

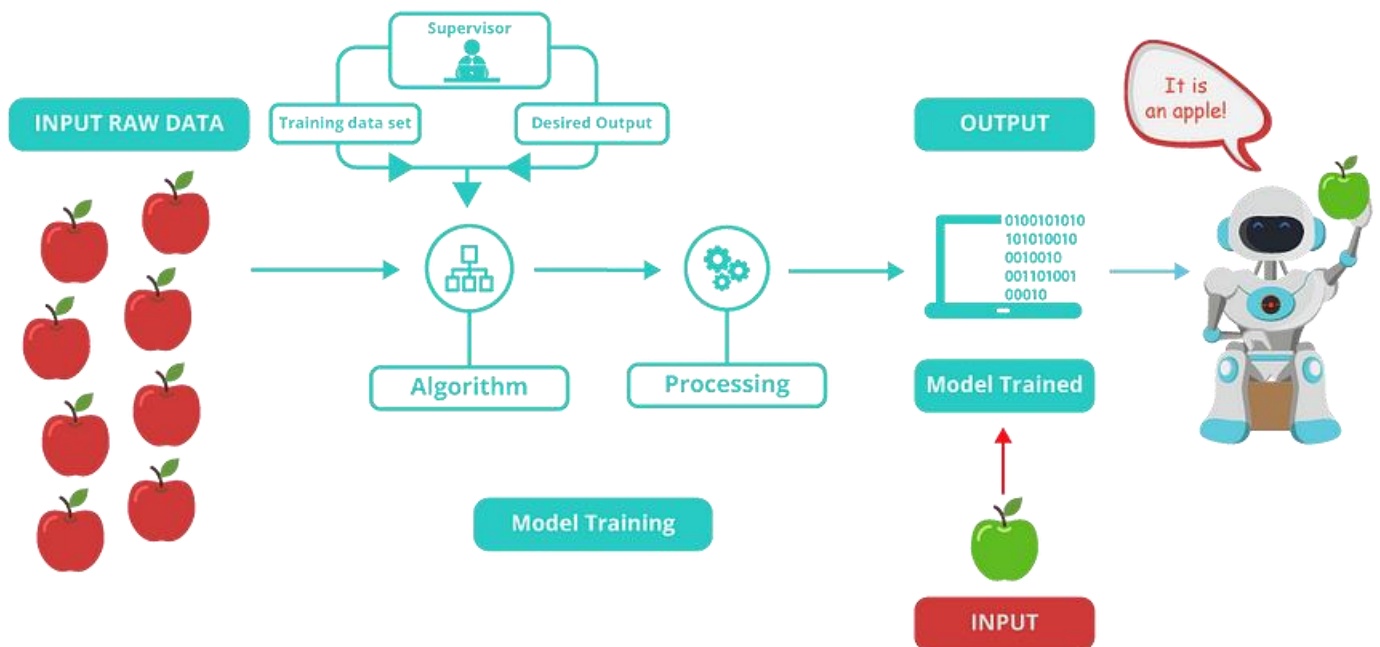
Apriori Algorithm

In this blog, I want you to give basic knowledge on one of the unsupervised learning algorithm called the **apriori algorithm** in machine learning. First of all, you need to have a clear idea of Supervised learning and unsupervised learning. So, What is **Supervised learning**?

Supervised learning is the one where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output. it,

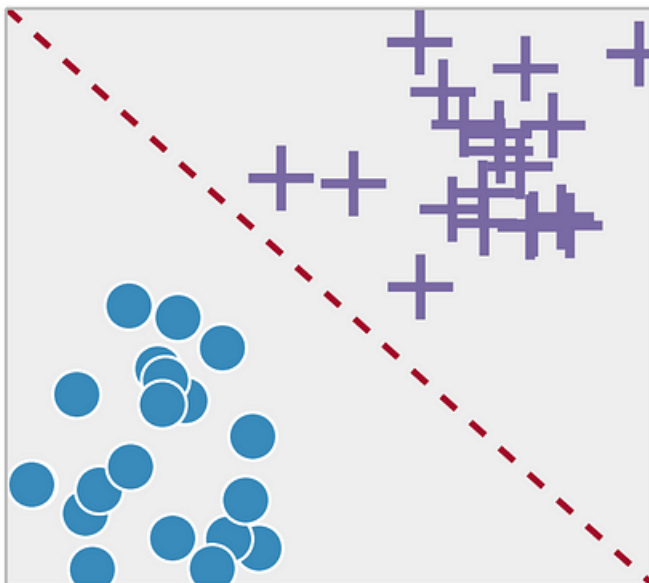
$$Y = f(X)$$

The goal is to approximate the mapping function so well that whenever you get some new input data (x), the machine can easily predict the output variables (Y) for that data.

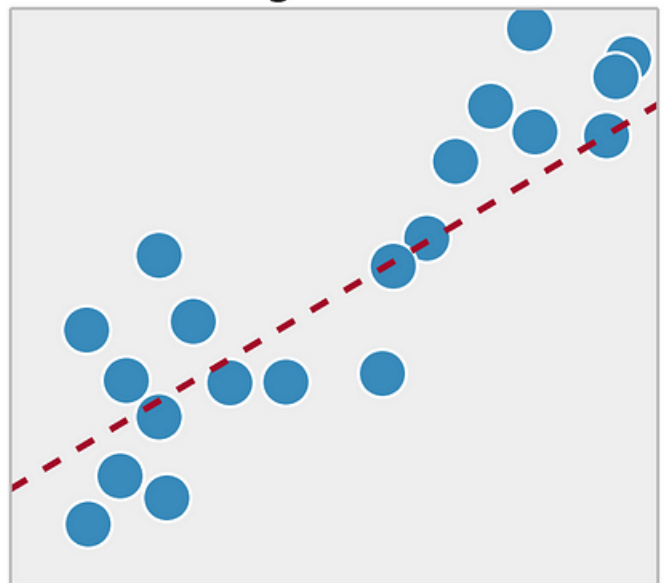


Supervised learning problems can be of two types:

Classification



Regression



a. Classification: To predict the outcome of a given sample where the output variable is in the form of categories. Examples include labels such as male and female, sick and healthy.

b. Regression: To predict the outcome of a given sample where the output variable is in the form of real values. Examples include real-valued labels denoting the amount of rainfall, the height of a person.

Let me rephrase you this in simple terms:

In the Supervised machine learning algorithm, every instance of the training dataset consists of input attributes and expected output. The training dataset can take any kind of data as an input like **values of a database row, the pixels of an image, or even an audio frequency histogram.**

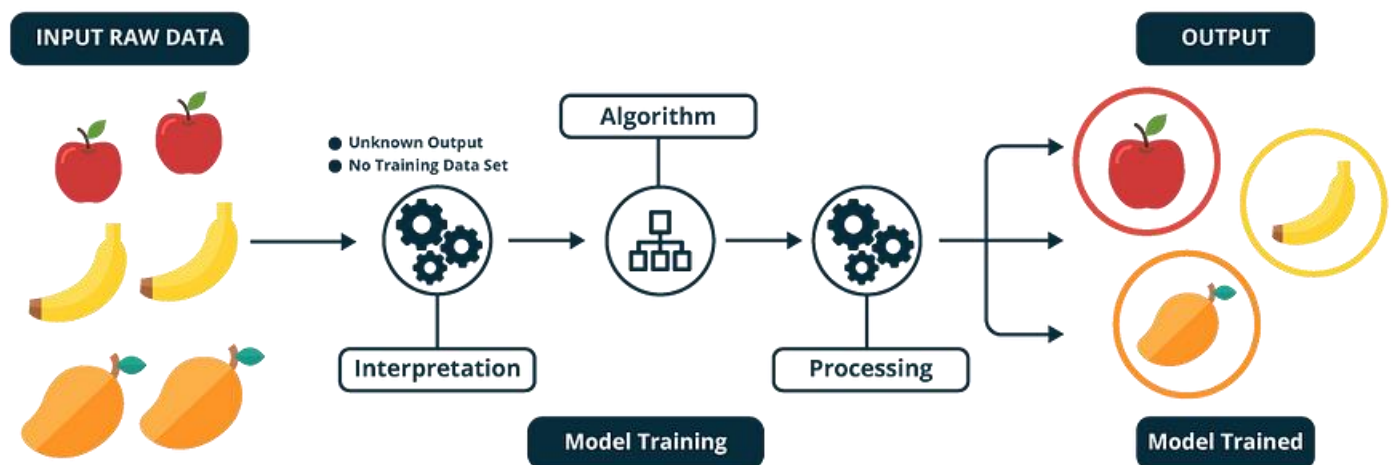
Now let me explain what is **supervised learning?**

The process of an algorithm learning from the training dataset can be thought of as a **teacher teaching his students.** The algorithm continuously predicts the result on the basis of training data and is continuously corrected by the teacher. The learning continues until the algorithm achieves an acceptable level of performance.

Now you have a clear idea about Supervised learning. Now let me explain what is **Unsupervised Learning**?

Mathematically, Unsupervised learning is where you only have input data (X) and no corresponding output variables.

The goal of unsupervised learning is to model the underlying structure or distribution of data in order to learn more about the data.



Unsupervised learning problems can be of two types:

a. Association: To discover the probability of the co-occurrence of items in a collection. It is extensively used in the market-basket analysis. Example: If a customer purchases bread, he is 80% likely to also purchase eggs.

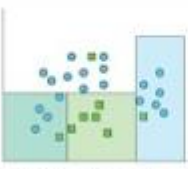

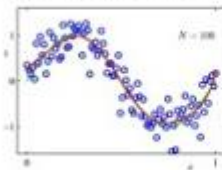

b. Clustering: To group samples such that objects within the same cluster are more similar to each other than to the objects from another cluster. Example: marketing research, image processing, data mining.

Let me rephrase it for you in simple terms:

In the unsupervised learning approach, the sample of a training data set does not have an expected output associated with them. Using the unsupervised learning algorithms you can detect patterns based on the typical characteristics of the input data. Clustering can be considered as an example of a machine learning task that uses the unsupervised learning approach. The machine then groups similar data samples and identifies different clusters within the data.

Now let me explain to you why this category of machine learning is known as unsupervised learning?

Well, In this category of machine learning is known as unsupervised because unlike in supervised learning there is no teacher. Algorithms are left on their own to discover and return the interesting structure of the data.

Supervised learning	Unsupervised Learning
Classification  Learns a method for predicting the instance class from pre-labeled (classified) instances	Clustering  Finds "natural" grouping of instances given un-labeled data
Regression  An attempt to predict a continuous attribute	Association Rules  Method for discovering interesting relations between variables in large DBs

There are many algorithms for generating association rules, some well-known algorithms are Apriori, Eclat, and FP-Growth.

Now let me tell you about the Apriori algorithm,

The Apriori algorithm is used in a transactional database to mine **frequent itemsets** and then generate **association rules**. The name of the algorithm is Apriori is because it uses prior knowledge of frequent itemset properties. It is popularly used in market basket analysis, where one checks for combinations of products that frequently co-occur in the database and it helps the customers in purchasing their items with more ease which increases the sales of the markets. It has also been used in the field of healthcare for the detection of adverse drug reactions. It produces association rules that indicate what all combinations of medications and patient characteristics lead to ADRs. In general, we write the association rule for, if a person purchases item X, then he purchases item Y, as: $X \rightarrow Y$.

Example: if a person purchases milk and sugar, then he is likely to purchase coffee powder. This could be written in the form of an association rule as {milk, sugar} -> coffee powder. Association rules are generated after crossing the threshold for support and confidence. The Support measure helps prune the number of candidate itemsets to be considered during frequent itemset generation. This support measure is guided by the Apriori principle. The **Apriori principle** states that **if an itemset is frequent, then all of its subsets must also be frequent.**

Let's consider some important terms,

Itemset: A set of items is referred to as itemset and an itemset containing k items is called k-itemset.























Frequent Itemset: Suppose min_sup is the minimum support threshold, an itemset satisfies minimum support if the occurrence frequency of the itemset is greater or equal to min_sup. If an itemset satisfies minimum support, then it is a frequent itemset.

What are association rules?

Association rules analysis is a technique to uncover how items are associated with each other. Association rule mining finds interesting associations and relationships among large sets of data items. This rule

shows how frequently an itemset occurs in a transaction. A typical example is a Market Based Analysis.

Market-Based Analysis is one of the key techniques used by large relations to show associations between items. It allows retailers to identify relationships between the items that people buy together frequently. Given a set of transactions, we can find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction.

Transaction 1	   
Transaction 2	  
Transaction 3	 
Transaction 4	 
Transaction 5	   
Transaction 6	  
Transaction 7	 
Transaction 8	 

There are three common ways to measure association. So that before we start defining the rule, let us first see the basic definitions.

Important Definitions:

Support

Support is an indication of how frequently the itemset appears in the dataset. It is one of the measures of interestingness. This tells about the usefulness and certainty of rules. **50% Support** means a total 50% of transactions in the database follow the rule.

$$\text{Support}\{X\} = \frac{\text{Number of transaction in which } X \text{ appears}}{\text{Total number of transactions}}$$

$$\text{Support}\{\text{🍎}\} = \frac{4}{8}$$

Confidence

Confidence is an indication of how often the rule has been found to be true. It signifies the likelihood of item Y being purchased when item X is purchased. A confidence of 75% means that 75% of the customers who purchased an apple also bought beer.

$$\text{Confidence}\{X \rightarrow Y\} = \frac{\text{Support}\{X \cup Y\}}{\text{Support}\{X\}}$$

$$\text{Confidence}\{\text{🍎} \rightarrow \text{🍺}\} = \frac{\text{Support}\{\text{🍎} \cup \text{🍺}\}}{\text{Support}\{\text{🍎}\}}$$

If a rule satisfies both **minimum support** and **minimum confidence**, it is a **strong rule**.

Lift

how likely item Y is purchased when item X is purchased while controlling for how popular item Y is. If the lift value is 1, which implies no association between items. A lift value greater than 1 means that item Y is *likely* to be bought if item X is bought, while a value less than 1 means that item Y is *unlikely* to be bought if item X is bought. A lift of 12.5% means that 12.5% of the customers who likely to be bought beer if an apple is bought.

$$\text{Lift}\{\text{🍏} \rightarrow \text{🍺}\} = \frac{\text{Support}\{\text{🍏} \cup \text{🍺}\}}{\text{Support}\{\text{🍏}\} \times \text{Support}\{\text{🍺}\}}$$

So, Let's learn about the Association Rules:

For this dataset, we can write the following association rules: (Rules are just for illustrations and understanding of the concept. They might not represent the actuals).

Rule 1: If apple is purchased, Then the beer is also purchased in 75% of the transactions.

Rule 2: If beer is purchased, Then the meat is also purchased in 33.33% of the transactions.

Generally, association rules are written in the “IF-THEN” format. We can also use the term “Antecedent” for IF and “Consequent” for THEN.

In order to understand the concept better, let's take a simple dataset and find **frequent itemsets** and generate **association rules** on this.

TID	Items
T1	I1,I2,I5
T2	I2,I4
T3	I2,I3
T4	I1,I2,I4
T5	I1,I3
T6	I2,I3
T7	I1,I3
T8	I1,I2,I3,I5
T9	I1,I2,I3

minimum support count is 2,
that means, minimum support threshold is
 $2/9 \times 100\% = 22.2\%$
minimum confidence is 60%

A minimum support threshold is applied to find all frequent itemsets in a database.
A minimum confidence constraint is applied to these frequent itemsets in order to form rules.

Step 1: Create a frequency table of all the items that occur in all the transactions. For our case:

Item set	Sup_count
I1	6
I2	7
I3	6
I4	2
I5	2

Step 2: We know that only those elements are significant for which the support is greater than or equal to the threshold support. Here, the support threshold is 22.2%, hence only those items are significant which occur in more than 2 transactions and such items are l1, l2, l3, l4, l5. Therefore, we are left with:

Item set	Sup_count
l1	6
l2	7
l3	6
l4	2
l5	2

The table above represents the single items that are purchased by the customers frequently.

Step 3: The next step is to make all the possible pairs of the significant items keeping in mind that the order doesn't matter, i.e., AB is same as BA. To do this, take the first item and pair it with all the others such as {l1, l2}, {l1, l3}, {l1, l4}, {l1, l5}. Similarly, consider the second item and pair it with preceding items, i.e., {l2, l3}, {l2, l4}, {l2, l5}. We are only considering the preceding items because {l2, l1} (same as {l1, l2}) already exists. So, all the pairs in our example are {l1, l2}, {l1, l3}, {l1, l4}, {l1, l5}, {l2, l3}, {l2, l4}, {l2, l5}, {l3, l4}, {l3, l5}, {l4, l5}.

Step 4: We will now count the occurrences of each pair in all the transactions.

Item set	Sup_count
l1, l2	4
l1, l3	4
l1, l4	1
l1, l5	2
l2, l3	4
l2, l4	2
l2, l5	2
l3, l4	0
l3, l5	1
l4, l5	0

Step 5: Again only those itemsets are significant which cross the support threshold, and those are {l1, l2}, {l1, l3}, {l1, l5}, {l2, l3}, {l2, l4} and {l2, l5}.

Step 6: Now let's say we would like to look for a set of three items that are purchased together. We will use the itemsets found in step 5 and create a set of 3 items.

To create a set of 3 items another rule, called self-join is required. It says that from the item pairs {l1, l2}, {l1, l3}, {l1, l5}, {l2, l3}, {l2, l4} and {l2, l5} we look for two pairs with the identical letter and so we get different set of items

- $\{l_1, l_2\}$ and $\{l_1, l_3\}$, this gives $\{l_1, l_2, l_3\}$
- $\{l_1, l_2\}$ and $\{l_1, l_5\}$, this gives $\{l_1, l_2, l_5\}$
- $\{l_1, l_2\}$ and $\{l_2, l_4\}$, this gives $\{l_1, l_2, l_4\}$
- $\{l_1, l_3\}$ and $\{l_1, l_5\}$, this gives $\{l_1, l_3, l_5\}$
- $\{l_2, l_3\}$ and $\{l_2, l_4\}$, this gives $\{l_2, l_3, l_4\}$
- $\{l_2, l_3\}$ and $\{l_2, l_5\}$, this gives $\{l_2, l_3, l_5\}$
- $\{l_2, l_4\}$ and $\{l_2, l_5\}$, this gives $\{l_2, l_4, l_5\}$

Next, we find the frequency for these itemsets. Among them only two item sets can get as frequent itemset, those are $\{l_1, l_2, l_3\}$ and $\{l_1, l_2, l_5\}$

Item set	Sup_count
$\{l_1, l_2, l_3\}$	2
$\{l_1, l_2, l_5\}$	2

Again we can apply the self-join rule, then we get $\{l_1, l_2, l_3, l_5\}$ item set, but it is not a frequent itemset or check all subsets of these itemsets are frequent or not (Here itemset formed by joining above table is $\{l_1, l_2, l_3, l_5\}$ so its subset contains $\{l_1, l_3, l_5\}$ which is not frequent). We stop here because no frequent itemset is found frequent further.

General Process of the Apriori algorithm

The entire algorithm can be divided into two steps: **Step 1:** Apply minimum support to find all the frequent sets with k items in a database. **Step 2:** Use the self-join rule to find the frequent sets with k+1 items with the help of frequent k-itemsets. Repeat this process from k=1 to the point when we are unable to apply the self-join rule.

Thus we discovered all frequent item-sets now the generation of strong association rule comes into the picture. For that, we need to calculate the confidence of each rule.

As an example a confidence of 60% means that 60% of the customers who purchased a milk and bread also bought the butter. So here By taking example of any frequent itemset we will show rule generation. Let's take Itemset {I1, I2, I3} ,So rules can be

- $[I1 \wedge I2] \Rightarrow [I3]$ //confidence = $\frac{\text{sup}(I1 \wedge I2 \wedge I3)}{\text{sup}(I1 \wedge I2)} = \frac{2}{4} * 100 = 50\%$ // **Rejected**
- $[I1 \wedge I3] \Rightarrow [I2]$ //confidence = $\frac{\text{sup}(I1 \wedge I2 \wedge I3)}{\text{sup}(I1 \wedge I3)} = \frac{2}{4} * 100 = 50\%$ // **Rejected**
- $[I2 \wedge I3] \Rightarrow [I1]$ //confidence = $\frac{\text{sup}(I1 \wedge I2 \wedge I3)}{\text{sup}(I2 \wedge I3)} = \frac{2}{4} * 100 = 50\%$ // **Rejected**
- $[I1] \Rightarrow [I2 \wedge I3]$ //confidence = $\frac{\text{sup}(I1 \wedge I2 \wedge I3)}{\text{sup}(I1)} = \frac{2}{6} * 100 = 33\%$ // **Rejected**

- $[I_2] \Rightarrow [I_1 \wedge I_3]$ //confidence = $\frac{\text{sup}(I_1 \wedge I_2 \wedge I_3)}{\text{sup}(I_2)} = \frac{2}{7} * 100 = 28\%$ // **Rejected**
- $[I_3] \Rightarrow [I_1 \wedge I_2]$ //confidence = $\frac{\text{sup}(I_1 \wedge I_2 \wedge I_3)}{\text{sup}(I_3)} = \frac{2}{6} * 100 = 33\%$ // **Rejected**

Let's take Itemset $\{I_1, I_2, I_5\}$,So rules can be

- $[I_1 \wedge I_2] \Rightarrow [I_5]$ //confidence = $\frac{\text{sup}(I_1 \wedge I_2 \wedge I_5)}{\text{sup}(I_1 \wedge I_2)} = \frac{2}{4} * 100 = 50\%$ // **Rejected**
- $[I_1 \wedge I_5] \Rightarrow [I_2]$ //confidence = $\frac{\text{sup}(I_1 \wedge I_2 \wedge I_5)}{\text{sup}(I_1 \wedge I_5)} = \frac{2}{2} * 100 = 100\%$ // **Selected**
- $[I_2 \wedge I_5] \Rightarrow [I_1]$ //confidence = $\frac{\text{sup}(I_1 \wedge I_2 \wedge I_5)}{\text{sup}(I_2 \wedge I_5)} = \frac{2}{2} * 100 = 100\%$ // **Selected**
- $[I_1] \Rightarrow [I_2 \wedge I_5]$ //confidence = $\frac{\text{sup}(I_1 \wedge I_2 \wedge I_5)}{\text{sup}(I_1)} = \frac{2}{6} * 100 = 33\%$ // **Rejected**
- $[I_2] \Rightarrow [I_1 \wedge I_5]$ //confidence = $\frac{\text{sup}(I_1 \wedge I_2 \wedge I_5)}{\text{sup}(I_2)} = \frac{2}{7} * 100 = 28\%$ // **Rejected**
- $[I_5] \Rightarrow [I_1 \wedge I_2]$ //confidence = $\frac{\text{sup}(I_1 \wedge I_2 \wedge I_5)}{\text{sup}(I_5)} = \frac{2}{2} * 100 = 100\%$ // **Selected**

The minimum confidence threshold is 60%. So, We have found three strong association rules.

References:

1. Apriori
algorithm: https://en.wikipedia.org/wiki/Apriori_algorithm
2. Apriori algorithm sample
code: <https://github.com/kaumadie/Machine-Learning/tree/master/Association>
3. More
examples: <https://kaumadiechamalka100.medium.com/apriori-algorithm-examples-be8915b01cf2>