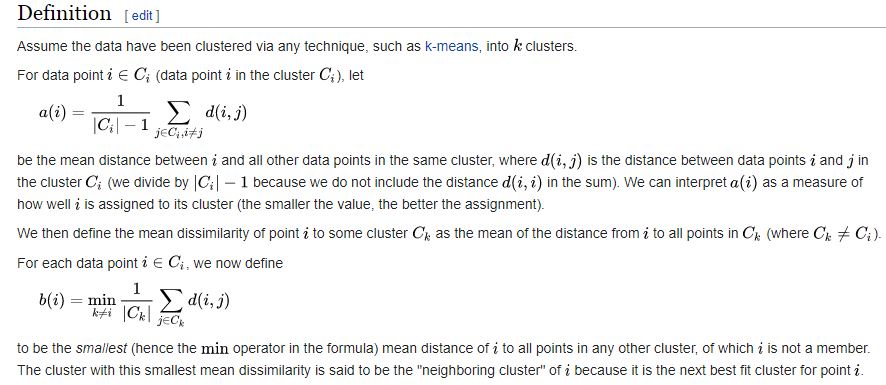
**Clustering**

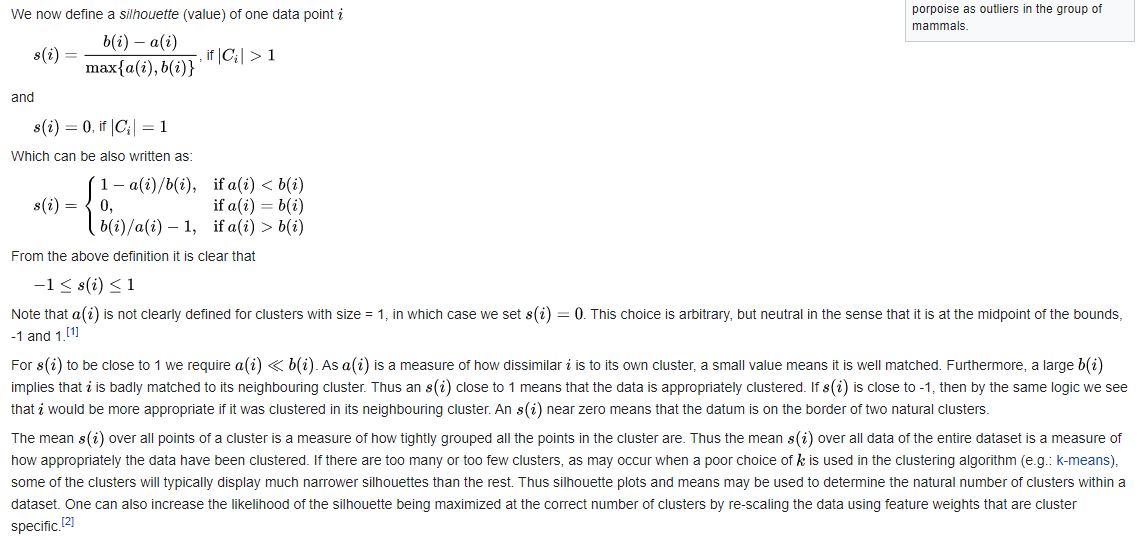
1. **Performance metric in clustering**

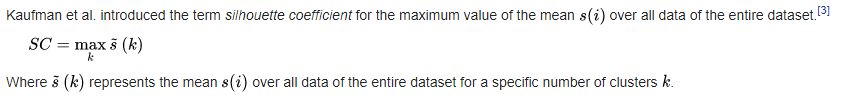
There are some performance metric which can help us evaluate the performance of clustering using any algorithm like k mean or k mean ++ etc.

We will be discussing 2 performance metric . they are:-

1. Dunn Index
2. Silhouette Coefficient:-







For more detail:-

<https://en.wikipedia.org/wiki/Silhouette_(clustering)>

**K – means:-**

It is randomly picking k – pts which leads to a problem called initialization sensitivity. So inorder to tackle this problem we repeat k – means multiple times with different initialization. Each time we compute inertia for each cluster and save it in dictionary. Same way we do it for all . then we plot graph for i iteration . we look for the elbow point in all graph and find the corresponding performance metric value . we can use dunn index or other methon. Then we compare which gives the better result by comparing metric value. The one which gives us best value is our best clustering.

**Drawback of using K-means:-**

Kmeans algorithm is good in capturing structure of the data if clusters have a spherical-like shape. It always try to construct a nice spherical shape around the centroid. That means, the minute the clusters have a complicated geometric shapes, kmeans does a poor job in clustering the data. We’ll illustrate three cases where kmeans will not perform well.

First, kmeans algorithm doesn’t let data points that are far-away from each other share the same cluster even though they obviously belong to the same cluster. Below is an example of data points on two different horizontal lines that illustrates how kmeans tries to group half of the data points of each horizontal lines together.

**Conclusion**

Kmeans clustering is one of the most popular clustering algorithms and usually the first thing practitioners apply when solving clustering tasks to get an idea of the structure of the dataset. The goal of kmeans is to group data points into distinct non-overlapping subgroups. It does a very good job when the clusters have a kind of spherical shapes. However, it suffers as the geometric shapes of clusters deviates from spherical shapes. Moreover, it also doesn’t learn the number of clusters from the data and requires it to be pre-defined. To be a good practitioner, it’s good to know the assumptions behind algorithms/methods so that you would have a pretty good idea about the strength and weakness of each method. This will help you decide when to use each method and under what circumstances. In this post, we covered both strength, weaknesses, and some evaluation methods related to kmeans.

**Below are the main takeaways:**

1. Scale/standardize the data when applying kmeans algorithm.
2. Elbow method in selecting number of clusters doesn’t usually work because the error function is monotonically decreasing for all ks.
3. Kmeans gives more weight to the bigger clusters.
4. Kmeans assumes spherical shapes of clusters (with radius equal to the distance between the centroid and the furthest data point) and doesn’t work well when clusters are in different shapes such as elliptical clusters.
5. If there is overlapping between clusters, kmeans doesn’t have an intrinsic measure for uncertainty for the examples belong to the overlapping region in order to determine for which cluster to assign each data point.
6. Kmeans may still cluster the data even if it can’t be clustered such as data that comes from uniform distributions.

**Limitation of K-means:-**

K-means has problem when cluster are of different:-

1.size

2. densities

3.non- globular shape.

4. k-means tends to have problem when data contain outliers.

**Note: -** evaluating clustering model is not easy primarily because it has no class label. All we depends upon intra and inter cluster distance.