# K-means Clustering Algorithm: Applications, Types, & How Does It Work?

Lesson 17 of 39

Last updated on Aug 13, 2024

## Table of Contents



Every Machine Learning engineer wants to achieve accurate predictions with their algorithms. Such learning algorithms are generally broken down into two types - supervised and unsupervised. K-Means clustering is one of the unsupervised algorithms where the available input data does not have a labeled response.

## 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:

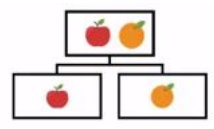
- Agglomerative clustering
- Divisive clustering

Partitioning clustering is further subdivided into:

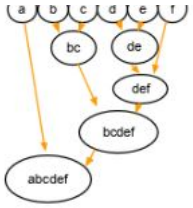
- K-Means clustering
- Fuzzy C-Means clustering

### Hierarchical Clustering

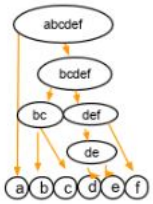
Hierarchical clustering uses a tree-like structure, like so:



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:



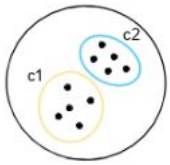

Divisive clustering is a top-down approach. We begin with the whole set and proceed to divide it into successively smaller clusters, as you can see below:




**Partitioning Clustering**

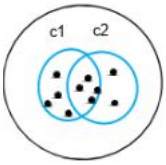
Partitioning clustering is split into two subtypes - K-Means clustering and Fuzzy C-Means.

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, c1 and c2, 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.




### What is Meant by the K-Means Clustering Algorithm?

K-Means clustering is an unsupervised learning algorithm. There is no labeled data for this clustering, unlike in supervised learning. K-Means performs the division of objects into clusters that share similarities and are dissimilar to the objects belonging to another cluster.

The term 'K' is a number. You need to tell the system how many clusters you need to create. For example, K = 2 refers to two clusters. There is a way of finding out what is the best or optimum value of K for a given data.

For a better understanding of k-means, let's take an example from cricket. Imagine you received data on a lot of cricket players from all over the world, which gives information on the runs scored by the player and the wickets taken by them in the last ten matches. Based on this information, we need to group the data into two clusters, namely batsman and bowlers.

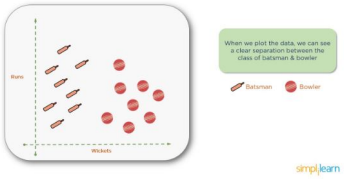
Let's take a look at the steps to create these clusters.

**Solution:**

Assign data points

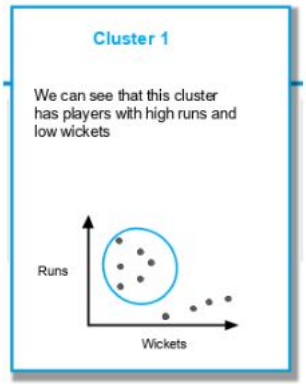
Here, we have our data set plotted on 'x' and 'y' coordinates. The information on the y-axis is about the runs scored, and on the x-axis about the wickets taken by the players.

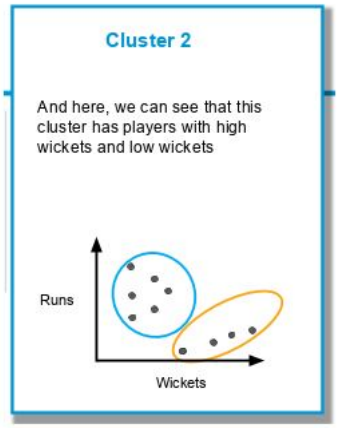
If we plot the data, this is how it would look:




**Perform Clustering**

We need to create the clusters, as shown below:



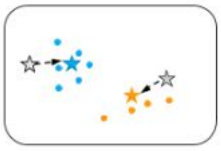




Considering the same data set, let us solve the problem using K-Means clustering (taking K = 2).

The first step in k-means clustering is the allocation of two centroids randomly (as K=2). Two points are assigned as centroids. Note that the points can be anywhere, as they are random points. They are called centroids, but initially, they are not the central point of a given data set.

The next step is to determine the distance between each of the randomly assigned centroids' data points. For every point, the distance is measured from both the centroids, and whichever distance is less, that point is assigned to that centroid. You can see the data points attached to the centroids and represented here in blue and yellow.

The next step is to determine the actual centroid for these two clusters. The original randomly allocated centroid is to be repositioned to the actual centroid of the clusters.



This process of calculating the distance and repositioning the centroid continues until we obtain our final cluster. Then the centroid repositioning stops.

As seen above, the centroid doesn't need anymore repositioning, and it means the algorithm has converged, and we have the two clusters with a centroid.

### Advantages of k-means

1. Simple and easy to implement: The k-means algorithm is easy to understand and implement, making it a popular choice for clustering tasks.
2. Fast and efficient: K-means is computationally efficient and can handle large datasets with high dimensionality.
3. Scalability: K-means can handle large datasets with a large number of data points and can be easily scaled to handle even larger datasets.
4. Flexibility: K-means can be easily adapted to different applications and can be used with different distance metrics and initialization methods.

### Disadvantages of K-Means:

1. Sensitivity to initial centroids: K-means is sensitive to the initial selection of centroids and can converge to a suboptimal solution.
2. Requires specifying the number of clusters: The number of clusters k needs to be specified before running the algorithm, which can be challenging in some applications.
3. Sensitive to outliers: K-means is sensitive to outliers, which can have a significant impact on the resulting clusters.

### Applications of K-Means Clustering

K-Means clustering is used in a variety of examples or business cases in real life, like:

- Academic performance
- Diagnostic systems
- Search engines
- Wireless sensor networks

#### Academic Performance

Based on the scores, students are categorized into grades like A, B, or C.

Diagnostic systems

The medical profession uses k-means in creating smarter medical decision support systems, especially in the treatment of liver ailments.

Search engines

Clustering forms a backbone of search engines. When a search is performed, the search results need to be grouped, and the search engines very often use clustering to do this.

Wireless sensor networks

The clustering algorithm plays the role of finding the cluster heads, which collect all the data in its respective cluster.

### Distance Measure

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

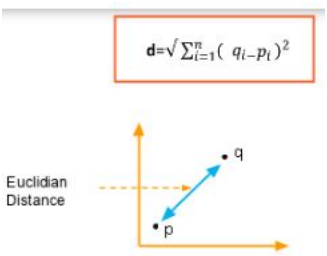
K-Means clustering supports various kinds of distance measures, such as:

- 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:




#### Squared Euclidean Distance Measure

This is identical to the Euclidean distance measurement but does not take the square root at the end. The formula is shown below:

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

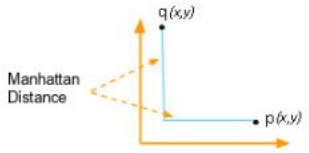
**Manhattan Distance Measure**

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:

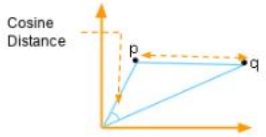
$$d = \sum_{i=1}^{n} |q_x - p_x| + |q_x - p_y|$$




**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:

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




### K-means on Geyser's Eruptions Segmentation

K-means can be used to segment the Geyser's Eruptions dataset, which records the duration and waiting time between eruptions of the Old Faithful geyser in Yellowstone National Park. The algorithm can be used to cluster the eruptions based on their duration and waiting time and identify different patterns of eruptions.

### K-means on Image Compression

K-means can also be used for image compression, where it can be used to reduce the number of colors in an image while maintaining its visual quality. The algorithm can be used to cluster the colors in the image and replace the pixels with the centroid color of each cluster, resulting in a compressed image.

### 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

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

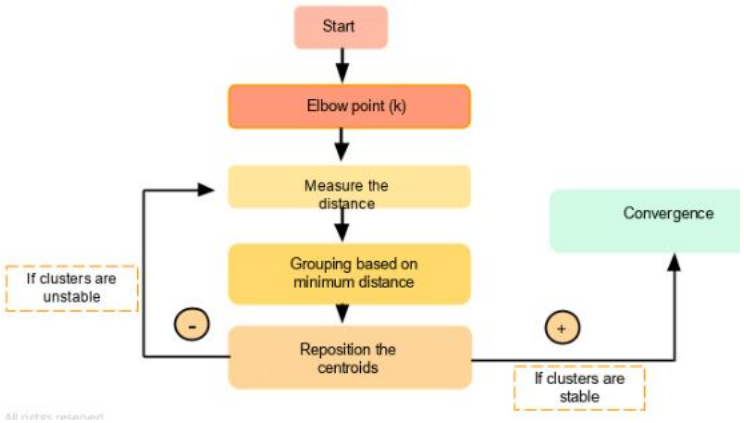
Silhouette Coefficient: This measures the similarity of a data point to its own cluster compared to other clusters. A high silhouette coefficient indicates that a data point is well-matched to its own cluster and poorly matched to neighboring clusters.

### 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:




The goal of the K-Means algorithm is to find clusters in the given input data. There are a couple of ways to accomplish this. We can use the trial and error method by specifying the value of K (e.g., 3,4, 5). As we progress, we keep changing the value until we get the best clusters.

Another method is to use the Elbow technique to determine the value of K. Once we get the K's value, the system will assign that many centroids randomly and measure the distance of each of the data points from these centroids. Accordingly, it assigns those points to the corresponding centroid from which the distance is minimum. So each data point will be assigned to the centroid, which is closest to it. Thereby we have a K number of initial clusters.

For the newly formed clusters, it calculates the new centroid position. The position of the centroid moves compared to the randomly allocated one.

Once again, the distance of each point is measured from this new centroid point. If required, the data points are relocated to the new centroids, and the mean position or the new centroid is calculated once again.

If the centroid moves, the iteration continues indicating no convergence. But once the centroid stops moving (which means that the clustering process has converged), it will reflect the result.

Let's use a visualization example to understand this better.

We have a data set for a grocery shop, and we want to find out how many clusters this has to be spread across. To find the optimum number of clusters, we break it down into the following steps:

**Step 1:**

The Elbow method is the best way to find the number of clusters. The elbow method constitutes running K-Means clustering on the dataset.

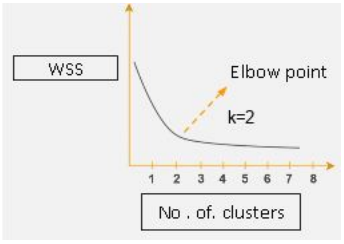
Next, we use within-sum-of-squares as a measure to find the optimum number of clusters that can be formed for a given data set. Within the sum of squares (WSS) is defined as the sum of the squared distance between each member of the cluster and its centroid.

$$WSS = \sum_{i=1}^{m} (x_i - c_i)^2$$

Where $x_i$ = data point and $c_i$ = closest point to centroid

The WSS is measured for each value of K. The value of K, which has the least amount of WSS, is taken as the optimum value.

Now, we draw a curve between WSS and the number of clusters.



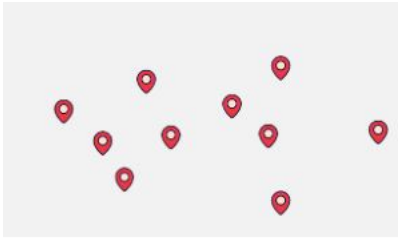

Here, WSS is on the y-axis and number of clusters on the x-axis.

You can see that there is a very gradual change in the value of WSS as the K value increases from 2.

So, you can take the elbow point value as the optimal value of K. It should be either two, three, or at most four. But, beyond that, increasing the number of clusters does not dramatically change the value in WSS, it gets stabilized.

**Step 2:**

Let's assume that these are our delivery points:




We can randomly initialize two points called the cluster centroids.

Here, C1 and C2 are the centroids assigned randomly.

**Step 3:**

Now the distance of each location from the centroid is measured, and each data point is assigned to the centroid, which is closest to it.

This is how the initial grouping is done:


**Step 4:**

Compute the actual centroid of data points for the first group.

**Step 5:**

Reposition the random centroid to the actual centroid.


**Step 6:**

Compute the actual centroid of data points for the second group.

**Step 7:**

Reposition the random centroid to the actual centroid.


**Step 8:**

Once the cluster becomes static, the k-means algorithm is said to be converged.

The final cluster with centroids c1 and c2 is as shown below:

### K-Means Clustering Algorithm

Let's say we have x1, x2, x3…… x(n) as our inputs, and we want to split this into K clusters.

The steps to form clusters are:

Step 1: Choose K random points as cluster centers called centroids.

Step 2: Assign each x(i) to the closest cluster by implementing euclidean distance (i.e., calculating its distance to each centroid)

Step 3: Identify new centroids by taking the average of the assigned points.

Step 4: Keep repeating step 2 and step 3 until convergence is achieved

Let's take a detailed look at it at each of these steps.

**Step 1:**

We randomly pick K (centroids). We name them c1,c2,..... ck, and we can say that

$$C = c_1, c_2, \ldots, c_k$$

Where C is the set of all centroids.

**Step 2:**

We assign each data point to its nearest center, which is accomplished by calculating the euclidean distance.

$$\arg\min_{c_i \in C} dist(c_i, x)^2$$

Where dist() is the Euclidean distance.

Here, we calculate each x value's distance from each c value, i.e. the distance between x1-c1, x1-c2, x1-c3, and so on. Then we find which is the lowest value and assign x1 to that particular centroid.

Similarly, we find the minimum distance for x2, x3, etc.

**Step 3:**

We identify the actual centroid by taking the average of all the points assigned to that cluster.

$$c_i = \frac{1}{|S_i|} \sum_{x_i \in S_i} x_i$$

Where Si is the set of all points assigned to the ith cluster.

It means the original point, which we thought was the centroid, will shift to the new position, which is the actual centroid for each of these groups.

**Step 4:**

Keep repeating step 2 and step 3 until convergence is achieved.

### How to Choose the Value of "K number of clusters" in K-Means Clustering?

Although there are many choices available for choosing the optimal number of clusters, the Elbow Method is one of the most popular and appropriate methods. The Elbow Method uses the idea of WCSS value, which is short for for Within Cluster Sum of Squares. WCSS defines the total number of variations within a cluster. This is the formula used to calculate the value of WCSS (for three clusters) provided courtesy of Javatpoint:

WCSS= ∑Pi in Cluster1 distance(Pi C1)2 +∑Pi in Cluster2distance(Pi C2)2+∑Pi in CLuster3 distance(Pi C3)2

### Python Implementation of the K-Means Clustering Algorithm

Here's how to use Python to implement the K-Means Clustering Algorithm. These are the steps you need to take:

- Data pre-processing
- Finding the optimal number of clusters using the elbow method
- Training the K-Means algorithm on the training data set
- Visualizing the clusters

1. Data Pre-Processing. Import the libraries, datasets, and extract the independent variables.

```
# importing libraries

import numpy as nm

import matplotlib.pyplot as mtp

import pandas as pd

# Importing the dataset

dataset = pd.read_csv('Mall_Customers_data.csv')

x = dataset.iloc[:, [3, 4]].values
```

2. Find the optimal number of clusters using the elbow method. Here's the code you use:

```
#finding optimal number of clusters using the elbow method

from sklearn.cluster import KMeans

wcss_list= []  #Initializing the list for the values of WCSS

#Using for loop for iterations from 1 to 10.

for i in range(1, 11):

    kmeans = KMeans(n_clusters=i, init='k-means++', random_state= 42)

    kmeans.fit(x)

    wcss_list.append(kmeans.inertia_)
```

```
    wcss_list.append(kmeans.inertia_)

mtp.plot(range(1, 11), wcss_list)

mtp.title('The Elobw Method Graph')

mtp.xlabel('Number of clusters(k)')

mtp.ylabel('wcss_list')

mtp.show()
```

3. Train the K-means algorithm on the training dataset. Use the same two lines of code used in the previous section. However, instead of using i, use 5, because there are 5 clusters that need to be formed. Here's the code:

```
#training the K-means model on a dataset

kmeans = KMeans(n_clusters=5, init='k-means++', random_state= 42)

y_predict= kmeans.fit_predict(x)
```

4. Visualize the Clusters. Since this model has five clusters, we need to visualize each one.

```
#visulaizing the clusters

mtp.scatter(x[y_predict == 0, 0], x[y_predict == 0, 1], s = 100, c = 'blue', label = 'Cluster 1') #for first cluster

mtp.scatter(x[y_predict == 1, 0], x[y_predict == 1, 1], s = 100, c = 'green', label = 'Cluster 2') #for second cluster

mtp.scatter(x[y_predict== 2, 0], x[y_predict == 2, 1], s = 100, c = 'red', label = 'Cluster 3') #for third cluster

mtp.scatter(x[y_predict == 3, 0], x[y_predict == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4') #for fourth cluster

mtp.scatter(x[y_predict == 4, 0], x[y_predict == 4, 1], s = 100, c = 'magenta', label = 'Cluster 5') #for fifth cluster

mtp.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s = 300, c = 'yellow', label = 'Centroid')

mtp.title('Clusters of customers')

mtp.xlabel('Annual Income (k$)')

mtp.ylabel('Spending Score (1-100)')

mtp.legend()

mtp.show()
```

Coding provided by Javatpoint.

### Demo: K-Means Clustering

Problem Statement - Walmart wants to open a chain of stores across the state of Florida, and it wants to find the optimal store locations to maximize revenue.

The issue here is if they open too many stores close to each other, they will not make a profit. But, if the stores are too far apart, they do not have enough sales coverage.

Solution - An organization like Walmart is an e-commerce giant. They already have the addresses of their customers in their database. So they can use this information and perform K-Means Clustering to find the optimal location.

### Conclusion

Considered as the Job of the future, Machine Learning engineers are in-demand as well as highly paid. A report by Marketwatch predicts a machine learning growth rate of over 45% for the period between 2017 and 2025. So, why restrict your learning to merely K-means clustering algorithms? Enroll in Simplilearn's Machine Learning Course and expand your knowledge in the broader concepts of Machine Learning. Get certified and become a part of the Artificial Intelligence talent that companies constantly look forward to.

### Find our Post Graduate Program in AI and Machine Learning Online Bootcamp in top cities:

| Name | Date | Place | |
|---|---|---|---|
| Post Graduate Program in AI and Machine Learning | Cohort starts on 8th Oct 2024, Weekend batch | Your City | View Deta |
| Post Graduate Program in AI and Machine Learning, Chicago | Cohort starts on 11th Oct 2024, Weekend batch | Chicago | View Deta |
| Post Graduate Program in AI and Machine Learning, Houston | Cohort starts on 25th Oct 2024, Weekend batch | Houston | View Deta |

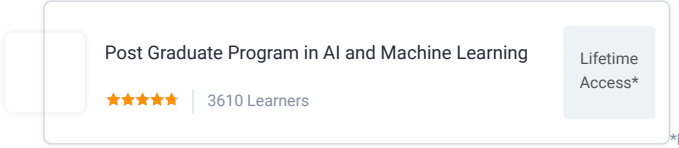
### About the Author

Mayank Banoula

Mayank is a Research Analyst at Simplilearn. He is proficient in Machine learning and Artificial intelligence with python.

View More

### Recommended Programs




*Lifetime access to high-quality, self-paced e-learning content.

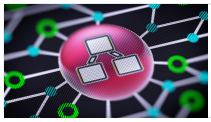
Explore Category

### Find Post Graduate Program in AI and Machine Learning in these cities

Post Graduate Program in AI and Machine Learning, Austin | Post Graduate Program in AI and Machine Learning,

Charlotte | Post Graduate Program in AI and Machine Learning, Chicago | Post Graduate Program in AI and Machine Learning, Dallas | Post Graduate Program in AI and Machine Learning, Houston | Post Graduate Program in AI and Machine Learning, Los Angeles | Post Graduate Program in AI and Machine Learning, NYC | Post Graduate Program in AI and Machine Learning, Raleigh | Post Graduate Program in AI and Machine Learning, San Diego | Post Graduate Program in AI and Machine Learning, San Francisco | Post Graduate Program in AI and Machine Learning, San Jose | Post Graduate Program in AI and Machine Learning, Seattle | Post Graduate Program in AI and Machine Learning, Washington, D.C.

## Recommended Resources



**What is Hierarchical Clustering and How D…**


**Free eBook: 2016 High Paying Certifications**

**K-Means C Algorithm:**