Association rules are "if-then" statements, that help to show the probability of relationships between data items, within large data sets in various types of databases. Association rule mining has a number of applications and is widely used to help discover sales correlations in [transactional data](https://whatis.techtarget.com/definition/transactional-data) or in medical data sets.

Below are a few real-world use cases for association rules:

* **Medicine.**Doctors can use association rules to help diagnose patients. There are many variables to consider when making a diagnosis, as many diseases share symptoms. By using association rules and machine learning-fueled data analysis, doctors can determine the conditional probability of a given illness by comparing symptom relationships in the data from past cases. As new diagnoses get made, the machine learning model can adapt the rules to reflect the updated data.
* **Retail.** Retailers can collect data about purchasing patterns, recording purchase data as item barcodes are scanned by point-of-sale systems. Machine learning models can look for co-occurrence in this data to determine which products are most likely to be purchased together. The retailer can then adjust marketing and sales strategy to take advantage of this information.
* **User experience (UX) design.** Developers can collect data on how consumers use a website they create. They can then use associations in the data to optimize the website user interface -- by analyzing where users tend to click and what maximizes the chance that they engage with a call to action, for example.
* **Entertainment.** Services like Netflix and Spotify can use association rules to fuel their content recommendation engines. Machine learning models analyze past user behavior data for frequent patterns, develop association rules and use those rules to recommend content that a user is likely to engage with, or organize content in a way that is likely to put the most interesting content for a given user first.

**How association rules work**

Association rule mining, at a basic level, involves the use of [machine learning](https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML) models to analyze data for patterns, or co-occurrences, in a database. It identifies frequent if-then associations, which themselves are the *association rules*.

An association rule has two parts: an antecedent (if) and a consequent (then). An antecedent is an item found within the data. A consequent is an item found in combination with the antecedent.

Association rules are created by searching data for frequent if-then patterns and using the criteria *support* and *confidence* to identify the most important relationships*.* Support is an indication of how frequently the items appear in the data. Confidence indicates the number of times the if-then statements are found true. A third metric, called *lift*, can be used to compare confidence with expected confidence, or how many times an if-then statement is expected to be found true.

Association rules are calculated from *itemsets*, which are made up of two or more items. If rules are built from analyzing all the possible itemsets, there could be so many rules that the rules hold little meaning. With that, association rules are typically created from rules well-represented in data.

**Measures of the effectiveness of association rules**

The strength of a given association rule is measured by two main parameters: support and confidence. Support refers to how often a given rule appears in the database being mined. Confidence refers to the amount of times a given rule turns out to be true in practice. A rule may show a strong correlation in a data set because it appears very often but may occur far less when applied. This would be a case of high support, but low confidence.

Conversely, a rule might not particularly stand out in a data set, but continued analysis shows that it occurs very frequently. This would be a case of high confidence and low support. Using these measures helps analysts separate causation from correlation, and allows them to properly value a given rule.

A third value parameter, known as the lift value, is the ratio of confidence to support. If the lift value is a negative value, then there is a negative correlation between datapoints. If the value is positive, there is a positive correlation, and if the ratio equals 1, then there is no correlation.

**Association rule algorithm**

With the Apriori algorithm, candidate itemsets are generated using only the large itemsets of the previous pass. The large itemset of the previous pass is joined with itself to generate all itemsets with a size that's larger by one. Each generated itemset with a subset that is not large is then deleted. The remaining itemsets are the candidates. The Apriori algorithm considers any subset of a frequent itemset to also be a frequent itemset. With this approach, the algorithm reduces the number of candidates being considered by only exploring the itemsets whose support count is greater than the minimum support count, according to Sayad.

**Uses of association rules in data mining**

In [data mining](https://searchsqlserver.techtarget.com/definition/data-mining), association rules are useful for analyzing and predicting customer behavior. They play an important part in [customer analytics](https://searchbusinessanalytics.techtarget.com/definition/customer-analytics), market basket analysis, product clustering, catalog design and store layout.

Programmers use association rules to build programs capable of machine learning. Machine learning is a type of artificial intelligence ([AI](https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence)) that seeks to build programs with the ability to become more efficient without being explicitly programmed.

**Examples of association rules in data mining**

A classic example of association rule mining refers to a relationship between diapers and beers. The example, which seems to be fictional, claims that men who go to a store to buy diapers are also likely to buy beer. Data that would point to that might look like this:

A supermarket has 200,000 customer transactions. About 4,000 transactions, or about 2% of the total number of transactions, include the purchase of diapers. About 5,500 transactions (2.75%) include the purchase of beer. Of those, about 3,500 transactions, 1.75%, include both the purchase of diapers and beer. Based on the percentages, that large number should be much lower. However, the fact that about 87.5% of diaper purchases include the purchase of beer indicates a link between diapers and beer.

**History**

While the concepts behind association rules can be traced back earlier, association rule mining was defined in the 1990s, when computer scientists Rakesh Agrawal, Tomasz Imieliński and Arun Swami developed an algorithm-based way to find relationships between items using point-of-sale (POS) systems. Applying the algorithms to supermarkets, the scientists were able to discover links between different items purchased, called *association rules*, and ultimately use that information to predict the likelihood of different products being purchased together.

For retailers, association rule mining offered a way to better understand customer purchase behaviors. Because of its retail origins, association rule mining is often referred to as *market basket analysis*.

As advances in data science, AI and machine learning, have occurred since the original use case for association rules -- and more devices generate data -- association rules can be used in wider breadth of use cases. More data is being generated, meaning more applications for association rules. AI and machine learning allow for larger and more complex data sets to be analyzed and mined for association rules.

# Apriori Algorithm in Machine Learning

The Apriori algorithm uses frequent itemsets to generate association rules, and it is designed to work on the databases that contain transactions. With the help of these association rule, it determines how strongly or how weakly two objects are connected. This algorithm uses a **breadth-first search** and **Hash Tree** to calculate the itemset associations efficiently. It is the iterative process for finding the frequent itemsets from the large dataset.

This algorithm was given by the **R. Agrawal** and **Srikant** in the year **1994**. It is mainly used for market basket analysis and helps to find those products that can be bought together. It can also be used in the healthcare field to find drug reactions for patients.

**What is Frequent Itemset?**

Frequent itemsets are those items whose support is greater than the threshold value or user-specified minimum support. It means if A & B are the frequent itemsets together, then individually A and B should also be the frequent itemset.

Suppose there are the two transactions: A= {1,2,3,4,5}, and B= {2,3,7}, in these two transactions, 2 and 3 are the frequent itemsets.

#### Note: To better understand the apriori algorithm, and related term such as support and confidence, it is recommended to understand the association rule learning.

### Steps for Apriori Algorithm

Below are the steps for the apriori algorithm:

**Step-1:** Determine the support of itemsets in the transactional database, and select the minimum support and confidence.

**Step-2:** Take all supports in the transaction with higher support value than the minimum or selected support value.

**Step-3:** Find all the rules of these subsets that have higher confidence value than the threshold or minimum confidence.

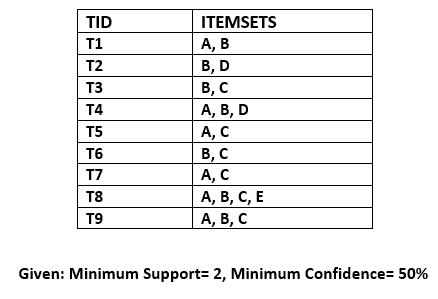
**Step-4:** Sort the rules as the decreasing order of lift.

### Apriori Algorithm Working

We will understand the apriori algorithm using an example and mathematical calculation:

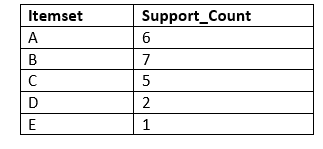
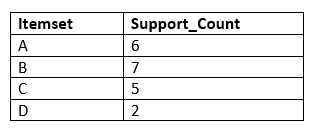
**Example:** Suppose we have the following dataset that has various transactions, and from this dataset, we need to find the frequent itemsets and generate the association rules using the Apriori algorithm:

:

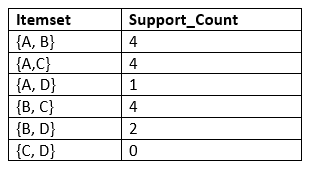
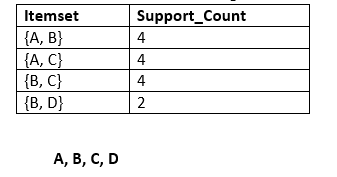


**Solution:**

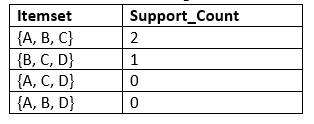
**Step-1: Calculating C1 and L1:**

* In the first step, we will create a table that contains support count (The frequency of each itemset individually in the dataset) of each itemset in the given dataset. This table is called the **Candidate set or C1.**  
  
* Now, we will take out all the itemsets that have the greater support count that the Minimum Support (2). It will give us the table for the **frequent itemset L1.**  
  Since all the itemsets have greater or equal support count than the minimum support, except the E, so E itemset will be removed.  
  

**Step-2: Candidate Generation C2, and L2:**

* In this step, we will generate C2 with the help of L1. In C2, we will create the pair of the itemsets of L1 in the form of subsets.
* After creating the subsets, we will again find the support count from the main transaction table of datasets, i.e., how many times these pairs have occurred together in the given dataset. So, we will get the below table for C2:  
  
* Again, we need to compare the C2 Support count with the minimum support count, and after comparing, the itemset with less support count will be eliminated from the table C2. It will give us the below table for L2  
  

**Step-3: Candidate generation C3, and L3:**

* For C3, we will repeat the same two processes, but now we will form the C3 table with subsets of three itemsets together, and will calculate the support count from the dataset. It will give the below table:  
  
* Now we will create the L3 table. As we can see from the above C3 table, there is only one combination of itemset that has support count equal to the minimum support count. So, the L3 will have only one combination, i.e., **{A, B, C}.**

**Step-4: Finding the association rules for the subsets:**

To generate the association rules, first, we will create a new table with the possible rules from the occurred combination {A, B.C}. For all the rules, we will calculate the Confidence using formula **sup( A ^B)/A.** After calculating the confidence value for all rules, we will exclude the rules that have less confidence than the minimum threshold(50%).

Consider the below table:

|  |  |  |
| --- | --- | --- |
| **Rules** | **Support** | **Confidence** |
| A ^B → C | 2 | Sup{(A ^B) ^C}/sup(A ^B)= 2/4=0.5=50% |
| B^C → A | 2 | Sup{(B^C) ^A}/sup(B ^C)= 2/4=0.5=50% |
| A^C → B | 2 | Sup{(A ^C) ^B}/sup(A ^C)= 2/4=0.5=50% |
| C→ A ^B | 2 | Sup{(C^( A ^B)}/sup(C)= 2/5=0.4=40% |
| A→ B^C | 2 | Sup{(A^( B ^C)}/sup(A)= 2/6=0.33=33.33% |
| B→ B^C | 2 | Sup{(B^( B ^C)}/sup(B)= 2/7=0.28=28% |

As the given threshold or minimum confidence is 50%, so the first three rules **A ^B → C, B^C → A, and A^C → B** can be considered as the strong association rules for the given problem.

**Advantages of Apriori Algorithm**

* This is easy to understand algorithm
* The join and prune steps of the algorithm can be easily implemented on large datasets.

**Disadvantages of Apriori Algorithm**

* The apriori algorithm works slow compared to other algorithms.
* The overall performance can be reduced as it scans the database for multiple times.
* The time complexity and space complexity of the apriori algorithm is O(2D), which is very high. Here D represents the horizontal width present in the database.

Assignment:

**Apriori: Support threshold=50%, Confidence= 60%**

**TABLE-1**

| **Transaction** | **List of items** |
| --- | --- |
| T1 | I1,I2,I3 |
| T2 | I2,I3,I4 |
| T3 | I4,I5 |
| T4 | I1,I2,I4 |
| T5 | I1,I2,I3,I5 |
| T6 | I1,I2,I3,I4 |

**Solution:**

Support threshold=50% => 0.5\*6= 3 => min\_sup=3

**1. Count Of Each Item**

**TABLE-2**

| **Item** | **Count** |
| --- | --- |
| I1 | 4 |
| I2 | 5 |
| I3 | 4 |
| I4 | 4 |
| I5 | 2 |

**2.** **Prune Step:** **TABLE -2** shows that I5 item does not meet min\_sup=3, thus it is deleted, only I1, I2, I3, I4 meet min\_sup count.

**TABLE-3**

| **Item** | **Count** |
| --- | --- |
| I1 | 4 |
| I2 | 5 |
| I3 | 4 |
| I4 | 4 |

**3.** **Join Step:** Form 2-itemset. From **TABLE-1** find out the occurrences of 2-itemset.

**TABLE-4**

| **Item** | **Count** |
| --- | --- |
| I1,I2 | 4 |
| I1,I3 | 3 |
| I1,I4 | 2 |
| I2,I3 | 4 |
| I2,I4 | 3 |
| I3,I4 | 2 |

**4.** **Prune Step:** **TABLE -4** shows that item set {I1, I4} and {I3, I4} does not meet min\_sup, thus it is deleted.

**TABLE-5**

| **Item** | **Count** |
| --- | --- |
| I1,I2 | 4 |
| I1,I3 | 3 |
| I2,I3 | 4 |
| I2,I4 | 3 |

**5.** **Join and Prune Step:** Form 3-itemset. From the **TABLE- 1** find out occurrences of 3-itemset. From **TABLE-5**, find out the 2-itemset subsets which support min\_sup.

We can see for itemset {I1, I2, I3} subsets, {I1, I2}, {I1, I3}, {I2, I3} are occurring in **TABLE-5** thus {I1, I2, I3} is frequent.

We can see for itemset {I1, I2, I4} subsets, {I1, I2}, {I1, I4}, {I2, I4}, {I1, I4} is not frequent, as it is not occurring in **TABLE-5** thus {I1, I2, I4} is not frequent, hence it is deleted.

**TABLE-6**

| **Item** |
| --- |
| I1,I2,I3 |
| I1,I2,I4 |
| I1,I3,I4 |
| I2,I3,I4 |

**Only {I1, I2, I3} is frequent**.

**6. Generate Association Rules:** From the frequent itemset discovered above the association could be:

{I1, I2} => {I3}

Confidence = support {I1, I2, I3} / support {I1, I2} = (3/ 4)\* 100 = 75%

{I1, I3} => {I2}

Confidence = support {I1, I2, I3} / support {I1, I3} = (3/ 3)\* 100 = 100%

{I2, I3} => {I1}

Confidence = support {I1, I2, I3} / support {I2, I3} = (3/ 4)\* 100 = 75%

{I1} => {I2, I3}

Confidence = support {I1, I2, I3} / support {I1} = (3/ 4)\* 100 = 75%

{I2} => {I1, I3}

Confidence = support {I1, I2, I3} / support {I2 = (3/ 5)\* 100 = 60%

{I3} => {I1, I2}

Confidence = support {I1, I2, I3} / support {I3} = (3/ 4)\* 100 = 75%

This shows that all the above association rules are strong if minimum confidence threshold is 60%.