



Introduction to Association Rule Mining - Market Basket Analysis (MBA)

- Association rules are **if-then** statements that show **the probability of relationships** between data items within large data sets in various types of databases.
- At a basic level, association rule mining involves the use of machine learning models to analyze data for patterns, called **co-occurrences**, in a database.
- It identifies **frequent if-then associations**, which themselves are the association rules.
- For example, if 75% of people who buy cereal also buy milk, then there is an evidence for pattern in the transactional data that customers who buy cereal often buy milk.

For example, the rule:

$$\{\text{cereal}\} \Rightarrow \{\text{milk}\}$$

Such information can be used as the basis for decisions about marketing activities such as, e.g., promotional pricing or product placements.

Market Basket Analysis



نلاستفسار فرعي 108 - 06/5814168			
=====			
رقم الفاتة	62464	الرقم الضريبي:	16650476
=====			
كاش	6	التاريخ	28/05/2025
الساعة :	3:34:00 pm		
المادة	الكمية	السعر	المجموع
ارم الذ همر اذفانس	1	1.350	1.350
ارم الذ همر معجون اسنان	1	1.350	1.350
معجون اورل بي 123	1	1.390	1.390
المراعي لينة 500 غم	1	2.250	2.250
صابون حمام نابلسي حبة و	1	0.550	0.550
عصير الربيع اناثاس 1 لتر	1	1.400	1.400
عصير الربيع كوكتيل 1 لتر	1	1.400	1.400
هاربيك سيترس 495 مل	1	1.200	1.200
هاربيك لافندر 495 مل	1	1.200	1.200
كلوف لاتيكنس اسود وسط*	1	2.490	2.490
تشيرورز غسل 375 غم	1	3.750	3.750
رايس كيك طبيعي 100 غم	2	0.950	1.900
رايس كيك سمسم 100 غم	1	0.950	0.950
غاستو مناديل مبللة 96	1	0.990	0.990
ساتينو معجون اسنان الحس	1	0.990	0.990
جبنة برمران ايطاني	0.166	12.990	2.156
جبنة برمران ايطاني	0.354	12.990	4.598
فلل قرن الغزال	0.552	0.650	0.359
بندورة علائيد كبيرة	0.7	1.100	0.770
مخلل هلاينو	0.284	1.990	0.565
خيار	0.44	0.790	0.348

المجموع النهائي:			31.956
فيزا :			31.956
المدفوع			0.000
الباقى			0.000
عدد المواد :			22

شكرا لزيارتكم
لا يتم تبديل البضاعة الا بوجود
فاتورة شراء خلال 24 ساعة

نلاستفسار فرعي 108 - 06/5814168			
=====			
رقم الفاتة	62465	الرقم الضريبي:	16650476
=====			
كاش	6	التاريخ	28/05/2025
الساعة :	3:35:00 pm		
المادة	الكمية	السعر	المجموع
نبيل دجاج تندر 900 غم	1	3.750	3.750
نبيل اصابع سمك محمد 40	2	2.390	4.780
خضار مشكلة السنبله 450	2	1.600	3.200
السنبله بامية اكسترا 400	2	2.590	5.180
سبانخ السنبله 400 غم	2	0.900	1.800
جبنه فاكيريوم روابي جرش 3	1	3.550	3.550
مكدوس بلدي التميمي 1 كغ	1	3.650	3.650
بردايس خبز طابون 4*	1	0.600	0.600
غيث كيز مسخن 5*	1	0.850	0.850
مخلل هلاينو	0.484	1.990	0.983
ديلايت قشقوان	0.3	8.190	2.457
مشمش حموي سوبر	0.796	1.750	1.393
سمسمية مصرية	0.816	1.750	1.428
مفروم عجل + خاروف	0.335	8.490	2.844
مفروم عجل + خاروف	0.355	8.490	3.014
دجاج تندوري مقطع متيل لا	0.98	3.250	3.185

المجموع النهائي:			42.664
فيزا :			42.664
المدفوع			0.000
الباقى			0.000
عدد المواد :			20

شكرا لزيارتكم
لا يتم تبديل البضاعة الا بوجود
فاتورة شراء خلال 24 ساعة

✓ The Apriori Algorithm

Apriori is the most common algorithm used for MBA. It identifies frequent itemsets and builds association rules from them by leveraging the principle that **subsets of frequent itemsets must also be frequent**.

Key Notes:

- Apriori is an algorithm for identifying frequent item-sets and association rule learning over relational databases.
- It was proposed by Agrawal and Srikant in 1994.
- Apriori is designed to operate on databases containing transactions (for example, collections of items bought by customers, or details of a website frequentation or IP addresses).
- It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database.
- The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database.
- This has applications in domains such as market basket analysis.
- Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data.
- The algorithm terminates when no further successful extensions are found.

.....

✓ Key Concepts:

1. Association Rules:

- These are if-then rules that identify relationships between items. For example, "If a customer buys bread, they are likely to buy butter."

✓ 2. Metrics:

2.1. Support:

Support is a measure that represents the frequency or proportion of transactions in the dataset that contain a given item set or pattern. In other words, it measures how often a particular combination of items appears together in the dataset. Support is a **Probability Based** measure that has a value between 0 and 1. $\text{Support}(X) = \frac{\text{Number of transactions containing } X}{\text{Total transactions}}$

Valid Values for Support:

- **Range:** The value of support ranges from **0% to 100%**:
- **0%:** The itemset does not appear in any transaction.

- **100%:** The itemset appears in all transactions.
- *Interpreting Support:**
- **High support:** The itemset is common and may represent a strong association.
- **Low support:** The itemset is rare and might not provide actionable insights.

Thresholds in Practice

- In practice, a minimum support threshold is specified to filter out infrequent itemsets, ensuring only itemsets that appear frequently in the dataset are considered for further analysis.
- Generally speaking, the minimum values for **support**, **confidence**, and **lift** depend on the specific dataset and the business context, but here are some common guidelines and starting points:
- Setting a very high support threshold can exclude less frequent but potentially interesting itemsets.
- If support is too low, the rule may represent rare occurrences, which might not be meaningful or actionable.
 - **Typical Threshold:**
- **5-10%:** Frequently used in retail datasets where only a small fraction of products are bought together.
- **Higher Thresholds (20-30%):** For smaller datasets or when focusing on very popular combinations.

Example:

If only 2 out of 10,000 transactions involve a rule, it's unlikely to be significant. Start with **5-10%** to ensure relevance.

.....

✓ 2.2. Confidence:

Confidence in Apriori is a measure that calculates the **probability** of the rule being true, in other words, it measures the reliability of the rule. Having higher confidence means that when the antecedent occurs, the consequent is likely to follow.

$$\text{Confidence}(X \rightarrow Y) = \frac{\text{Support}(X \cap Y)}{\text{Support}(X)}$$

Valid Values for Confidence

- **Range:** Confidence values range from **0 to 1**:

- **0**: The rule $X \rightarrow Y$ never holds true.
- **1**: The rule $X \rightarrow Y$ always holds true (perfect confidence).

Interpreting Confidence

1. High Confidence (≈ 1):

- Indicates that Y almost always occurs when X occurs.
- Example: If $X = \{\text{bread}\}$ and $Y = \{\text{butter}\}$, a high confidence implies that customers buying bread are very likely to buy butter.

2. Low Confidence (≈ 0):

- Indicates that Y rarely occurs when X occurs.
- Example: If $X = \{\text{bread}\}$ and $Y = \{\text{cereal}\}$, a low confidence implies that buying bread is not strongly associated with buying cereal.

Thresholds in Practice

The minimum values for **confidence** depend on the specific dataset and the business context, but here are some common guidelines and starting points:

- **Typical Threshold:**

- **70-80%**: Commonly used as a starting point for reliable rules.
- Adjust based on your use case:
 - Lower threshold (50-60%) for exploratory insights.
 - Higher threshold (90%+) for high-accuracy recommendations.

- **Reason:** Rules with low confidence may not reliably predict the consequent item, making them less actionable.

Example:

A rule with 30% confidence means that only 30% of the time, the consequent is bought when the antecedent is bought. This may not justify a marketing strategy.

.....

Confidence vs. Other Metrics

- Having **High Confidence** results can lead to misleading rules if Y is very frequent in the dataset. For example, a high-confidence rule might occur simply because Y is common, not because X and Y are strongly related.
- To address this, metrics like **lift** or **conviction** are often used alongside confidence.

.....

✓ 2.3. Lift:

Lift measures the **ratio** or the strength of the association between two items or itemsets.

Specifically, it compares the observed co-occurrence of items to their expected co-occurrence if they were independent.

In other words, lift shows whether the items are occurring together more often than expected by chance. Higher values are better.

$$\text{Lift}(X \rightarrow Y) = \frac{\text{Confidence}(X \rightarrow Y)}{\text{Support}(Y)}$$

$$\text{Lift}(X \rightarrow Y) = \frac{\text{Support}(X \cap Y)}{\text{Support}(X) \times \text{Support}(Y)}$$

Valid Values for Lift

1. Greater than 1:

- Indicates a **positive association** between X and Y .
- X and Y are more likely to occur together than if they were independent.

2. Equal to 1:

- Indicates **no association** between X and Y .
- X and Y occur together just as often as would be expected under independence.

3. Less than 1:

- Indicates a **negative association** between X and Y .
- X and Y are less likely to occur together than if they were independent.

Thresholds in Practice

The minimum values for **lift** depend on the specific dataset and the business context, but here are some common guidelines and starting points:

• Typical Threshold:

- **Lift > 1**: Indicates a positive association between items (start considering rules here).
- **Lift > 3**: Strong and actionable association.
- **Lift = 1**: No association (ignore these rules).
- **Lift < 1**: Negative association (discard unless relevant for avoiding recommendations).

Example:

If Lift = 1.5, buying the antecedent increases the likelihood of buying the consequent by 50%. If Lift = 4, it's a much stronger association.

✓ Why Lift is Useful

Lift helps overcome a major limitation of confidence alone. Consider this scenario: if bread appears in 80% of all transactions, then any rule "X → bread" will have high confidence simply because bread is popular, not because there's a meaningful association. Lift corrects for this by normalizing against the baseline probability of the consequent.

Practical Example Let's say we're analyzing a grocery store dataset with 1,000 transactions: Data:
 Bread appears in 600 transactions (Support = 0.6) Butter appears in 400 transactions (Support = 0.4) Bread and Butter together appear in 300 transactions (Support = 0.3)

For the rule "Bread → Butter":

Confidence = $300/600 = 0.5$ (50%) Lift = $0.5/0.4 = 1.25$

Interpretation: People who buy bread are 25% more likely to also buy butter than the average customer. This suggests a meaningful positive association. Comparison with a weak association: If we had a rule "Bread → Milk" where:

Milk appears in 300 transactions (Support = 0.3) Bread and Milk together appear in 180 transactions

Then:

Confidence = $180/600 = 0.3$ (30%) Lift = $0.3/0.3 = 1.0$

This lift of 1.0 indicates no special association between bread and milk beyond what we'd expect by chance.

.....

✓ Suggested Starting Points for General Use:

- **Support ≥ 5-10%**
- **Confidence ≥ 70-80%**
- **Lift > 1 (preferably > 3 for strong rules)**

.....

✓ Rule Prioritization

To prioritize rules, combine the metrics:

1. High Lift + High Confidence + Acceptable Support: These are the most actionable rules.

- Example: If Lift = 4, Confidence = 85%, and Support = 12%, this rule is strong and impacts many customers.

2. Trade-offs:

- Rules with **low support** but **high confidence and lift** might still be valuable for niche promotions.
- Rules with **high support** but **low lift** may indicate popular combinations that are obvious and less insightful.

✓ Step-by-Step Implementation for the Apriori Algorithm

We will go step-by-step through how the **Apriori algorithm** would process this dataset. We'll compute frequent itemsets and generate rules using the given data, along with thresholds for **support**, **confidence**, and **lift**.

Let's assume a small dataset with 12 transactions:

Transaction ID	Item 1	Item 2	Item 3	Item 4	Item 5
1	Milk	Egg	Bread	Butter	
2	Milk	Butter	Egg	Ketchup	
3	Bread	Butter	Ketchup		
4	Milk	Bread	Butter		
5	Bread	Butter	Cookies		
6	Milk	Bread	Butter	Cookies	
7	Milk	Cookies			
8	Milk	Bread	Butter		
9	Bread	Butter	Egg	Cookies	
10	Milk	Butter	Bread		
11	Milk	Bread	Butter		
12	Milk	Bread	Cookies	Ketchup	

We set out thresholds as following:

- **Minimum Support:** 50% (focus on items present in at least half of the transactions), this value can be set to 5% or 10% in real-life situations with large datasets.
- **Minimum Confidence:** 70%.
- **Lift:** Lift > 1 for actionable rules.

Step 1: Dataset Transformation

Convert the dataset into a binary format:

Transaction	Milk	Egg	Bread	Butter	Ketchup	Cookies
1	1	1	1	1	0	0
2	1	1	0	1	1	0
3	0	0	1	1	1	0
4	1	0	1	1	0	0
5	0	0	1	1	0	1
6	1	0	1	1	0	1
7	1	0	0	0	0	1
8	1	0	1	1	0	0
9	0	1	1	1	0	1
10	1	0	1	1	0	0
11	1	0	1	1	0	0
12	1	0	1	0	1	1

Step 2: Identify Frequent 1-Itemsets

The algorithm starts by calculating the support for each single item.

Item	Support Count	Support (%)	Frequent?
Milk	9	75%	Yes
Egg	3	25%	No
Bread	10	83%	Yes
Butter	10	83%	Yes
Ketchup	3	25%	No
Cookies	5	42%	No

Frequent 1-itemsets: {Milk}, {Bread}, {Butter}.

Step 3: Generate 2-Itemsets

Now, combine frequent 1-itemsets into 2-itemsets and calculate their support.

Itemset	Support Count	Support (%)	Frequent?
{Milk, Bread}	7	58%	Yes
{Milk, Butter}	7	58%	Yes
{Bread, Butter}	9	75%	Yes

Frequent 2-itemsets: {Milk, Bread}, {Milk, Butter}, {Bread, Butter}.

Step 4: Generate 3-Itemsets

Combine frequent 2-itemsets into 3-itemsets and calculate their support.

Itemset	Support Count	Support (%)	Frequent?
{Milk, Bread, Butter}	6	50%	Yes

Frequent 3-itemset: {Milk, Bread, Butter}.

Step 5: No Larger Itemsets Since there are no frequent 4-itemsets (support would drop below 50%), we stop here.

Step 6: Generate Association Rules

Now, use the frequent itemsets to generate association rules and calculate their metrics: **support**, **confidence**, and **lift**.

Rules from {Milk, Bread, Butter}:

1. Rule:

Milk, Bread \rightarrow Butter

- **Support:**

$$P(\text{Milk, Bread, Butter}) = 6/12 = 50\%$$

- **Confidence:**

$$P(\text{Butter}|\text{Milk, Bread}) = \frac{P(\text{Milk, Bread, Butter})}{P(\text{Milk, Bread})} = \frac{6}{7} = 85$$

- **Lift:**

$$\text{Lift} = \frac{P(\text{Butter}|\text{Milk, Bread})}{P(\text{Butter})} = \frac{0.85}{0.83} \approx 1.02$$

2. Rule:

Milk, Butter \rightarrow Bread

- **Support:**

$$P(\text{Milk, Bread, Butter}) = 6/12 = 50\%$$

- **Confidence:**

$$P(\text{Bread}|\text{Milk, Butter}) = \frac{P(\text{Milk, Bread, Butter})}{P(\text{Milk, Butter})} = \frac{6}{7} = 85$$

- **Lift:**

$$\text{Lift} = \frac{P(\text{Bread}|\text{Milk, Butter})}{P(\text{Bread})} = \frac{0.85}{0.83} \approx 1.02$$

3. Rule:

Bread, Butter \rightarrow Milk

- **Support:**

$$P(\text{Milk, Bread, Butter}) = 6/12 = 50\%$$

- **Confidence:**

$$P(\text{Milk}|\text{Bread, Butter}) = \frac{P(\text{Milk, Bread, Butter})}{P(\text{Bread, Butter})} = \frac{6}{9} \approx 66$$

- **Lift:**

$$\text{Lift} = \frac{P(\text{Milk}|\text{Bread, Butter})}{P(\text{Milk})} = \frac{0.66}{0.75} \approx .88$$

✓ Final Results

Frequent Itemsets:

- **1-itemsets:** {Milk}, {Bread}, {Butter}
- **2-itemsets:** {Milk, Bread}, {Milk, Butter}, {Bread, Butter}
- **3-itemset:** {Milk, Bread, Butter}

Rules: | Rule | Support | Confidence | Lift | Actionable? | |-----|-----|-----|-----|-----|

Milk, Bread → Butter

| 50% | 85% | 1.02 | Yes | |

Milk, Butter → Bread

| 50% | 85% | 1.02 | Yes | |

Bread, Butter → Milk

| 50% | 66% | .88 | No |

Interpretation and Actionable Insights | Rule | Explanation | |-----|-----|

Milk, Bread → Butter

| Customers buying Milk and Bread are highly likely (85%) to also buy Butter. Consider bundling these items. | |

Milk, Butter → Bread

| Strong association; placing these items together could increase sales. | |

Bread, Butter → Milk

| Suggests that Milk is a complementary product to Bread and Butter, Weak lift thought |

✓ Python Code Example

We will implement the **Apriori algorithm** using the `m1xtend` library for frequent itemset mining and rule generation.

a. Install Required Libraries Make sure you have the required libraries installed:

```
pip install pandas mlxtend
```

b. Load the Dataset We will represent the dataset as a **binary transaction matrix** where each row is a transaction, and each column is an item (1 = purchased, 0 = not purchased).

```
1 import pandas as pd
2
3 # Define the dataset
4 data = {
5     "Milk":    [1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1],
6     "Egg":     [1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
7     "Bread":   [1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1],
8     "Butter":  [1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0],
9     "Ketchup": [0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1],
10    "Cookies": [0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1],
11 }
12
13 # Create a DataFrame
14 df = pd.DataFrame(data)
15 print("Transaction Dataset:")
16 df
```



Transaction Dataset:

	Milk	Egg	Bread	Butter	Ketchup	Cookies
0	1	1	1	1	0	0
1	1	1	0	1	1	0
2	0	0	1	1	1	0
3	1	0	1	1	0	0
4	0	0	1	1	0	1
5	1	0	1	1	0	1
6	1	0	0	0	0	1
7	1	0	1	1	0	0
8	0	1	1	1	0	1
9	1	0	1	1	0	0
10	1	0	1	1	0	0
11	1	0	1	0	1	1

c. Apply the Apriori Algorithm We will use the `mlxtend` library to find frequent itemsets and generate association rules.

```
1 ! pip install mlxtend
```



Collecting mlxtend

```
Downloading mlxtend-0.23.4-py3-none-any.whl.metadata (7.3 kB)
Requirement already satisfied: scipy>=1.2.1 in /home/me/myenv/lib/python3.12/site-packages (1.14.0)
Requirement already satisfied: numpy>=1.16.2 in /home/me/myenv/lib/python3.12/site-packages (2.0.2)
Requirement already satisfied: pandas>=0.24.2 in /home/me/myenv/lib/python3.12/site-packages (2.2.3)
Requirement already satisfied: scikit-learn>=1.3.1 in /home/me/myenv/lib/python3.12/site-packages (1.6.0)
Requirement already satisfied: matplotlib>=3.0.0 in /home/me/myenv/lib/python3.12/site-packages (3.9.2)
Requirement already satisfied: joblib>=0.13.2 in /home/me/myenv/lib/python3.12/site-packages (1.4.2)
Requirement already satisfied: contourpy>=1.0.1 in /home/me/myenv/lib/python3.12/site-packages (1.3.0)
Requirement already satisfied: cycler>=0.10 in /home/me/myenv/lib/python3.12/site-packages (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /home/me/myenv/lib/python3.12/site-packages (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /home/me/myenv/lib/python3.12/site-packages (1.4.7)
Requirement already satisfied: packaging>=20.0 in /home/me/myenv/lib/python3.12/site-packages (25.0)
Requirement already satisfied: pillow>=8 in /home/me/myenv/lib/python3.12/site-packages (11.0.0)
Requirement already satisfied: pyparsing>=2.3.1 in /home/me/myenv/lib/python3.12/site-packages (3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in /home/me/myenv/lib/python3.12/site-packages (2.9.0)
Requirement already satisfied: pytz>=2020.1 in /home/me/myenv/lib/python3.12/site-packages (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /home/me/myenv/lib/python3.12/site-packages (2024.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /home/me/myenv/lib/python3.12/site-packages (3.5.0)
Requirement already satisfied: six>=1.5 in /home/me/myenv/lib/python3.12/site-packages (1.17.0)
Downloading mlxtend-0.23.4-py3-none-any.whl (1.4 MB)
1.4/1.4 MB 5.5 MB/s eta 0:00:00[36m0:00:01m
Installing collected packages: mlxtend
Successfully installed mlxtend-0.23.4
```

```
1 from mlxtend.frequent_patterns import apriori, association_rules
2
3 # Step 1: Generate frequent itemsets with Apriori
4 frequent_itemsets = apriori(df, min_support=0.5, use_colnames=True)
5
6 # Display frequent itemsets
7 print("\nFrequent Itemsets:")
8 print(frequent_itemsets)
```



```
Frequent Itemsets:
   support  itemsets
0  0.750000  (Milk)
1  0.833333  (Bread)
2  0.833333  (Butter)
3  0.583333  (Bread, Milk)
4  0.583333  (Milk, Butter)
5  0.750000  (Bread, Butter)
6  0.500000  (Bread, Milk, Butter)
/home/me/myenv/lib/python3.12/site-packages/mlxtend/frequent_patterns/fpcommon.py:161: [
warnings.warn(
```

d. Generate Association Rules We generate rules based on **confidence** and calculate **lift**.

Generate rules based on confidence

```
1 # Step 2: Generate association rules based on confidence
2 rules = association_rules(frequent_itemsets, num_itemsets=len(df), metric="confidence",
3
4 # Display the rules
5 print("\nAssociation Rules:")
6 print(rules[['antecedents', 'consequents', 'support', 'confidence', 'lift']])
```



```
Association Rules:
   antecedents consequents  support  confidence    lift
0      (Bread)      (Milk)  0.583333    0.700000  0.933333
1      (Milk)      (Bread)  0.583333    0.777778  0.933333
2    (Butter)      (Milk)  0.583333    0.700000  0.933333
3      (Milk)    (Butter)  0.583333    0.777778  0.933333
4    (Butter)      (Bread)  0.750000    0.900000  1.080000
5      (Bread)    (Butter)  0.750000    0.900000  1.080000
6 (Milk, Bread)    (Butter)  0.500000    0.857143  1.028571
7 (Butter, Milk)      (Bread)  0.500000    0.857143  1.028571
```

We can also generate rules based on lift

```
1 # Step 2: Generate association rules based on lift
2 rules = association_rules(frequent_itemsets, num_itemsets=len(df), metric="lift", min_lift=1
3
4 # Display the rules
5 print("\nAssociation Rules:")
6 print(rules[['antecedents', 'consequents', 'support', 'confidence', 'lift']])
```



```
Association Rules:
   antecedents consequents  support  confidence    lift
0    (Butter)      (Bread)    0.75    0.900000  1.080000
1    (Bread)      (Butter)    0.75    0.900000  1.080000
2 (Milk, Bread)    (Butter)    0.50    0.857143  1.028571
3 (Butter, Milk)      (Bread)    0.50    0.857143  1.028571
4      (Bread) (Butter, Milk)    0.50    0.600000  1.028571
5    (Butter) (Milk, Bread)    0.50    0.600000  1.028571
```

e. Filter and Interpret Rules You can filter rules for high lift or specific itemsets.

```
1 # Filter rules with Lift > 1
2 filtered_rules = rules[rules['lift'] > 1]
```

```
3 print("\nFiltered Rules with Lift > 1:")
4 print(filtered_rules[['antecedents', 'consequents', 'support', 'confidence', 'lift']])
```



Filtered Rules with Lift > 1:

	antecedents	consequents	support	confidence	lift
0	(Butter)	(Bread)	0.75	0.900000	1.080000
1	(Bread)	(Butter)	0.75	0.900000	1.080000
2	(Milk, Bread)	(Butter)	0.50	0.857143	1.028571
3	(Butter, Milk)	(Bread)	0.50	0.857143	1.028571
4	(Bread)	(Butter, Milk)	0.50	0.600000	1.028571
5	(Butter)	(Milk, Bread)	0.50	0.600000	1.028571

Interpretation

1. **Milk, Bread \rightarrow Butter**
 - Support = 58%, Confidence = 87.5%, Lift = 1.05.
 - Suggest bundling these items for promotions.
2. **Milk, Butter \rightarrow Bread**
 - Similar metrics as above, showing strong relationships.
3. **Bread, Butter \rightarrow Milk**
 - High confidence but slightly weaker lift, still actionable.

✓ Two Items Rules

If desired, association rules at the 1-item values of X could be specified. Consider the rule: Milk \rightarrow Bread.

Out of total 12 transactions Milk occurs 9 times while (Milk, Bread) occurs 7 times.

The rule has a support level of 7/12 (or 58 percent) and a confidence level of 7/9 (or 77 percent).

Thus, the next valid association rule is as follows:

Milk \rightarrow Bread {S = 58%, C = 77%}.

2-Item Rules vs 3-Items Rules 2-Item Rules

- **Simplicity:** Easier to interpret and implement in business decisions.
- **Higher Support and Confidence:** More likely to appear frequently in transactions due to fewer constraints.
- **Broad Applicability:** Useful for general trends, such as Milk \rightarrow Bread, which can inform store layout or promotions.

3-Item Rules

- **Deeper Insights:** Reveal more specific patterns, such as Milk, Bread \rightarrow Butter, which may not be evident in 2-item rules.
- **Lower Support:** Less likely to occur because all three items must appear together.
- **Targeted Application:** Better suited for niche marketing strategies or tailored product bundles.

Which is Better?

- **2-item rules** are ideal for identifying general trends and implementing broader strategies.
- **3-item rules** are more appropriate for uncovering specific patterns and enabling targeted interventions.

The choice between the two depends on the dataset's characteristics and the business objectives.

Association Rule Use Cases and Domains

Association rule mining can be applied in various domains beyond market basket analysis. Here are some examples:

1. Healthcare

- **Disease Diagnosis:** Identifying associations between symptoms and diseases. For example, a rule like $\{fever, cough\} \rightarrow \{flu\}$ can help predict diseases.
- **Drug Interactions:** Discovering relationships between medications that are frequently prescribed together or identifying combinations that lead to adverse reactions.

2. Web Usage Mining

- **Website Optimization:** Analyzing user navigation patterns to determine common paths or clicks, e.g., $\{homepage \rightarrow product\ page\} \rightarrow \{checkout\}$.
- **Recommendation Systems:** Suggesting content or products based on frequently co-accessed items, e.g., $\{clicked\ 'smartphone'\} \rightarrow \{clicked\ 'smartphone\ accessories'\}$.

3. Education

- **Student Behavior Analysis:** Discovering patterns in course enrollment, such as $\{math101, cs101\} \rightarrow \{stat101\}$.
- **Learning Paths:** Identifying sequences of topics that students study, helping design better curricula.

4. Telecommunications

- **Customer Churn Analysis:** Detecting combinations of usage patterns that are associated with customers leaving the service, e.g., $\{low\ data\ usage, few\ calls\} \rightarrow$

{churn}.

- **Service Bundling:** Identifying services that are commonly purchased together, like *{broadband, mobile} → {TV subscription}*.

5. Banking and Finance

- **Fraud Detection:** Uncovering patterns associated with fraudulent transactions, e.g., *{high transaction frequency, odd hours} → {fraud}*.
- **Loan Approvals:** Identifying attributes of successful loan applications, such as *{high income, good credit score} → {loan approved}*.

6. Manufacturing

- **Fault Detection:** Identifying combinations of machine conditions that frequently result in faults, e.g., *{high temperature, low pressure} → {equipment failure}*.
- **Supply Chain Optimization:** Discovering patterns in material usage, e.g., *{material A, material B} → {product C}*.

7. Retail Beyond Market Basket

- **Shelf Placement:** Finding products that are often bought together to optimize store layout.
- **Customer Segmentation:** Identifying customer groups with similar purchasing behaviors, e.g., *{frequent discount purchases} → {low brand loyalty}*.

8. Social Media Analysis

- **Trending Topics:** Discovering associations between hashtags, e.g., *{#climatechange, #sustainability} → {#renewableenergy}*.
- **User Behavior Patterns:** Understanding engagement behaviors, such as *{likes post, comments on post} → {shares post}*.

9. Energy Sector

- **Usage Patterns:** Identifying associations in energy usage, like *{high A/C usage, weekend} → {peak energy consumption}*.