DTSA 5510 Project

April 17, 2024

Instacart Market Basket Analysis

Details for this dataset: https://www.kaggle.com/datasets/psparks/instacart-market-basket-analysis/data

Instacart Introduction

Select location

Select Available Store

Select Grocery Products and Pay

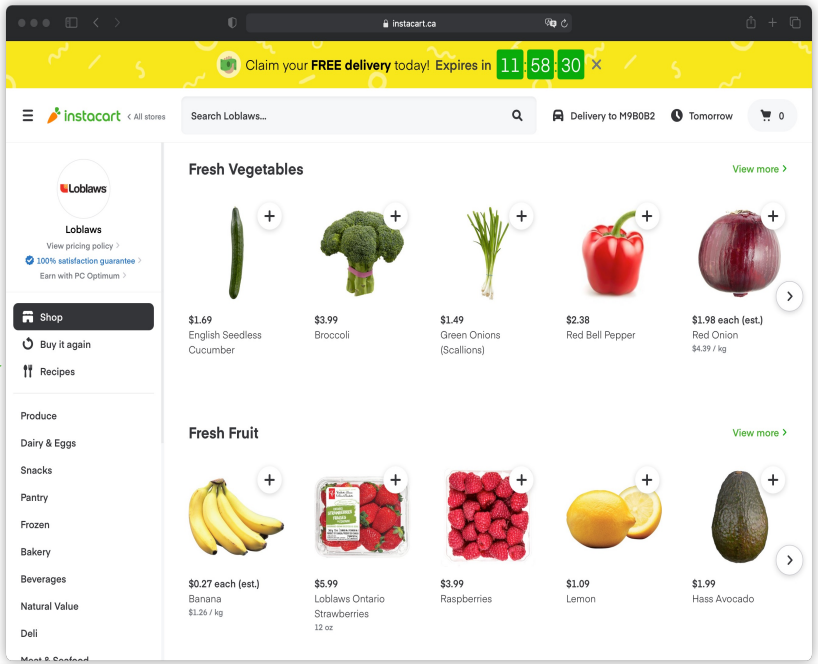
Buyer

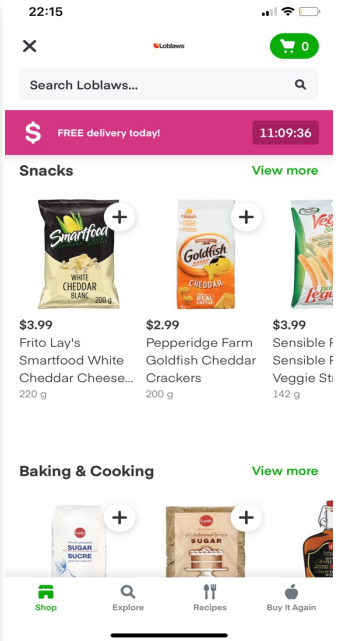
Delivery to your doorstep

Vendor

Receive order

Create Shop and List Grocery products





Transfer Amount (payment and commission)

Receive payments

Manage Website

Admin

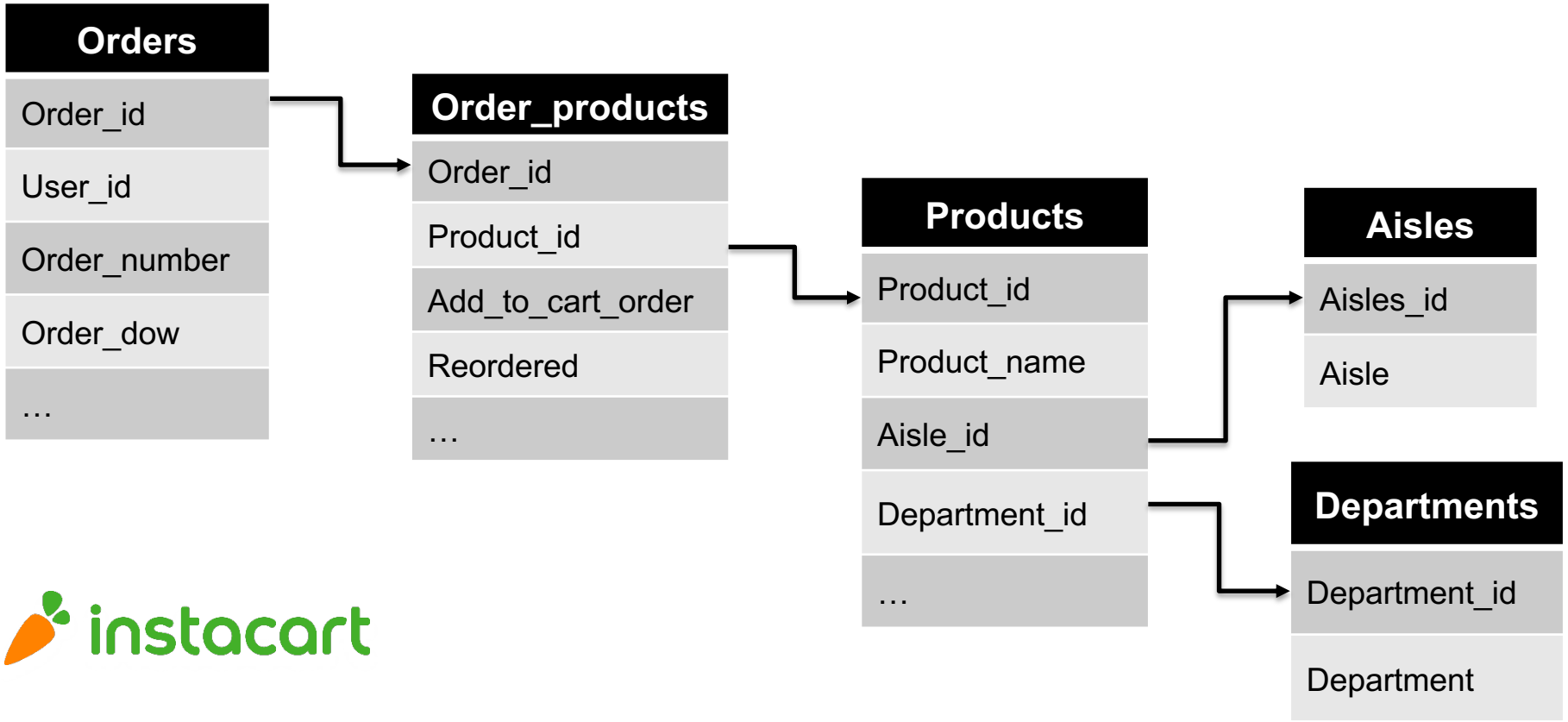
Data Overview

- This dataset has 6 different csv files.
- The orders dataset has more than 3.4 million orders.
- Over 40000 different products
- Large dataset, great for unsupervised machine learning

Problem Statement

Explore this dataset and build a recommended system for users to predict their next purchase using unsupervised machine learning algorithms.

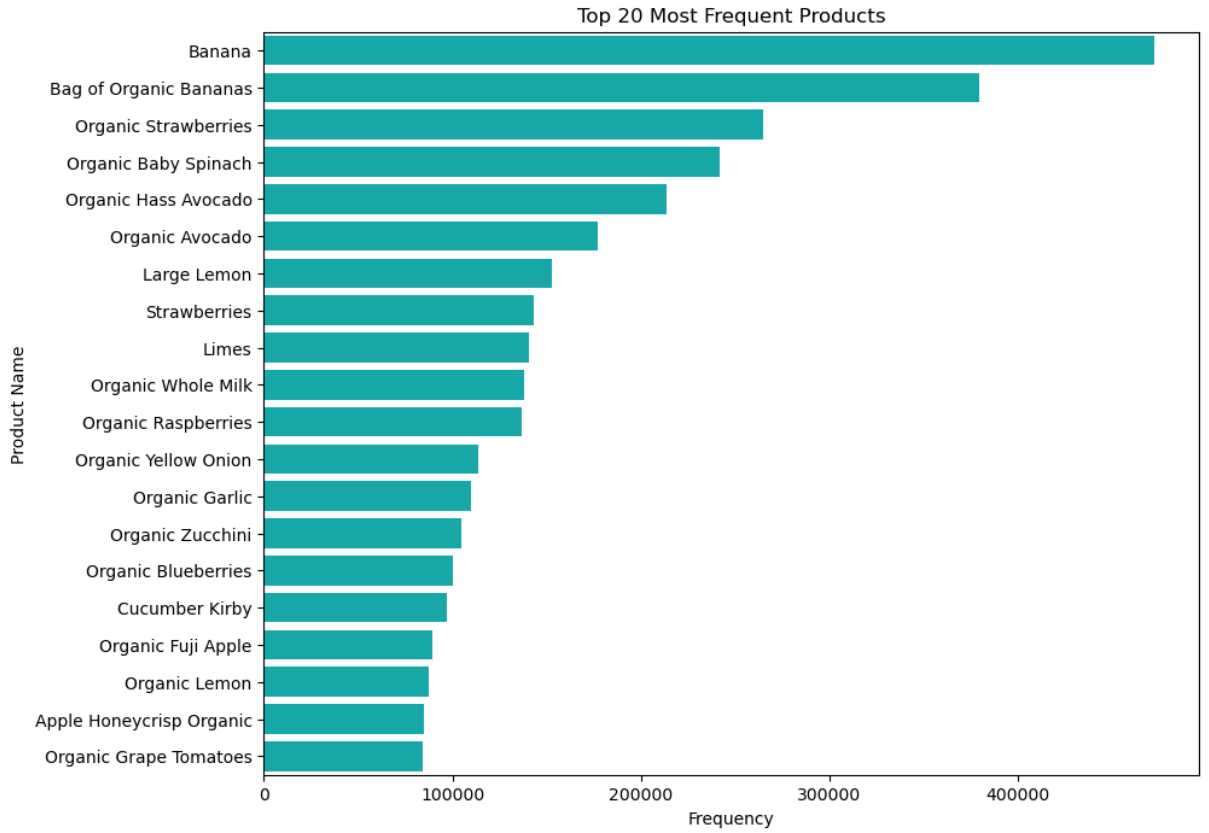
Data Description



Most Frequent Products

Bananas

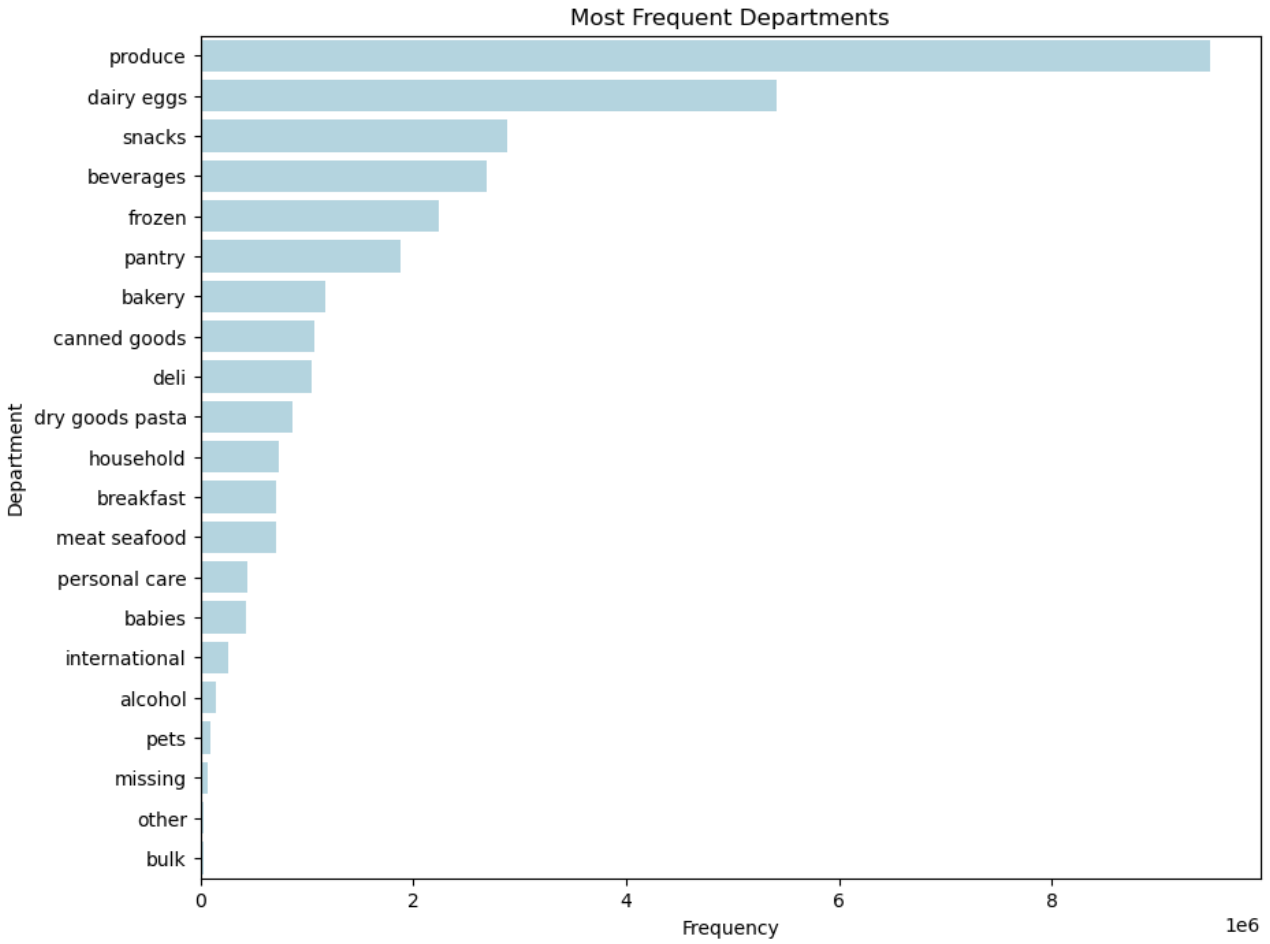
- Banana is the single most frequently ordered product
- Organic Bananas are the second most frequently ordered product
- Top 20 products are all from similar categories



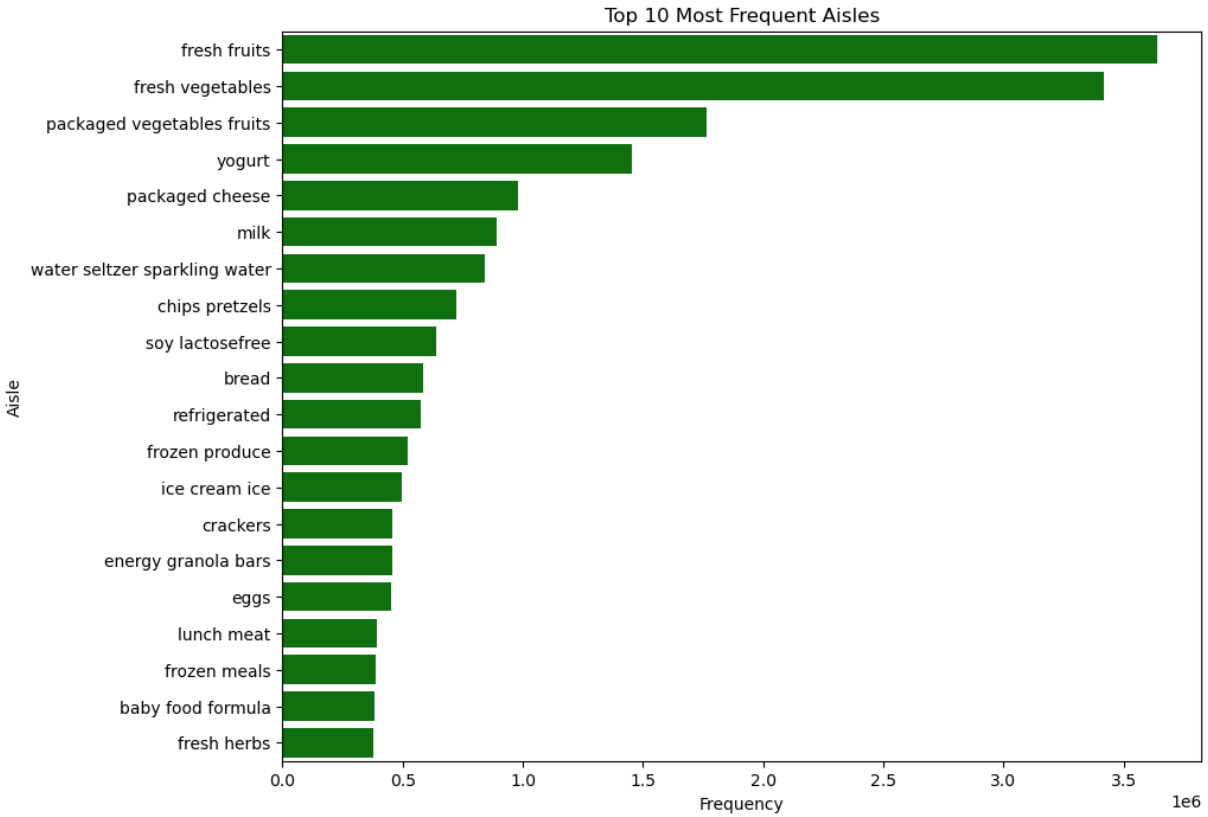
Most Frequent Department

Produce & Dairy Eggs

- The produce department is the single most frequent department
- Dairy eggs department is the second most frequent department
- These two department dominates the whole dataset



Real World Problem



Imbalanced data

- Fresh fruits and vegetables are the most frequent aisles with very large amounts values compared with others.
- It may impact the training of unsupervised machine learning algorithms.

My Solution

```
# Calculate total number of transactions
total_transactions = len(transactions)

# Calculate normalized support for each item
normalized_supports = df.sum(axis=0) / total_transactions

# Define the minimum support value as needed
min_support = 0.005

# Filter items by minimum support threshold
items_to_keep = normalized_supports[normalized_supports >= min_support].index
df_filtered = df[items_to_keep]
```

✓ 1.5s

Normalization

Filter some data from the training data frame.

Models

FP-Growth

Details: https://rasbt.github.io/mlxtend/user_guide/frequent_patterns/fpgrowth/

- Frequent item-set mining
- Build a recommender system based on this algorithm

Model Performance

```
current_cart = ['28204', '47766']
print('Current cart:', get_product_names(current_cart))
recommended_products = recommend_products(current_cart, rules)
print("Recommended products based on current cart:", get_product_names(recommended_products))
```

✓ 0.0s

```
Current cart: {'28204': 'Organic Fuji Apple', '47766': 'Organic Avocado'}
Recommended products based on current cart: {26209: 'Limes', 47626: 'Large Lemon', 21903: 'Organic Baby Spinach', 21137: 'Organic Strawberries', 24852: 'Banana', 13176: 'Bag of Organic Bananas'}
```

```
current_cart = transactions[10000]
print('Current cart:', get_product_names(current_cart))
recommended_products = recommend_products(current_cart, rules)
print("Recommended products based on current cart:", get_product_names(recommended_products))
```

✓ 0.0s

```
Current cart: {47141: 'Cola', 47877: 'Coke Zero', 20361: 'Winterfrost Sugar-Free Gum'}
Recommended products based on current cart: None
```

- Recommend new products based on the current shopping cart
- But not every transaction can be recommended by the algorithm

Model Evaluation

(Transactions that can be recommended) / (Total transactions)

FP growth can recommend 56.73% of transactions from the test dataset.

Conclusion & Final thoughts

- This dataset is very large, so it is great for unsupervised learning
- The imbalance nature of this dataset makes the recommended system less efficient
- The FP-growth algorithm alone is not enough for this task
- Matrix factorization and collaborating filtering may be used to build a more robust recommended system.

Thank You!