Clustering in Machine Learning

Clustering or cluster analysis is a machine learning technique, which groups the unlabelled dataset. It can be defined as "A way of grouping the data points into different clusters, consisting of similar data points. The objects with the possible similarities remain in a group that has less or no similarities with another group."

It does it by finding some similar patterns in the unlabelled dataset such as shape, size, color, behavior, etc., and divides them as per the presence and absence of those similar patterns.

It is an unsupervised learning method, hence no supervision is provided to the algorithm, and it deals with the unlabeled dataset.

After applying this clustering technique, each cluster or group is provided with a cluster-ID. ML systems can use this id to simplify the processing of large and complex datasets.

The clustering technique is commonly used for statistical data analysis.

Some most common uses of this technique are:

Market Segmentation

Statistical data analysis

Social network analysis

Image segmentation

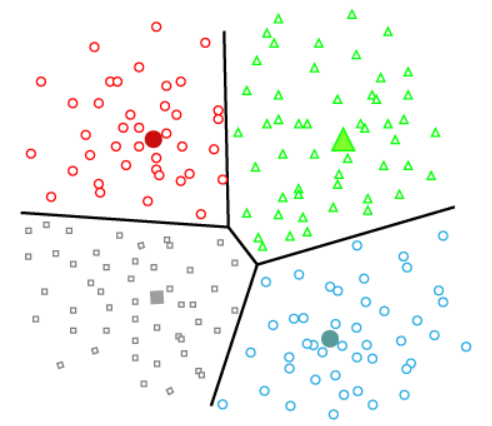
Anomaly detection, etc.

Below are the main clustering methods used in Machine learning:

Partitioning Clustering

It is a type of clustering that divides the data into non-hierarchical groups. Given a data set of N points, a partitioning method constructs K (N ≥ K) partitions of the data, with each partition representing a cluster. It is also known as the centroid-based method. The most common example of partitioning clustering is the K-Means Clustering algorithm.

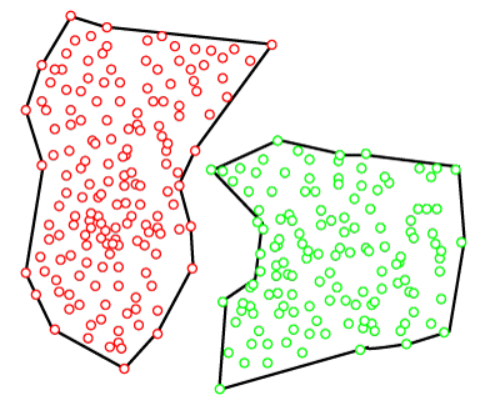
In this type, the dataset is divided into a set of k groups, where K is used to define the number of pre-defined groups. The cluster center is created in such a way that the distance between the data points of one cluster is minimum as compared to another cluster centroid.



Density-Based Clustering

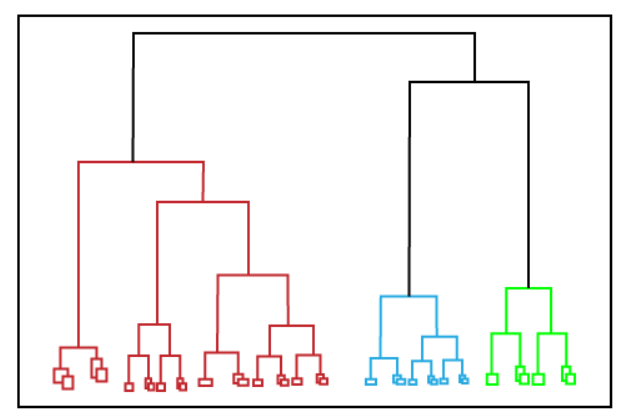
The density-based clustering method connects the highly-dense areas into clusters, and the arbitrarily shaped distributions are formed as long as the dense region can be connected. This algorithm does it by identifying different clusters in the dataset and connects the areas of high densities into clusters. The dense areas in data space are divided from each other by sparser areas.

These algorithms can face difficulty in clustering the data points if the dataset has varying densities and high dimensions.



Hierarchical Clustering

Hierarchical clustering can be used as an alternative for the partitioned clustering as there is no requirement of pre-specifying the number of clusters to be created. In this technique, the dataset is divided into clusters to create a tree-like structure, which is also called a dendrogram. The observations or any number of clusters can be selected by cutting the tree at the correct level. The most common example of this method is the Agglomerative Hierarchical algorithm.



Partitioning clustering:

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of predefined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

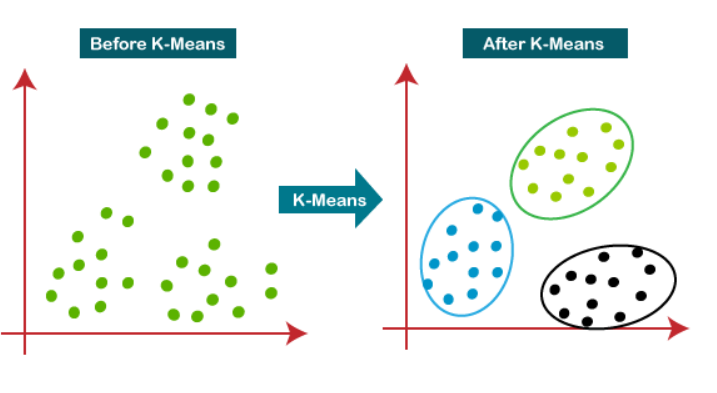
It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means clustering algorithm mainly performs two tasks:

Determines the best value for K center points or centroids by an iterative process.

Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.



How does the K-Means Algorithm Work?

The working of the K-Means algorithm is explained in the below steps:

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be different from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

Step-5: Repeat the third steps, which means assign each datapoint to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

Step-7: The model is ready.

How to choose the value of "K number of clusters" in K-means Clustering?

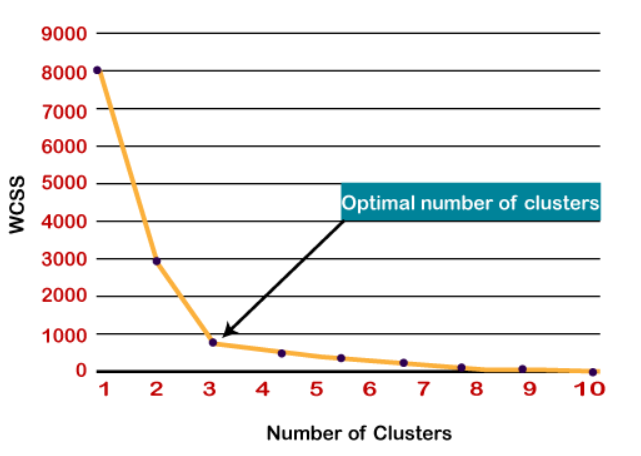
The performance of the K-means clustering algorithm depends upon highly efficient clusters that it forms. But choosing the optimal number of clusters is a big task. There are some different ways to find the optimal number of clusters, but here we are discussing the most appropriate method to find the number of clusters or value of K. The method is given below:

Elbow Method

The Elbow method is one of the most popular ways to find the optimal number of clusters. This method uses the concept of WCSS value. WCSS stands for Within Cluster Sum of Squares, which defines the total variations within a cluster.

To measure the distance between data points and centroid, we can use any method such as Euclidean distance or Manhattan distance.

Since the graph shows the sharp bend, which looks like an elbow, hence it is known as the elbow method. The graph for the elbow method looks like the below image:



K-mediods:

K-medoids is an unsupervised method with unlabelled data to be clustered. It is an improvised version of the K-Means algorithm mainly designed to deal with outlier data sensitivity. Compared to other partitioning algorithms, the algorithm is simple, fast, and easy to implement.

The partitioning will be carried on such that:

Each cluster must have at least one object

An object must belong to only one cluster

A medoid is a most centrally located object in the Cluster or whose average dissimilarity to all the objects is minimum.

Algorithm:

Given the value of k and unlabelled data:

Choose k number of random points from the data and assign these k points to k number of clusters. These are the initial medoids.

For all the remaining data points, calculate the distance from each medoid and assign it to the cluster with the nearest medoid.

Calculate the total cost (Sum of all the distances from all the data points to the medoids)

Select a random point as the new medoid and swap it with the previous medoid. Repeat 2 and 3 steps.

If the total cost of the new medoid is less than that of the previous medoid, make the new medoid permanent and repeat step 4.

If the total cost of the new medoid is greater than the cost of the previous medoid, undo the swap and repeat step 4.

The Repetitions have to continue until no change is encountered with new medoids to classify data points.

Hierarchical:

Data objects are grouped in a bottom-up fashion.

Initially each data object is in its own cluster.

Then merge these atomic clusters into larger and larger clusters, until all of the objects are in a single cluster or until certain termination conditions are satisfied.

Termination condition can be specified by the user, as the desired number of clusters.

## Difference between agglomerative clustering and Divisive clustering :

| S.No. | Parameters | Agglomerative Clustering | Divisive Clustering |
| --- | --- | --- | --- |
| 1. | Category | Bottom-up approach | Top-down approach |
| 2. | Approach | each data point starts in its own cluster, and the algorithm recursively merges the closest pairs of clusters until a single cluster containing all the data points is obtained. | all data points start in a single cluster, and the algorithm recursively splits the cluster into smaller sub-clusters until each data point is in its own cluster. |
| 3. | Complexity level | Agglomerative clustering is generally more computationally expensive, especially for large datasets as this approach requires the calculation of all pairwise distances between data points, which can be computationally expensive. | Comparatively less expensive as divisive clustering only requires the calculation of distances between sub-clusters, which can reduce the computational burden. |
| 4. | Outliers | Agglomerative clustering can handle outliers better than divisive clustering since outliers can be absorbed into larger clusters | divisive clustering may create sub-clusters around outliers, leading to suboptimal clustering results. |
| 5. | Interpretability | Agglomerative clustering tends to produce more interpretable results since the dendrogram shows the merging process of the clusters, and the user can choose the number of clusters based on the desired level of granularity. | Divisive clustering can be more difficult to interpret since the dendrogram shows the splitting process of the clusters, and the user must choose a stopping criterion to determine the number of clusters. |
| 6. | Implementation | Scikit-learn provides multiple linkage methods for agglomerative clustering, such as “ward,” “complete,” “average,” and “single,” | divisive clustering is not currently implemented in Scikit-learn. |
| 7. | Example | Here are some of the applications in which Agglomerative Clustering is used :  Image segmentation, Customer segmentation, Social network analysis, Document clustering, Genetics, genomics, etc., and many more. | Here are some of the applications in which Agglomerative Clustering is used :  Market segmentation, Anomaly detection, Biological classification, Natural language processing, etc. |

A dendrogram is a tree-like diagram used to visualize the relationships between objects, such as clusters of data points. It is commonly used in hierarchical clustering analysis, which is a technique for grouping similar objects or data points based on their proximity to one another. In a dendrogram, each object or cluster is represented by a leaf node, and the distance between the objects or clusters is represented by the length of the branches that connect them. The height of each branch represents the distance between the objects or clusters being connected. A dendrogram can be useful in identifying natural groupings or patterns within a dataset.

Cluster Distance Measures

Single link: smallest distance between an element in one cluster and an element in the other, i.e., d(Ci, Cj) = min{d(xip, xjq)}

Complete link: largest distance between an element in one cluster and an element in the other, i.e., d(Ci, Cj) = max{d(xip, xjq)}

Average: avg distance between elements in one cluster and elements in the other, i.e.,

d(Ci, Cj) = avg{d(xip, xjq)}

DBSCAN:

The DBSCAN algorithm is based on this intuitive notion of “clusters” and “noise”. The key idea is that for each point of a cluster, the neighborhood of a given radius has to contain at least a minimum number of points.

Partitioning methods (K-means, PAM clustering) and hierarchical clustering work for finding spherical-shaped clusters or convex clusters. In other words, they are suitable only for compact and well-separated clusters. Moreover, they are also severely affected by the presence of noise and outliers in the data.

Real life data may contain irregularities, like:

Clusters can be of arbitrary shape such as those shown in the figure below.

Data may contain noise.

DBSCAN algorithm requires two parameters:

eps : It defines the neighborhood around a data point i.e. if the distance between two points is lower or equal to ‘eps’ then they are considered neighbors. If the eps value is chosen too small then large part of the data will be considered as outliers. If it is chosen very large then the clusters will merge and the majority of the data points will be in the same clusters. One way to find the eps value is based on the k-distance graph.

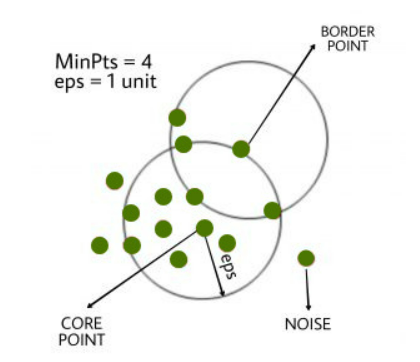
MinPts: Minimum number of neighbors (data points) within eps radius. Larger the dataset, the larger value of MinPts must be chosen. As a general rule, the minimum MinPts can be derived from the number of dimensions D in the dataset as, MinPts >= D+1. The minimum value of MinPts must be chosen at least 3.

In this algorithm, we have 3 types of data points.

Core Point: A point is a core point if it has more than MinPts points within eps.

Border Point: A point which has fewer than MinPts within eps but it is in the neighborhood of a core point.

Noise or outlier: A point which is not a core point or border point



OUTLIERS:

In data mining, an outlier refers to a data point that differs significantly from other data points in the dataset. It is an observation that lies an abnormal distance away from other values in a random sample from a population. They can be caused by measurement or execution error. The analysis of outlier data is referred to as outlier analysis or outlier mining. Outliers can have a significant impact on data analysis, as they can skew results and lead to inaccurate conclusions.

Types of Outliers:

Univariate Outliers: A univariate outlier is a data point that is anomalous with respect to a single variable. For example, in a dataset of exam scores, a student who scores 100% while all other students score below 90% would be a univariate outlier.

Multivariate Outliers: A multivariate outlier is a data point that is anomalous with respect to multiple variables. For example, in a dataset of housing prices, a house that is much larger than all other houses in the neighborhood and also much more expensive would be a multivariate outlier.

Global Outliers: A global outlier is a data point that is anomalous with respect to the entire dataset. This type of outlier is also called a point anomaly, as it represents a single data point that is significantly different from all other data points in the dataset. Global outliers can be either univariate or multivariate.

Detecting Outlier:

Clustering based outlier detection using distance to the closest cluster:

In the K-Means clustering technique, each cluster has a mean value. Objects belong to the cluster whose mean value is closest to it. In order to identify the Outlier, firstly we need to initialize the threshold value such that any distance of any data point greater than it from its nearest cluster identifies it as an outlier for our purpose. Then we need to find the distance of the test data to each cluster mean. Now, if the distance between the test data and the closest cluster to it is greater than the threshold value then we will classify the test data as an outlier.

Algorithm:

Calculate the mean of each cluster

Initialize the Threshold value

Calculate the distance of the test data from each cluster mean

Find the nearest cluster to the test data

If (Distance > Threshold) then, Outlier

Outliers can pose several challenges to data analysis and modeling, including:

Biased Estimates: Outliers can cause bias in the estimated parameters of a model or analysis. Since outliers are often far from the typical observations, they can heavily influence the estimates of central tendency, dispersion, and other measures of the data distribution.

Distorted Visualization: Outliers can also distort the visual representation of data. For example, if a scatter plot includes an outlier, it may obscure the relationship between the other data points.

Skewed Statistical Tests: Outliers can also skew statistical tests such as t-tests or ANOVA, leading to incorrect conclusions about the relationship between variables.

Reduced Model Accuracy: Outliers can reduce the accuracy of predictive models by pulling the model's predictions towards extreme values. For example, a regression model may be thrown off by a single outlier, leading to inaccurate predictions.

Difficulty in Handling: Outliers can be difficult to handle in data analysis because they can arise due to a variety of reasons, such as measurement error, data entry errors, or legitimate extreme values. Determining which outliers are legitimate and which are not can be a challenging task.

Outlier detection is a key task in data mining and can be performed using various methods. Some of the commonly used methods for outlier detection are:

Supervised Outlier Detection: Supervised outlier detection methods rely on labeled data, where the data is divided into two classes: normal and outlier. A model is then trained using the labeled data to classify new data points as normal or outlier. Examples of supervised outlier detection methods include support vector machines, decision trees, and logistic regression.

Unsupervised Outlier Detection: Unsupervised outlier detection methods do not require labeled data and rely on the assumption that outliers are rare and significantly different from normal data points. These methods try to identify the most dissimilar data points from the rest of the data. Examples of unsupervised outlier detection methods include clustering-based methods, density-based methods, and distance-based methods.

Semi-Supervised Outlier Detection: Semi-supervised outlier detection methods are a combination of supervised and unsupervised methods. In semi-supervised methods, some labeled data is used to train a model, and then the model is used to detect outliers in the remaining unlabeled data.

Proximity-Based Outlier Detection: Proximity-based outlier detection methods use the distance or similarity measure between data points to identify outliers. These methods assume that outliers are located far away from the rest of the data points. Examples of proximity-based outlier detection methods include k-nearest neighbor (kNN) and local outlier factor (LOF).

Clustering-Based Outlier Detection: Clustering-based outlier detection methods are based on the assumption that outliers belong to clusters that are significantly different from the majority of the data points. These methods first group data points into clusters and then identify outliers as data points that do not belong to any cluster or belong to a very small cluster. Examples of clustering-based outlier detection methods include DBSCAN and OPTICS.