Introduction to Association Rule Mining - Market Baset 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

للاستفسار فرعي 108 - 06/5814168
==========
رقم الفاتورة 62465 الرقم الضريبي :16650476
==========
كاشير 6
التاريخ 28/05/2025 الساعة : 3:35:00 pm

ف المادة	الكمية	السعر	المجموع
نبيل دجاج تندر 900 غم	1	3.750	3.750
نبيل اصابع سمك مجمد 40	2	2.390	4.780
خضار مشكلة السنبلة 450	2	1.600	3.200
السنبلة بامية اكسترا 400	2	2.590	5.180
سبانخ السنبلة 400 غم	2	0.900	1.800
جبنة فاكيوم روابي جرش (	1	3.550	3.550
مكدوس بلدي التميمي 1 كغ	1	3.650	3.650
برداريس خبز طابون *4	1	0.600	0.600
غيث خبز مسخن *5	1	0.850	0.850
مكلل هلابينو	0.494	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 متبل لـ	0.98	3.250	3.185

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

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

للاستفسار فرعي 108 - 06/5814168
==========
رقم الفاتورة 62464 الرقم الضريبي :16650476
==========
كاشير 6
التاريخ 28/05/2025 الساعة : 3:34:00 pm

ف المادة	الكمية	السعر	المجموع
ارم الند همر الدقاس	1	1.350	1.350
ارم الند همر معجون اسنان	1	1.350	1.350
معجون اورل بي 123	1	1.390	1.390
المراعي لبنة 500 غم	1	2.250	2.250
صابون حمام نابلسي حبة و	1	0.550	0.550
عصير الربيع الناس 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 لاتيكس اسود وسط*3	1	2.490	2.490
تشيروز عسل 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
سانيتو معجون اسنان الحسـ	1	0.990	0.990
جبنة برمزان ايطالي	0.166	12.990	2.156
جبنة برمزان ايطالي	0.354	12.990	4.598
فلفل قرن الغزال	0.552	0.650	0.359
بندورة عناقيد كبيرة	0.7	1.100	0.770
مكلل هلابينو	0.284	1.990	0.565
خيار	0.44	0.790	0.348

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

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

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} → {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 → product page} → {checkout}*.
- **Recommendation Systems:** Suggesting content or products based on frequently co-accessed items, e.g., *{clicked 'smartphone'} → {clicked 'smartphone accessories'}*.

3. **Education**

- **Student Behavior Analysis:** Discovering patterns in course enrollment, such as *{math101, cs101} → {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} → {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}*.
- **Smart Grid Analysis:** Detecting patterns for predictive maintenance, e.g., *{low voltage, high demand} → {power outage}*.

10. **Transportation and Logistics**

- **Traffic Analysis:** Discovering patterns in traffic conditions, e.g., *{morning rush hour, bad weather} → {traffic jam}*.
- **Route Optimization:** Identifying frequently used delivery routes, such as *{route A, route B} → {delivered faster}*.

11. **E-commerce**

- **User Preferences:** Identifying patterns in user preferences for personalized recommendations.
- **Cross-Selling:** Suggesting related products based on purchase history.

12. **Sports Analytics**

- **Performance Metrics:** Discovering combinations of player actions that lead to victories, e.g., *{high possession, accurate passes} → {win}*.
- **Injury Prevention:** Identifying conditions that precede injuries, such as *{high training load, lack of rest} → {injury risk}*.

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.

.....

Creating Association Rules

To create association rules, we use 3 metrics:

1. Support: finds the popular items and builds initial rules.
2. Confidence: Enhance initial rules by finding true relationships.
3. Lift: Enhance rules by acomodating for popular items effect (x -> y where x is popular e.g. bread in a bakery)

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.

$$\text{Support}(\text{X}) = \frac{\text{Number of transactions containing X}}{\text{Total transactions}}$$

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

Let's assume that we have 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	

First, we set out the **Minimum Support** value to:

- 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.

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

Itemset	Support Count	Support (%)	Frequent?
{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 Can Be Created
Since there are no frequent 4-itemsets (support would drop below 50%), we stop here.

Step 6: Generate Initial Association Rules

Now, use the frequent itemsets to generate association rules, we will focus on the largest item set, we can create rules from smaller ones if needed.

Rules from `{Milk, Bread, Butter}`:

1. **Rule One**:

$$\text{Milk, Bread} \rightarrow \text{Butter}$$

- **Support**:

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

2. **Rule Two**:

$$\text{Milk, Butter} \rightarrow \text{Bread}$$

- **Support**:

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

3. **Rule Three**:

$$\text{Bread, Butter} \rightarrow \text{Milk}$$

- **Support**:

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

Notice that the support for all values is the same, now we need to use **confidence and lift** to enhance these initial rules.

2 Confidence:

In order to validate which rule is correct, we need to examine the confidence of the rule, for example, we might have a frequent item set that have two items (Milk, Bread), we can establish two rules from that set:

$$\text{Milk} \rightarrow \text{Bread}$$

$$\text{Bread} \rightarrow \text{Milk}$$

To determine which rule is correct for that frequenct item set (the rule that says when x occurs y also most probably occurs), we can use to confidence establish that assertion.

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.

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

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.

Step 7: Validate the Rules using Confidence

First, let us set the **Minimum Confidence** value to: 70%.

Rules from `{Milk, Bread, Butter}`:

1. **Rule One**:

$$\text{Milk, Bread} \rightarrow \text{Butter}$$

- **Support**:

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

- **Confidence**:

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

2. **Rule Two**:

$$\text{Milk, Butter} \rightarrow \text{Bread}$$

- **Support**:

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

- **Confidence**:

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

3. **Rule Three**:

$$\text{Bread, Butter} \rightarrow \text{Milk}$$

- **Support**:

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

- **Confidence**:

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

2.3. Lift

Confidence alone can be misleading, especially when the consequent (Y) is very frequent in the dataset. In that case, a rule (X \Rightarrow Y) may have high confidence simply because (Y) is common, not because there is a strong dependency between (X) and (Y).

Example (bakery: cake and bread) Suppose we analyze orders in a bakery and find:

- 80% of all orders contain **bread.**
- 20% of all orders contain **cake**.
- 16% of all orders contain **both cake and bread.**

Then the confidence of the rule $\text{cake} \Rightarrow \text{bread}$ is:
$\text{conf}(\text{cake} \Rightarrow \text{bread}) = \frac{P(\text{cake} \cap \text{bread})}{P(\text{cake})} = \frac{0.16}{0.20} = 0.8.$

A confidence of **80%** sounds strong, but notice that **bread is already in 80% of all orders**. So customers who buy cake are **not more likely** to buy bread than an average customer—bread is just very common in the bakery.

To address this, we use additional measures such as **lift** alongside confidence.

Lift compares $P(Y \mid X)$ to the overall frequency $P(Y)$:
$\text{Lift}(X \Rightarrow Y) = \frac{\text{Confidence}(X \Rightarrow Y)}{\text{Support}(Y)} = \frac{P(Y|X)}{P(Y)}.$

Equivalently, in terms of supports:

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

In the example:
$\text{lift}(\text{cake} \Rightarrow \text{bread}) = \frac{P(\text{bread}|\text{cake})}{P(\text{bread})} = \frac{0.8}{0.8} = 1,$

which indicates **no real association** beyond what we'd expect by chance.

Lift measures how strongly two items or itemsets are associated:

- $(\text{lift} > 1)$: items occur together **more often than expected** (positive association).
- $(\text{lift} = 1)$: items occur together **as expected** under independence (no association).
- $(\text{lift} < 1)$: items occur together **less often than expected** (negative association).

So lift tells us whether items co-occur more (or less) frequently than chance, making it a useful complement to confidence.

Interpreting Lift Values

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.

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.

Step 8: Validate Rules using the Lift

First, we set the **Lift > 1** for actionable rule

1. **Rule One**:

$$\text{Milk, Bread} \rightarrow \text{Butter}$$

   - **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 Two**:

$$\text{Milk, Butter} \rightarrow \text{Bread}$$

   - **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 Three**:

$$\text{Bread, Butter} \rightarrow \text{Milk}$$

- **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?
$\text{Milk, Bread} \rightarrow \text{Butter}$	50%	85%	1.02	Yes
$\text{Milk, Butter} \rightarrow \text{Bread}$	50%	85%	1.02	Yes
$\text{Bread, Butter} \rightarrow \text{Milk}$	50%	66%	.88	No

Interpretation and Actionable Insights

Rule	Explaniation
$\text{Milk, Bread} \rightarrow \text{Butter}$	Customers buying Milk and Bread are highly likely (85%) to also buy Butter. Consider bundling these items.
$\text{Milk, Butter} \rightarrow \text{Bread}$	Strong association; placing these items together could increase sales.
$\text{Bread, Butter} \rightarrow \text{Milk}$	Suggests that Milk is a complementary product to Bread and Butter, Weak lift thought

---

Python Example

We will implement the **Apriori algorithm** using the `mlxtend` 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).

In [1]:
```
import pandas as pd

# Define the dataset
data = {
    "Milk":    [1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1],
    "Egg":     [1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0],
    "Bread":   [1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1],
    "Butter":  [1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 0],
    "Ketchup": [0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1],
    "Cookies": [0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1],
}

# Create a DataFrame
df = pd.DataFrame(data)
print("Transaction Dataset:")
df
```

```
Transaction Dataset:
```

Out[1]:

	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.

In [3]:
```
! 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-pa
ckages (from mlxtend) (1.13.1)
Requirement already satisfied: numpy>=1.16.2 in /home/me/myenv/lib/python3.12/site-p
ackages (from mlxtend) (1.26.4)
Requirement already satisfied: pandas>=0.24.2 in /home/me/myenv/lib/python3.12/site-
packages (from mlxtend) (2.2.3)
Requirement already satisfied: scikit-learn>=1.3.1 in /home/me/myenv/lib/python3.12/
site-packages (from mlxtend) (1.5.2)
Requirement already satisfied: matplotlib>=3.0.0 in /home/me/myenv/lib/python3.12/si
te-packages (from mlxtend) (3.9.2)
Requirement already satisfied: joblib>=0.13.2 in /home/me/myenv/lib/python3.12/site-
packages (from mlxtend) (1.4.2)
Requirement already satisfied: contourpy>=1.0.1 in /home/me/myenv/lib/python3.12/sit
e-packages (from matplotlib>=3.0.0->mlxtend) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /home/me/myenv/lib/python3.12/site-pa
ckages (from matplotlib>=3.0.0->mlxtend) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /home/me/myenv/lib/python3.12/si
te-packages (from matplotlib>=3.0.0->mlxtend) (4.54.1)
Requirement already satisfied: kiwisolver>=1.3.1 in /home/me/myenv/lib/python3.12/si
te-packages (from matplotlib>=3.0.0->mlxtend) (1.4.7)
Requirement already satisfied: packaging>=20.0 in /home/me/myenv/lib/python3.12/site
-packages (from matplotlib>=3.0.0->mlxtend) (24.2)
Requirement already satisfied: pillow>=8 in /home/me/myenv/lib/python3.12/site-packa
ges (from matplotlib>=3.0.0->mlxtend) (11.0.0)
Requirement already satisfied: pyparsing>=2.3.1 in /home/me/myenv/lib/python3.12/sit
e-packages (from matplotlib>=3.0.0->mlxtend) (3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in /home/me/myenv/lib/python3.1
2/site-packages (from matplotlib>=3.0.0->mlxtend) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /home/me/myenv/lib/python3.12/site-pa
ckages (from pandas>=0.24.2->mlxtend) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /home/me/myenv/lib/python3.12/site-
packages (from pandas>=0.24.2->mlxtend) (2024.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /home/me/myenv/lib/python3.1
2/site-packages (from scikit-learn>=1.3.1->mlxtend) (3.5.0)
Requirement already satisfied: six>=1.5 in /home/me/myenv/lib/python3.12/site-packag
es (from python-dateutil>=2.7->matplotlib>=3.0.0->mlxtend) (1.16.0)
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 eta 0:00:01
Installing collected packages: mlxtend
Successfully installed mlxtend-0.23.4
```

In [4]:
```python
from mlxtend.frequent_patterns import apriori, association_rules

# Step 1: Generate frequent itemsets with Apriori
frequent_itemsets = apriori(df, min_support=0.5, use_colnames=True)

# Display frequent itemsets
print("\nFrequent Itemsets:")
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: DeprecationWarning: DataFrames with non-bool types result in worse computationalperformance and their support might be discontinued in the future.Please use a DataFrame with bool type
  warnings.warn(
```

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

Generate rules based on confidence

In [ ]:
```
# Step 2: Generate association rules based on confidence
rules = association_rules(frequent_itemsets, num_itemsets=len(df), metric="confiden

# Display the rules
print("\nAssociation Rules:")
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

In [ ]:
```
# Step 2: Generate association rules based on lift
rules = association_rules(frequent_itemsets, num_itemsets=len(df), metric="lift", m

# Display the rules
print("\nAssociation Rules:")
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.

In [ ]:
```
# Filter rules with Lift > 1
filtered_rules = rules[rules['lift'] > 1]
print("\nFiltered Rules with Lift > 1:")
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. $\text{Milk, Bread} \rightarrow \text{Butter}$
   - Support = 58%, Confidence = 87.5%, Lift = 1.05.
   - Suggest bundling these items for promotions.
2. $\text{Milk, Butter} \rightarrow \text{Bread}$
   - Similar metrics as above, showing strong relationships.
3. $\text{Bread, Butter} \rightarrow \text{Milk}$
   - High confidence but slightly weaker lift, still actionable.

Two-Item Rules

If desired, we can focus on association rules where the antecedent (X) contains just **one item** (i.e., 1-item antecedent, 2 items in the rule overall).

Consider the rule:

$$\text{Milk} \rightarrow \text{Bread}$$

Suppose we have **12 transactions**, and we observe:

- **Milk** appears in **9** transactions.
- **Milk and Bread together** appear in **7** transactions.

Then:

- **Support** of the rule is the fraction of all transactions that contain both Milk and Bread: $\text{supp}(\text{Milk} \rightarrow \text{Bread}) = \frac{7}{12} \approx 58$
- **Confidence** of the rule is the fraction of Milk-transactions that also contain Bread: $\text{conf}(\text{Milk} \rightarrow \text{Bread}) = \frac{7}{9} \approx 78$

So we can write the rule as:

$\text{Milk} \rightarrow \text{Bread}$ {S = 58%, C = 78%}.

2-Item Rules vs 3-Item Rules

2-Item Rules (e.g., $\text{Milk} \rightarrow \text{Bread}$)

- **Simplicity**: Easy to read, explain, and act on.
- **Higher support and confidence (typically)**: Fewer items need to co-occur, so these rules tend to appear more often in the data.
- **Broad applicability**: Capture general trends such as $\text{Milk} \rightarrow \text{Bread}$, which can inform store layout (placing items nearby) or simple promotions.

3-Item Rules (e.g., $\text{Milk}, \text{Bread} \rightarrow \text{Butter}$)

- **Deeper insights**: Reveal more specific patterns, like $\text{Milk}, \text{Bread} \rightarrow \text{Butter}$ that might not be visible in 2-item rules.
- **Lower support (usually)**: All three items must appear together, so these combinations are naturally less frequent.
- **Targeted application**: Well-suited for niche marketing, personalized recommendations, or special bundle offers.

Which Is Better?

- Use **2-item rules** when you want:
  - General trends,
  - Simple explanations,
  - Broad business strategies (e.g., product placement, popular combos).
- Use **3-item rules** when you want:
  - More specific patterns,
  - Finer-grained understanding of customer behavior,
  - Targeted campaigns or tailored bundles.

There is no universally “better” rule size—the choice depends on your **data characteristics** (how dense/sparse the transactions are) and your **business objectives** (broad policies vs. targeted actions).