**Association Rule Mining & Market Basket Analysis**

Association Rule Mining is a powerful data mining technique that uncovers datasets' patterns and relationships. This technique has broad applications across various industries, enabling businesses to make informed decisions by understanding the relationships between different items or events.

In this tutorial, we will explore how Association Rule Mining can help in different industries. Then, we will focus on using Market Basket Analysis, a specific implementation of Association Rule Mining, to analyse retail data.

Applications of Association Rule Mining in Different Industries

Market Basket Analysis and Association Rule Mining have various applications across industries. Below are some notable use cases:

**1. Retail Industry**

**Use Case**: *Market Basket Analysis -* Identifies products frequently purchased together **(e.g., "Bread → Butter").** Helps in:

* + - Optimising store layouts (placing related items closer).
    - Planning promotions and cross-selling campaigns.
    - Designing combo offers to increase sales.

**2. Healthcare**

**Use Case**: *Medical Diagnosis and Treatment* - Discovers relationships between symptoms and diseases **(e.g., "High Fever & Rash → Measles").**

Assists in:

* + - Identifying comorbidities.
    - Developing personalised treatment plans.
    - Enhancing disease prevention strategies.

**3. E-commerce**

**Use Case**: *Product Recommendations* - Suggests additional items based on purchase history (**e.g., "Laptop → Mouse & Keyboard").**

Improves:

* + - Customer experience with personalised recommendations.
    - Average order value through upselling and cross-selling.

**4. Banking**

**Use Case**: *Fraud Detection* - Detects unusual transaction patterns **(e.g., "High Transaction Amounts & Foreign Locations → Fraud").**

Benefits include:

* + - Strengthening fraud prevention mechanisms.
    - Reducing risks and improving customer trust.

**5. Telecommunications**

**Use Case**: *Churn Analysis* - Identifies patterns leading to customer churn (e.g., "High Data Usage & Frequent Complaints → Plan Change or Churn").

Enables:

* + - Designing targeted retention strategies.
    - Offering customized service plans.

**7. Education**

**Use Case**: *Curriculum Optimization* - Identifies complementary subjects frequently enrolled together **(e.g., "Data Science → Machine Learning")**.

Supports:

* + - Designing bundled programs.
    - Targeting resources to student needs.

**8. Logistics**

**Use Case**: *Demand analysis* - Finds relationships between product demand and other factors **(e.g., "Rainy Season → High Umbrella Sales")**.

Benefits include:

* + - Inventory optimization.
    - Streamlining supply chain management.

**9. Grocery Stores**

**Use Case**: *Sales Insights* - Reveals customer purchasing behaviour & trends **(e.g., "Weekend → Snacks & Beverages").**

Allows for:

* + - Tailoring promotions for peak times.
    - Enhancing customer satisfaction with targeted offers.

**10. Food Delivery Services**

**Use Case**: *Order Customization* - Identifies popular meal combinations (e.g., "Pizza → Soft Drink & Dessert").

Boosts:

* + - Customer satisfaction through combo deals.
    - Revenue through strategic bundling.

Algorithms Used in Association Rule Mining

* Apriori Algorithm
* Frequent Pattern (FP) Growth Algorithm
* Equivalence Class Transformation (ECLAT) Algorithm
* Agrawal, Imielinski, and Swami (AIS) Algorithm

In this tutorial, we're going to focus on the **Apriori Algorithm**. Let's dive into the details:

Apriori Algorithm for MBA

The Apriori algorithm is a foundational data mining technique used to identify frequent itemsets and generate association rules in a dataset. It operates on the principle that all non-empty subsets of a frequent itemset must also be frequent. Here's an explanation of how it works:

1. **Generate Candidate Itemsets:** The algorithm starts by generating candidate itemsets of size 1. It then scans the dataset to count the support of each itemset and removes those that do not meet *the minimum support threshold*.
2. **Generate Larger Itemsets:** Using the frequent itemsets of size 1, the algorithm generates candidate itemsets of size 2, then size 3, and so on. For each size, it scans the dataset to count support and filters out infrequent itemsets.
3. **Generate Association Rules:** Once all frequent itemsets are identified, the algorithm generates association rules from these itemsets. Each rule is evaluated based on *its confidence and lift*, and only those that meet the minimum thresholds are retained.

Key Metrics in Association rule Mining

We're focusing on the key metrics because they have the most impact on the results:

1. **Support:**

**Definition:** Support measures how frequently an itemset appears in the dataset.

**Impact and interpretation:** If item A appears in 30 out of 100 transactions, the support is 0.3 or 30%. Support is used as a measure of significance (importance) of an itemset. Since it uses the count of transactions, it is often called a frequency constraint. An itemset with support greater than a set minimum support threshold is called a frequent or large itemset.

**Range:** [0,1]

1. **Confidence:**

**Definition:** Confidence measures the likelihood that an item B is purchased when item A is purchased.

**Impact and interpretation:** If 20 out of the 30 transactions containing item A also contain item B, the confidence is 0.67 or 67%. Association rules have to satisfy a minimum confidence constraint. Support is first used to find frequent (significant) itemsets exploiting its downward closure property to prune the search space. Then, confidence is used in a second step to produce rules from the frequent itemsets that exceed a min confidence threshold.

**Range:** [0,1]

1. **Lift:**

**Definition:** Lift measures how much more likely items A and B are to be purchased together than expected if they were independent.

**Impact and interpretation:** If the support of item B is 0.2 and the confidence is 0.67, the lift is 3.35.A lift ratio of 1 means that bundling A and B does not add value to the business as if they are sold individually. However, a lift larger than 1.0 implies that linking the antecedent (A) and the consequent (B) in a rule (i.e. promotion or location) is more significant than having the two items independently. The larger the lift ratio, the more significant the association. Less than one implies a negative association. Item B is unlikely to be bought together with item A if bought.

**Range:**[0,∞] (1 means independence)

Supporting Metrics in Association rule Mining

1. **Leverage**

**Definition and use:**Leverage measures the difference between the observed frequency of the itemset and the expected frequency if the items were independent. One first can use an algorithm to find all itemsets with min. support of 0.01% and then filter the found item sets using the leverage constraint

**Impact and interpretation:** Higher leverage indicate a stronger association, useful for guiding marketing strategies. The rationale in a sales setting is to find out how many more units (items A and B together) are sold than expected from the independent sells (A alone) and (B alone).

**Range:** [−1,1] (0 indicates independence)

1. **Conviction**

**Definition:** Conviction measures the ratio of the expected frequency of the antecedent occurring without the consequent to the observed frequency of the antecedent occurring with the consequent.

**Impact and interpretation:** Higher conviction values indicate stronger rules, useful for confident decision-making in promotions. Conviction compares the probability that A appears without B if they were dependent on the actual frequency of the appearance of A without B. In that respect, it is similar to lift . However, in contrast to lift, it is a directed measure since it also uses the information of the absence of the consequent. An interesting fact is that conviction is monotone in confidence and lift.

**Range:** [0,∞] (1 indicates independence; rules that always hold have ∞)

1. **Jaccard**

**Definition:** Jaccard similarity measures the overlap between the antecedent and consequent.

**Impact:** Higher Jaccard similarity suggests a stronger overlap, useful for decisions on product placement and cross-promotion.

**Range:** [0,1]

1. **Certainty Factor**

**Definition:** Certainty factor quantifies how much more likely the consequent is given the antecedent compared to its baseline likelihood.

**Impact and interpretation:** High certainty factors suggest a robust relationship, helpful in deciding on confident marketing actions. An increasing CF means it is less probable that item B is NOT in a transaction that contains A. Negative CFs have a similar interpretation.

**Range:** [−1,1] (0 indicates independence)

1. **Kulczynski**

**Definition:** Kulczynski measures the average of the two confidence values (antecedent to consequent and consequent to antecedent).

**Impact and interpretation:** It quantifies the strength of the association between two itemsets, A and B. A Kulczynski value closer to 0 or 1 indicates a stronger relationship (positive or negative association), while a value closer to 0.5 suggests a weaker or uninteresting relationship. It helps identify "interesting" rules by considering both directions of the association (A→B and B→A). 0.5 means neutral and typically uninteresting.

**Range:** [0,1] (0.5 means neutral and typically uninteresting)

Applying the Apriori Algorithm with Python

By understanding and applying the Apriori Algorithm, retailers can uncover valuable insights into customer purchasing patterns and optimise their marketing and sales strategies accordingly. These insights can help retailers make informed decisions about product placement, promotions, and inventory management to optimise sales and enhance the customer shopping experience.

The following example demonstrates how to produce a Market Basket Analysis with key and supporting metrics.

*Support* tells us the frequency of itemsets. *Confidence* measures how likely it is that one item will be bought when another item is bought. *Lift* evaluates the strength of the association compared to random chance. By interpreting these metrics, we can gain insights into which items tend to be bought together and uncover potential marketing and inventory management opportunities.

Now that we have explored the fundamentals of association rule mining through a simple example let’s move to the next step: applying the algorithm to a real dataset. In real-world scenarios, datasets come from various sources such as sales transactions, e-commerce logs, or inventory management systems. For educational purposes, we will use a transactional dataset from a supermarket.

**1) Load your Dataset:** We upload the customer shopping transactions to our Google Colab environment using the Pandas package. We have almost 1000 transactions.

import pandas as pd

Dataset = pd.read\_csv('/content/basket\_analysis.csv', index\_col=0)

Dataset

A screenshot of a computer

Description automatically generated

Notice that the dataset is already one-hot-encoded and ready to use by the algorithm, but in a real-world scenario, data preprocessing and transformation are necessary.

**2) Generate the frequent itemsets:** We will use the ***apriori***function from the ***mlxtend***library to find frequent itemsets and generate association rules.

Our dataset consists of 999 transactions. For a dataset of this size, setting a ***min\_support*** threshold that is too high might exclude interesting patterns, while a threshold that is too low could result in too many itemsets, making the analysis computationally intensive.

We aim to capture a meaningful number of itemsets while maintaining computational efficiency. A min\_support threshold of 0.02 means that an itemset must appear in at least 2% of the transactions to be considered frequent. For our dataset, this translates to approximately 20 transactions. It ensures that we do not miss out on significant itemsets that could provide valuable insights for decision-making.

from mlxtend.frequent\_patterns import apriori, association\_rules

# Apply the apriori algorithm to find frequent itemsets

frequent\_itemsets = apriori(Dataset, min\_support=0.02, use\_colnames=True)

# Display the frequent itemsets

frequent\_itemsets

A screenshot of a computer

Description automatically generated

**3) Generate the association rules:** Now we can generate association rules from the frequent itemsets. These rules must meet a minimum lift to demonstrate interestingness.

# Generate association rules from the frequent itemsets

rules = association\_rules(frequent\_itemsets, metric="lift", min\_threshold=1.01, num\_itemsets=len(frequent\_itemsets))

# Calculate itemset size as a total number of items in both antecedents and consequents

rules['itemset\_size'] = rules['antecedents'].apply(len) + rules['consequents'].apply(len)

# Display the association rules

rules.sort\_values(by="lift", ascending=False)

A screenshot of a computer

Description automatically generated

**4) Visualise the strength of association rules:** After we get all the associations meeting the minimum lift with their results, we can visualise the key metrics to better understand and interpret the data. Visualisation helps identify patterns, trends, and insights that might not be immediately apparent from the raw data. Here are some visualisations we can create:

### Support vs Lift by Itemset Size

import matplotlib.pyplot as plt

import seaborn as sns

# Create a scatter plot to visualise the support against lift for the size of itemsets

plt.figure(figsize=(12, 8))

scatter = plt.scatter(

rules['support'],

rules['lift'],

c=rules['itemset\_size'], # Color based on itemset size

cmap='jet', # Color map

alpha=0.2, # Slight transparency

s=50) # Marker size

# Add color bar (legend) with intervals of 1

cbar = plt.colorbar(scatter, label='Itemset Size')

cbar.set\_ticks(list(range(2, rules['itemset\_size'].max() + 1)))

# Labels and title

plt.xlabel('support')

plt.ylabel('lift')

plt.title('Association Rules: Support vs lift, Colored by Itemset Size')

# Add grid for readability

plt.grid(True, linestyle='--', alpha=0.4)

# Show the plot

plt.show()

A graph showing different colored shapes

Description automatically generated with medium confidence

From the previous plot, we observe having 5 groups of rules; these are characterised itemsets sizes increases.

So, as the itemset size increases:

* **Support decreases**: Larger itemsets (containing more items) are less frequent, as it's rarer for multiple specific items to be purchased together.
* **Lift increases**: Larger itemsets often have higher lift, indicating stronger associations among the items within those sets when they do occur.

**5) Filter the association rules:** Let's filter the rules dataset to show the top point with the *highest lift* and *support* for each itemset size (cluster). This way, the business decision will be more specific. We will filter the strongest association rule (with the highest lift) for each group of rules out of the five groups.

### Top itemset by Lift and Support

# Sort by itemset size, then by lift and support

rules\_sorted = rules.sort\_values(by=['itemset\_size', 'lift', 'support'], ascending=[True, False, False])

# Group by itemset size and select the top itemset in each group

top\_rules = rules\_sorted.groupby('itemset\_size').head(1).reset\_index(drop=False)

# Display the filtered rules

top\_rules

A screenshot of a black screen

Description automatically generated

Let's also visualise how the average confidence of association rules varies with the size of the itemsets.

### Average Confidence by Itemset Size

# Group by itemset size and calculate average confidence for each group of item size

avg\_confidence = rules.groupby('itemset\_size')['confidence'].mean().reset\_index()

# Create a line plot for Average Confidence by Itemset Size

plt.figure(figsize=(10, 6))

plt.plot(avg\_confidence['itemset\_size'], avg\_confidence['confidence'], marker='o', linestyle='-', color='b')

# Adding labels and title

plt.xlabel('Itemset Size')

plt.ylabel('Average Confidence')

plt.title('Average Confidence by Itemset Size')

# Add grid for readability

plt.grid(True, linestyle='--', alpha=0.4)

# Show the plot

plt.show()

A graph with a line

Description automatically generated

The graph indicates that as the itemset size decreases, the average confidence tends to increase.

This makes sense because smaller itemsets are likely to have higher confidence levels, as the likelihood of a smaller set of items being bought together is higher compared to larger itemsets.

**6) Summarising insights and making business recommendations:** Here, we analyse each group of rules based on their calculated support, lift and confidence. Let’s look at each group in turns:

* **Association rules with itemset size of 6**

**Support:** Very low (around 0.025).

**Lift:** Ranges from 1.01 to just under 3.5.

**Confidence:** Average confidence is lower compared to smaller itemsets, about 0.26

**Example itemsets and rules:**

Support 0.022, Lift 3.256, Confidence 0.244

**Business Decision:**These rules have a high lift but low support and confidence, indicating that while they are not frequent, they are very strong when they do occur. Focus on *niche marketing* or targeted promotions. For example:

* + Create special bundles that include Sugar, Kidney Beans, Corn, and Unicorn, Apple, Cheese to attract customers interested in diverse meal preparations.
  + Offer exclusive deals for these combinations to targeted segments, perhaps through loyalty programs or personalised offers.
* **Association rules with itemset size of 5**

**Support:** Slightly higher than the red cluster (around 0.025 to 0.05).

**Lift:** Ranges from 1.01 to just under 2.5.

**Confidence:** Moderate average confidence at the lowest

**Example itemsets and rules:**

Support 0.030, Lift 2.357, Confidence 0.441

**Business Decision:** These rules are more frequent than the red cluster but still have a relatively high lift and moderate confidence. Use these associations for more frequent but still targeted marketing campaigns. For instance:

* + Promote breakfast combos that include Unicorn, Apple, Eggs, and Sugar, Corn to cater to morning shoppers.
  + Use these combinations in loyalty programs or personalised email marketing to encourage repeat purchases.
* **Association rules with itemset size of 4**

**Support:** Ranges from 0.025 to 0.075.

**Lift:** Ranges from 1.0 to 2.0.

**Confidence:** Moderately high confidence.

**Example Itemset and rules:**

Support 0.055, Lift 1.826, Confidence 0.311

**Business Decision:** These rules are moderately frequent and have a moderate lift and confidence. Use these associations for regular promotions or bundling strategies. For example:

* + Bundle **Dill, Cheese**, and**Kidney Beans, Onion** together in promotional displays or online banners.
  + Encourage customers to buy these items together by offering slight discounts or special offers when they purchase all four items.
* **Association rules with itemset size of 3**

**Support:** Ranges from 0.075 to 0.125.

**Lift:** Ranges from 1.0 to 1.5.

**Confidence:** High confidence.

**Example Itemset:**

Support 0.102, Lift 1.428, Confidence 0.576

**Business Decision:** These rules are quite frequent and have a high confidence but a lower lift. Use these associations for general promotions or discounts. For instance:

* + Feature Dill, Cheese, and Onion in weekly flyers or general discount campaigns to attract a broader audience.
  + Offer discounts or bundle deals on these items to encourage bulk purchases.
* **Association rules with itemset size of 2**

**Support:** Ranges from 0.125 to 0.2.

**Lift:** Ranges from 1.0 to 1.25.

**Confidence:** Very high confidence.

**Example Itemset:**

Support 0.211, Lift 1.236, Confidence 0.521

**Business Decision:** These rules are the most frequent and have very high confidence but the lowest lift. Use these associations for everyday promotions or standard product placements. For example:

* + Place Milk and chocolate together in high-traffic areas of the store to boost impulse purchases.
  + Feature these items in standard marketing materials and daily deals to maintain steady sales.

**7) Summary of Business Decisions and Recommended Strategies**

* **Niche Marketing:** Focus on rare but strong associations (red cluster) for targeted promotions.
* **Loyalty Programs:** Use moderately frequent itemsets with high lift and moderate confidence (orange cluster) to encourage repeat purchases.
* **Regular Promotions:** Promote moderately frequent itemsets with moderate lift and confidence (green cluster) through regular marketing channels.
* **General Discounts:** Feature frequent itemsets with high confidence but lower lift (blue cluster) in general promotions to attract a broad audience.
* **Standard Placements:** Use very frequent itemsets with very high confidence but the lowest lift (dark blue cluster) for everyday promotions and product placements.