FinalProjectPart4

August 11, 2024

1 BIA 678 - Final Project

Part 4 Product Recommendations Assignment

For this assignment, you will analyze Instacart order data to uncover product associations that can inform recommendation systems to suggest complementary products to shoppers.

Tasks: 1. Prepare the Instacart order data for association rule mining using an algorithm like Apriori. This involves aggregating orders per user, transforming orders into product sets, and filtering low occurrence products.

2. Apply association rule mining to identify product relationships with minimum support,minimum confidence and lift. Analyze the rules to find the most relevant associations for recommendations.

3. Propose a recommendation system interface that suggests additional products to shoppers during checkout based on your association rule results. Prioritize rules with higher lift values. Visualize what this recommender experience could look like.

Additionally, select one or more of the following tasks:

4a. Smart Basket Recommendations: Design a smart basket system that tracks products shoppers add in real-time and provides live suggestions based on the association rules to prompt additional purchases.

4b. Content-Based Recommendations: Build a content-based recommender that uses product descriptions and properties to match people to similar products. What product attributes are most meaningful to use for similarity?

4c. Collaborative Filtering Recommendations: Implement a collaborative filtering system that uses historical purchase data to identify shoppers with similar buying patterns and generates recommendations based on what similar shoppers purchased.

Key Deliverables: - A 3-5 page project report documenting your analysis, association rules, recommendation system proposal, and selected extended task. - Code and output for the association rule mining - Mockups or diagrams for the proposed recommendation interfaces

The goal is to demonstrate you can analyze basket data, discover product relationships, and design compelling product recommendation experiences. What associations exist in grocery shopping data and how can retailers leverage recommendations to encourage larger purchases?

[1]:
```
# Import packages
import numpy as np
import pandas as pd

from mlxtend.frequent_patterns import apriori, association_rules
```

```
[2]: df_aisles = pd.read_csv('aisles.csv')
     df_departments = pd.read_csv('departments.csv')
     df_orders = pd.read_csv('orders.csv')
     df_products = pd.read_csv('products.csv')
     df_order_products = pd.read_csv('order_products__prior.csv')
     df_order_products = df_order_products [:50000]
```

```
[3]: df1 = pd.merge(df_order_products, df_orders, on= 'order_id')
     df1.head()
```

```
[3]:    order_id  product_id  add_to_cart_order  reordered  user_id eval_set  \
     0         2       33120                  1          1   202279    prior
     1         2       28985                  2          1   202279    prior
     2         2        9327                  3          0   202279    prior
     3         2       45918                  4          1   202279    prior
     4         2       30035                  5          0   202279    prior

        order_number  order_dow  order_hour_of_day  days_since_prior_order
     0             3          5                  9                     8.0
     1             3          5                  9                     8.0
     2             3          5                  9                     8.0
     3             3          5                  9                     8.0
     4             3          5                  9                     8.0
```

```
[4]: prod_aisles = pd.merge(df_products, df_aisles, on = 'aisle_id')
     df2 = pd.merge(prod_aisles, df_departments, on = 'department_id')
     df2.head
```

```
[4]: <bound method NDFrame.head of        product_id
     product_name  \
     0               1                          Chocolate Sandwich Cookies
     1              78                   Nutter Butter Cookie Bites Go-Pak
     2             102                               Danish Butter Cookies
     3             172     Gluten Free All Natural Chocolate Chip Cookies
     4             285                            Mini Nilla Wafers Munch Pack
     …            …                                                      …
     49683       22827                              Organic Black Mission Figs
     49684       28655                              Crystallized Ginger Chunks
     49685       30365                                         Vegetable Chips
     49686       38007                          Naturally Sweet Plantain Chips
     49687       48778  Fit Super A Juice, Cold Pressed, Carrot/Apple/…

            aisle_id  department_id                         aisle department
```

```
0            61            19                cookies cakes     snacks
1            61            19                cookies cakes     snacks
2            61            19                cookies cakes     snacks
3            61            19                cookies cakes     snacks
4            61            19                cookies cakes     snacks
…          …             …                             …         …
49683        18            10   bulk dried fruits vegetables       bulk
49684        18            10   bulk dried fruits vegetables       bulk
49685        18            10   bulk dried fruits vegetables       bulk
49686        18            10   bulk dried fruits vegetables       bulk
49687        18            10   bulk dried fruits vegetables       bulk

[49688 rows x 6 columns]>
```

[5]:
```
combined_df = pd.merge(df1, df2, on = 'product_id').reset_index(drop=True)
combined_df.head()
```

[5]:
```
   order_id  product_id  add_to_cart_order  reordered  user_id eval_set  \
0         2       33120                  1          1   202279    prior
1        26       33120                  5          0   153404    prior
2       120       33120                 13          0    23750    prior
3       327       33120                  5          1    58707    prior
4       390       33120                 28          1   166654    prior

   order_number  order_dow  order_hour_of_day  days_since_prior_order  \
0             3          5                  9                     8.0
1             2          0                 16                     7.0
2            11          6                  8                    10.0
3            21          6                  9                     8.0
4            48          0                 12                     9.0

         product_name  aisle_id  department_id aisle  department
0  Organic Egg Whites        86             16  eggs  dairy eggs
1  Organic Egg Whites        86             16  eggs  dairy eggs
2  Organic Egg Whites        86             16  eggs  dairy eggs
3  Organic Egg Whites        86             16  eggs  dairy eggs
4  Organic Egg Whites        86             16  eggs  dairy eggs
```

[6]:
```
df2 = combined_df.sample(n=1000)[['user_id','product_name']]
basket = pd.crosstab(df2['user_id'],df2['product_name']).astype('bool').
  ↪astype('int')
```

[7]:
```
#Checking and removing index.
basket=basket.reset_index(drop=True)
basket.index
```

[7]:
```
RangeIndex(start=0, stop=869, step=1)
```

[20]:
```
#Calling apriori algorithm on dummified data - basket.
frequent_itemsets=apriori(basket, min_support=0.00002, use_colnames=True).
 ↪sort_values('support', ascending=False)

frequent_itemsets.head(10)
```

```
/Library/Frameworks/Python.framework/Versions/3.11/lib/python3.11/site-
packages/mlxtend/frequent_patterns/fpcommon.py:109: DeprecationWarning:
DataFrames with non-bool types result in worse computationalperformance and
their support might be discontinued in the future.Please use a DataFrame with
bool type
  warnings.warn(
```

[20]:
```
      support                   itemsets
48   0.021864                   (Banana)
45   0.010357  (Bag of Organic Bananas)
422  0.008055    (Organic Hass Avocado)
470  0.006904     (Organic Raspberries)
39   0.005754                (Asparagus)
529  0.005754        (Organic Zucchini)
623  0.005754           (Russet Potato)
476  0.004603        (Organic Red Onion)
502  0.004603    (Organic Strawberries)
494  0.004603       (Organic Sour Cream)
```

[21]:
```
rules = association_rules(frequent_itemsets, metric="confidence",␣
 ↪min_threshold=0.5)
rules.head(20)
```

[21]:
```
                                         antecedents  \
0                (Large Brown Eggs, Honey Wheat Bread)
1                   (Large Brown Eggs, Hamburger Buns)
2                  (Honey Wheat Bread, Hamburger Buns)
3                                   (Large Brown Eggs)
4                                  (Honey Wheat Bread)
5                                     (Hamburger Buns)
6    (Organic Light Agave Nectar, Sharp Cheddar Che…
7    (Organic Light Agave Nectar, Newman O's Creme …
8    (Newman O's Creme Filled Mint Chocolate Cookie…
9                         (Organic Light Agave Nectar)
10                              (Sharp Cheddar Cheese)
11    (Newman O's Creme Filled Mint Chocolate Cookies)
12      (Organic Reduced Fat 2% Milk, Original Hummus)
13   (Organic Reduced Fat 2% Milk, Organic Madagasc…
14   (Original Hummus, Organic Madagascar Vanilla W…
15                       (Organic Reduced Fat 2% Milk)
16                                   (Original Hummus)
17   (Organic Madagascar Vanilla Wafer Ice Cream Sa…
18  (Large Brown Eggs, Fat Free Sour Cream, Honey …
19  (Large Brown Eggs, Fat Free Sour Cream, Hambur…

                                      consequents  antecedent support  \
0                                (Hamburger Buns)            0.001151
1                              (Honey Wheat Bread)            0.001151
2                               (Large Brown Eggs)            0.001151
3                (Honey Wheat Bread, Hamburger Buns)            0.002301
4                 (Large Brown Eggs, Hamburger Buns)            0.001151
5              (Large Brown Eggs, Honey Wheat Bread)            0.001151
6   (Newman O's Creme Filled Mint Chocolate Cookies)            0.001151
7                            (Sharp Cheddar Cheese)            0.001151
8                      (Organic Light Agave Nectar)            0.001151
9   (Newman O's Creme Filled Mint Chocolate Cookie…          0.001151
10  (Organic Light Agave Nectar, Newman O's Creme …          0.002301
11  (Organic Light Agave Nectar, Sharp Cheddar Che…          0.001151
12  (Organic Madagascar Vanilla Wafer Ice Cream Sa…          0.001151
13                               (Original Hummus)            0.001151
14                   (Organic Reduced Fat 2% Milk)            0.001151
15  (Original Hummus, Organic Madagascar Vanilla W…          0.001151
16  (Organic Reduced Fat 2% Milk, Organic Madagasc…          0.002301
17    (Organic Reduced Fat 2% Milk, Original Hummus)            0.001151
18                                (Hamburger Buns)            0.001151
19                             (Honey Wheat Bread)            0.001151

    consequent support   support  confidence   lift  leverage  conviction  \
0             0.001151  0.001151         1.0  869.0  0.001149         inf
1             0.001151  0.001151         1.0  869.0  0.001149         inf
2             0.002301  0.001151         1.0  434.5  0.001148         inf
3             0.001151  0.001151         0.5  434.5  0.001148    1.997699
4             0.001151  0.001151         1.0  869.0  0.001149         inf
5             0.001151  0.001151         1.0  869.0  0.001149         inf
6             0.001151  0.001151         1.0  869.0  0.001149         inf
7             0.002301  0.001151         1.0  434.5  0.001148         inf
8             0.001151  0.001151         1.0  869.0  0.001149         inf
9             0.001151  0.001151         1.0  869.0  0.001149         inf
10            0.001151  0.001151         0.5  434.5  0.001148    1.997699
11            0.001151  0.001151         1.0  869.0  0.001149         inf
12            0.001151  0.001151         1.0  869.0  0.001149         inf
13            0.002301  0.001151         1.0  434.5  0.001148         inf
14            0.001151  0.001151         1.0  869.0  0.001149         inf
15            0.001151  0.001151         1.0  869.0  0.001149         inf
16            0.001151  0.001151         0.5  434.5  0.001148    1.997699
17            0.001151  0.001151         1.0  869.0  0.001149         inf
18            0.001151  0.001151         1.0  869.0  0.001149         inf
19            0.001151  0.001151         1.0  869.0  0.001149         inf
```

```
    zhangs_metric
0        1.000000
1        1.000000
2        0.998848
3        1.000000
4        1.000000
5        1.000000
6        1.000000
7        0.998848
8        1.000000
9        1.000000
10       1.000000
11       1.000000
12       1.000000
13       0.998848
14       1.000000
15       1.000000
16       1.000000
17       1.000000
18       1.000000
19       1.000000
```

[22]:
```
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
rules.head(10)
```

[22]:
```
                                        antecedents  \
0               (Large Brown Eggs, Honey Wheat Bread)
1                  (Large Brown Eggs, Hamburger Buns)
2                 (Honey Wheat Bread, Hamburger Buns)
3                                  (Large Brown Eggs)
4                                 (Honey Wheat Bread)
5                                    (Hamburger Buns)
6   (Organic Light Agave Nectar, Sharp Cheddar Che…
7   (Organic Light Agave Nectar, Newman O's Creme …
8   (Newman O's Creme Filled Mint Chocolate Cookie…
9                        (Organic Light Agave Nectar)

                                        consequents  antecedent support  \
0                                    (Hamburger Buns)            0.001151
1                                 (Honey Wheat Bread)            0.001151
2                                  (Large Brown Eggs)            0.001151
3                 (Honey Wheat Bread, Hamburger Buns)            0.002301
4                  (Large Brown Eggs, Hamburger Buns)            0.001151
5               (Large Brown Eggs, Honey Wheat Bread)            0.001151
6    (Newman O's Creme Filled Mint Chocolate Cookies)            0.001151
7                              (Sharp Cheddar Cheese)            0.001151
8                        (Organic Light Agave Nectar)            0.001151
```

```
9  (Newman O's Creme Filled Mint Chocolate Cookie…           0.001151

   consequent support   support  confidence   lift  leverage  conviction  \
0            0.001151  0.001151         1.0  869.0  0.001149         inf
1            0.001151  0.001151         1.0  869.0  0.001149         inf
2            0.002301  0.001151         1.0  434.5  0.001148         inf
3            0.001151  0.001151         0.5  434.5  0.001148    1.997699
4            0.001151  0.001151         1.0  869.0  0.001149         inf
5            0.001151  0.001151         1.0  869.0  0.001149         inf
6            0.001151  0.001151         1.0  869.0  0.001149         inf
7            0.002301  0.001151         1.0  434.5  0.001148         inf
8            0.001151  0.001151         1.0  869.0  0.001149         inf
9            0.001151  0.001151         1.0  869.0  0.001149         inf

   zhangs_metric
0       1.000000
1       1.000000
2       0.998848
3       1.000000
4       1.000000
5       1.000000
6       1.000000
7       0.998848
8       1.000000
9       1.000000
```

[11]:
```
rules[(rules['lift'] >= 5) & (rules['confidence']>= 0.5)]
```

[11]:
```
                                           antecedents  \
0                (Large Brown Eggs, Honey Wheat Bread)
1                   (Large Brown Eggs, Hamburger Buns)
2                  (Honey Wheat Bread, Hamburger Buns)
3                                   (Large Brown Eggs)
4                                  (Honey Wheat Bread)
..                                                 …
437  (Spanish Pitted Manzanilla Cocktail Olives, Or…
438                                        (Red Mango)
439                             (Organic Diced Tomatoes)
440        (Spanish Pitted Manzanilla Cocktail Olives)
441                                (Organic Skim Milk)

                                           consequents  antecedent support  \
0                                     (Hamburger Buns)            0.001151
1                                  (Honey Wheat Bread)            0.001151
2                                   (Large Brown Eggs)            0.001151
3                  (Honey Wheat Bread, Hamburger Buns)            0.002301
4                   (Large Brown Eggs, Hamburger Buns)            0.001151
```

```
..                                                   …                …
437                (Red Mango, Organic Diced Tomatoes)          0.001151
438  (Organic Diced Tomatoes, Spanish Pitted Manzan…          0.001151
439  (Red Mango, Spanish Pitted Manzanilla Cocktail…          0.001151
440  (Red Mango, Organic Skim Milk, Organic Diced T…          0.001151
441  (Red Mango, Spanish Pitted Manzanilla Cocktail…          0.001151

     consequent support   support  confidence   lift  leverage  conviction  \
0              0.001151  0.001151         1.0  869.0  0.001149         inf
1              0.001151  0.001151         1.0  869.0  0.001149         inf
2              0.002301  0.001151         1.0  434.5  0.001148         inf
3              0.001151  0.001151         0.5  434.5  0.001148    1.997699
4              0.001151  0.001151         1.0  869.0  0.001149         inf
..                  …         …         …    …       …          …
437            0.001151  0.001151         1.0  869.0  0.001149         inf
438            0.001151  0.001151         1.0  869.0  0.001149         inf
439            0.001151  0.001151         1.0  869.0  0.001149         inf
440            0.001151  0.001151         1.0  869.0  0.001149         inf
441            0.001151  0.001151         1.0  869.0  0.001149         inf

     zhangs_metric
0         1.000000
1         1.000000
2         0.998848
3         1.000000
4         1.000000
..             …
437       1.000000
438       1.000000
439       1.000000
440       1.000000
441       1.000000

[393 rows x 10 columns]
```

[12]:
```python
from sklearn.metrics.pairwise import cosine_similarity

combined_df.head()
```

[12]:
```
   order_id  product_id  add_to_cart_order  reordered  user_id eval_set  \
0         2       33120                  1          1   202279    prior
1        26       33120                  5          0   153404    prior
2       120       33120                 13          0    23750    prior
3       327       33120                  5          1    58707    prior
4       390       33120                 28          1   166654    prior

   order_number  order_dow  order_hour_of_day  days_since_prior_order  \
0              3          5                  9                      8.0
1              2          0                 16                      7.0
2             11          6                  8                     10.0
3             21          6                  9                      8.0
4             48          0                 12                      9.0

         product_name  aisle_id  department_id aisle  department
0  Organic Egg Whites        86             16  eggs  dairy eggs
1  Organic Egg Whites        86             16  eggs  dairy eggs
2  Organic Egg Whites        86             16  eggs  dairy eggs
3  Organic Egg Whites        86             16  eggs  dairy eggs
4  Organic Egg Whites        86             16  eggs  dairy eggs
```

[13]:
```
processed_df = combined_df.drop(columns=['eval_set', 'product_name', 'aisle',␣
 ↪'department'])
processed_df.head()
```

[13]:
```
   order_id  product_id  add_to_cart_order  reordered  user_id  order_number  \
0         2       33120                  1          1   202279             3
1        26       33120                  5          0   153404             2
2       120       33120                 13          0    23750            11
3       327       33120                  5          1    58707            21
4       390       33120                 28          1   166654            48

   order_dow  order_hour_of_day  days_since_prior_order  aisle_id  \
0          5                  9                     8.0        86
1          0                 16                     7.0        86
2          6                  8                    10.0        86
3          6                  9                     8.0        86
4          0                 12                     9.0        86

   department_id
0             16
1             16
2             16
3             16
4             16
```

[14]:
```
processed_df.shape
```

[14]:
```
(50000, 11)
```

[15]:
```
processed_df['days_since_prior_order'].
 ↪fillna(processed_df['days_since_prior_order'].mean(), inplace=True)
print(processed_df.isna().sum())
```

```
order_id                  0
product_id                0
```

```
add_to_cart_order         0
reordered                 0
user_id                   0
order_number              0
order_dow                 0
order_hour_of_day         0
days_since_prior_order    0
aisle_id                  0
department_id             0
dtype: int64
```

[16]:
```
user_item_matrix = processed_df.pivot_table(index='user_id',␣
 ↪columns='product_id', values='add_to_cart_order', fill_value=0)
```

[17]:
```
similarity_matrix = cosine_similarity(user_item_matrix)
similarity_matrix_df = pd.DataFrame(similarity_matrix, index=user_item_matrix.
 ↪index, columns=user_item_matrix.index)
```

[18]:
```
def recommend_products(user_id, similarity_matrix_df, user_item_matrix, top_n):
    similar_users = similarity_matrix_df[user_id].sort_values(ascending=False).
 ↪index[1:]
    user_purchases = set(user_item_matrix.columns[user_item_matrix.loc[user_id]␣
 ↪> 0])

    recommendations = []
    for similar_user in similar_users:
        similar_user_purchases = set(user_item_matrix.columns[user_item_matrix.
 ↪loc[similar_user] > 0])
        recommended_products = similar_user_purchases - user_purchases
        recommendations.extend(list(recommended_products))
        if len(recommendations) >= top_n:
            break

    return recommendations[:top_n]
```

[19]:
```
userId = 202279
top_n = 10

recommended_products = recommend_products(userId, similarity_matrix_df,␣
 ↪user_item_matrix, top_n)
product_dict = pd.Series(df_products.product_name.values, index=df_products.
 ↪product_id).to_dict()
product_names = list(map(product_dict.get, recommended_products))

print(f"Top {top_n} Product Recommendations for the user {userId}:")
print('-------------')
for i, name in enumerate(product_names):
    print(i, name)
```

```
Top 10 Product Recommendations for the user 202279:
-------------
0 Pure Sparkling Water
1 Half & Half
2 Freeze Dried Strawberry Slices
3 Double Chocolate Cake
4 Tiny Fruits Blueberry Apple
5 Organic Freeze Dried Strawberries
6 Organic Freeze-Dried Mango
7 Berry Medley
8 Organic Garlic
9 Organic Small Bunch Celery
```

[ ]:

```

```