basket analysis

Elizabeth Naameh
lizzynaameh@ucla.edu
Data Scientist

Client:

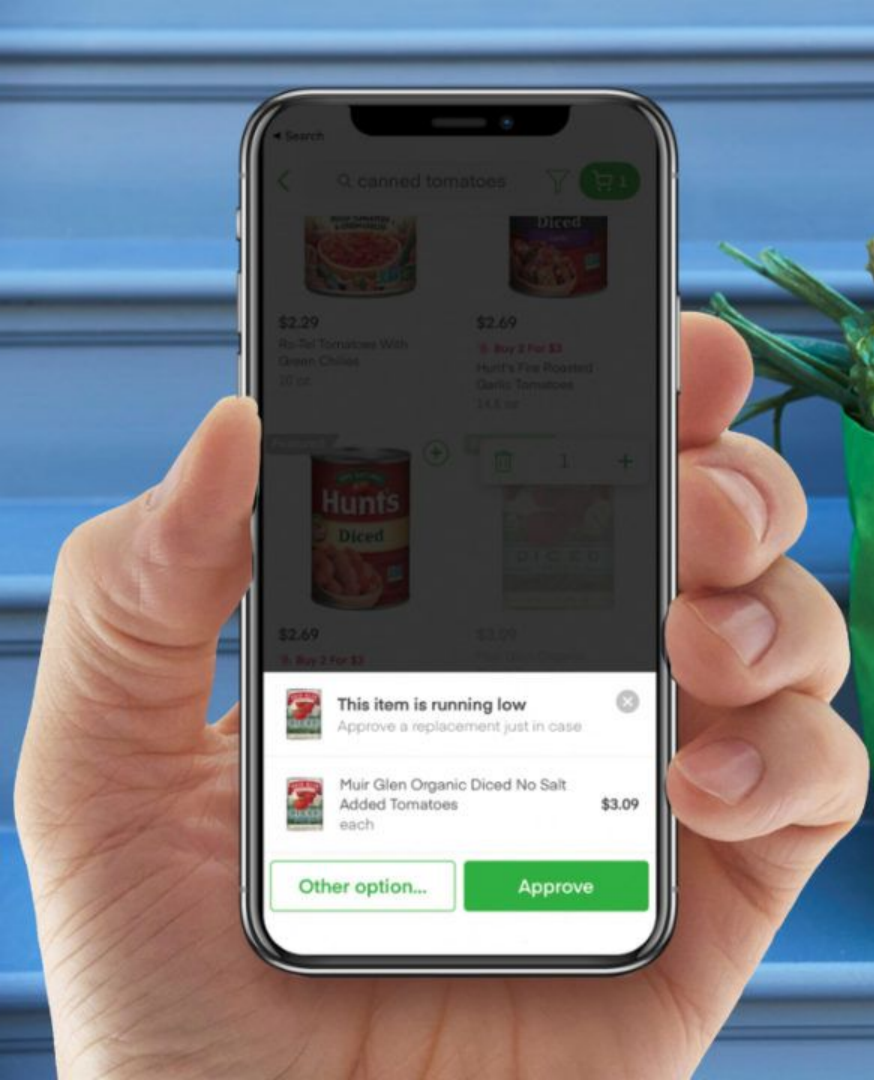
Instacart is an online grocery delivery service and app.

Goal:

Predict which products will be in a user's next order.

Product:

A classification model that predicts whether a user will reorder a product from their purchase history in their next order.



Data:

The dataset is anonymized and contains a sample of over 3 million grocery orders from more than 200,000 Instacart users.

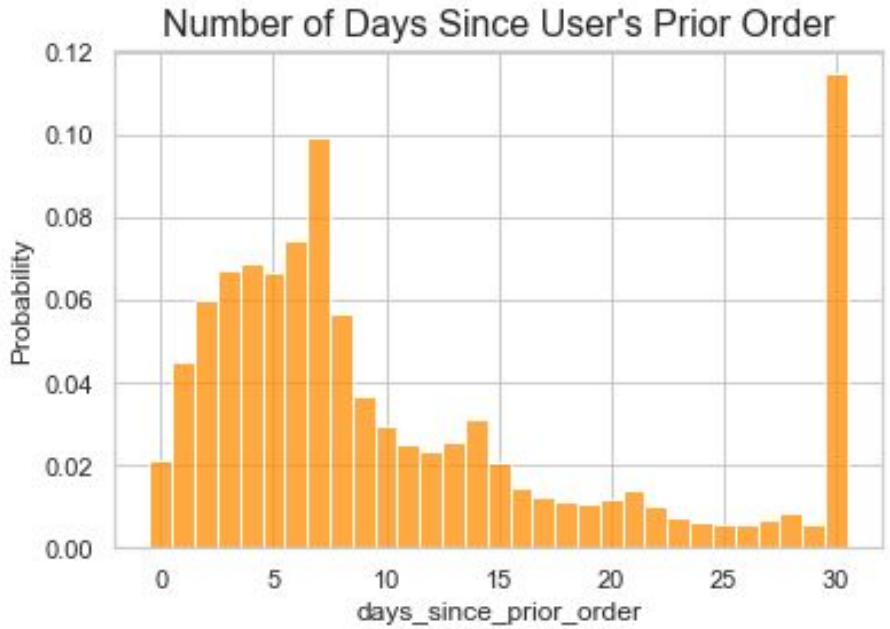
Dictionary available here.

Tools Used:

- Numpy & Pandas for data processing
- Seaborn for visualization
- Scikit-learn for machine learning

Exploratory Data Analysis

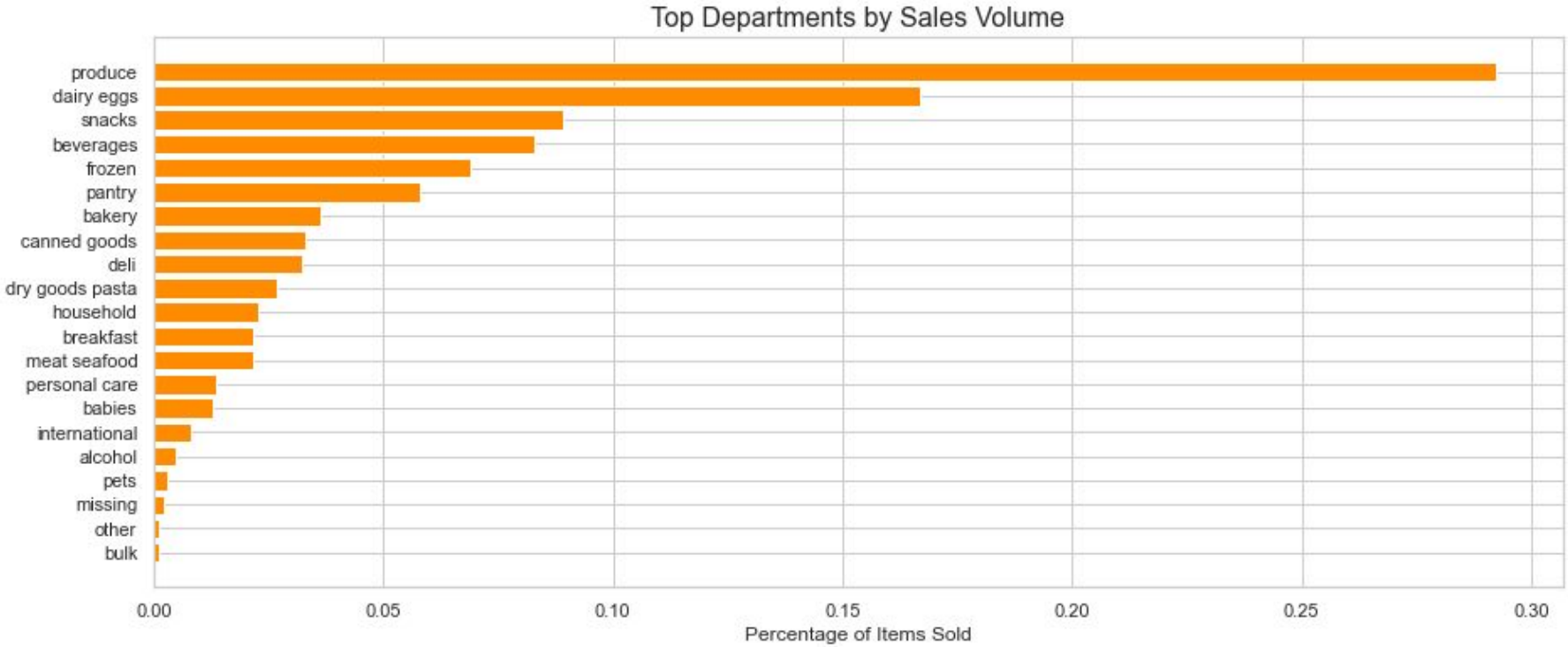
Users tend to make place orders on a weekly basis or more frequently.

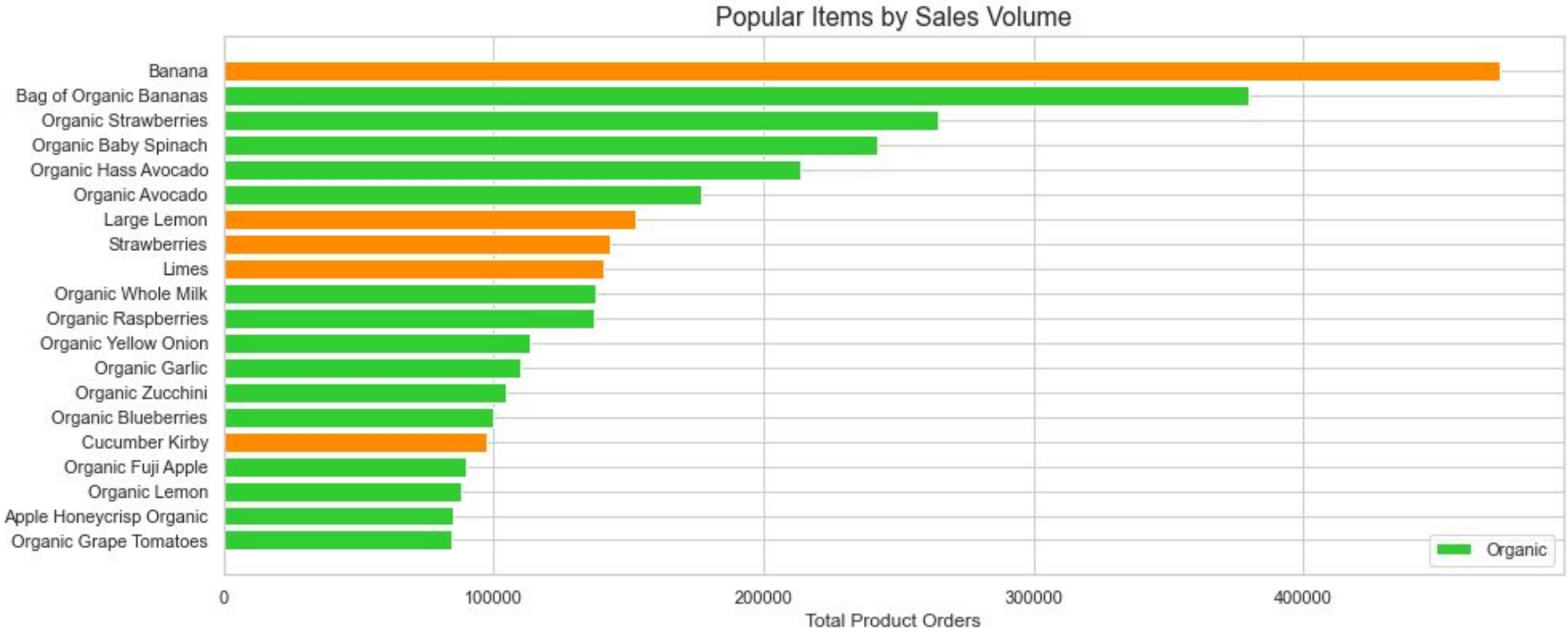


Findings

Peak shopping hours are between 10am and 4p.

Peak shopping days are Saturday and Sunday (coded 0, 1).


Top Departments by Sales Volume
produce
dairy eggs
snacks
beverages
frozen
pantry
bakery
canned goods
deli
dry goods pasta
household
breakfast
meat seafood
personal care
babies
international
alcohol
pets
missing
other
bulk
0.00
0.05
0.10
0.15
0.20
0.25
0.30
Percentage of Items Sold


Popular Items by Sales Volume
Banana
Bag of Organic Bananas
Organic Strawberries
Organic Baby Spinach
Organic Hass Avocado
Organic Avocado
Large Lemon
Strawberries
Limes
Organic Whole Milk
Organic Raspberries
Organic Yellow Onion
Organic Garlic
Organic Zucchini
Organic Blueberries
Cucumber Kirby
Organic Fuji Apple
Organic Lemon
Apple Honeycrisp Organic
Organic Grape Tomatoes
Organic
0
100000
200000
300000
400000
Total Product Orders

Goal: Use user's purchase history to predict whether a product will appear in their next order.

Optimize for the Best Shopping Experience

Precision: % of products we predict to be reordered that actually are.

Low precision means users see suggested products they are actually not interested in.

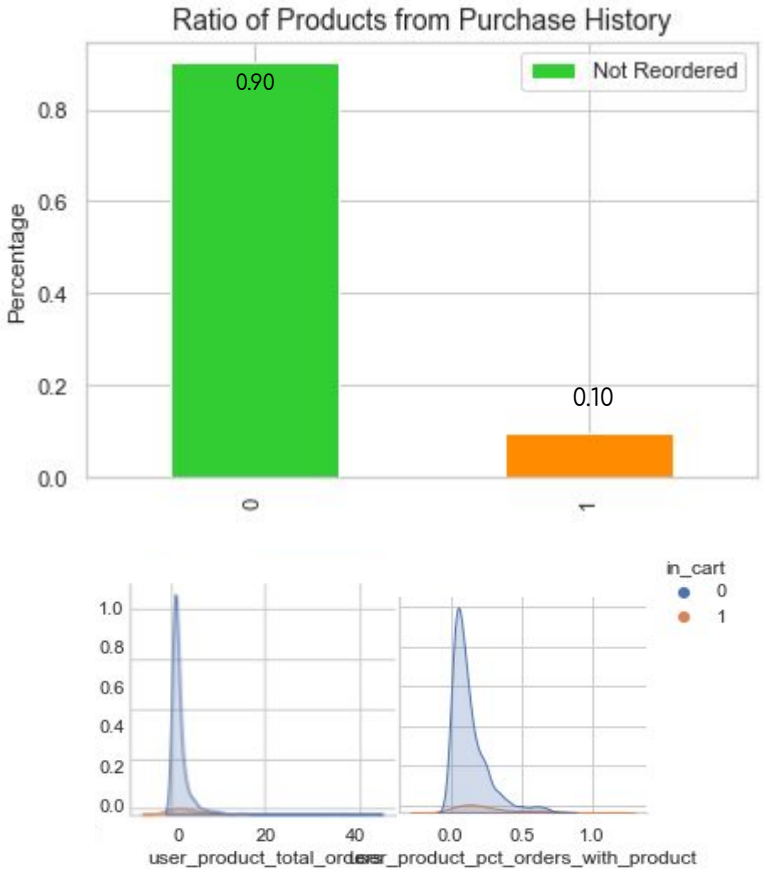
Recall: % of products that are actually reordered that we predicted.

Low recall means our model is missing out on showing products that our user is interested in.

Balance to keep suggestions relevant and promote purchases.

Baseline Model

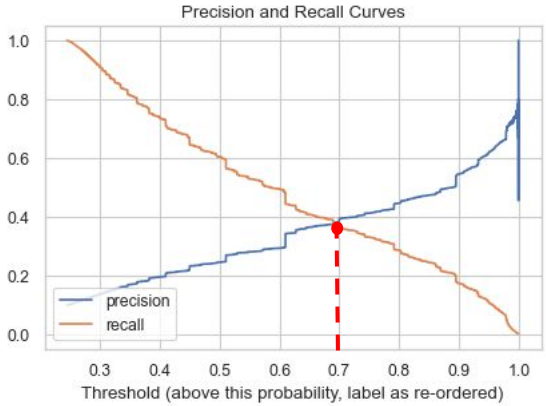
- Correct for class imbalance
- **Feature 1:** user's total number of orders for product
- **Feature 2:** percent of user's prior orders that included a given product



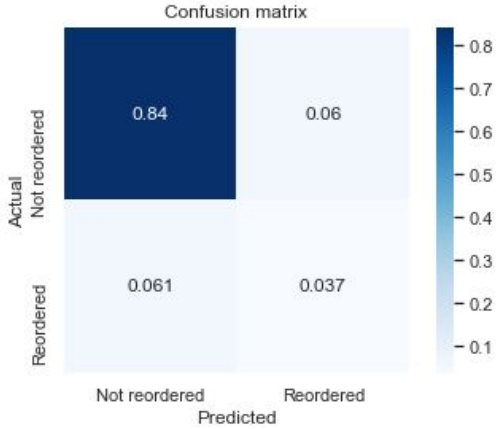
Tuning Our Baseline Model

- **Model:** Logistic Regression using a 20-20-60 train-validate-test split
- Tune for optimum threshold
- Maximize for F1 score, balance precision and recall

F1 = 0.38

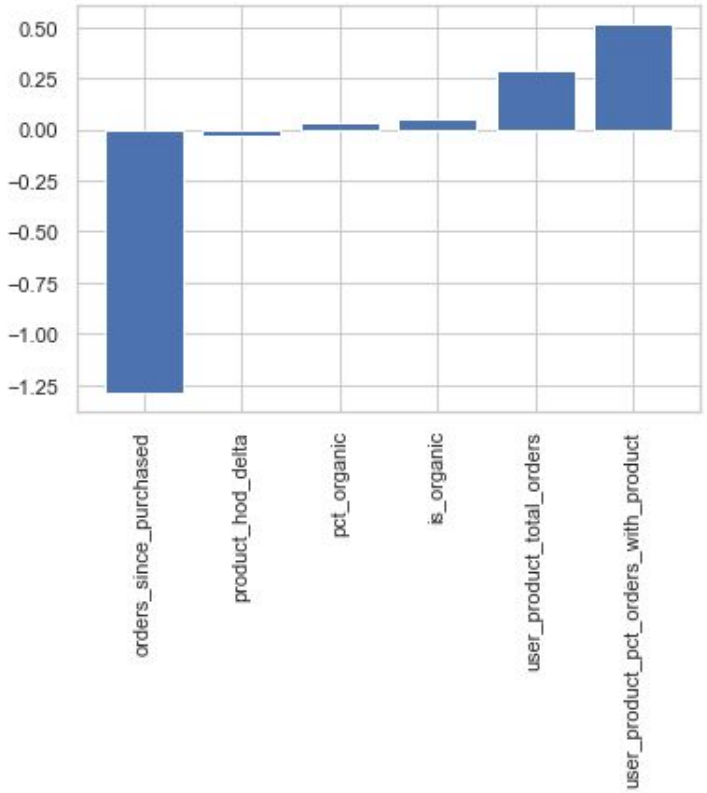



```
Threshold of 0.694:
Precision: 0.3791,   Recall: 0.376
F1 score:  0.3778946847131695
```



Feature Engineering

- Number of orders since user last purchased product
- Average difference between current order time and typical order time for product
- Percentage of user's prior products that are organic
- Product is organic
- User's total count of product orders
- Percent of user's prior orders that include product

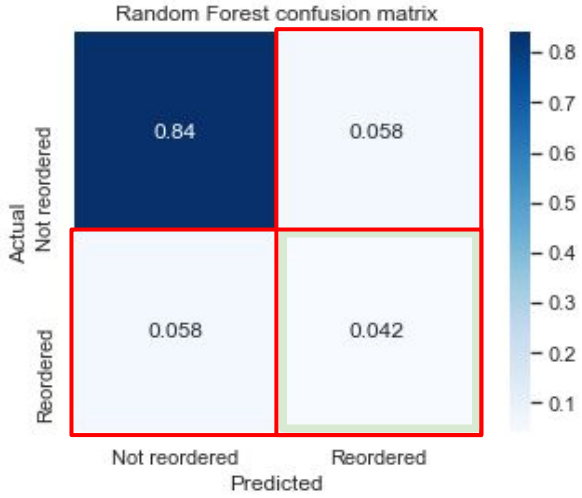


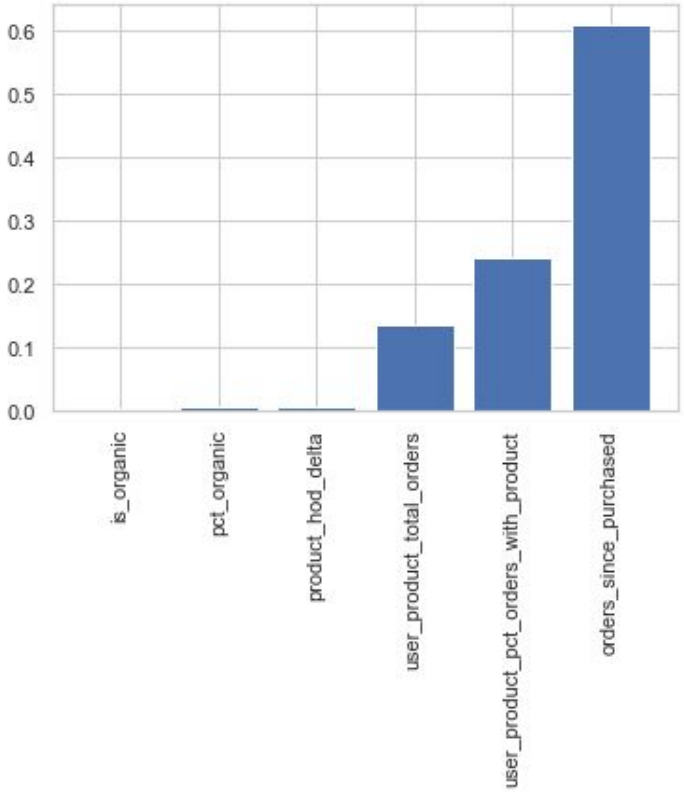
Final Model

Random Forest using an 80-20 train-holdout split.

```
Precision: 0.4179,   Recall: 0.4162
```

F1 = 0.42





Conclusions

- Our classification model can serve as the basis for recommendation system.

- Engineer more predictive features.
  - Time-series data
  - Aisle/category information

- Try more sophisticated modeling techniques.

- Reduce memory usage.

- Extend functionality: recommend recipes to users that align with their purchase history.


HIGH Brew
Mexican VANILLA
HAPPY