D599: Data Exploration and Preparation Task 3

By: Staphon Smith

Part I: Research Question

A1. Research Question:

What combinations of product purchases frequently occur together across different customer segments, and how can these insights help Allias Megastore tailor its product bundles and marketing strategies?

A2. Goal of the Data Analysis:

The goal of this analysis is to use market basket analysis to identify common product pairings or groupings within different customer segments. These insights will help Allias Megastore develop targeted promotions, optimize cross-selling opportunities, and increase overall customer satisfaction and sales revenue.

Part II: Market Basket Justification

B1. Explanation of Market Basket Technique:

Market basket analysis (MBA) is a data mining technique used to uncover associations or co-occurrences between items in large transactional datasets. By using algorithms such as Apriori, we can identify which items are frequently bought together and quantify their relationships through metrics like **support**, **confidence**, and **lift**. Applying MBA to the Allias Megastore dataset allows us to identify product affinity patterns across customer purchases and market segments.

Expected Outcomes:

- Discovery of frequently co-purchased products.
- Identification of strong association rules to guide bundling and promotions.
- Insights to improve customer targeting and cross-selling.

B2: Transaction Example:

One example of a transaction from the dataset is **Order ID 536370**. The items bought together in this order include:

- WHITE HANGING HEART T-LIGHT HOLDER
- WHITE METAL LANTERN
- CREAM CUPID HEARTS COAT HANGER
- KNITTED UNION FLAG HOT WATER BOTTLE
- RED WOOLLY HOTTIE WHITE HEART

This order represents a typical group of items that a customer might purchase together. Market basket analysis looks at transactions like this to find patterns in what customers often buy at the same time. For instance, if customers who purchase the “WHITE HANGING HEART T-LIGHT HOLDER” also regularly buy the “WHITE METAL LANTERN,” the algorithm might highlight that as a strong association. Insights like this can help the company decide which products to bundle together or recommend to customers, improving sales and marketing strategies.

B3. Assumption of Market Basket Analysis:

One key assumption of market basket analysis is that past purchase behavior is predictive of future purchase behavior. This means that if two or more items are frequently purchased together historically, they are likely to be bought together again, providing a reliable basis for bundling or recommendation strategies.

Part III: Data Preparation and Analysis:

C1a: Categorical Variables

An ordinal variable is a categorical variable that has a natural order. The first ordinal variable I will consider is “**order priority”**, which can be categorized as “high” or “medium,” allowing one data point to be ranked higher than the other. The second ordinal variable is **“Customer Satisfaction”**, which is ranked from 0 to 4 based on satisfaction level:
“Prefer not to respond” (0), “Dissatisfied” (1), “Very Dissatisfied” (2), “Satisfied” (3), and “Very Satisfied” (4). This variable clearly shows a progression in satisfaction.

For nominal variables, I will use “**Segment”**, which can either be corporate or consumer, and “**Payment Method”**, which can be PayPal or credit card.

C1b:Encoding:

Before applying the Apriori algorithm, we must convert categorical data into a numerical format. The algorithm cannot interpret raw text values, so encoding is necessary.

For ordinal variables, I used ordinal encoding.

Specifically:

Ordinal Variables (Natural Order Exists):

- **Order Priority**: Encoded using ordinal encoding:
    - Medium = 1
    - High = 2

This preserves the natural ranking of priority.

- **Customer Satisfaction**: Also encoded using ordinal encoding:
    - Prefer not to respond = 0
    - Dissatisfied = 1
    - Very Dissatisfied = 2
    - Satisfied = 3
    - Very Satisfied = 4

This encoding reflects the increasing level of customer satisfaction.

For nominal variables, which have no inherent order, I applied one-hot encoding to convert them into binary flags. For example, Segment was transformed into Segment_Consumer and Segment_Corporate, each with 0 or 1 values depending on the record.

Using ordinal encoding for variables with a defined order and one-hot encoding for those without ensures the dataset is ready for analysis by algorithms like Apriori, which require binary input.

```
# Ordinal Encoding - Order Priority and Expedited Shipping
df['OrderPriority_Encoded'] = df['OrderPriority'].map({'Medium': 1, 'High': 2})
customer_satisfaction_mapping = {
    'Prefer to not respond': 0,
    'Dissatisfied': 1,
    'Very dissatisfied': 2,
    'Satisfied': 3,
    'Very Satisfied': 4
}
df['CustomerOrderSatisfaction_Encoded'] = df['CustomerOrderSatisfaction'].map(customer_satisfaction_mapping)


# One-hot Encoding - Segment and Payment Method
df_encoded = pd.get_dummies(df, columns=['Segment', 'PaymentMethod'])

# Add ordinal Encoded columns
df_encoded['OrderPriority_Encoded'] = df['OrderPriority_Encoded']
df_encoded['CustomerOrderSatisfaction_Encoded'] = df['CustomerOrderSatisfaction_Encoded']
```

C1c:Transactionalize Data and C1d: Justification

After encoding, I needed to turn the product data into a format Apriori can use — a transactional matrix. In this matrix, each row represents a unique order, and each column is a product. A True means the product was part of that order; False means it wasn't.

Here's what I did:

1. Grouped the data by OrderID and ProductName to sum the quantity of each item per order.

2. Unstacked it so that each product became its own column.

3. Replaced any missing values with 0.

4. Convert the numbers to True/False using .gt(0).astype(bool) — so Apriori knows what was purchased.

The result is a binary matrix ready for analysis.

```
# Transactionalize Data
basket = df.groupby(['OrderID', 'ProductName'])['Quantity'].sum().unstack().fillna(0)
basket = basket.gt(0).astype(bool)
```

```
# Transactional basket visual
print(basket)
```

```
ProductName   50S CHRISTMAS GIFT BAG LARGE   DOLLY GIRL BEAKER  \
OrderID
536370                               False              False
536852                               False              False
536974                               False              False
537065                               False              False
537463                               False              False
...                                    ...                ...
581001                               False              False
581171                               False              False
581279                               False              False
581316                               False              False
581587                               False              False

ProductName   I LOVE LONDON MINI BACKPACK   NINE DRAWER OFFICE TIDY  \
OrderID
536370                             False                     False
536852                             False                     False
```

C2:Clean Dataset Copy:

See Attached.

C3:Execute Code:

Here is the code I used to run the Apriori algorithm, as well as a sample of the output:

```
#Start of Market Basket Analysis
from mlxtend.frequent_patterns import apriori, association_rules

# apply the Apriori algorithm
frequent_itemsets = apriori(basket, min_support=0.01, use_colnames=True)

# Display the itemsets
print(frequent_itemsets)
```

```
      support                                           itemsets
0    0.020408                          ( DOLLY GIRL BEAKER)
1    0.011338                ( I LOVE LONDON MINI BACKPACK)
2    0.013605            ( SET 2 TEA TOWELS I LOVE LONDON )
3    0.036281                     ( SPACEBOY BABY GIFT SET)
4    0.027211                      (10 COLOUR SPACEBOY PEN)
...       ...                                           ...
8335 0.011338  (SET6 RED SPOTTY PAPER CUPS, SET OF 9 HEART SH...
8336 0.011338  (SET6 RED SPOTTY PAPER CUPS, SET OF 9 HEART SH...
8337 0.011338  (SET6 RED SPOTTY PAPER CUPS, SET OF 9 HEART SH...
8338 0.011338  (ALARM CLOCK BAKELIKE GREEN, SET6 RED SPOTTY P...
8339 0.011338  (SET6 RED SPOTTY PAPER CUPS, SET OF 9 HEART SH...

[8340 rows x 2 columns]
```

```python
# Generate association rules from frequent itemsets
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1.0)

# Display the first few rules
print(rules.head())
```

```
                        antecedents                       consequents  \
0  (CHARLOTTE BAG DOLLY GIRL DESIGN)              ( DOLLY GIRL BEAKER)
1               ( DOLLY GIRL BEAKER)  (CHARLOTTE BAG DOLLY GIRL DESIGN)
2        (DOLLY GIRL CHILDRENS BOWL)              ( DOLLY GIRL BEAKER)
3               ( DOLLY GIRL BEAKER)        (DOLLY GIRL CHILDRENS BOWL)
4         (DOLLY GIRL CHILDRENS CUP)              ( DOLLY GIRL BEAKER)

   antecedent support  consequent support   support  confidence       lift  \
0            0.058957            0.020408  0.011338    0.192308   9.423077
1            0.020408            0.058957  0.011338    0.555556   9.423077
2            0.040816            0.020408  0.015873    0.388889  19.055556
3            0.020408            0.040816  0.015873    0.777778  19.055556
4            0.036281            0.020408  0.013605    0.375000  18.375000

   representativity  leverage  conviction  zhangs_metric   jaccard  certainty  \
0               1.0  0.010135    1.212828       0.949880  0.166667   0.175481
1               1.0  0.010135    2.117347       0.912500  0.166667   0.527711
2               1.0  0.015040    1.602968       0.987842  0.350000   0.376157
3               1.0  0.015040    4.316327       0.967262  0.350000   0.768322
4               1.0  0.012865    1.567347       0.981176  0.315789   0.361979
```

C4:Support, Lift, And Confidence Values and C5:Relevant Rules:

Using the frequent itemsets generated from the dataset, I created association rules to uncover how products relate to each other in customer transactions. The key metrics—support, confidence, and lift—help quantify the strength and significance of these rules.

The top rules identified all share the same high performance across metrics:

- **Support:** 0.011338
- **Confidence:** 1.0
- **Lift:** 88.2

These values indicate strong and highly reliable item relationships. A **confidence of 1.0** means that when the antecedent items are purchased, the consequent items are always purchased as well. A **lift of 88.2** suggests that these combinations are **88 times more likely** to occur together than randomly — a significant insight. Although the **support is relatively low**, meaning the exact combinations are rare, their **predictive strength and consistency** make them extremely valuable for targeted marketing and bundling strategies.

```
top_rules = rules.sort_values(by='lift', ascending=False).head(3)
top_rules.head()
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	representativity	leverage	conviction	zhangs_metric	jaccard	certainty	kulczyn
82744	(SET6 RED SPOTTY PAPER PLATES, ROUND SNACK BOX...	(ALARM CLOCK BAKELIKE GREEN, SET6 RED SPOTTY P...	0.011338	0.011338	0.011338	1.0	88.2	1.0	0.011209	inf	1.0	1.0	1.0	
82991	(ALARM CLOCK BAKELIKE PINK, SPACEBOY BIRTHDAY ...	(ALARM CLOCK BAKELIKE RED , CARD DOLLY GIRL , ...	0.011338	0.011338	0.011338	1.0	88.2	1.0	0.011209	inf	1.0	1.0	1.0	
84849	(ALARM CLOCK BAKELIKE RED , SET6 RED SPOTTY PA...	(ALARM CLOCK BAKELIKE GREEN, SET6 RED SPOTTY P...	0.011338	0.011338	0.011338	1.0	88.2	1.0	0.011209	inf	1.0	1.0	1.0	

Rule 1

Antecedents: (SET6 RED SPOTTY PAPER PLATES, ROUND SNACK BOXES SET OF 4 WOODLAND)

Consequents: (ALARM CLOCK BAKELIKE GREEN, SET6 RED SPOTTY PAPER NAPKINS)

This rule reflects a clear party-related purchasing trend. Customers who buy paper plates and snack boxes are highly likely to also purchase napkins and a matching clock. The combination suggests a coordinated theme or event, making it ideal for promoting **party bundles** or recommending add-on items at checkout. The association is extremely strong, with a **confidence of 1.0** and a **lift of 88.2**, meaning that the presence of the antecedent items almost perfectly predicts the purchase of the consequent items. However, the **support is low (0.011338)**, indicating that this exact combination of items is relatively rare in the overall dataset.

Rule 2

Antecedents: (ALARM CLOCK BAKELIKE PINK, SPACEBOY BIRTHDAY CARD)

Consequents: (ALARM CLOCK BAKELIKE RED, CARD DOLLY GIRL)

This rule suggests themed gift purchases — blending clocks with birthday cards. It likely represents customers shopping for children's gifts or themed events. With such a high lift and perfect confidence, this rule could guide **gift set curation** or **seasonal marketing** for birthdays and events. The **confidence is 1.0** and the **lift is extremely high**, highlighting a very strong predictive relationship. Still, this combination appears **infrequently in the data (low support)**, making it a niche but dependable pattern.

Rule 3
Antecedents: (ALARM CLOCK BAKELIKE RED, SET6 RED SPOTTY PAPER PLATES)

Consequents: (ALARM CLOCK BAKELIKE GREEN, SET6 RED SPOTTY PAPER NAPKINS)

This rule highlights another strong pattern involving color-themed clocks and matching party supplies. The antecedents and consequents suggest that customers buying red-themed items are also purchasing green items in the same category, indicating coordinated shopping behavior. It provides a great opportunity to **bundle color-themed party supplies** or to design automated suggestions for related items. The association is very strong with **high lift and confidence values**, confirming a tight pattern among these products. Like the previous rules, it has **low support**, meaning the scenario is specific but very consistent when it does occur.

Part IV Data Summary and Implications

D1:Significance of Support, Lift and Confidence from results

Support, lift, and confidence are important measures that helped me understand how strong and relevant the item relationships are in the dataset. In this analysis, the confidence levels were perfect (1.0), which tells me that when customers bought the first group of items, they also always bought the second group. The lift values were also extremely high, which means these purchases are strongly connected—way more than just random chance. However, the support was low, which shows that while these item combinations don't happen often, they're very consistent when they do.

D2:Significance of findings and D3. Recommended course of action

What stood out in the results is that people tend to buy themed items together like clocks, lunch boxes, paper cups, and cards; which points to situations like planning a kid's birthday party or putting together a gift. These products seem to go hand in hand, even if the combo doesn't come up super often. This tells me that certain customers are looking for coordinated items, which is something the company can take advantage of. There's a clear opportunity here to make shopping easier for them by offering related products together.

Based on these findings, I recommend the company create themed bundles or kits using the items that show up in the rules—like pairing lunch boxes with matching snack boxes, clocks, and birthday cards. These could be marketed as party sets or gift ideas. It would also be smart to use this info in online product suggestions. For example, if someone adds a lunch box to their cart, the site could recommend the other connected items. Even though not every shopper buys this combo, the ones who do are clearly following a pattern, and that gives the company a way to boost sales and improve the shopping experience.

Sources

No sources were used except for WGU materials.