Market Basket Analysis Case Study:

Question:

Based on the results of the Market Basket Analysis using the Online Retail dataset, provide insights into the frequent itemsets and association rules discovered. What are the key patterns of items being purchased together, and how do these patterns contribute to our understanding of customer behaviour? Additionally, discuss the significance of the selected thresholds (e.g., minimum support, lift, confidence) in shaping the identified association rules.

Solution:

Certainly! Let's break down the Market Basket Analysis case study step by step, explaining the algorithmic concepts behind each stage.

1. **Install Required Libraries**

```python
!pip install pandas matplotlib seaborn mlxtend
```

This step installs the necessary Python libraries: pandas for data manipulation, matplotlib and seaborn for data visualization, and mlxtend for implementing the Apriori algorithm.

2. **Import Libraries**

```python
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from mlxtend.frequent_patterns import apriori

from mlxtend.frequent_patterns import association_rules
```

Import the required libraries for data manipulation, visualization, and implementing the Apriori algorithm for market basket analysis.

3. **Load and Explore the Dataset**

```python
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/00352/Online%20Retail.xlsx"

df = pd.read_excel(url)

print(df.head())
```

Load the Online Retail dataset, which contains transaction data from an online retail store. Display the first few rows of the dataset to understand its structure.

4. **Data Preprocessing**

```python
df['Description'] = df['Description'].str.strip()
df.dropna(axis=0, subset=['InvoiceNo'], inplace=True)
df = df[df['Country'] == 'France']
basket = (df.groupby(['InvoiceNo', 'Description'])['Quantity']
          .sum().unstack().reset_index().fillna(0)
          .set_index('InvoiceNo'))
```

Clean the data by removing leading/trailing spaces in descriptions, dropping rows with missing invoice numbers, and filtering transactions for the country 'France'. Consolidate items into 1 transaction per row.

5. **One-Hot Encoding and Apriori Algorithm**

```python
def encode_units(x):
    if x <= 0:
        return 0
    if x >= 1:
        return 1

basket_sets = basket.applymap(encode_units)
frequent_itemsets = apriori(basket_sets, min_support=0.07, use_colnames=True)
```

Perform one-hot encoding, converting quantities to 0 or 1 (binary encoding). Use the Apriori algorithm to find frequent itemsets in the dataset, considering a minimum support of 0.07.

6. **Association Rules**

```python
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)
print(rules)
```

Generate association rules from the frequent itemsets using the Apriori algorithm. Display the resulting rules, including support, confidence, and lift for each rule.

7. **Results and Visualization**

```python

filtered_rules = rules[(rules['lift'] >= 6) & (rules['confidence'] >= 0.8)]

print(filtered_rules)


sns.scatterplot(x='support', y='confidence', data=rules, size='lift', sizes=(10, 200))

plt.title('Association Rules - Support vs Confidence')

plt.xlabel('Support')

plt.ylabel('Confidence')

plt.show()

```

Filter the association rules based on specific conditions (e.g., lift > 6, confidence > 0.8). Display and visualize the association rules using a scatter plot, where support and confidence are plotted against each other, and the size of the points represents the lift of the rule. This case study demonstrates how to perform market basket analysis using the Apriori algorithm, identify association rules, and visualize the results to understand item relationships in transaction data. Adjusting parameters and conditions allows customization based on specific business requirements.

The Apriori algorithm is a classic algorithm in data mining and machine learning used for association rule mining. Association rule mining aims to discover interesting relationships, patterns, or associations among a set of items in large datasets. The Apriori algorithm is particularly useful in analyzing transaction data, such as market basket analysis, where the goal is to find patterns of items that are frequently bought together.

Here's a step-by-step explanation of how the Apriori algorithm works:

1. **Support Counting:**

  - **Support:** It measures the frequency of occurrence of an itemset in the dataset.

  - **Support Count:** The number of transactions that contain a particular itemset.

2. **Frequent Itemset Generation:**

  - An itemset is considered "frequent" if its support count is above a specified minimum support threshold.

  - The algorithm starts by identifying frequent individual items (1-itemsets).

  - Then, it iteratively generates candidate itemsets of higher lengths based on the frequent itemsets of the previous iteration.

- The process continues until no new frequent itemsets can be generated.

3. **Association Rule Generation:**

- Once frequent itemsets are identified, association rules are generated based on them.

- An association rule has the form "A => B," where A and B are itemsets.

- The algorithm generates rules with a confidence above a specified minimum confidence threshold.

4. **Confidence Calculation:**

- **Confidence:** It measures the reliability or certainty of a rule. It is calculated as the support of the combined itemset divided by the support of the antecedent (preceding) itemset.

- High confidence indicates a strong association between items in the rule.

5. **Pruning:**

- To efficiently explore the combinatorial space of itemsets and rules, the algorithm uses a pruning step.

- If an itemset is not frequent, its supersets (larger itemsets that contain it) are also not considered frequent. This helps reduce the search space.

Example:

Consider a transaction dataset:

```

Transaction 1: {bread, milk}

Transaction 2: {bread, diaper, beer, eggs}

Transaction 3: {milk, diaper, beer, cola}

Transaction 4: {bread, milk, diaper, beer}

```

Suppose the minimum support is set to 50%. The Apriori algorithm would proceed as follows:

1. **1-itemset generation:** Identify frequent individual items.

- {bread}: 3

- {milk}: 3

- {diaper}: 3

- {beer}: 3

2. **2-itemset generation:** Create candidate 2-itemsets from frequent 1-itemsets.

- {bread, milk}: 2 (not frequent)

   - {bread, diaper}: 3 (frequent)

   - {bread, beer}: 2 (not frequent)

   - {milk, diaper}: 2 (not frequent)

   - ...

3. **3-itemset generation:** Create candidate 3-itemsets from frequent 2-itemsets.

   - {bread, diaper, beer}: 2 (not frequent)

   - ...

4. **Association Rule Generation:**

   - Calculate confidence for rules like {bread, diaper} => {beer}.

5. **Pruning:**

   - Eliminate itemsets and rules that do not meet the support and confidence thresholds.

The Apriori algorithm efficiently discovers frequent itemsets and association rules, making it a powerful tool for uncovering patterns in large transactional datasets.