**Unsupervised learning**

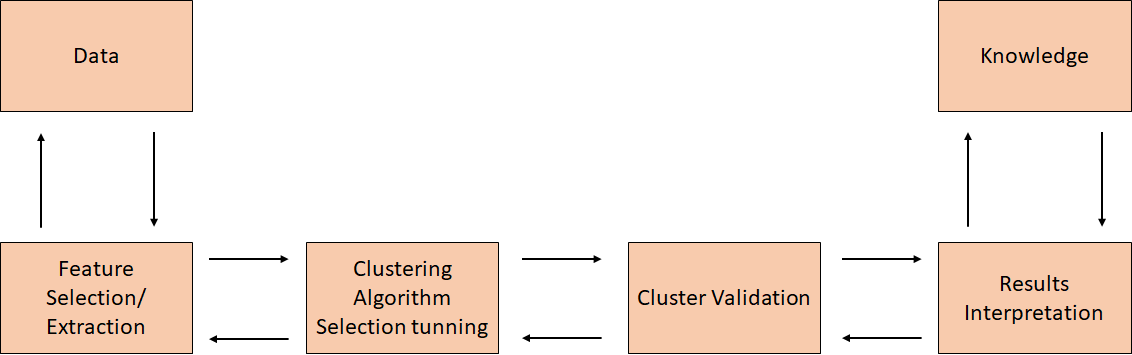
Unsupervised learning is a type of self-organized [Hebbian learning](https://en.wikipedia.org/wiki/Hebbian_learning) that helps find previously unknown patterns in data set without pre-existing labels. It is also known as self-organization and allows modeling probability densities of given inputs. It is one of the main three categories of machine learning, along with supervised and reinforcement learning.

Up to know, we have only explored supervised Machine Learning algorithms and techniques to develop models where the data had labels previously known. In other words, our data had some target variables with specific values that we used to train our models.

However, when dealing with real-world problems, most of the time, data will not come with predefined labels, so we will want to develop machine learning models that can classify correctly this data, by finding by themselves some commonality in the features, that will be used to predict the classes on new data.

Unsupervised Learning Analysis Process

The overall process that we will follow when developing an unsupervised learning model can be summarized in the following chart:



Picture1 unsupervised learning model

Unsupervised learning main applications are:

1.Segmenting datasets by some shared attributes.

2.Detecting anomalies that do not fit to any group.

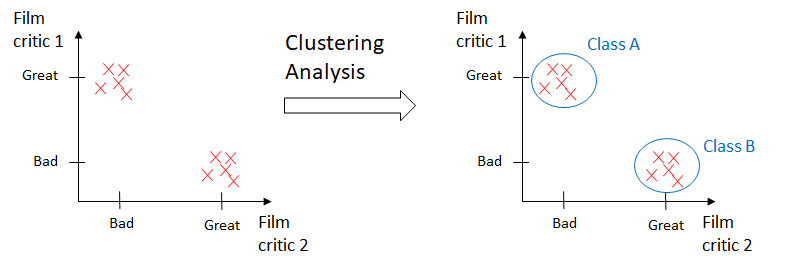
3.Simplify datasets by aggregating variables with similar attributes.

In summary, the main goal is to study the intrinsic (and commonly hidden) structure of the data. In this article, we will focus on the clustering problems.

**Clustering Analysis**

In basic terms, the objective of clustering is to find different groups within the elements in the data. To do so, clustering algorithms find the structure in the data so that elements of the same cluster (or group) are more similar to each other than to those from different clusters.

In a visual way: Imagine that we have a dataset of movies and want to classify them. We have the following reviews of films:



Picture2 examples of clustering analysis

This article will focus on K-Means and Hierarchichal Clustering algorithm.

**K-Means Clustering**

The K-Means algorithms aims to find and group in classes the data points that have high similarity between them. In the terms of the algorithm, this similiarity is understood as the opposite of the distance between data points. The closer the data points are, the more similar and more likely to belong to the same cluster they will be.

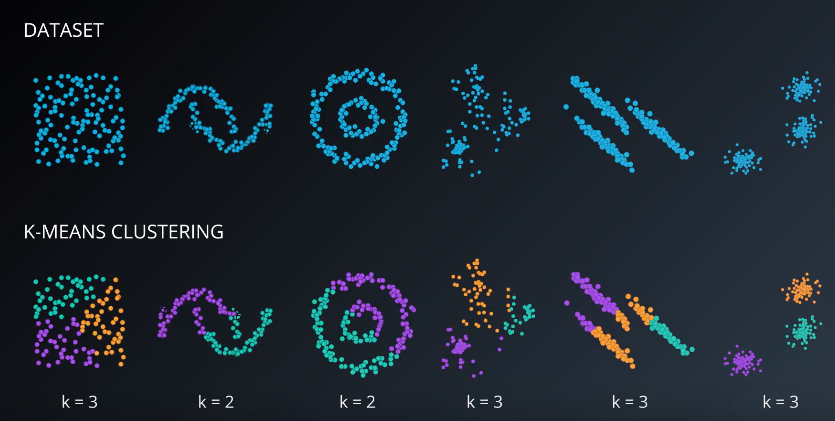
The most commonly used distance in K-Means is the squared Euclidean distance. An example of this distance between two points *x and y*in *m-dimensional*space is:

Here,*j*is the *jth* dimension (or feature column) of the sample points *x and y.*

Cluster inertia is the name given to the Sum of Squared Errors within the clustering context, and is represented as follows:

Where u(j) is the centroid for cluster j, and w(i,j) is 1 if the sample x(i) is in cluster j and 0 otherwise.

K-Means can be understood as an algorithm that will try to minimize the cluster inertia factor.



Picture3 using K-Means to dataset

**PROS AND CONS**

**Advantages of K-Means Clustering**

1.Good at dealing with the situation where features must be measured on the same scale, so it may be necessay to perform z-score standardization or max-min scaling.

2. dealing with categorical data, we will use the get dummies function.

3. it is most useful when we know beforehand the exact number of clusters and when we are dealing with spherical-shaped distributions.

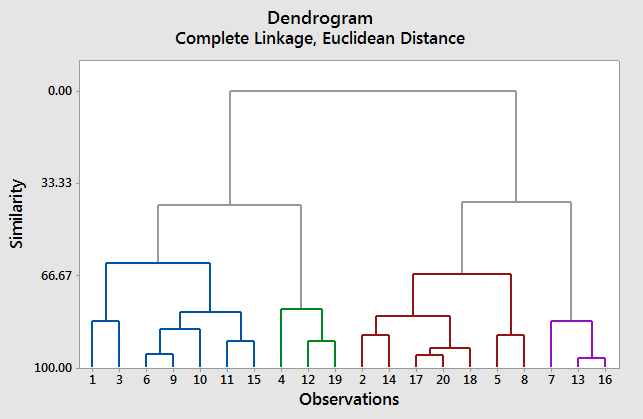
**Disadvantages of K-Means Clustering**

1. The output for any fixed training set won’t be always the same, because the initial centroids are set randomly and that will influence the whole algorithm process.

2. due to the nature of Euclidean distance, it is not a suitable algorithm when dealing with clusters that adopt non-spherical shapes.

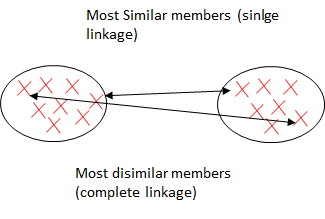
**Hierarchichal Clustering**

Hierarchichal clustering is an alternative to prototyope-based clustering algorithms. The main advantage of Hierarchichal clustering is that we do not need to specify the number of clusters, it will find it by itself. In addition, it enables the plotting of dendograms. Dendograms are visualizations of a binary hierarchichal clustering.



Picture4 Dendrogram for hierarchichal clustering

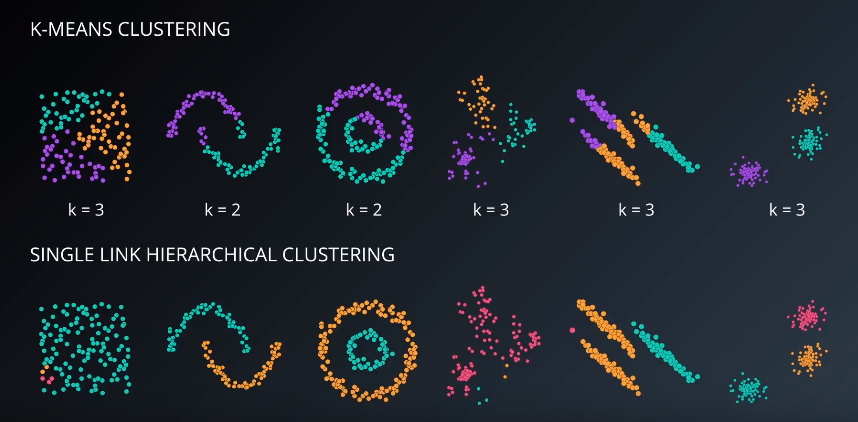
These are the most common algorithms used for agglomerative hierarchichal clustering.



Picture5 single and complete linkage

**Single linkage**

As being an agglomerative algorithm, single linkage starts by assuming that each sample point is a cluster. Then, it computes the distances between the most similar members for each pair of clusters and merge the two clusters for which the distance between the most similar members is the smallest.



Picture6 single link Hierachichal VS k-menas

**Complete Linkage**

Although being similar to its brother (single linkage) its philosophy is exactly the opposite, it compares the most dissimilar datapoints of a pair of clusters to perform the merge.

**PROS AND CONS**

**Advantages of Hierarchichal Clustering**

1.The resulting hierarchichal representations can be very informative.

2.Dendograms provide an interesting and informative way of visualization.

3.They are specially powerful when the dataset contains real hierarchichal relationships.

**Disadvantages of Hierarchichal Clustering**

1.They are very sensitive to outliers and, in their presence, the model performance decreases significantly.

2,They are very expensive, computationally speaking.

k-means can be applied to data that has a smaller number of dimensions, is numeric, and is continuous. such as document clustering, identifying crime-prone areas, customer segmentation, insurance fraud detection, public transport data analysis, clustering of IT alerts.