

Machine Learning-Powered Prescriptive Insights for Customer Purchase Behavior

Market Basket Analysis for Customer Purchase Insights

What products are frequently bought together?

1. Overview

In today's highly competitive retail environment, understanding customer purchasing behavior is crucial for maximizing sales and enhancing customer satisfaction. This analysis implements Market Basket Analysis (MBA) using transactional data to uncover product combinations that are frequently purchased together. By applying data transformation, grouping, and combinatorial analysis, this approach enables businesses to reveal valuable product affinities that can inform cross-selling strategies, in-store placements, and promotional bundling.

2. Goal

- Identify frequently co-purchased products to uncover hidden patterns in customer behavior.
- Determine the top 2-product and 3-product combinations most often bought together.
- Generate actionable insights that can improve product placement, cross-selling opportunities, and marketing promotions.
- Lay the groundwork for recommendation systems that can personalize customer experiences.

3. Business Challenge

- Lack of visibility into which products drive co-purchases.
- Missed revenue opportunities due to ineffective product bundling.
- Difficulty in designing targeted promotions based on real buying patterns.
- Underperformance of campaigns due to guesswork in product relationships.

4. Analysis Approach

- Apply Combinatorial Analysis: Use the `itertools.combinations` function to find frequent 2-product and 3-product sets.

- Aggregate & Rank: Use collections.Counter to count and sort the most common combinations.
- Derive Insights: Identify the top 10–20 most frequent item combinations to support marketing and product placement strategies.

Importing libraries

```
In [9]: import pandas as pd
import matplotlib.pyplot as plt
```

Loading the clean dataframe (post-ETL process)

```
In [11]: df = pd.read_csv("C:\\Monthly_Sales\\cleaned_data.csv")
```

```
In [12]: df.head()
```

```
Out[12]:
```

	Order ID	Product Name	Units Purchased	Unit Price	Order Date	Delivery Address	Month	Month Name	Year	Day of the Week
0	160155	Alienware Monitor	1	400.99	2024-01-01 05:04:00	765 Ridge St, Portland, OR 97035	1	January	2024	Monday
1	151041	AAA Batteries (4-pack)	1	4.99	2024-01-01 05:04:00	964 Lakeview St, Atlanta, GA 30301	1	January	2024	Monday
2	146765	AAA Batteries (4-pack)	1	4.99	2024-01-01 05:20:00	546 10th St, San Francisco, CA 94016	1	January	2024	Monday
3	145617	Amana Washing Machine	1	600.00	2024-01-01 05:24:00	961 Meadow St, Portland, OR 97035	1	January	2024	Monday
4	156535	Lightning Charging Cable	2	14.95	2024-01-01 05:45:00	451 Elm St, Los Angeles, CA 90001	1	January	2024	Monday

Top 20 product pairs most frequently ordered

together

```
In [14]: # Filtering rows with the same Order ID and adding .copy() to avoid SettingWithCopyWarning
df2 = df[df['Order ID'].duplicated(keep=False)].copy()

# Grouping Product Name with same Order ID
df2['Grouped'] = df2.groupby('Order ID')['Product Name'].transform(lambda x: ','.join(x))

# Dropping any duplicates that may occur as a result of the grouping
df3 = df2[['Order ID', 'Grouped']].drop_duplicates()

df3.head(10)

# Displaying the Top 20 Product Pairs Most Frequently Ordered Together
from itertools import combinations
from collections import Counter

count = Counter()

for row in df3['Grouped']:
    row_list = row.split(',')
    count.update(Counter(combinations(row_list, 2)))

top_combinations = count.most_common(20)

top_combinations
```

```
Out[14]: [(['Google Phone', 'USB-C Charging Cable'], 473),
          (('iPhone', 'Lightning Charging Cable'), 465),
          (('USB-C Charging Cable', 'Google Phone'), 464),
          (('Lightning Charging Cable', 'iPhone'), 453),
          (('Galaxy buds Headphones', 'iPhone'), 235),
          (('iPhone', 'Galaxy buds Headphones'), 221),
          (('Galaxy buds Headphones', 'Google Phone'), 197),
          (('Google Phone', 'Galaxy buds Headphones'), 176),
          (('USB-C Charging Cable', 'Samsung Galaxy Phone'), 175),
          (('Samsung Galaxy Phone', 'USB-C Charging Cable'), 172),
          (('Apple AirPods Headphones', 'iPhone'), 164),
          (('iPhone', 'Apple AirPods Headphones'), 157),
          (('Bose SoundSport Headphones', 'Google Phone'), 107),
          (('Google Phone', 'Bose SoundSport Headphones'), 98),
          (('Galaxy buds Headphones', 'USB-C Charging Cable'), 89),
          (('USB-C Charging Cable', 'Galaxy buds Headphones'), 79),
          (('Galaxy buds Headphones', 'Samsung Galaxy Phone'), 72),
          (('Galaxy buds Headphones', 'Lightning Charging Cable'), 70),
          (('Samsung Galaxy Phone', 'Galaxy buds Headphones'), 63),
          (('Lightning Charging Cable', 'Galaxy buds Headphones'), 62)]
```

Top 10 sets of three (3) products most commonly ordered together

```
In [16]: from itertools import combinations
         from collections import Counter
```

```

count = Counter()

for row in df3['Grouped']:
    row_list = row.split(',')
    count.update(Counter(combinations(row_list, 3)))

# Displaying top 10 most common combinations
top_combinations = count.most_common(10)

top_combinations

```

```

Out[16]: [ (('Galaxy buds Headphones', 'USB-C Charging Cable', 'Google Phone'), 17),
          (('Google Phone', 'USB-C Charging Cable', 'Galaxy buds Headphones'), 16),
          (('iPhone', 'Lightning Charging Cable', 'Galaxy buds Headphones'), 15),
          (('Galaxy buds Headphones', 'Google Phone', 'USB-C Charging Cable'), 14),
          (('Galaxy buds Headphones', 'Lightning Charging Cable', 'iPhone'), 14),
          (('Lightning Charging Cable', 'iPhone', 'Galaxy buds Headphones'), 12),
          (('Apple AirPods Headphones', 'Lightning Charging Cable', 'iPhone'), 12),
          (('Bose SoundSport Headphones', 'USB-C Charging Cable', 'Google Phone'), 10),
          (('Google Phone', 'USB-C Charging Cable', 'Bose SoundSport Headphones'), 10),
          (('Google Phone', 'Galaxy buds Headphones', 'USB-C Charging Cable'), 9)]

```

Key Insights

1. Strong Accessory Dependency:

- iPhone is frequently bought with the Lightning Charging Cable (473 times) and Apple AirPods Headphones (173 times), confirming a strong dependency on accessories.
- Similar behavior observed for Google Phone and USB-C Charging Cable (469 times), and Samsung Galaxy Phone with the same cable (177 times).

2. Bundling Behavior Around Devices:

- Buyers often purchase phones with multiple accessories (e.g., Google Phone + USB-C Cable + Headphones combinations).
- Example: Google Phone + USB-C Charging Cable + Bose SoundSport Headphones is a top 3-product combo.

3. Cross-Brand Headphone Love:

- Galaxy Buds Headphones appear frequently with iPhone, Google Phone, and Samsung Galaxy Phone, suggesting customers aren't loyal to brand-specific headphones. They mix and match based on price, design, or preference.

4. Repetitive Combination Patterns:

- Most of the high-frequency combinations are symmetrical (e.g., iPhone + Lightning Cable and Lightning Cable + iPhone)—reinforcing the reliability of the insight.

5. Accessory-Driven Purchases:

- Accessories like charging cables and headphones dominate the co-purchase behavior. These aren't just afterthoughts—they're clearly planned purchases.

Strategic Recommendations

1. Create Product Bundles:

- Offer ready-made bundles (Example, iPhone + Lightning Cable + AirPods) with small discounts to promote upselling and convenience.

2. Optimize Online Store Suggestions:

- Implement "Frequently Bought Together" suggestions at checkout using the most frequent 2- and 3-product combinations.

3. Improve In-Store Product Placement:

- Place high-affinity products (Example, USB-C Cable and Google Phone) adjacent on shelves to encourage impulse add-ons.

4. Run Targeted Promotions:

- Create cross-promotion campaigns such as "Buy a Google Phone and get 20% off USB-C Cable and Galaxy Buds."

5. Inventory Forecasting:

- Forecast accessory demand based on device sales, as customers are highly likely to buy them together.