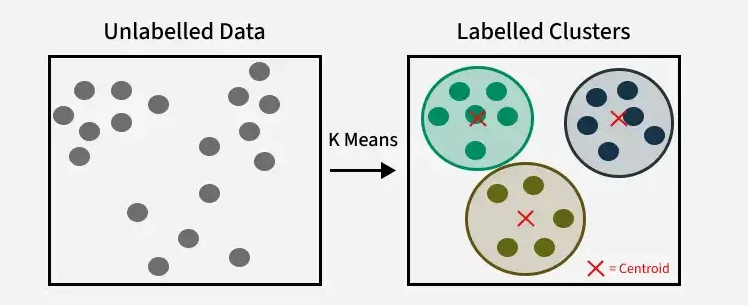
# K-means clustering

[K-Means Clustering](https://www.geeksforgeeks.org/k-means-clustering-introduction/) is an[Unsupervised Machine Learning](https://www.geeksforgeeks.org/ml-types-learning-part-2/) algorithm which groups the unlabeled dataset into different clusters. The article aims to explore the fundamentals and working of k means clustering along with its implementation.

K-means clustering is a technique used to organize data into **groups based on their similarity**. For example **online store uses K-Means to group customers based on purchase frequency and spending creating segments like Budget Shoppers, Frequent Buyers and Big Spenders for personalized marketing.**

The algorithm works by first randomly picking some central points called**centroids**and each data point is then assigned to the closest centroid forming a cluster. After all the points are assigned to a cluster the centroids are updated by finding the average position of the points in each cluster. This process repeats until the centroids stop changing forming clusters. The goal of clustering is to divide the data points into clusters so that similar data points belong to same group.



The algorithm will categorize the items into k groups or clusters of similarity. To calculate that similarity, we will use the [Euclidean distance](https://www.geeksforgeeks.org/euclidean-distance/) as a measurement. The algorithm works as follows:

1. First, we randomly initialize k points, called means or cluster centroids.
2. We categorize each item to its closest mean, and we update the mean’s coordinates, which are the averages of the items categorized in that cluster so far.
3. We repeat the process for a given number of iterations and at the end, we have our clusters.

The “points” mentioned above are called means because they are the mean values of the items categorized in them. To initialize these means, we have a lot of options. An intuitive method is to initialize the means at random items in the data set. Another method is to initialize the means at random values between the boundaries of the data set. For example for a feature *x* the items have values in [0,3] we will initialize the means with values for *x* at [0,3].