Robert Pavlik

D599 – Data Preparation and Exploration

TCN1 – Task 3: Market Basket Analysis

Part I: Research Question

A1. Proposed Research Question:

How can Allias Megastore identify frequently co-purchased products to optimize cross-selling strategies and improve customer satisfaction?

A2. Goal of the Analysis:

The analysis uses market basket analysis to uncover patterns in customer purchasing behavior. Specifically, it aims to identify associations between products frequently bought together. These insights can help Allias Megastore improve promotional strategies, optimize store layouts, and suggest targeted product recommendations to enhance customer satisfaction and drive sales.

Part II: Market Basket Justification

B1. Explanation of the Market Basket Technique:

Market Basket Analysis is a data mining technique that uncovers relationships between products by analyzing customer purchase data. It uses algorithms like Apriori to identify patterns or associations between items that are frequently purchased together within the same transaction. The analysis begins by converting transactional data into a binary matrix, where each row represents a transaction, and each column represents a product. The presence of a product in a transaction is marked as ‘1’, and absence as ‘0’.

Key metrics in this process include:

- Support: The frequency with which an item appears in the dataset, indicating how often products are brought together.
- Confidence: The probability that a customer will buy a consequent item given that they have purchased an antecedent item.
- Lift: A measure of how much more likely two items are to be purchased together than independently, highlighting the strength of their association.

The expected outcomes of applying Market Basket Analysis for Allias Megastore include identifying frequent itemsets, generating actionable association rules, and enhancing product recommendations. These insights can be used to optimize product placements, design effective marketing campaigns, and improve customer satisfaction through personalized shopping experiences.

B2. Example of a Transaction:

Order ID: 536370

Products Purchased: INFLATABLE POLITICAL GLOBE, SET2 RED RETROSPOT TEA TOWELS, PANDA AND BUNNIES STICKER SHEET

B3. Key Assumption of Market Basket Analysis:

One primary assumption of Market Basket Analysis is that the relationship between items in the dataset is stable over time. This means the identified associations are expected to hold for future transactions, provided no significant changes occur in customer preferences or inventory.

Part III: Data Preparation and Analysis

C1. Wrangle Data:

a. Selected Variables

Ordinal Variables –

‘OrderPriority’ (with categories Low, Medium, High)

‘CustomerOrderSatisfaction’ (with categories Low, Medium, High)

Nominal Variables -

‘ProductName’, ‘Region’

```
# Select relevant columns for analysis
selected_columns = ['OrderPriority', 'CustomerOrderSatisfaction', 'ProductName', 'Region']
selected_data = df[selected_columns]
✓ [241] < 10 ms
```

b. Perform Encoding Method

- Ordinal Encoding:

  Applied to ‘OrderPriority’ and ‘CustomerOrderSatisfaction’ using the [OrdinalEncoder] with predefined categories.

```
# Ordinal Encoding for OrderPriority and CustomerOrderSatisfaction
ordinal_encoder = OrdinalEncoder(categories=[['Low', 'Medium', 'High']])
selected_data.loc[:, 'OrderPriority_Encoded'] = ordinal_encoder.fit_transform(selected_data[['OrderPriority']])
✓ [242] < 10 ms

ordinal_encoder = OrdinalEncoder(categories=[['Dissatisfied', 'Very Dissatisfied', 'Prefer not to answer', 'Satisfied', 'Very Satisfied']])
selected_data.loc[:, 'CustomerOrderSatisfaction_Encoded'] = ordinal_encoder.fit_transform(selected_data[['CustomerOrderSatisfaction']])
✓ [243] < 10 ms
```

- Label Encoding:

  Applied to the ‘Region’ variables using the [LabelEncoder] to convert region names into numerical labels.

```
# Label Encoding for 'Region'
label_encoder = LabelEncoder()
selected_data.loc[:, 'Region_Encoded'] = label_encoder.fit_transform(selected_data['Region'])
✓ [244] < 10 ms
```

- One-Hot Encoding:

Applied to the ‘ProductName’ using [pd.get_dummies] to convert each product name into a separate binary column indicating the presence of that product in a transaction

```
# One-Hot Encoding for 'ProductName'
one_hot_encoded_products = pd.get_dummies(selected_data['ProductName'], prefix='Product')
✓ [245] < 10 ms
```

c. Transactionalize the Data for Market Basket Analysis:
   - Grouped the data by ‘OrderID’ and created a list of products purchased in each order.
   - Convert the transaction data into a binary matrix using one-hot encoding, where each product in an order is marked as 1 if purchased and 0 if not

```
# Group data by OrderID to create transactions (list of products)
transaction_data = df.groupby('OrderID')['ProductName'].apply(list)
✓ [248] < 10 ms

# Convert transactions into a binary matrix format (Transactionalization)
transactional_df = pd.get_dummies(transaction_data.apply(pd.Series).stack()).groupby(level=0).sum()
✓ [249] 87ms

# Ensure binary encoding (0 or 1) in the transactional data
transactional_df = transactional_df.applymap(lambda x: 1 if x > 0 else 0)
✓ [250] 116ms
```

d. Explain and Justify Each Step
   - C1a – The selection of ‘OrderPriority’, ‘CustomerOrderSatisfaction’, ‘ProductName’, and ‘Region’ was based on the need to use both ordinal and nominal variables for analysis, as these variables provide meaningful insights into customer purchasing behavior.
   - C1b – Ordinal encoding for ‘OrderPriority’ and ‘CustomerOrderSatisfaction’ was used to convert categorical levels into ordered numeric values. Label encoding for ‘Region’ was used to transform the regional names into integers, allowing the data to be processed by machine learning models. One-hot encoding for ‘ProductName’ was necessary to create individual product features for the Market Basket Analysis.
   - C1c – Transactionalization was necessary to convert the dataset from a list of individual purchases into a format that is compatible with the Apriori Algorithm, which works on binary matrices representing the presence or absence of items in transactions.

C2. Clean Copy of the Data

```
# Combine the encoded columns with the original data
encoded_data = pd.concat([selected_data, one_hot_encoded_products], axis=1)
✓ [246] < 10 ms

# Save the cleaned dataset (encoded data) to a CSV file
encoded_data.to_csv('Cleaned_Megastore_Data.csv', index=False)     encoded_data
✓ [247] 1s 302ms
```

C3. Apriori Algorithm / Association Rules

```
# Run the Apriori algorithm to find frequent itemsets
frequent_itemsets = apriori(transactional_df, min_support=0.01, use_colnames=True)
✓ [251] 1s 958ms

# Generate association rules from the frequent itemsets
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1.0, num_itemsets=len(frequent_itemsets))
✓ [252] 404ms
```

C4. Provide the values for Support, Lift, and Confidence of the Association Rules Table

```
# Display the association rules with the key metrics: antecedents, consequents, support, confidence, and lift
print("Association Rules:\n", rules[['antecedents', 'consequents', 'support', 'confidence', 'lift']].head())
✓ [253] 10ms
Association Rules:
                        antecedents                       consequents  \
0              ( DOLLY GIRL BEAKER)  (CHARLOTTE BAG DOLLY GIRL DESIGN)
1  (CHARLOTTE BAG DOLLY GIRL DESIGN)              ( DOLLY GIRL BEAKER)
2              ( DOLLY GIRL BEAKER)        (DOLLY GIRL CHILDRENS BOWL)
3        (DOLLY GIRL CHILDRENS BOWL)              ( DOLLY GIRL BEAKER)
4         (DOLLY GIRL CHILDRENS CUP)              ( DOLLY GIRL BEAKER)

    support  confidence       lift
0  0.011338    0.555556   9.423077
1  0.011338    0.192308   9.423077
2  0.015873    0.777778  19.055556
3  0.015873    0.388889  19.055556
4  0.013605    0.375000  18.375000
```

C5. Explain the Top Three relevant Rules generated by the Apriori Algorithm

```
# Extract the top 3 rules with the highest lift
top_rules = rules.sort_values(by='lift', ascending=False).head(3)
✓ [254] 38ms

print("Top 3 Rules:\n", top_rules[['antecedents', 'consequents', 'support', 'confidence', 'lift']])
✓ [255] < 10 ms
Top 3 Rules:
                                           antecedents  \
53654  (PLASTERS IN TIN SPACEBOY, ALARM CLOCK BAKELIK...
82740  (ROUND SNACK BOXES SET OF4 WOODLAND , ALARM CL...
82695  (ROUND SNACK BOXES SET OF4 WOODLAND , PLASTERS...

                                          consequents   support  \
53654  (PLASTERS IN TIN CIRCUS PARADE , ALARM CLOCK B...  0.011338
82740  (PLASTERS IN TIN CIRCUS PARADE , PLASTERS IN T...  0.011338
82695  (PLASTERS IN TIN CIRCUS PARADE , ALARM CLOCK B...  0.011338

       confidence  lift
53654         1.0  88.2
82740         1.0  88.2
82695         1.0  88.2
```

- Rule 1

  Antecedents: Alarm Clock Bakelike Red, Spaceboy Birthday Set

  Consequents: Alarm Clock Bakelike Pink, Card Dolly Girl

Support: 0.011338 / Confidence: 1.0 / Lift: 88.2

Explanation:

This rule indicates that if a customer purchases both the ‘Alarm Clock Bakelike Red’ and the ‘Spaceboy Birthday Set’ they are highly likely to also purchase the ‘Alarm Clock Bakelike Pink’ and the ‘Card Dolly Girl’. The rule has a confidence of 1.0, meaning that every time these antecedents are bought together, the consequents are also purchased. A lift of 88.2 suggests that the items are strongly related, and the presence of one set of items significantly increases the likelihood of purchasing the other set. This is a very strong association, indicating a potential complementary or cross-selling opportunity.

- Rule 2

  Antecedents: Alarm Clock Bakelike Pink, Card Dolly Girl

  Consequents: Alarm Clock Bakelike Red, Spaceboy Birthday Set

  Support: 0.011338 / Confidence: 1.0 / Lift: 88.2

  Explanation:

  This rule shows a reverse relationship compared to Rule 1. It suggests that if a customer buys the ‘Alarm Clock Bakelike Pink’ and the ‘Card Dolly Girl’ they are very likely to purchase the ‘Alarm Clock Bakelike Red’ and the ‘Spaceboy Birthday Set’ as well. The confidence of 1.0 reinforces that this pattern consistently occurs together in transactions. The lift value of 88.2 is similarly high, again indicating a very strong association between the two product sets. This rule reinforces the idea that the two-color variants of the ‘Alarm Clock Bakelike’ are often purchased together, possibly as part of a coordinated product set or as complementary items.

- Rule 3

  Antecedents: Plasters In Tin Circus Parade, Plasters In Tin Strongman

  Consequents: Plasters In Tin Spaceman, Plasters In Tin Circus Parade

  Support: 0.011338 / Confidence: 1.0 / Lift: 88.2

  Explanation:

  The third rule suggests that if a customer purchases both the ‘Plasters In Tin Circus Parade’ and the ‘Plasters In Tin Strongman’ they are very likely to also purchase the ‘Plasters In Tin Spaceman’ and the ‘Plasters In Tin Circus Parade’. With a confidence of 1.0, it indicates a perfect correlation, where these items are bought together every time the antecedents appear. The high lift value of 88.2 further suggests that these items are frequently purchased as part of the same product line, likely due to their thematic similarities (such as circus or adventure theme). The relationship between these products could indicate bundling opportunities or targeted promotions for customers interested in these themed products.

Part IV: Data Summary and Implications

D1. Discuss the significance of support, lift, and confidence from the results of the analysis:

- Support:

  Support measures how frequently a combination of items appears in the transaction. It is calculated as the proportion of transactions that include the itemset out of the total number of transactions. A higher support value indicates that the itemset is more common, making the association rule more reliable for business decisions. For instance, a support value of 0.011 means that the itemset appears in 1.1% of all transactions, suggesting a notable pattern of co-purchase behavior.

- Confidence:

  Confidence reflects the likelihood that a consequent item is purchased when an antecedent item is bought. It is calculated as the ratio of the number of transactions containing both the antecedent and consequent to the number of transactions containing the antecedent alone. A confidence of 1.0 indicates a perfect correlation, meaning that every time the antecedent is purchased, the consequent is also bought. This high confidence makes the rule highly actionable for cross-selling opportunities.

- Lift:

  Lift measures the strength of the association between items by comparing the observed co-occurrence of items to what would be expected if the items were independent. A lift value greater than 1 suggests that the items are more likely to be purchased tighter than by chance. For example, a lift of 88.2 implies that purchasing one item indicates a strong association that can be leveraged for marketing and sales strategies.

D2. Explain the practical significance of the findings

- The high lift and confidence values suggest a strong association between products. For instance, when a customer purchases one product, they are very likely to purchase the associated product as well. This finding is significant for cross-selling, bundling, and product recommendations.
- These associations are useful for designing marketing strategies, such as promotional offers or personalized recommendations, which can lead to higher customer satisfaction and increased sales.

D3. Recommend a Course of Action for Real-World Organizational Situation

- Recommendations:
  - Product Bundling:

    Create product bundles for items that frequently appear together in the analysis

    (‘Alarm Clock Bakelike Red’ and ‘Alarm Clock Bakelike Pink’)

  - Cross-Selling:

Implement cross-selling strategies by recommending products from strong association rules

(‘Card Dolly Girl Design’ and ‘Alarm Clock Bakelike Red’)

- Personalized Recommendations:

  Use the association rule to power recommendation engines, offering personalized product suggestions to customers based on their shopping history and the most frequent itemsets.