As described in our Module 7 lectures, the DBSCAN data clustering algorithm groups together data points based on density. Points that are gathered close together on a graph can be assigned to specific clusters, which are separated from other clusters by areas that are comparatively less dense. The DBSCAN algorithm requires only two parameters: epsilon, i.e. the radius or “neighborhood” around a given point p; and minPts, i.e. the minimum number of points that should be contained within epsilon. Does a point’s epsilon radius contain more data points than the minPts parameter amount? If so, we can be assured that that point is part of the interior of a cluster. Does a point’s epsilon radius contain less points than minPts? If true, that point is likely either a border point (meaning it is reachable from a cluster core but not actually part of that core), or a noise point that is not reachable at all.

How are these two parameters chosen? Module 7 proposes a k-nearest neighbor method for selecting parameters for DBSCAN. Using k-NN, we can calculate the average of each data point’s distance to its’ nearest k neighbors. This average of values will serve as our minPts parameter – so the result of the k-NN calculation doubles as the minimum number of points that should reside within epsilon. Once we have that value, we can now utilize dbscan’s kNNdistplot() function to generate a k-distance plot in which all distances are plotted ascendingly against the number of points sampled. This function produces a graph which should show at a certain point a drastic curve upward; the point on the Y-axis where that curve appears on the graph is what we will ultimately use for our value of epsilon, i.e. the ideal size of our radiuses.