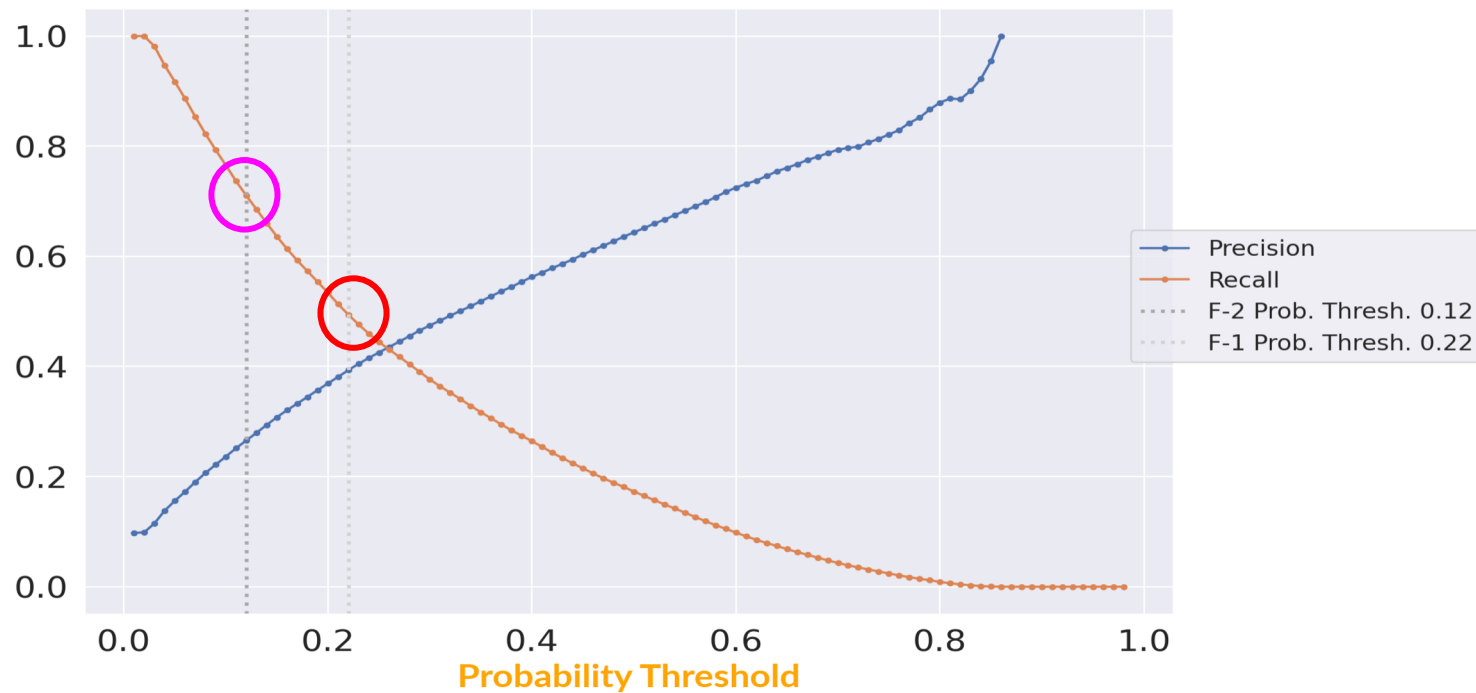
Not Much.

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

Optimal F-1 threshold places too much emphasis on precision

	no_reorder	reorder
no_reorder	1400745 / 82.77%	126160 / 7.46%
reorder	83649 / 4.94%	81706 / 4.83%

Actual (vertical axis) / **Prediction** (horizontal axis)

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

Adjusted F-2 Score

0.53

Probability Threshold

0.12

Previous threshold (0.22)

Optimized F-2...much better!

	Optimal F-1 threshold places too much emphasis on precision	
no_reorder	1400745 / 82.77%	126160 / 7.46%
reorder	83649 / 4.94%	81706 / 4.83%

Prediction / Actual

Actual \ Prediction	no_reorder	reorder
no_reorder	1201741 / 71.01%	325164 / 19.21%
reorder	47886 / 2.83%	117469 / 6.94%

Prediction
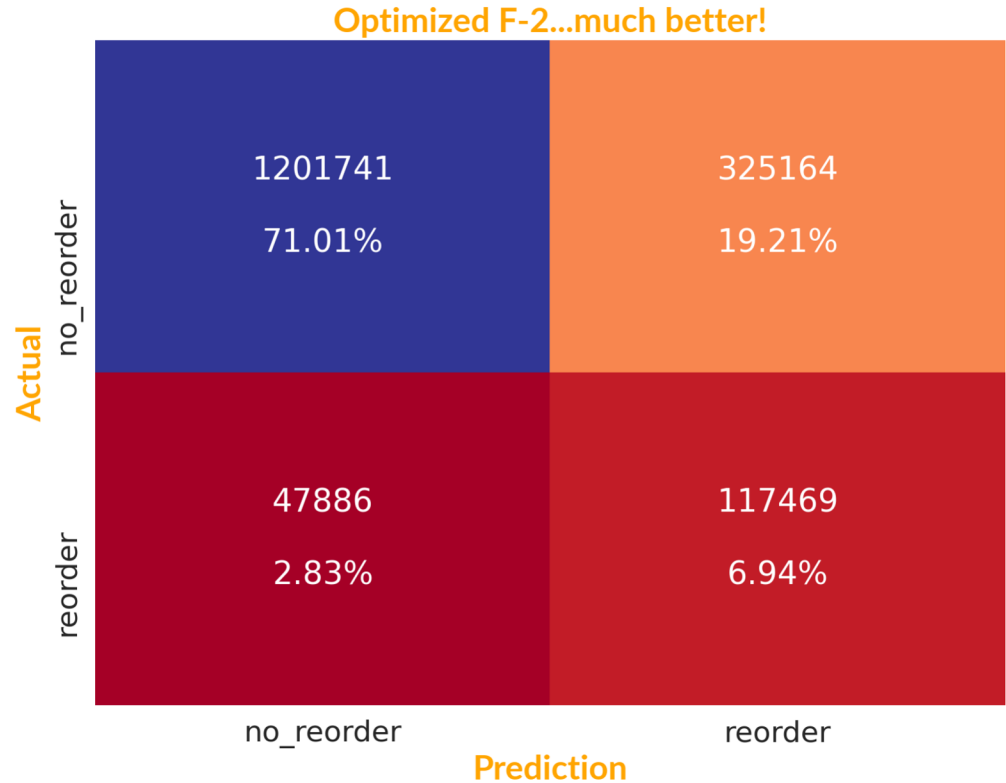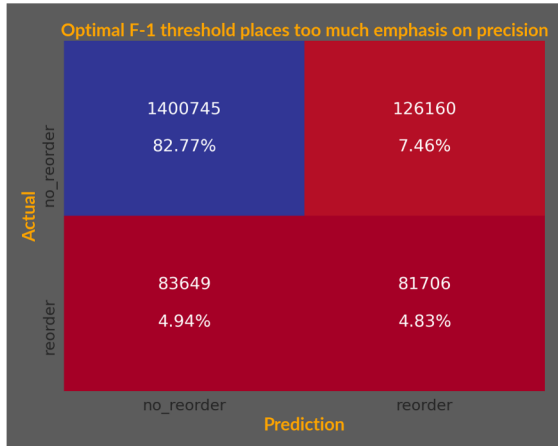
THANK YOU!

Any questions?

You can find me...

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

On GitHub: www.github.com/edubu2

wilensel@gmail.com