Unsupervised Machine Learning: Clustering Analysis

Victor Roman Follow
Mar 6, 2019 · 12 min read ★

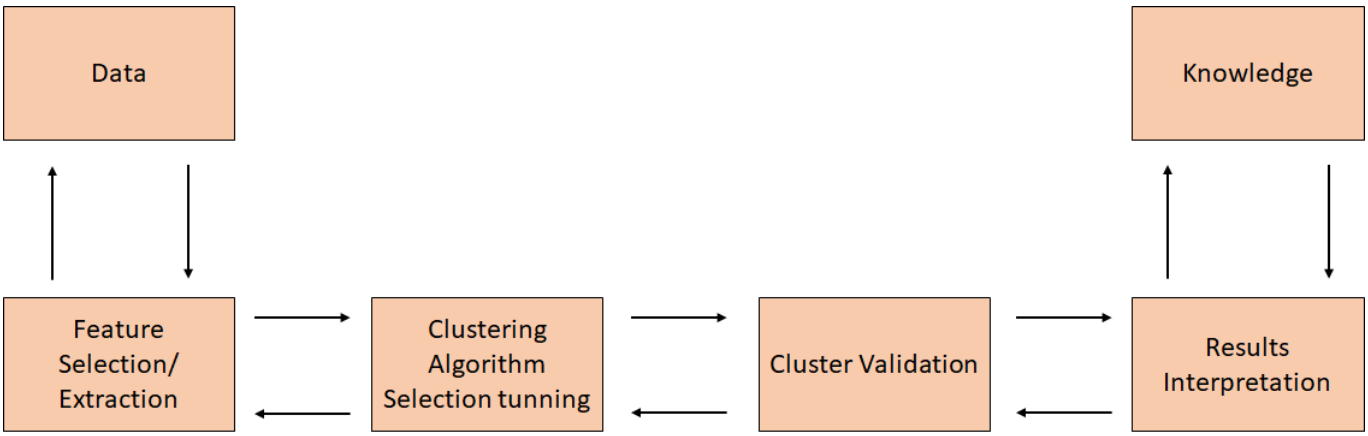
Introduction to Unsupervised Learning

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:



Unsupervised learning main applications are:

- Segmenting datasets by some shared atributes.
- Detecting anomalies that do not fit to any group.
- Simplify datasets by aggregating variables with similar atributes.

In summary, the main goal is to study the intrinsic (and commonly hidden) structure of the data.

This techniques can be condensed in two main types of problems that unsupervised learning tries to solve. This problems are:

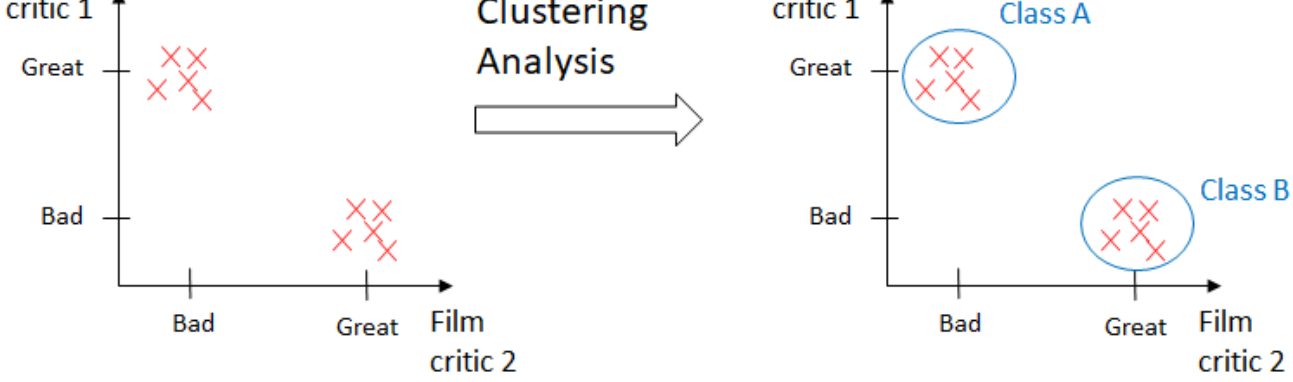
- Clustering
- Dimensionality Reduction

Throughout this article we will focus on clustering problems and we will cover dimensionality reduction in future articles.

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:



The machine learning model will be able to infere that there are two different classes without knowing anything else from the data.

These unsupervised learning algorithms have an incredible wide range of applications and are quite useful to solve real world problems such as anomaly detection, recommending systems, documents grouping, or finding customers with common interests based on their purchases.

Some of the most common clustering algorithms, and the ones that will be explored thourghout the article, are:

- K-Means
- Hierarchichal Clustering
- Density Based Scan Clustering (DBSCAN)
- Gaussian Clustering Model

K-Means Clustering

K-Means algorithms are extremely easy to implement and very efficient computationally speaking. Those are the main reasons that explain why they are so popular. But they are not very good to identify classes when dealing with in groups that do not have a spherical distribution shape.

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 datapoints. The closer the data points are, the more similar and more likely to belong to the same cluster they will be.

Key Concepts

- Squared Euclidean Distance

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:

$$d(\boldsymbol{x}, \boldsymbol{y})^2 = \sum_{j=1}^{m} \left(x_j - y_j\right)^2 = \|\boldsymbol{x} - \boldsymbol{y}\|_2^2$$

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

- Cluster Inertia

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

$$SSE = \sum_{i=1}^{n} \sum_{j=1}^{k} w^{(i,j)} \left\|\boldsymbol{x}^{(i)} - \boldsymbol{\mu}^{(j)}\right\|_2^2$$

Where *μ(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.

Algorithm Steps

1. First, we need to choose k, the number of clusters that we want to be finded.
2. Then, the algorithm will select randomly the the centroids of each cluster.
3. It will be assigned each datapoint to the closest centroid (using euclidean distance).
4. It will be computed the cluster inertia.
5. The new centroids will be calculated as the mean of the points that belong to the centroid of the previous step. In other words, by calculating the minimum quadratic error of the datapoints to the center of each cluster, moving the center towards that point
6. Back to step 3.

K-Means Hyperparameters

- Number of clusters: The number of clusters and centroids to generate.
- Maximum iterations: Of the algorithm for a single run.
- Number initial: The numbe rof times the algorithm will be run with different centroid seeds. The final result will be the best output of the number defined of consecutives runs, in terms of inertia.

Challeges of K-Means

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

Points to be Considered When Applying K-Means

- Features must be measured on the same scale, so it may be necessay to perform z-score standardization or max-min scaling.
- When dealing with categorical data, we will use the get dummies function.
- Exploratory Data Analysis (EDA) is very helpful to have an overview of the data and determine if K-Means is the most appropiate algorithm.
- The minibatch method is very useful when there is a large number of columns, however, it is less accurate.

How to Choose the Right K Number

Choosing the right number of clusters is one of the key points of the K-Means algorithm. To find this number there are some methods:

- Field knowledge
- Bussiness decision
- Elbow Method

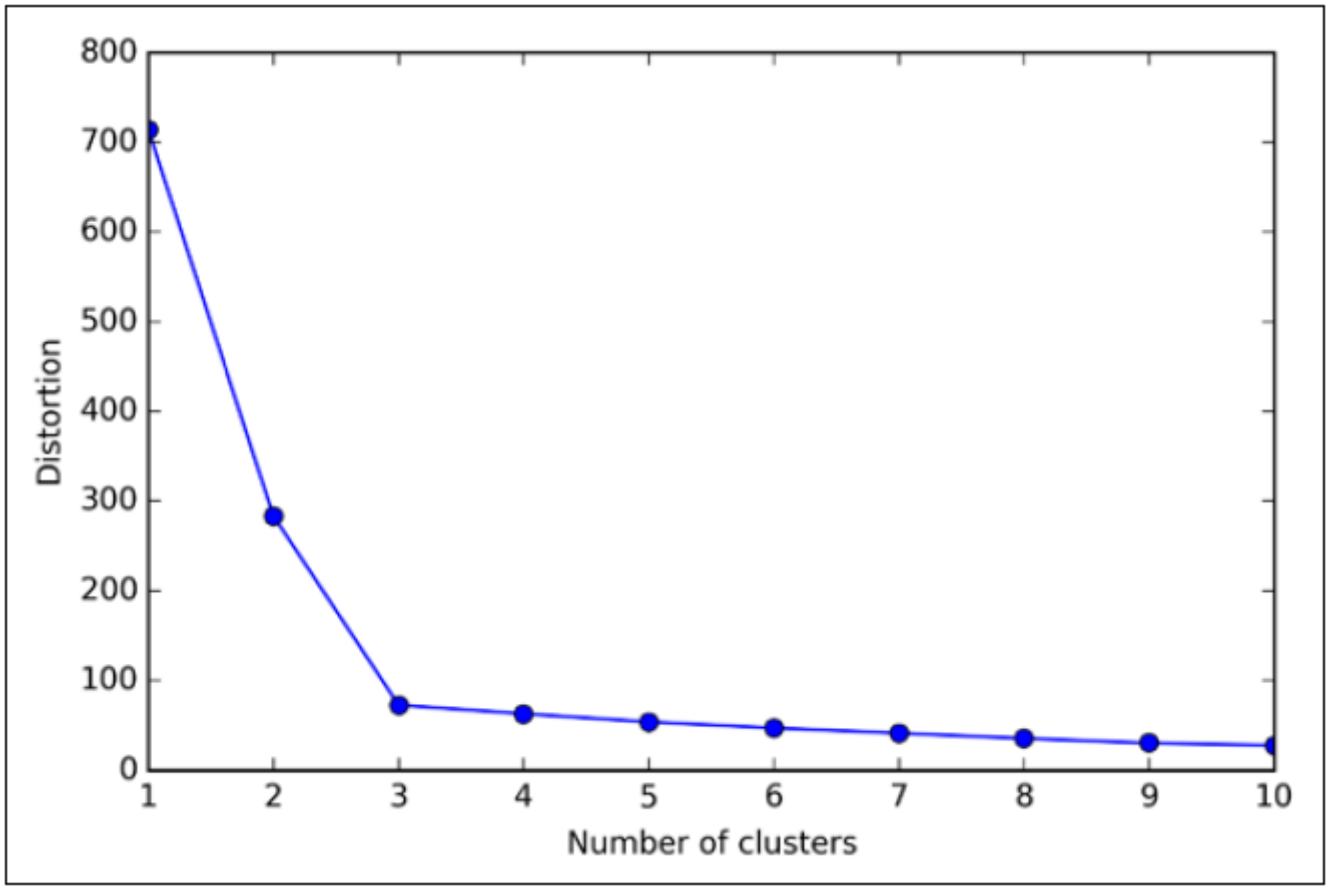
As being aligned with the motivation and nature of Data Science, the elbow mehtod is the prefered option as it relies on an analytical method backed with data, to make a decision.

Elbow Method

The elbow method is used for determining the correct number of clusters in a dataset. It works by plotting the ascending values of K versus the total error obtained when using that K.

$$\% \text{ Variance} = \frac{\text{Variance between groups}}{\text{Total variance}}$$

The goal is to find the k that for each cluster will not rise significantly the variance

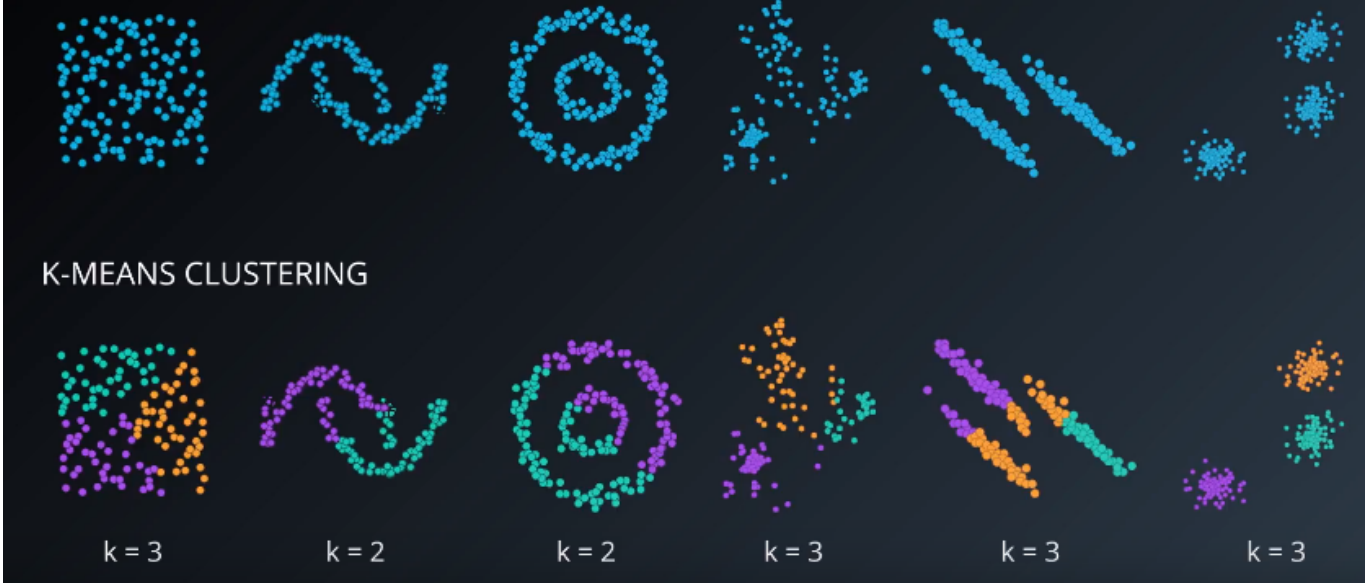


In this case, we will choose the k=3, where the elbow is located.

K-Means Limitations

Although K-Means is a great clustering algorithm, it is most useful when we know beforehand the exact number of clusters and when we are dealing with spherical-shaped distributions.

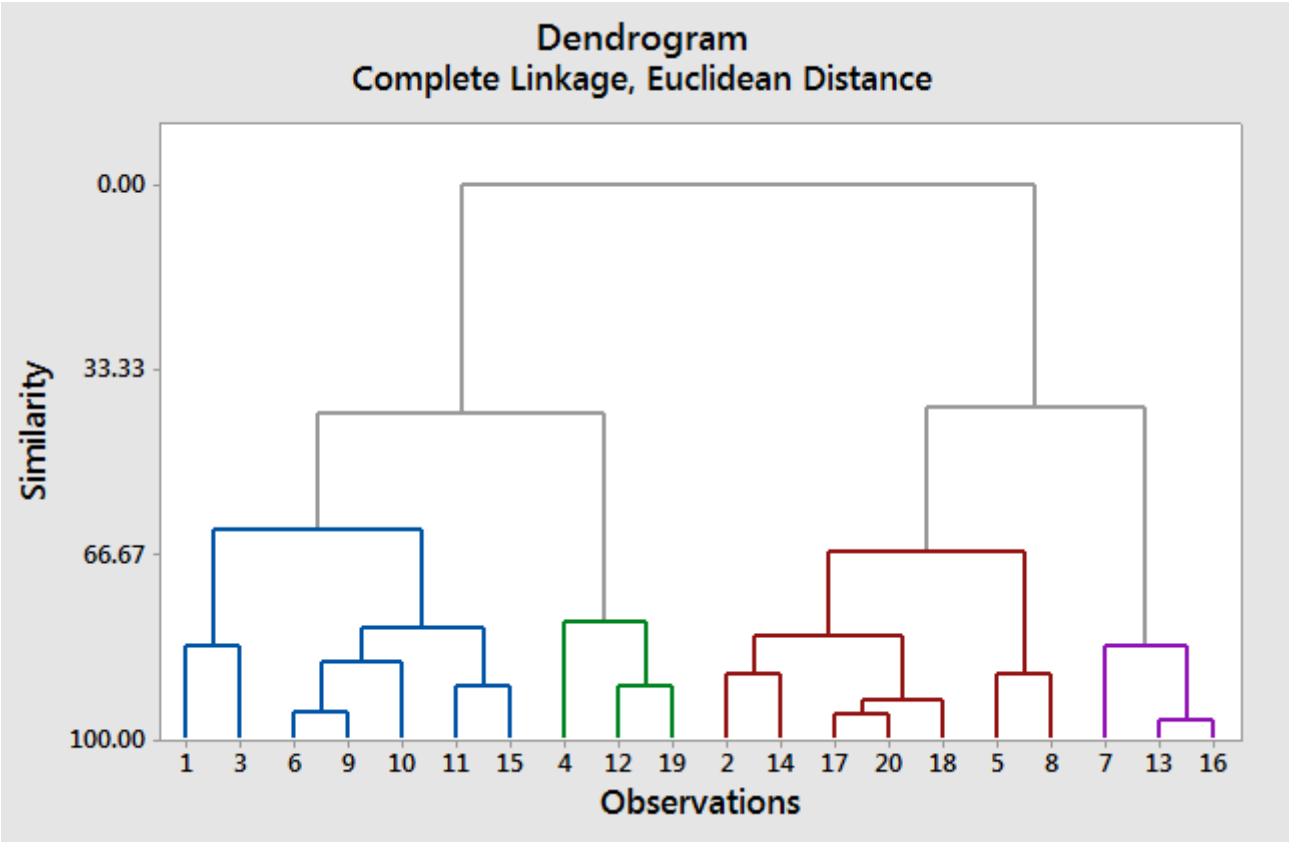
The following picture show what we would obtain if we use K-means clustering in each dataset even if we knew the exact number of clusters beforehand:



It is quite common to take the K-Means algorithm as a benchmark to evaluate the performance of other clustering methods.

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.



Observations that fuse at the bottom are similarm while those that are at the top are quite different. With dendograms, conclutions are made based on the location of the vertical axis rather than on the horizontal one.

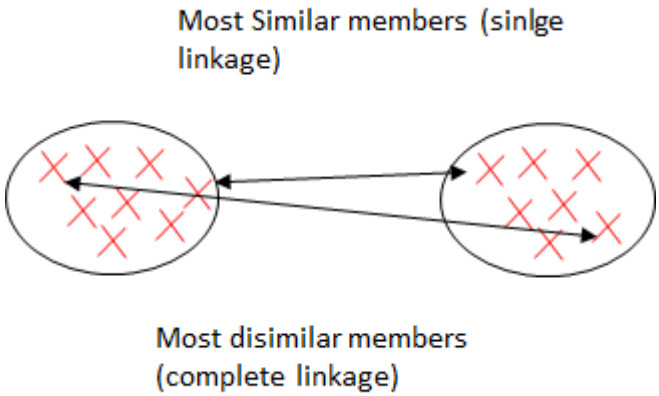
Kinds of Hierarchichal Clustering

There are two approaches to this type of clustering: Aglomerative and divisive.

- Divisive: this method starts by englobing all datapoints in one single cluster. Then, it will split the cluster iteratively into smaller ones until each one of them contains only one sample.
- Agglomerative: this method starts with each sample being a different cluster and then merging them by the ones that are closer from each other until there is only one cluster.

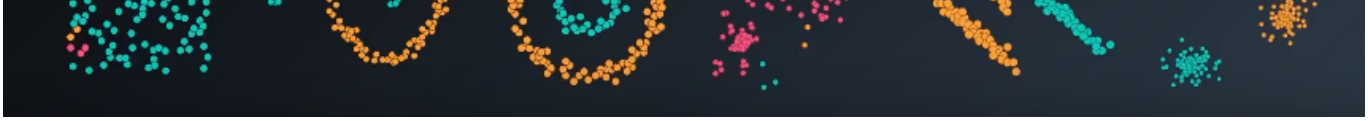
Single Linkage & Complete Linkage

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



- 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.



- Complete Linkage

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

Advantages of Hierarchichal Clustering

- The resulting hierarchichal representations can be very informative.
- Dendograms provide an interesting and informative way of visualization.
- They are specially powerful when the dataset comtains real hierarchichal relationships.

Disadvantages of Hierarchichal Clustering

- They are very sensitive to outliers and, in their presence, the model performance decreases significantly.
- They are very expensive, computationally speaking.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

Density-Based Spatial Clustering of Applications with Noise, or DBSCAN, is another clustering algorithm specially useful to correctly identify noise in data.

DBSCAN Assigning Criteria

It is based on a number of points with a specified radius ε and there is a special label assigned to each datapoint. The process of assigning this label is the following:

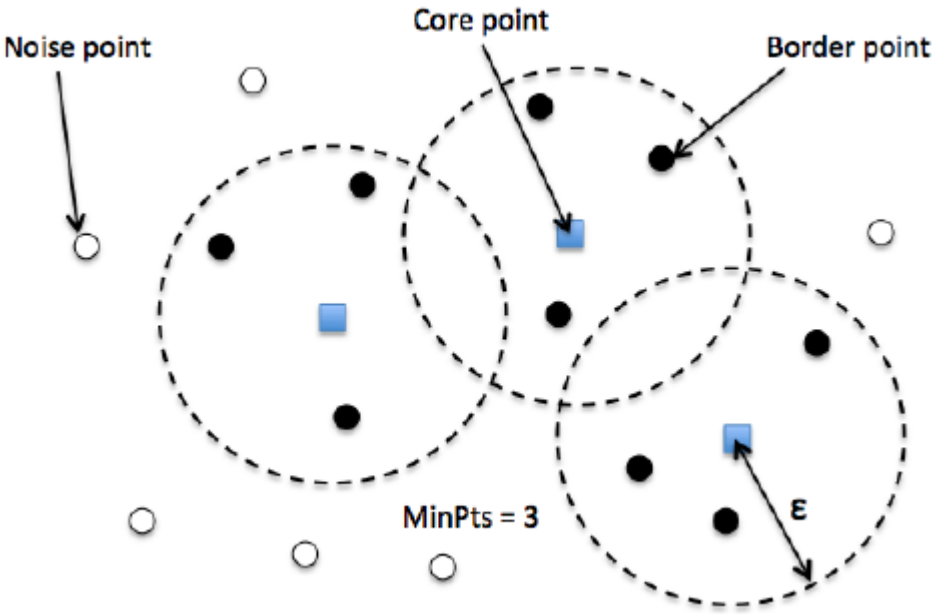
- It is a specified number (MinPts) of neighbour points. A core point will be assigned if there is this MinPts number of points that fall in the ε radius.
- A border point will fall in the ε radius of a core point, but will have less neighbors than the MinPts number.
- Every other point will be noise points.

DBSCAN Algorithm

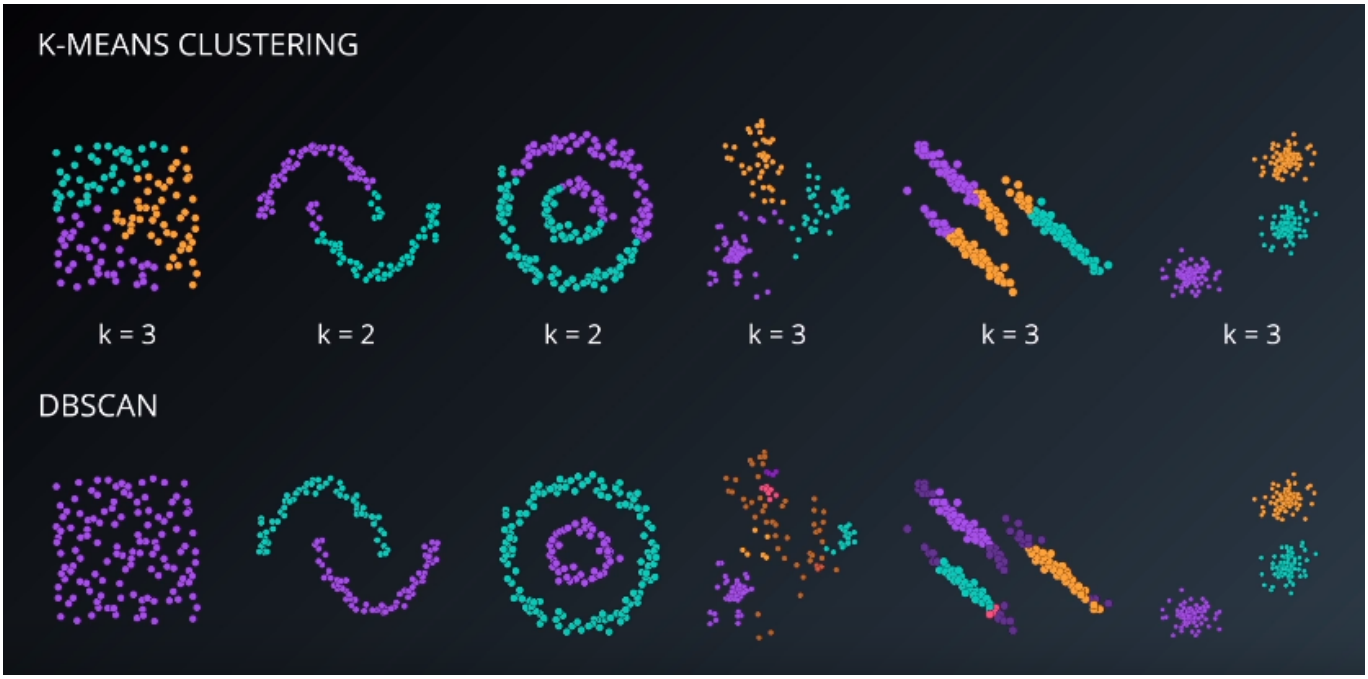
The algorithm follows the logic:

1. Identify a core point and make a group for each one, or for each connected gorup of core points (if they staisfy the criteria to be core point).
2. Identify and assign border points to their respective core points.

The following figure summarize very well this process and the commented notation.



DBSCAN vs K-Means Clustering



DBDSCAN Advantages

- We do not need to specify the number of clusters.

- There is high flexibility in the shapes and sizes that the clusters may adopt.
- It is very useful to identify and deal with noise data and outliers.

DBSCAN Disadvantages

- It faces difficulties when dealing with boirder points that are reachable by two clusters.
- It doesn't find well clusters of varying densities.

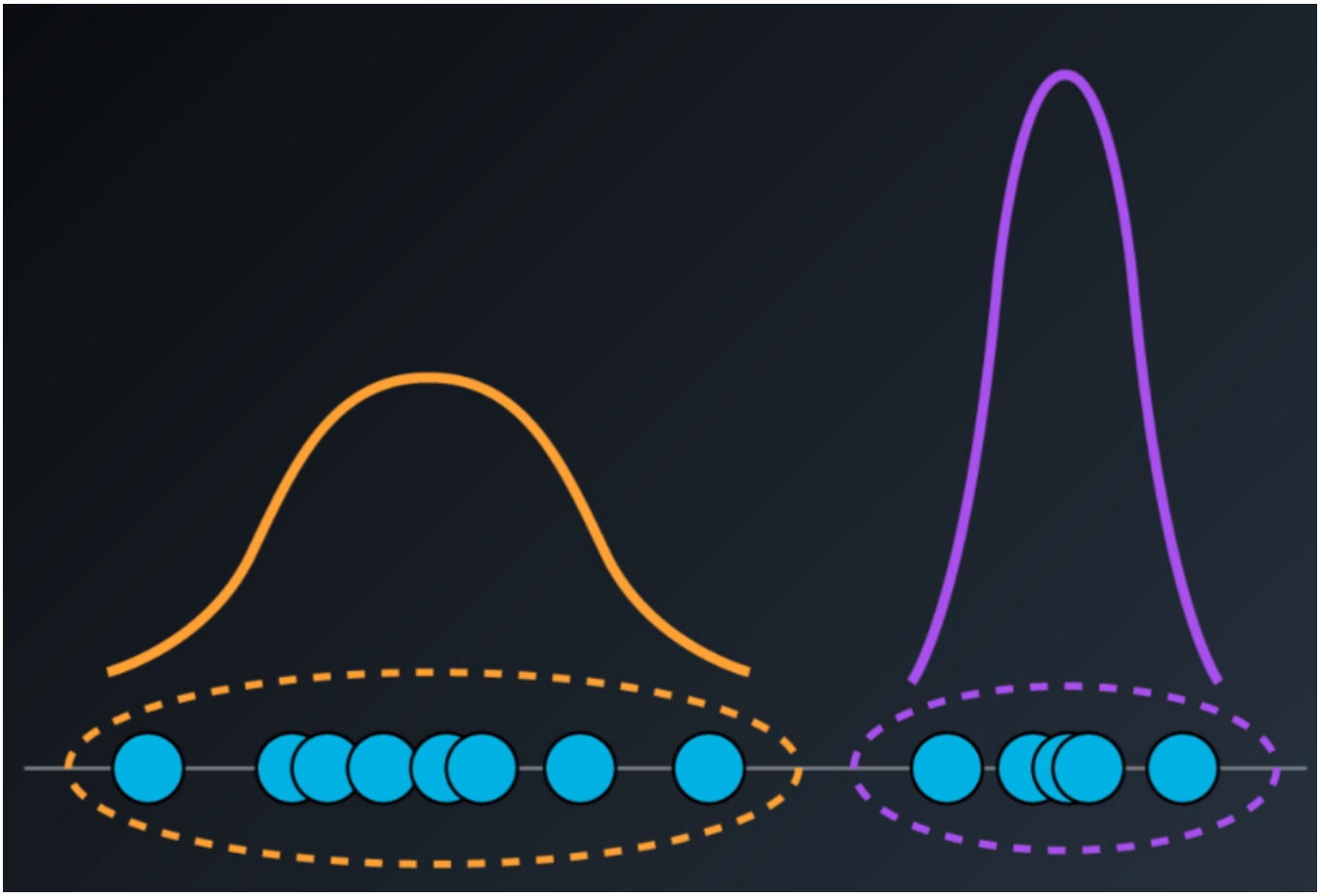
Gaussian Mixture Models (GMM)

Gaussian Mixture Models are probabilistic models that assume that all samples are generated from a mix of a finitite number of Gaussian distribution with unkown parameters.

It belongs to the group of soft clustering algorithms in which every data point will belong to every cluster existing in the dataset, but with different levels of membership to each cluster. This membership is assigned as the probability of belonging to a certain cluster, ranging from 0 to 1.

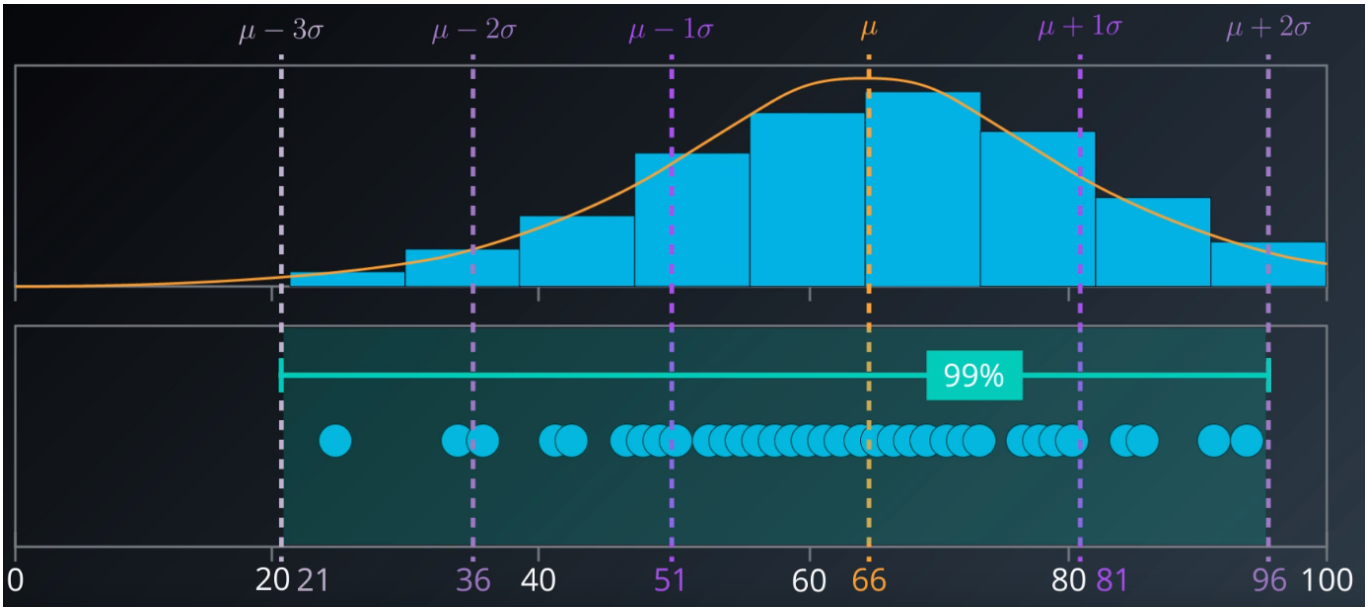
For example, the highlighted point will belong to clusters A and B simultaneoulsy, but with higher membership to the group A, due to its closeness to it.

GMM is one of the most advanced clustering methods that we will study in this series, it assumes that each cluster follows a probabilistic distribution that can be Gaussian or Normal. It is a generalization of K-Means clustering that includes information about the covariance structure of the data as well as the centers of the latent Gaussians.



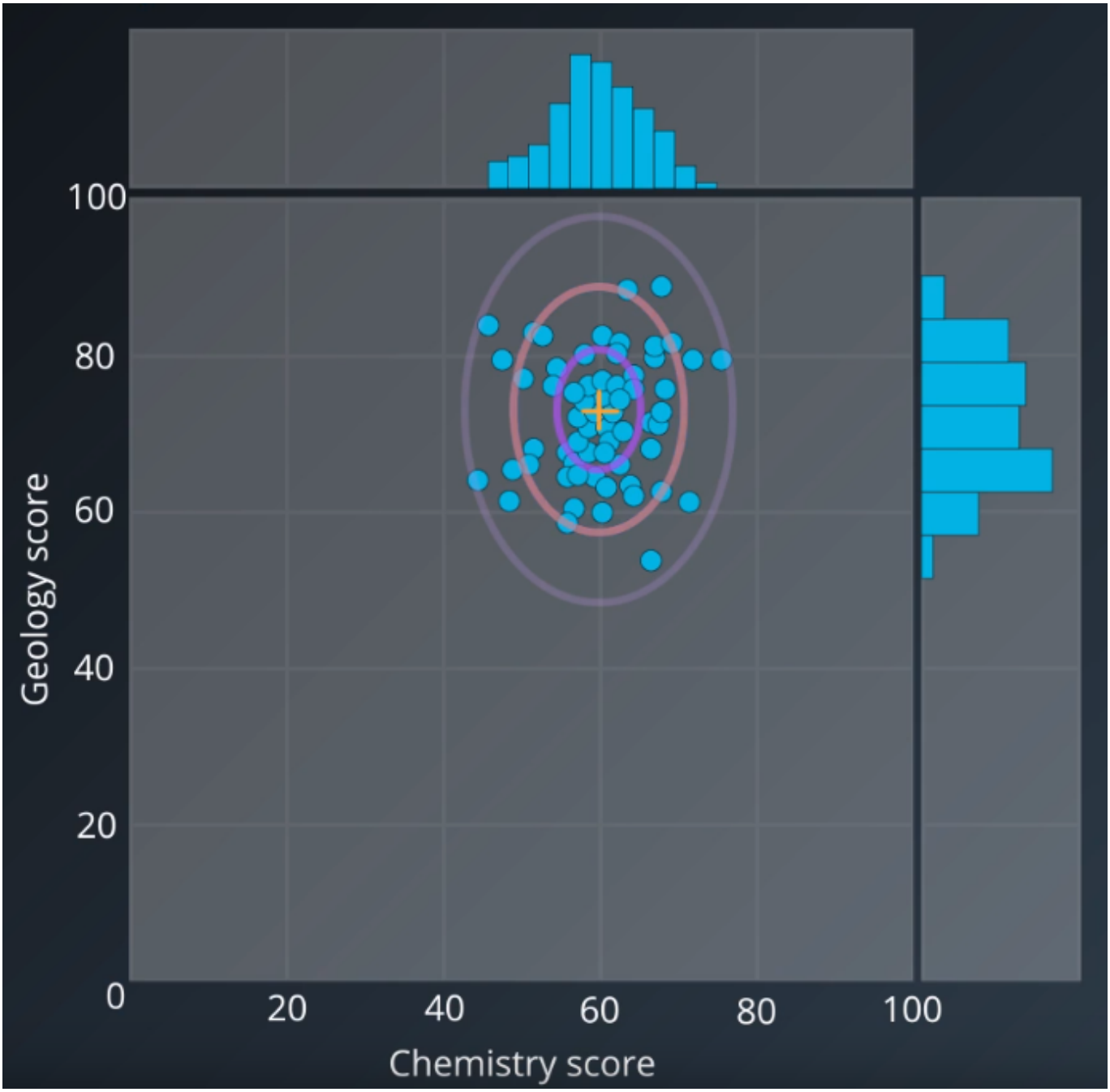
GMM Distribution in One Dimension

The GMM will search for gaussian distributions in the dataset and mixture them.



GMM in Two Dimensions

When having multivariate distributions as the following one, the mean centre would be μ + σ, for each axis of the de dataset distribution.



GMM Algorithm

It is an expectation-maximization algorithm which process could be summarize as follows:

1. Initialize K Gaussian distributions. It does this with the μ (mean) and σ (standard deviation) values. They can be taken from the dataset (naive method) or by applying K-Means.

2. Soft cluster the data: this is the 'Expectation' phase in which all datapoints will be assigned to every cluster with their respective level of membership.

3. Re-estimate the gaussians: this is the 'Maximization' phase in which the expectations are checked and they are used to calculate new parameters for the gaussians: new μ and σ.

4. Evaluate the log-likelihood of the data to check for convergence. The higher the log-likehood is, the more probable is that the mixture of the model we created is likely to fit our dataset. So, this is the function to maximize.

5. Repeat from step 2 until convergence.

GMM Advantages

- It is a soft-clustering method, which assign sample membersips to multiple clusters. This characteristic makes it the fastest algorithm to learn mixture models
- There is high flexibility in the number and shape of the clusters.

GMM Disadvantages

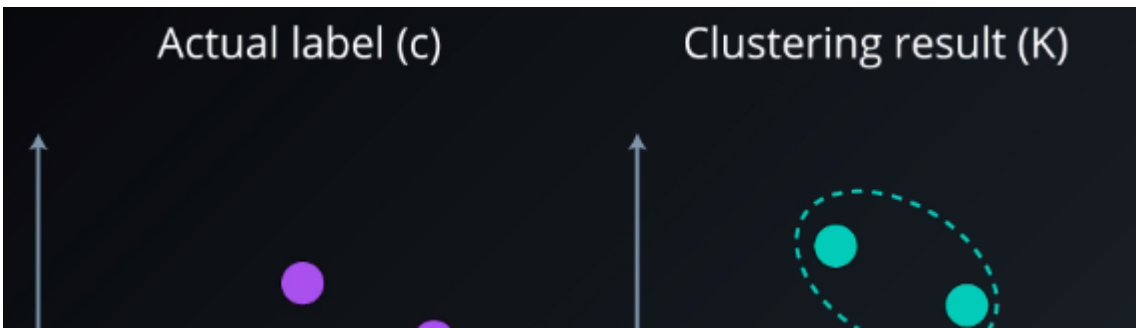
- It is very sensitive to the initial values which will condition greatly its performance.
- GMM may converge to a local minimum, which would be a sub-optimal solution.
- When having insufficient points per mixture, the algorithm diverges and finds solutions with infinite likelihood unless we regularize the covariances between the data points artificially.

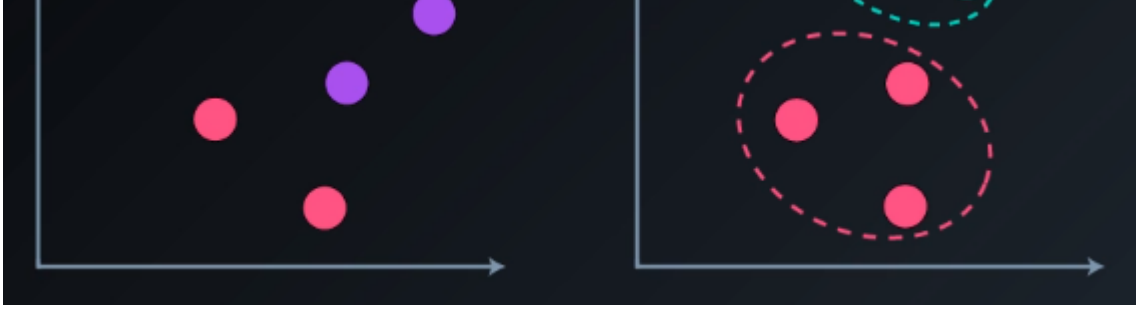
Clustering Validation

Clustering validation is the process of evaluating the result of a cluster objectively and quantitatively. We will do this validation by applying cluster validation indices. There are three main categories:

External Indices

These are scoring methods that we use if the original data was labelled, which is not the most frequent case in this kind of problems. We will match a clusering structure to information known beforehand.





The most used index is the Adjusted Rand index.

- Adjusted Rand Index (ARI) € [-1,1]

To understand it we should first define its components:

$$\text{Rand Index} = \frac{a + b}{\binom{n}{2}}$$

- a: is the number of points that are in the same cluster both in C and in K
- b: is the number of points that are in the different cluster both in C and in K.
- n = is the total number of samples

$$\text{ARI} = \frac{\text{RI} - \text{Expected Index}}{\text{Max(RI)} - \text{Expected Index}}$$

The ARI can get values ranging from -1 to 1. The higher the value, the better it matches the original data.

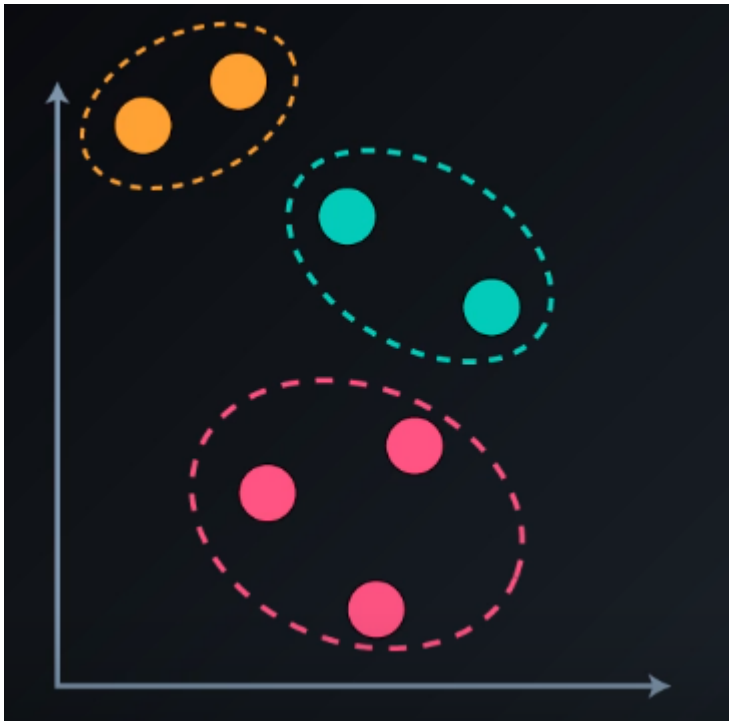
Internal Validation Indices

In unsupervised learning, we will work with unlabeled data and this is when internal indices are more useful.

One of the most common indices is the Silhouette Coefficient.

- Silhouette Coefficient:

There is a Silhouette Coefficient for each data point.



$$Si = \frac{bi - ai}{Max(bi - ai)}$$

- a = average distance to other sample i in the same cluster
- b = average distance to other sample i in closest neighbouring cluster

Sc = average of the sum of Si for each point

The Silhouette Coefficient (SC) can get values from -1 to 1. The higher the value, the better the K selected is. It penalized more if we surpass the ideal K than if we fall short.

It is only suitable for certain algorithms such as K-Means and hierarchical clustering. It is not suitable to work with DBSCAN, we will use DBCV instead.

Conclusion

We have made a first introduction to unsupervised learning and the main clustering algorithms.

In the next article we will walk through an implementation that will serve as an example to build a K-means model and will review and put in practice the concepts explained.

Stay tuned!

Machine Learning Data Science Unsupervised Learning Clustering K Means