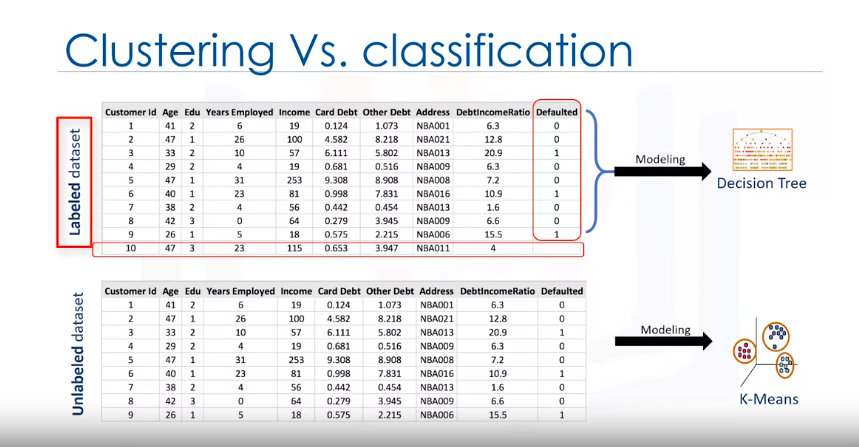
**Clustering Algorithms**

**Week 5**

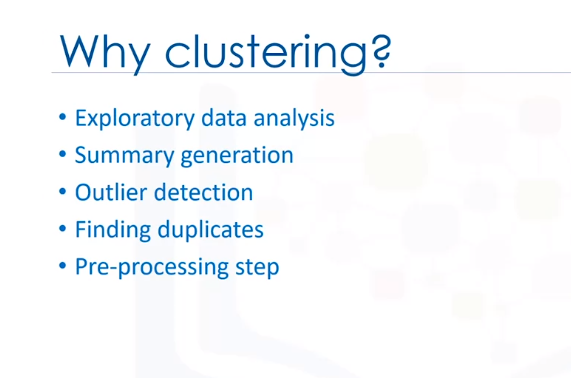
Clustering is an unsupervised machine learning task. You might also hear this referred to as cluster analysis because of the way this method works. Using a clustering algorithm means you're going to give the algorithm a lot of input data with no labels and let it find any groupings in the data it can.

As the examples are unlabeled, clustering relies on unsupervised machine learning. If the examples are labeled, then clustering becomes classification.



**Why Clustering?**

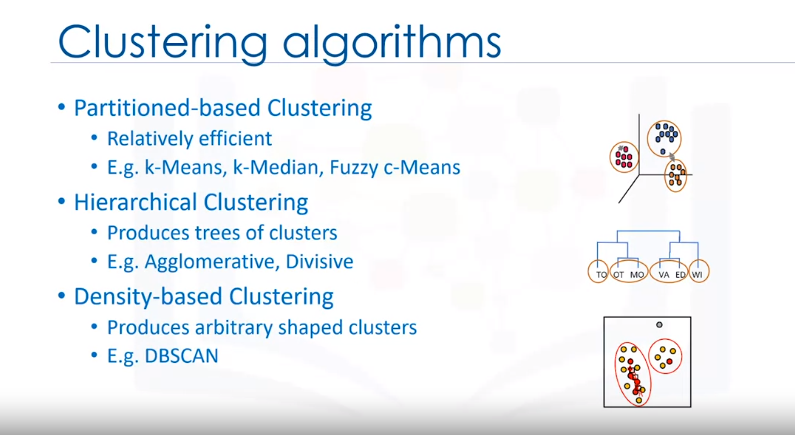
If you look around you can find many other applications of clustering, but generally clustering can be used for one of the following purposes: exploratory data analysis, summary generation or reducing the scale, outlier detection- especially to be used for fraud detection or noise removal, finding duplicates and datasets or as a pre-processing step for either prediction, other data mining tasks or as part of a complex system.



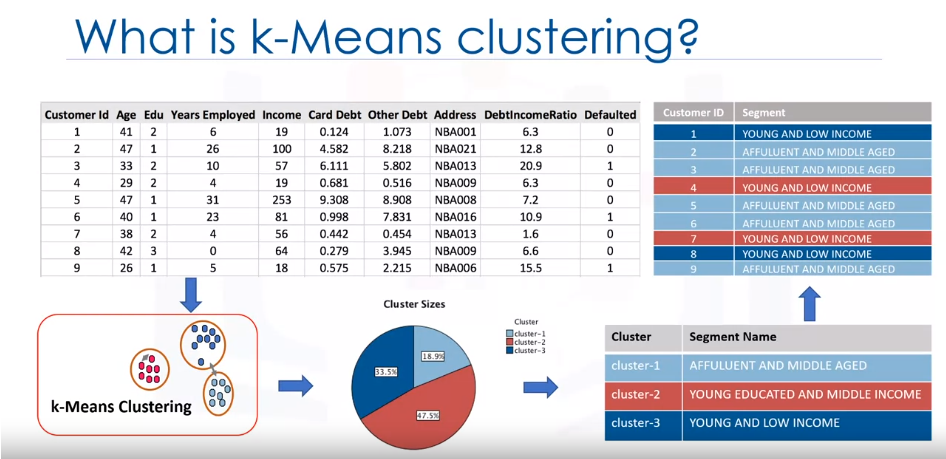
**Partition-based clustering** is a group of clustering algorithms that produces sphere-like clusters, such as; K-Means, K-Medians or Fuzzy c-Means. These algorithms are relatively efficient and are used for medium and large sized databases.

**Hierarchical clustering** algorithms produce trees of clusters, such as agglomerative and divisive algorithms. This group of algorithms are very intuitive and are generally good for use with small size datasets.

**Density-based clustering** algorithms produce arbitrary shaped clusters. They are especially good when dealing with spatial clusters or when there is noise in your data set. For example, the DB scan algorithm.



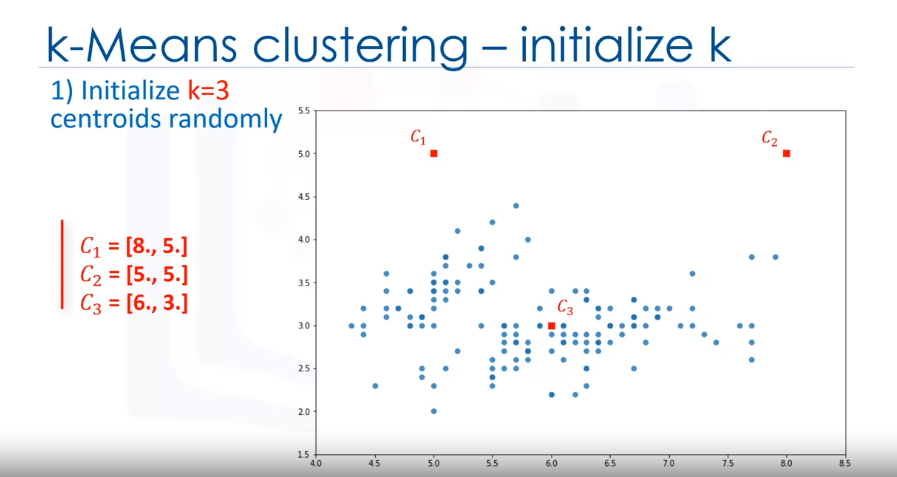
**K-Means Clustering Algorithm/Model**

**Customer segmentation** is the practice of partitioning a customer base into groups of individuals that have similar characteristics. One of the algorithms that can be used for customer segmentation is K-Means clustering. K-Means can group data only unsupervised based on the similarity of customers to each other.

The key concept of the K-Means algorithm is that it randomly picks a center point for each cluster. It means we must initialize K which represents number of clusters. Essentially, determining the number of clusters in a dataset or K is a hard problem in K-Means, that we will discuss later.

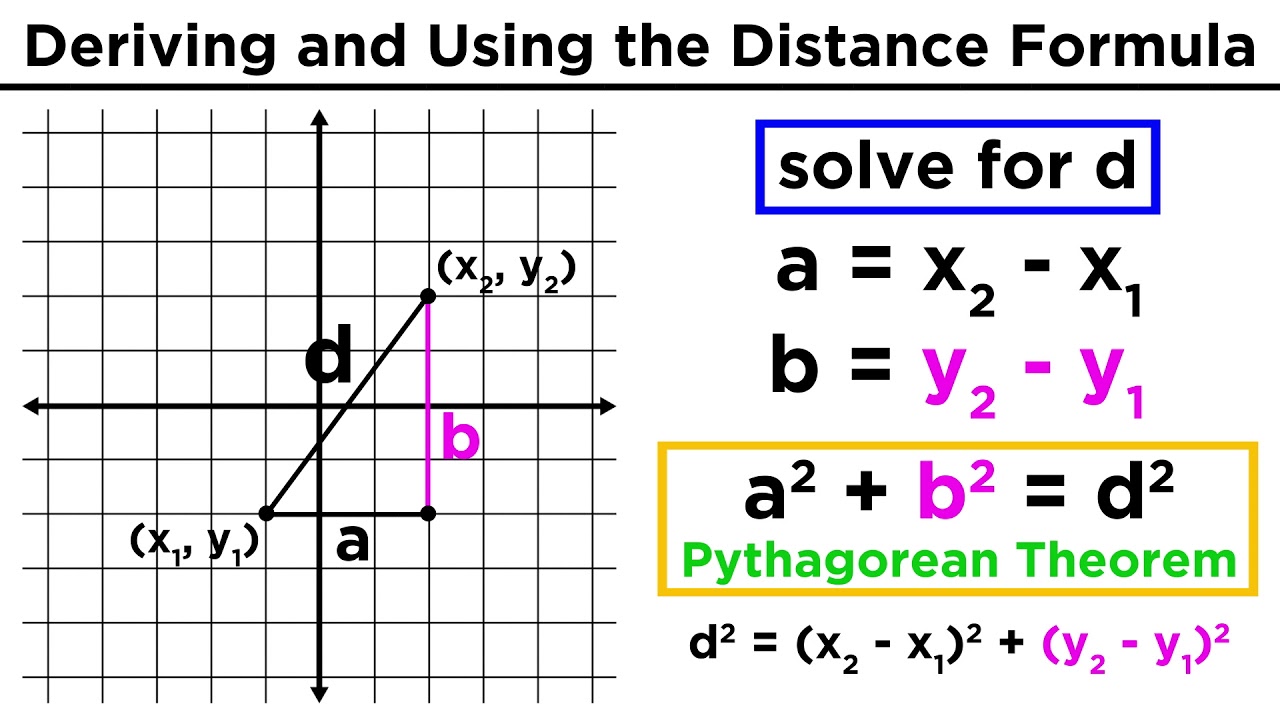
**Step 1 -Initialize K**

For now, let's put K equals three here for our sample dataset. It is like we have three representative points for our clusters. These three data points are called centroids of clusters and should be of same feature size of our customer feature set.

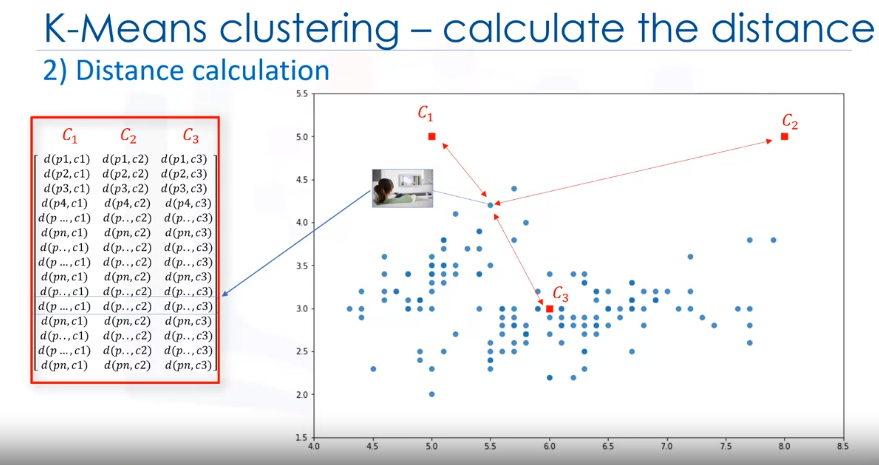


There are two approaches to choose these centroids.

* One, we can randomly choose three observations out of the dataset and use these observations as the initial means.
* Or two, we can create three random points as centroids of the clusters which is our choice that is shown in the plot with red color.

**Step 2 -Distance Calculation**

Therefore, you will form a matrix where each row represents the distance of a customer from each centroid. It is called the Distance Matrix.

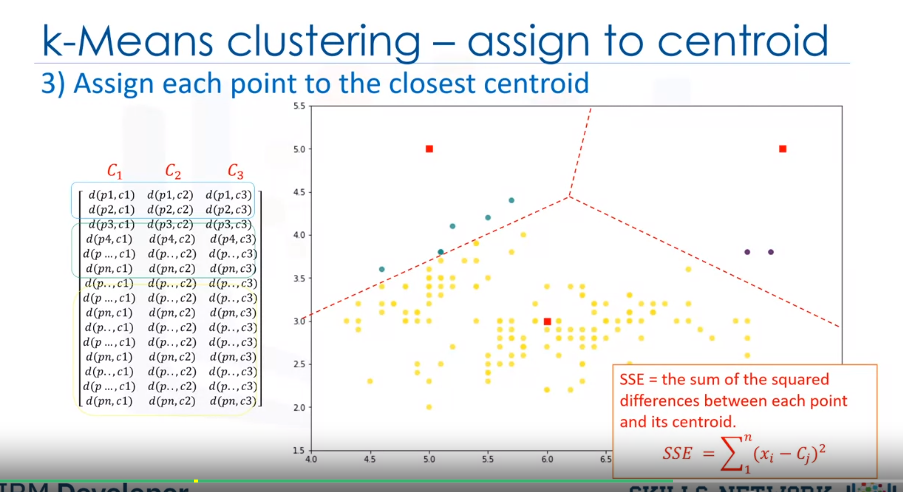


**Step 3 -Assign each point to the closest centroid**

The main objective of K-Means clustering is to minimize the distance of data points from the centroid of this cluster and maximize the distance from other cluster centroids. So, in this step, we have to find the closest centroid to each data point. We can use the distance matrix to find the nearest centroid to datapoints. Finding the closest centroids for each data point, we assign each data point to that cluster. In other words, all the customers will fall to a cluster based on their distance from centroids.

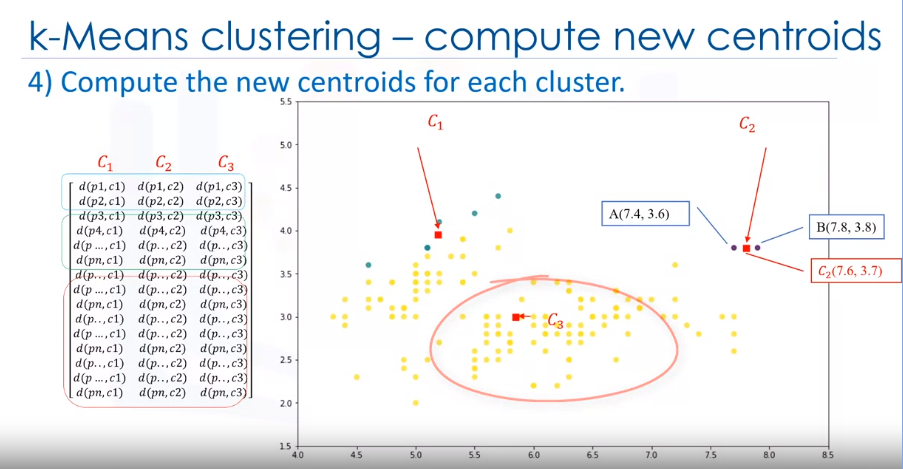
Finding the closest centroids for each data point, we assign each data point to that cluster. In other words, all the customers will fall to a cluster based on their distance from centroids. We can easily say that it does not result in good clusters because the centroids were chosen randomly from the first.

It can be shown as within-cluster sum of squares error. Intuitively, we try to reduce this error. It means we should shape clusters in such a way that the total distance of all members of a cluster from its centroid be minimized.



**Step 4 -Compute new centroids for each cluster**

In the next step, each cluster center will be updated to be the mean for datapoints in its cluster. Indeed, each centroid moves according to their cluster members.

In other words the centroid of each of the three clusters becomes the new mean. For example, if point A coordination is 7.4 and 3.6, and B point features are 7.8 and 3.8, the new centroid of this cluster with two points would be the average of them, which is 7.6 and 3.7. Now, we have new centroids.

**Step 5 -Repeat until there are no more changes**

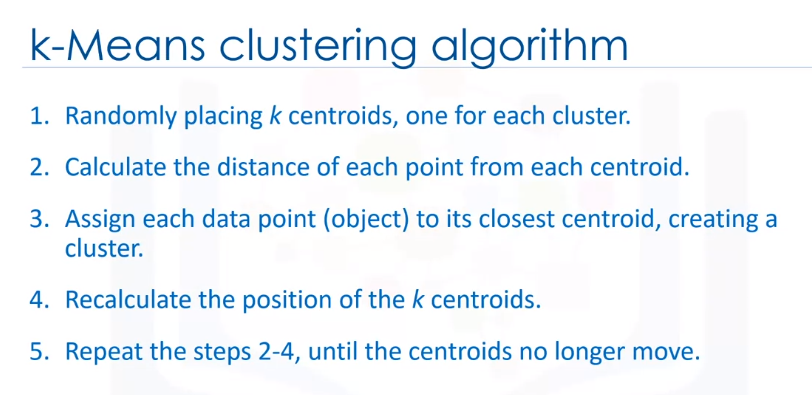
As you can guess, once again we will have to calculate the distance of all points from the new centroids. The points are re-clustered and the centroids move again.

This continues until the centroids no longer move. Please note that whenever a centroid moves, each point's distance to the centroid needs to be measured again.

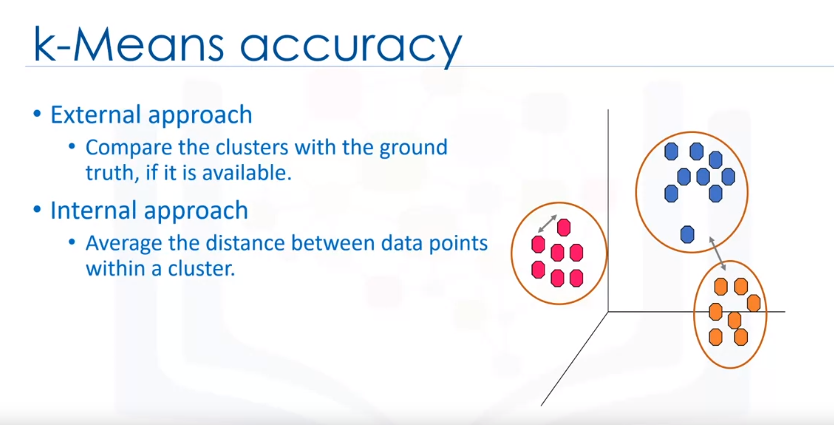


Yes, K-Means is an iterative algorithm and we have to repeat steps two to four until the algorithm converges. In each iteration, it will move the centroids, calculate the distances from new centroids and assign data points to the nearest centroid. It results in the clusters with minimum error or the most dense clusters.

**Summary: k-Means Clustering Algorithm Steps**

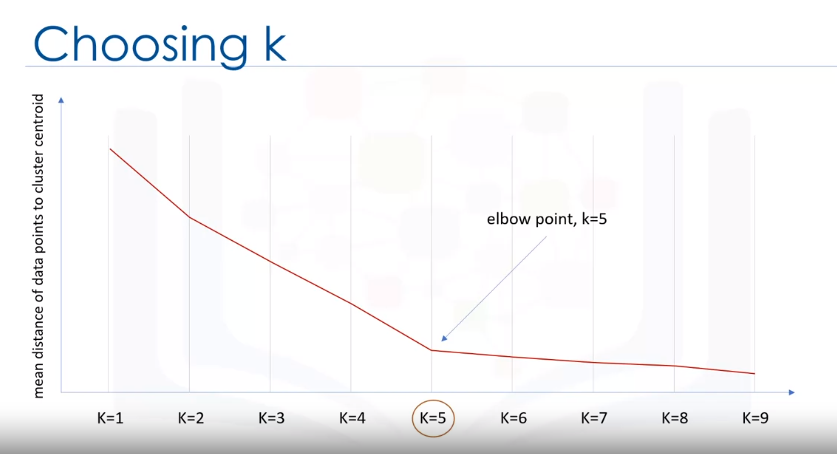


**How can we evaluate the k-means clustering accuracy?**



**Choosing number of clusters k:**

Essentially, determining the number of clusters in a data set, or k as in the k-Means algorithm, is a frequent problem in data clustering. The correct choice of K is often ambiguous because it's very dependent on the shape and scale of the distribution of points in a dataset. There are some approaches to address this problem, but one of the techniques that is commonly used is to run the clustering across the different values of K and looking at a metric of accuracy for clustering.



This metric can be mean, distance between data points and their cluster's centroid (y-axis), which indicate how dense our clusters are or, to what extent we minimize the error of clustering. Then, looking at the change of this metric, we can find the best value for K (x-axis).

But the problem is that with increasing the number of clusters, the distance of centroids to data points will always reduce. This means increasing K will always decrease the error.

So, the value of the metric as a function of K is plotted and the elbow point is determined where the rate of decrease sharply shifts. It is the right K for clustering. This method is called the elbow method.

**Elbow Method: it is a method used to determine the best value of k where rate of decrease sharply shifts.**