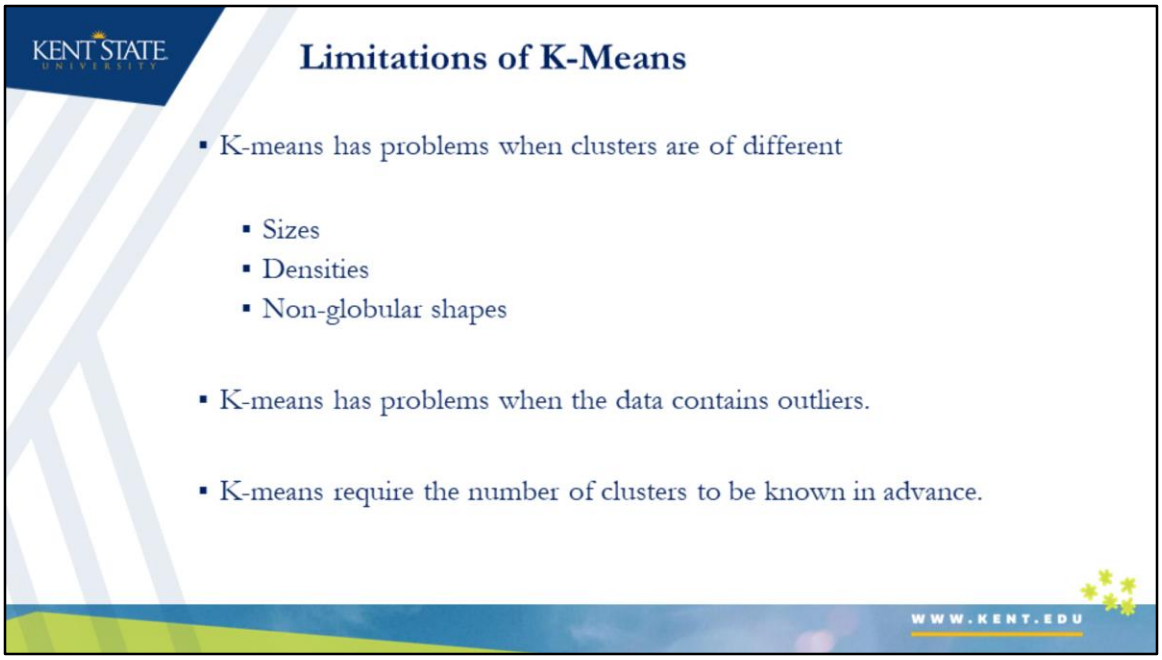
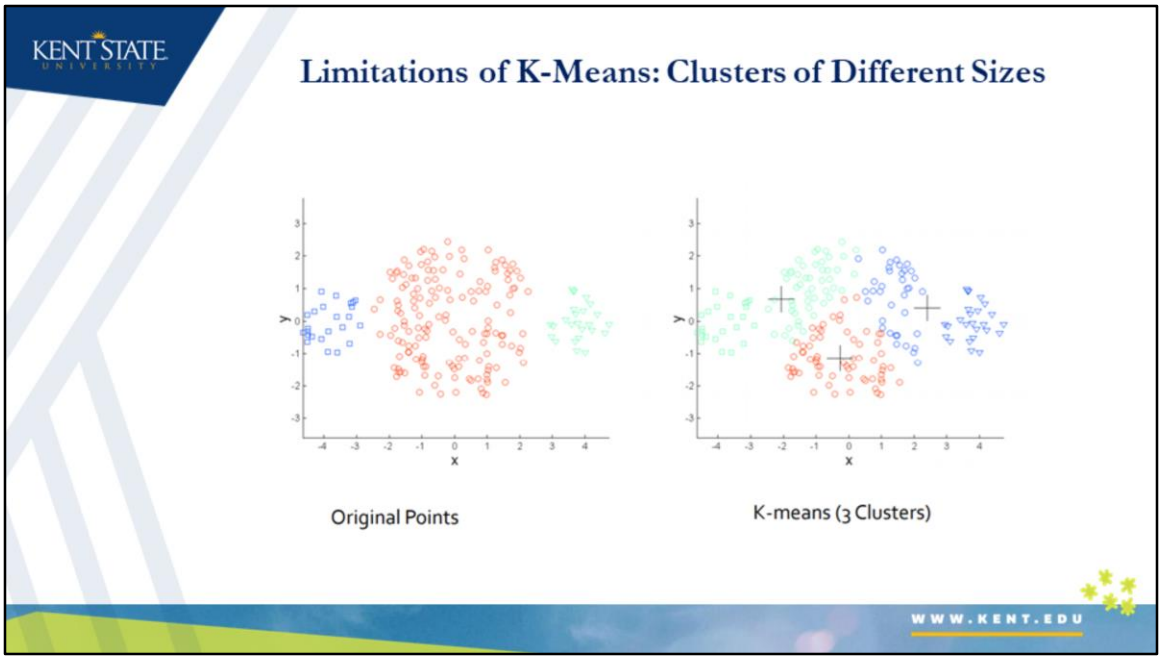
Let us now consider in a little more details the relative advantages and disadvantages of using DBSCAN versus k-means



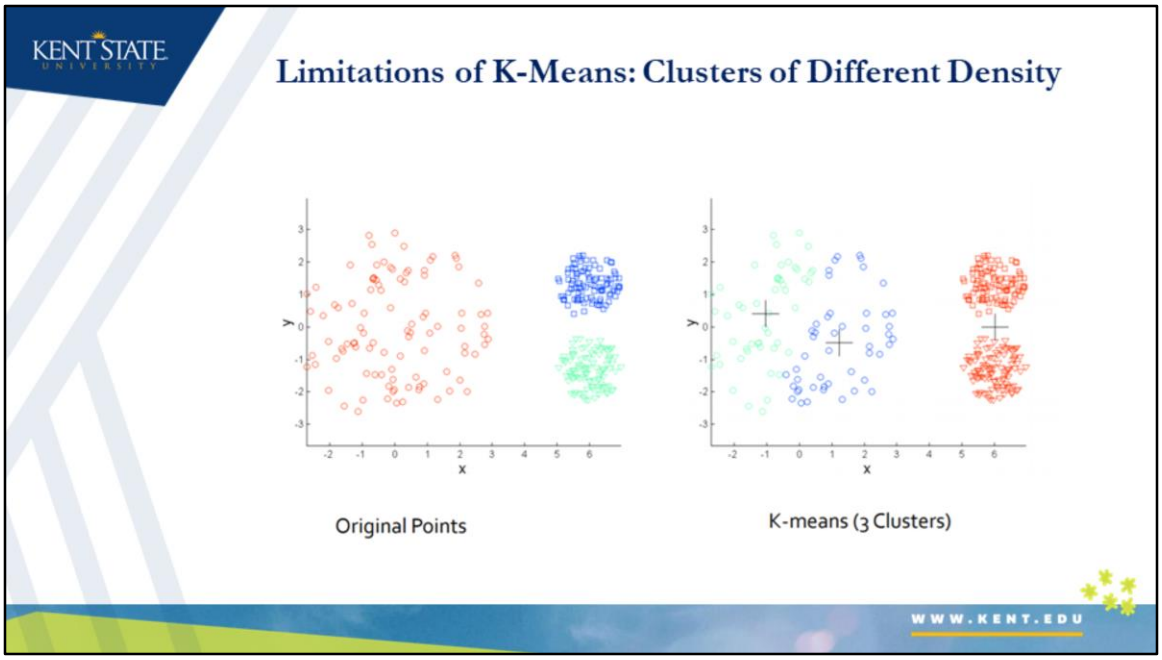
K-means is a computationally efficient algorithm. It is also stochastic, in that each time the algorithm is run, we get a slightly different result. In terms of parameters, the primary parameter is to choose the value of k. But, there are several limitations of using k-means.

1. k-means is sensitive to outliers, as most distance calculation based metrics are
2. k-means cannot handle non-linear clusters well
3. k-means does not do well when there are different densities and shapes of clusters in data

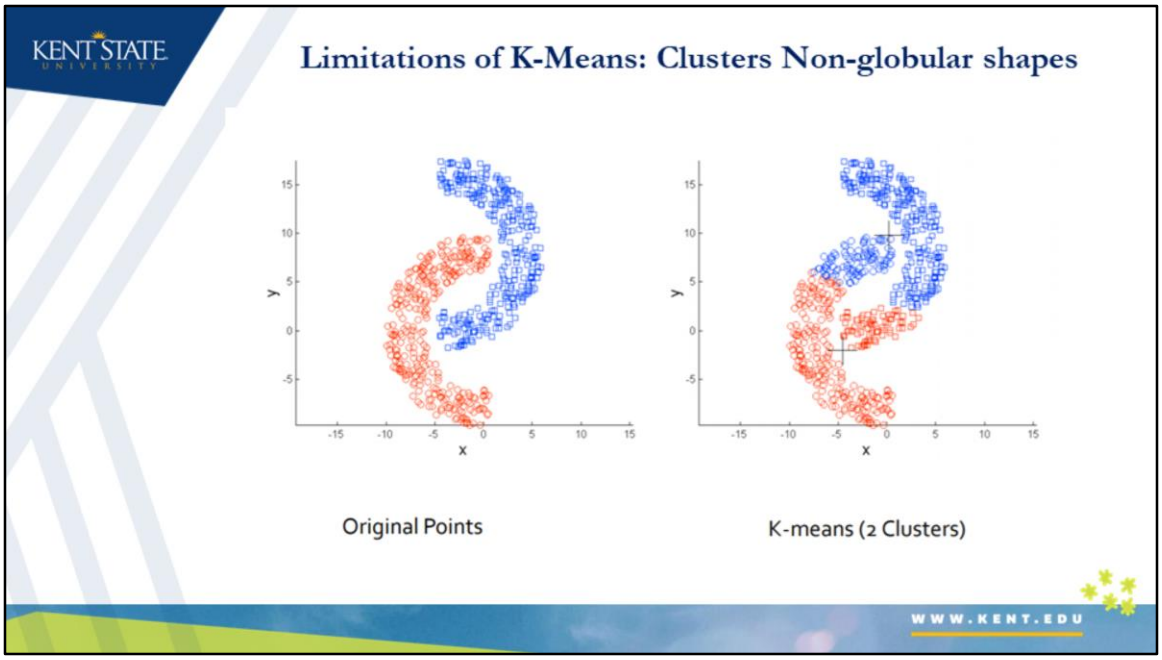
Let us see some examples to illustrate each of these limitations.



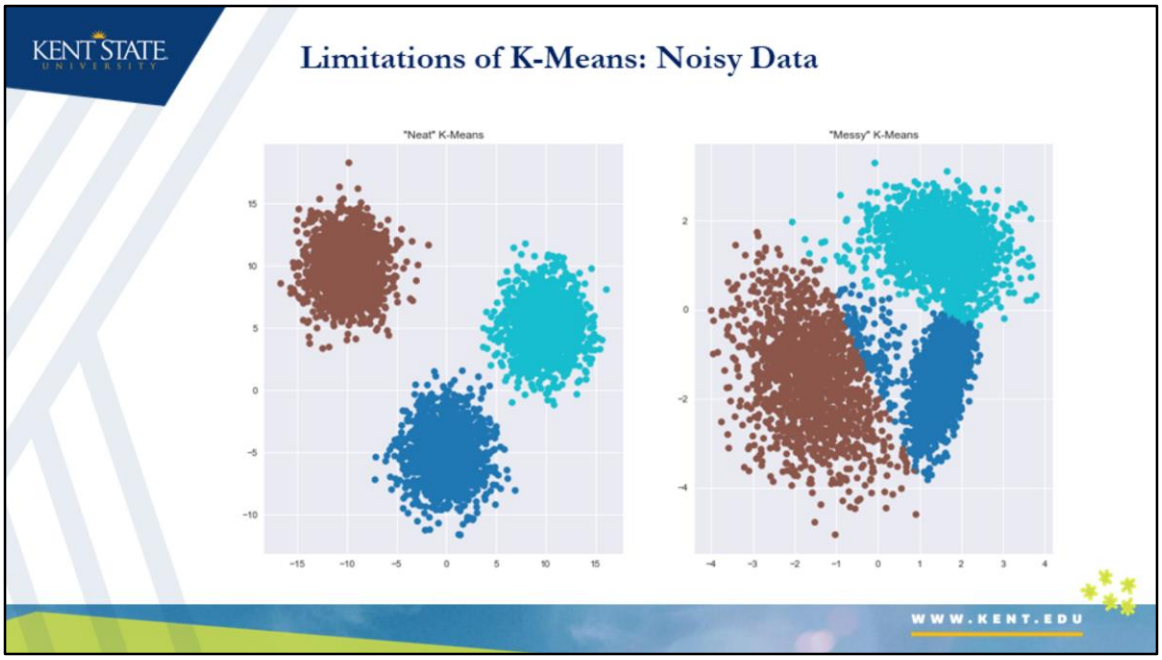
Notice that the original data clearly indicates three clusters, but the k-means algorithm, with k=3, provides totally different clusters. This is related to the measure of distance, and also how distances are calculated between clusters. For example, we typically use the centroid distances as the measure, but we could as easily use the minimum, average, or maximum distance as a measure. Nevertheless, we can't still guarantee that it would have been able to identify the three true clusters for this data.



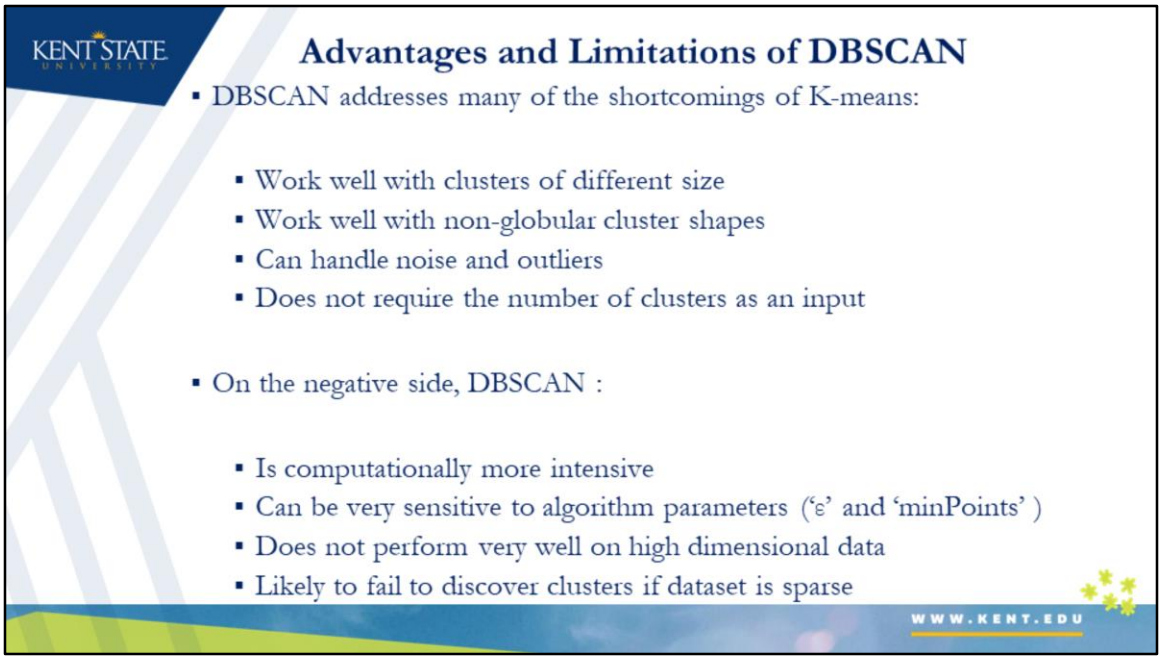
This is an example where the density affects the k-means clustering algorithm. Again, note the different clusters chosen by k-means compared to the expected.



This is a clear example of the lack of ability of k-means to identify non-linear clusters.



k-means is also affected by outliers, or noise.



There are several advantages of using DBSCAN

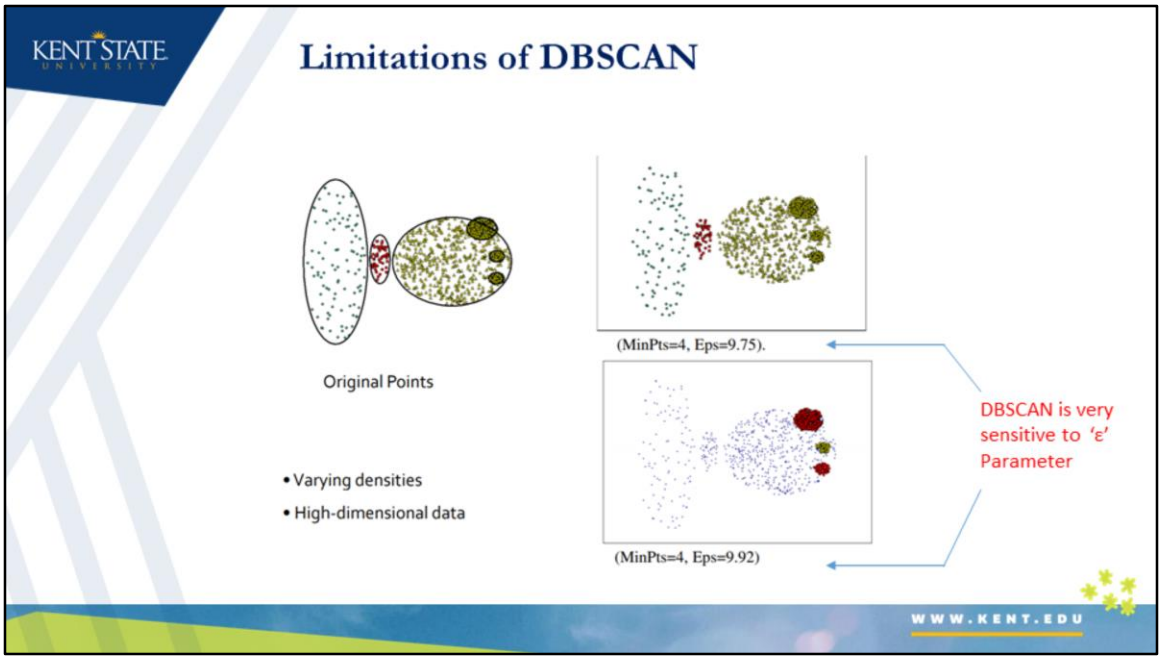
1. DBSCAN does not require one to specify the number of clusters in the data a priori, as opposed to k-means.
2. DBSCAN can find arbitrarily shaped clusters. It can even find a cluster completely surrounded by (but not connected to) a different cluster. Due to the MinPts parameter, the so-called single-link effect (different clusters being connected by a thin line of points) is reduced.
3. DBSCAN has a notion of noise, and is robust to outliers.
4. DBSCAN requires just two parameters and is mostly insensitive to the ordering of the points in the database.
5. The parameters minPts and ε can be set by a domain expert, if the data is well understood.

Nevertheless, DBSCAN does also have a few disadvantages:

1. DBSCAN is not entirely deterministic: border points that are reachable from more than one cluster can be part of either cluster, depending on the order the

data are processed. For most data sets and domains, this situation does not arise often and has little impact on the clustering result:[5] both on core points and noise points, DBSCAN is deterministic.

2. The quality of DBSCAN depends on the distance measure . The most common distance metric used is Euclidean distance. Especially for high-dimensional data, this metric can be rendered almost useless due to the so-called "Curse of dimensionality", making it difficult to find an appropriate value for ε. This effect, however, is also present in any other algorithm based on Euclidean distance.
3. DBSCAN cannot cluster data sets well with large differences in densities, since the minPts-ε combination cannot then be chosen appropriately for all clusters.[8]
4. If the data and scale are not well understood, choosing a meaningful distance threshold ε can be difficult.



This concludes our discussion on the comparison between k-means and DBSCAN. In the next module, we discuss the implementation of DBSCAN.