

# The Ultimate Guide to K-Means Clustering: Definition, Methods and Applications

I love working on [recommendation engines](#). Whenever I encounter any recommendation engine on a website, I can't wait to break it down and understand how it works.

It's one of the many great things about being a data scientist! Artificial Intelligence has revolutionized how we approach data analysis and has led to the development of powerful algorithms such as k-means clustering.

K-means clustering, a part of the unsupervised learning family in AI, is used to group similar data points together in a process known as clustering.

Clustering helps us understand our data in a unique way – by grouping things together into – you guessed it – clusters.

## What is Clustering?

Cluster analysis is a technique used in data mining and machine learning to group similar objects into clusters. K-means clustering is a widely used method for cluster analysis where the aim is to partition a set of objects into K clusters in such a way that the sum of the squared distances between the objects and their assigned cluster mean is minimized.

# K means Clustering

[Unsupervised Machine Learning](#) is the process of teaching a computer to use unlabelled, unclassified data and enabling the algorithm to operate on that data without supervision. Without any previous data training, the machine's job in this case is to organize unsorted data according to parallels, patterns, and variations.

The goal of [clustering](#) is to divide the population or set of data points into a number of groups so that the data points within each group are more comparable to one another and different from the data points within the other groups. It is essentially a grouping of things based on how similar and different they are to one another.

## Types of Clustering

Clustering is a type of unsupervised learning wherein data points are grouped into different sets based on their degree of similarity.

**The various types of clustering are:**

- **Hierarchical clustering**
- **Partitioning clustering**

**Hierarchical clustering is further subdivided into:**

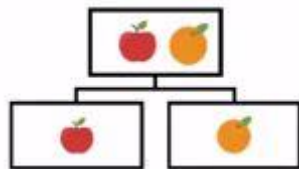
- **Agglomerative clustering**
- **Divisive clustering**

**Partitioning clustering is further subdivided into:**

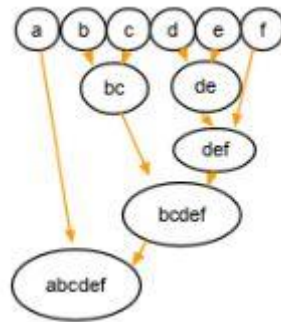
- **K-Means clustering**

## **Hierarchical Clustering**

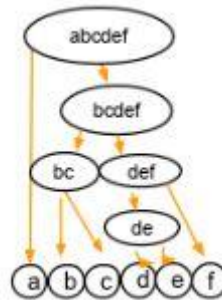
[Hierarchical clustering](#) uses a **tree-like structure**, like so:



**In agglomerative clustering, there is a bottom-up approach. We begin with each element as a separate cluster and merge them into successively more massive clusters, as shown below:**



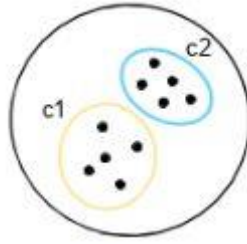
**Divisive clustering is a top-down approach. We begin with the whole set and proceed to divide it into successively smaller clusters, as you can see below:**



## Partitioning Clustering

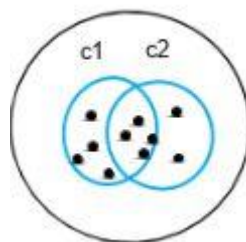
**Partitioning clustering is split into two subtypes - K-Means clustering and Fuzzy C-Means.**

**In k-means clustering, the objects are divided into several clusters mentioned by the number 'K.'** So if we say  $K = 2$ , the objects are divided into two clusters,  $c_1$  and  $c_2$ , as shown:



**Here, the features or characteristics are compared, and all objects having similar characteristics are clustered together.**

**Fuzzy c-means is very similar to k-means in the sense that it clusters objects that have similar characteristics together. In k-means clustering, a single object cannot belong to two different clusters. But in c-means, objects can belong to more than one cluster, as shown.**



## Distance Measure

Distance measure determines **the similarity between two elements and influences the shape of clusters.**

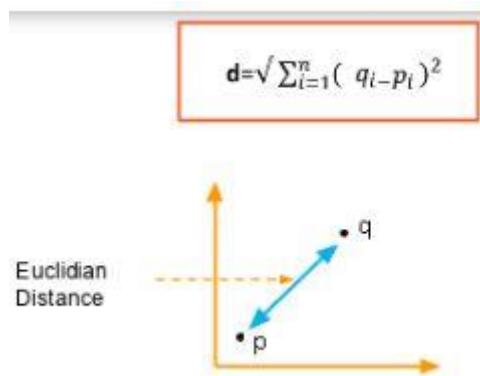
K-Means clustering supports various kinds of distance measures, such as:

- Euclidean distance measure
- Manhattan distance measure
- A squared Euclidean distance measure
- Cosine distance measure

### Euclidean Distance Measure

The most common case is determining the distance between two points. **If we have a point P and point Q, the Euclidean distance is an ordinary straight line.** It is the distance between the two points in Euclidean space.

The formula for distance between two points is shown below:

$$d = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$$


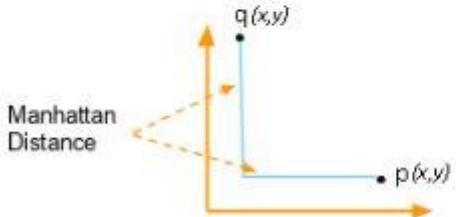
Euclidian Distance

## Manhattan Distance Measure

The Manhattan distance **is the simple sum of the horizontal and vertical components** or the distance between two points measured along axes at right angles.

Note that we are taking the absolute value so that the negative values don't come into play.

The formula is shown below:

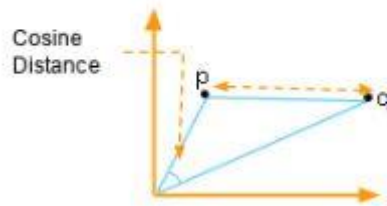
$$d = \sum_{i=1}^n |q_x - p_x| + |q_y - p_y|$$


The diagram shows a 2D coordinate system with x and y axes. Two points are plotted: p(x,y) in the first quadrant and q(x,y) in the second quadrant. A dashed line connects the two points. A right-angled path is highlighted with blue arrows: one horizontal arrow from p(x,y) to the y-axis, and one vertical arrow from that point on the y-axis to q(x,y). An orange arrow points from the text 'Manhattan Distance' to this path. The axes are labeled with orange arrows.

## Cosine Distance Measure

In this case, **we take the angle between the two vectors formed by joining the origin point**. The formula is shown below:

$$d = \frac{\sum_{i=0}^{n-1} q_i - p_x}{\sum_{i=0}^{n-1} (q_i)^2 \times \sum_{i=0}^{n-1} (p_i)^2}$$



## Evaluation Methods

Evaluation methods are used to measure the performance of clustering algorithms. Common evaluation methods include:

**Sum of Squared Errors (SSE):** This measures the sum of the squared distances between each data point and its assigned centroid.

**Silhouette Coefficient:** This measures the similarity of a data point to its own cluster compared to other clusters. A high silhouette coefficient indicates that a data point is well-matched to its own cluster and poorly matched to neighbouring clusters.

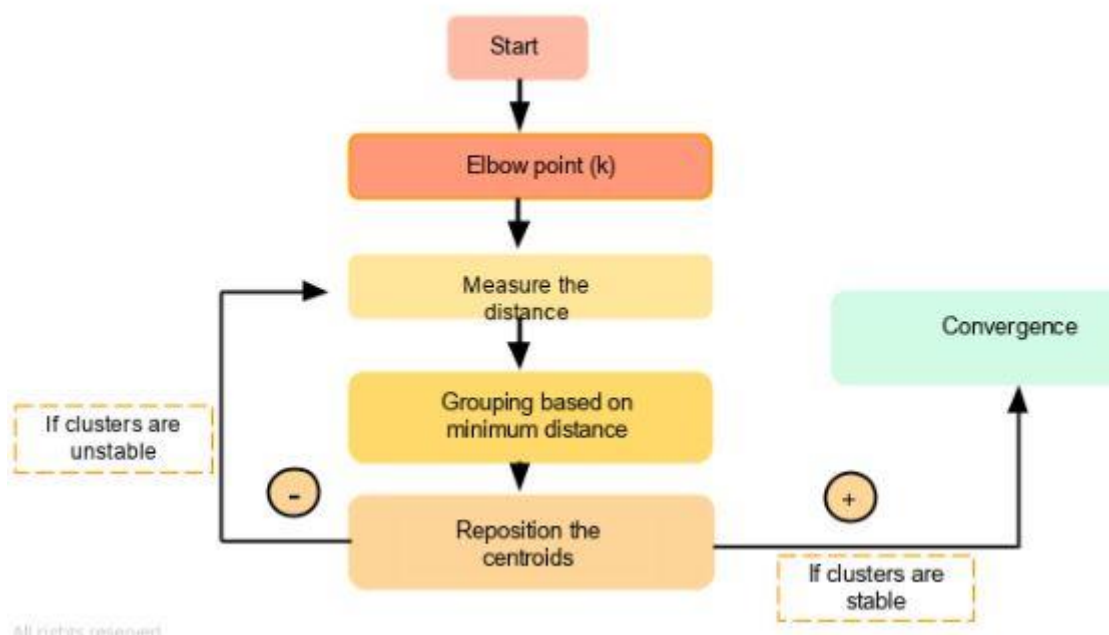


## Silhouette Analysis

Silhouette analysis is a graphical technique used to evaluate the quality of the clusters generated by a clustering algorithm. **It involves calculating the silhouette coefficient for each data point and plotting them in a histogram.** The width of the histogram indicates the quality of the clustering. A wide histogram indicates that the clusters are well-separated and distinct, while a narrow histogram indicates that the clusters are poorly separated and may overlap.

## How Does K-Means Clustering Work?

The flowchart below shows how k-means clustering works:



## Overview

- DBSCAN clustering is an underrated yet super useful clustering algorithm for unsupervised learning problems

### What Exactly is DBSCAN Clustering?

**DBSCAN** stands for **Density-Based Spatial Clustering of Applications with Noise**.

**DBSCAN requires only two parameters: *epsilon* and *minPoints*.**

***Epsilon*** is the radius of the circle to be created around each data point to check the density and ***minPoints*** is the minimum number of data points required inside that circle for that data point to be classified as a **Core** point.

DBSCAN creates a circle of *epsilon* radius around every data point and classifies them into **Core** point, **Border** point, and **Noise**. A data point is a **Core** point if the circle around it contains at least '*minPoints*' number of points. If the number of points is less than *minPoints*, then it is classified as **Border** Point, and if there are no other data points around any data point within *epsilon* radius, then it is treated as **Noise**.

