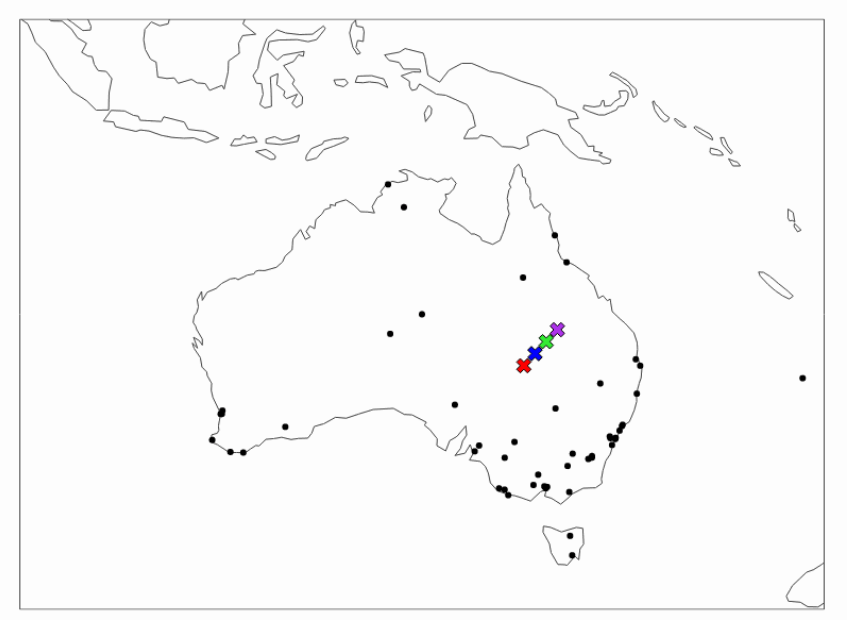
**K-Means.**

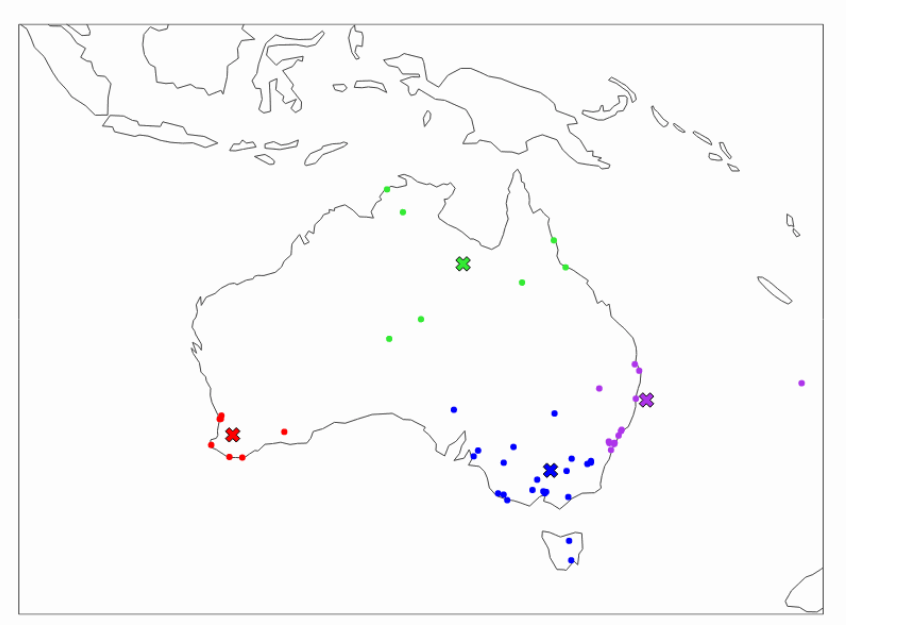
K-Means is one the most well-known and most commonly used algorithms due to its simplicity and speed. Although, at the same time, it has some limitations.

It is designed to make use of a few straightforward steps that are repeated through multiple iterations.

1. The first step for the algorithm is to initialize a defined number of centroids (denoted as X’s in the chart).

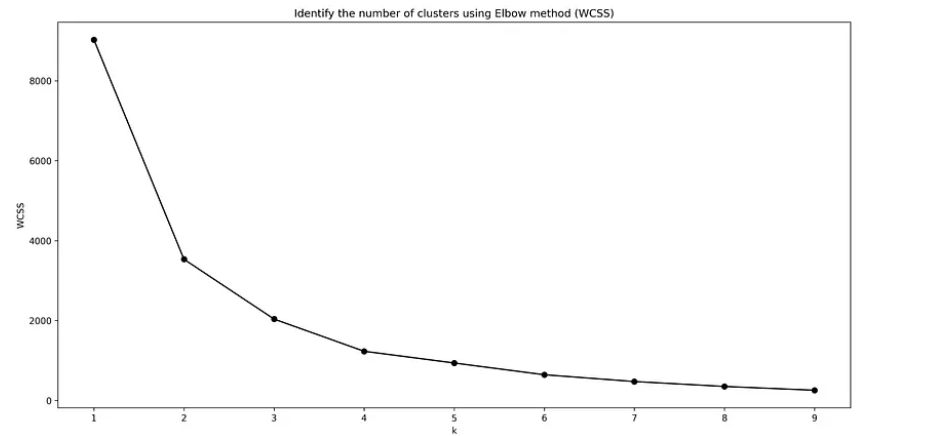


1. The distance between each point (vector) and each centroid is calculated, and then points are assigned to the nearest centroid.
2. Each centroid is then assigned a new position based on the mean of all the points (vectors) in the same group (cluster).
3. The process then repeats with points re-assigned based on new centroid positions and a new mean calculated, leading to new positions for centroids.
4. The iterations continue until the centroids no longer change their position or the defined number of max iterations is reached.



1. **K-Means limitations and what to do about it**
2. **Defining the number of clusters**
3. Before you start the clustering process with K-Means, you need to define how many clusters you want to have. If you have a simple data set with only 2 or 3 dimensions, you can plot the data and visually analyze it to decide the most appropriate number of clusters.
4. However, that is not ideal, especially if you have a much higher number of attributes (dimensions). To help decide on the appropriate number of clusters, you can employ the “elbow” method. For this, you run K-Means clustering multiple times, trying a different number of clusters each time, and record the value of WCSS (Within Cluster Sum of Squares). Note, WCSS is called “inertia” in sklearn.

Once you have done that, plot the WCSS value onto a line chart like the one below:



In the above case, the “elbow” is around k=3 or k=4. Hence, 3 or 4 is the max number of clusters you want to have.

## Centroid initialization

The basic K-Means algorithm initializes centroids randomly. Hence, the result may differ based on the initial position of your centroids and may not be optimal.

To fix this, you can employ the following tactics:

* Initialize centroids multiple times and pick the best run (lowest WCSS)
* Use smart initialization available with k-means++

Both of the above are implemented as default settings in:

<https://scikit-learn.org/stable/modules/generated/sklearn.cluster.k_means.html>

## Distance-based algorithm

K-Means uses the Euclidean distance between different points in space (vectors) to identify which ones belong together. Because of that, you also need to be aware of the following:

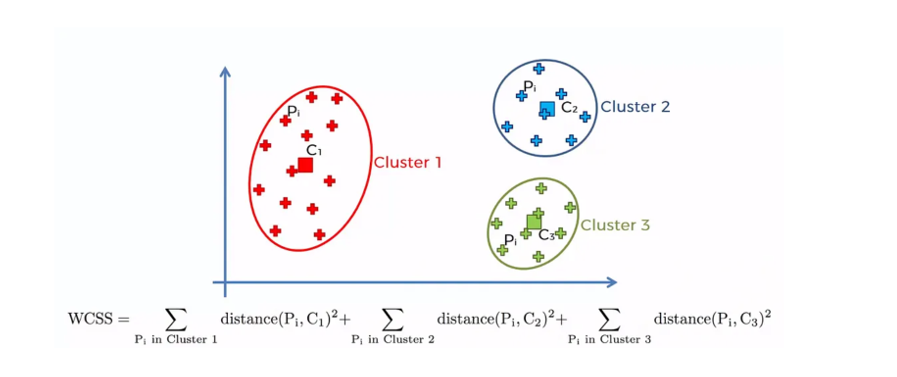
* You can only use numerical attributes in the K-Means algorithm. If you want to use categorical ones for your clustering, you will first need to convert them to numerical. There are multiple ways of doing this, such as using [sklearn’s Ordinal Encoder](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OrdinalEncoder.html" \t "_blank) or similar methods.
* Be mindful of the scale and distribution of the attributes that you are using in your clustering. You may want to first exclude outliers by removing them or introducing “range caps.” You may also want to use [Power Transformation](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PowerTransformer.html#sklearn.preprocessing.PowerTransformer) or [Min-Max scaling](https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.minmax_scale.html#sklearn.preprocessing.minmax_scale) to transform and fit the distribution into the same range for each of your attributes.
* Finally, depending on the type of your data and your use case, you may prefer to use a density-based algorithm instead of a distance-based one.

Density-Based Clustering refers to unsupervised machine learning methods that identify distinctive clusters in the data, based on the idea that a cluster/group in a data space is a contiguous region of high point density, separated from other clusters by sparse regions.

**#the next parameter is n\_init which is the number of times the #K\_means algorithm will be run with different initial centroid.**

**#kmeans inertia\_ attribute is: Sum of squared distances of samples #to their closest cluster center.**

**WCSS, inertia in sklearn is defined as the sum of the squared distance between each member of the cluster and its centroid.**

****

Disadvantages:

* K-Means is highly scale dependent and is not suitable for data of varying shapes and densities.
* Evaluating results is more subjective. It requires much more human evaluation than trusted metrics.

Parameters: <https://scikit-learn.org/0.19/modules/generated/sklearn.cluster.KMeans.html>