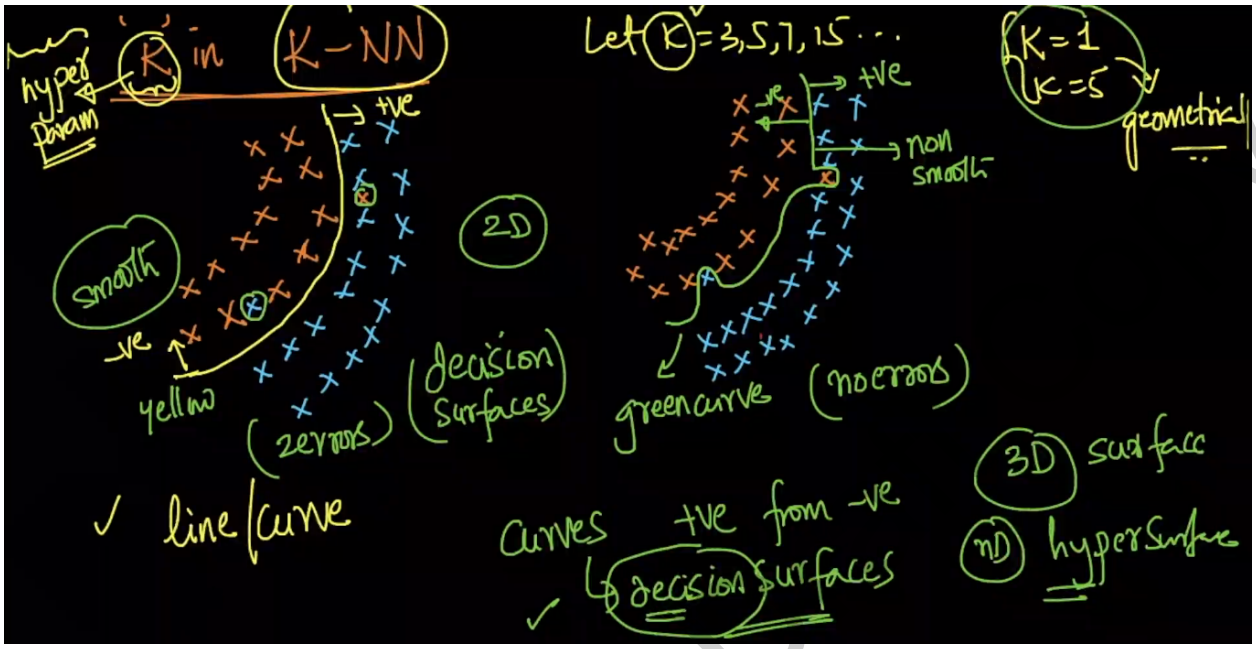
29.11 Decision Surface for K-NN as 'K' changes



'K' in K-NN is referred to as a Hyperparameter. Let us assume we have two different datasets and their training points are as shown above.

In the first dataset, we could see a smooth curve separating the '+ve' and the '-ve' classes. There are a few misclassifications, but still the curve is smooth. Whereas in the second dataset, the curve is not smooth, but all the points are classified perfectly. So when the curve is smooth, there are chances for a few classifications in this context, whereas if the curve is non-smooth, there are more chances for proper classifications.

The line/curve that separates the points belonging to two different classes is called a decision surface.

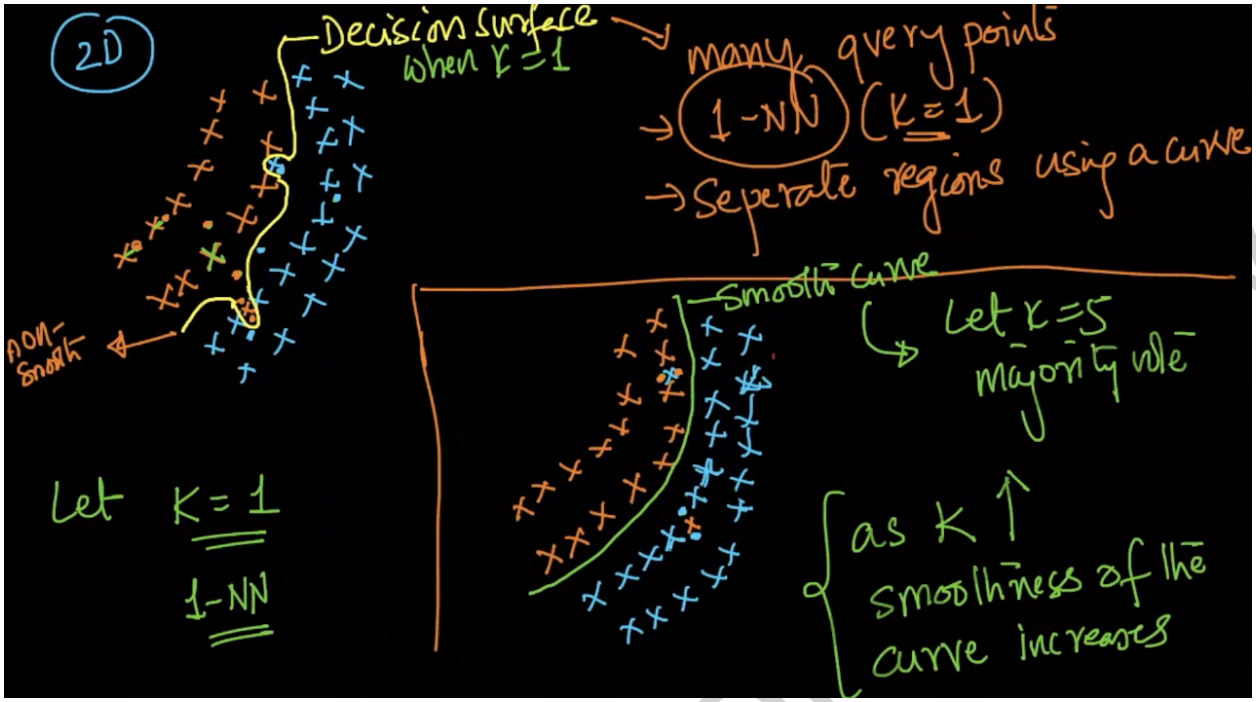
In 2-D space, we call the decision surface a line/curve.

In 3-D space, we call the decision surface a surface.

In n-D space, we call the decision surface a hyper-surface.

Let us examine the datasets in detail. When we look at the dataset, where the decision surface is non-smooth, we see all the points are classified correctly. In this case, if we consider K=1, as we have a '+ve' point lying in between a group of '-ve' points, and also a '-ve' point lying in between a group of '+ve' points.

So when K=1, if any query point has this exceptional blue point(ie., the blue point lying in between the range points) as it's 1-nearest neighbor, then the class assigned for this query point will be the same as that of the exceptional blue point.

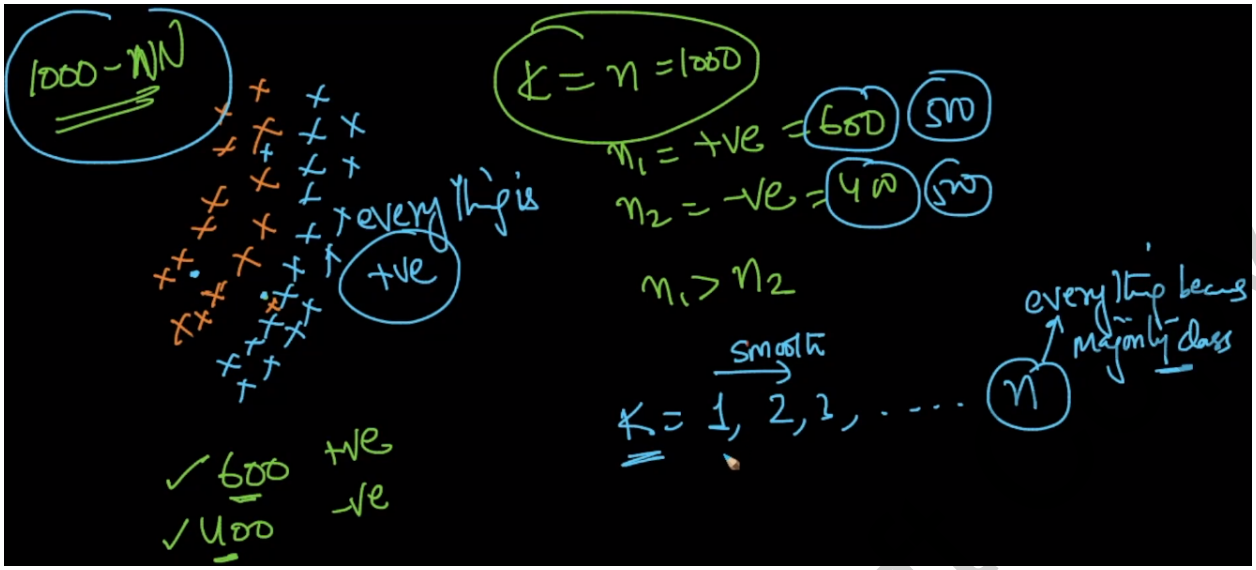


In case, if a new query point arrives with the exceptional orange point(ie.,the orange point lying in between the group of blue points) as its i-nearest neighbor, then the class label assigned for this query point will be the same as this exceptional orange point.

In both the above mentioned scenarios, if these exceptional points are not the 1-nearest neighbors, then the assigned class labels would definitely change. So here we can say that, for small values of 'K', the class labels of the points keep changing as we get different points into the neighborhood, and also the decision surface is non-smooth.

In the second example, we see the decision surface is smooth. Let us assume the 'K' value as 5, then even if we have one or two exceptional points, we do not see much changes in the polarities of those points which have these exceptional points as their neighbors. Because having these one or two couldn't show much impact when we consider 5 nearest neighbors.

So we can say, as the 'K' value keeps increasing, the decision surface becomes smoother.



Let us now consider the case, where K=n(where 'n' is the total number of points in the training dataset). Here we do not see the model putting much effort for predictions. It blindly goes with the majority class.

For example, if our dataset has 1000 data points, out of which 600 belong to the '+ve' class, and the remaining 400 belong to the 've' class, then for every query point, the KNN model assigns the majority class label. In this case, all the query points will be assigned with the '+ve' class label. This happens in case of an imbalance dataset(where the number of points belonging to a particular class differ from that of another classes)

For example, if our dataset has 1000 data points, out of which 600 belong to the '+ve' class, and the remaining 400 belong to the 've' class, then for every query point, the KNN model assigns the majority class label. In this case, all the query points will be assigned with the '+ve' class label.

For example, if our dataset has 1000 data points, out of which 500 belong to the '+ve' class, and the remaining 500 belong to the 've' class, then the model becomes indecisive and couldn't assign the accurate class label. It picks one of the labels randomly and assigns it to the query point.