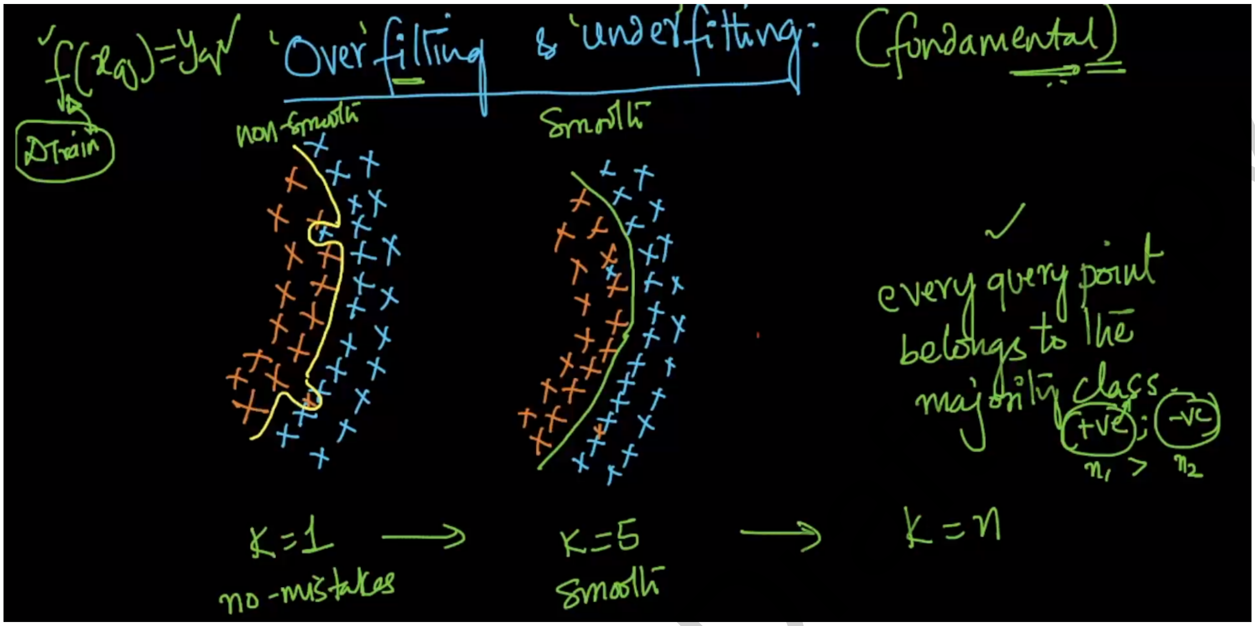
29.12 Overfitting and Underfitting



In the above scenarios, we can clearly observe that the decision surface becomes complex, if the ‘K’ value is low. It is because the KNN tries to take the least number of neighbors into consideration while predicting the class label of a query point. As we are taking only a small number of points in the neighborhood for consideration, the prediction on the query point can easily get affected.

For K=1, we take the nearest point into consideration. If we slightly increase the ‘K’ value (say K=3), then if the other class points(not the class label that was assigned when K=1) are in majority, then the class label prediction changes. So for small ‘K’ values, the decision surface keeps changing easily. Hence we see a complex decision surface. As the value of ‘K’ keeps increasing, the decision surface keeps getting smoother.

For example, if we consider K=5, then we could see the decision surface has become smoother when compared to that of K=1. Here the prediction we make would be of more confidence when compared to K=1. So as the ‘K’ value keeps increasing, we can give our predicted result with more confidence.

Similarly, if K = n (where ‘n’ is the total number of points in the training dataset), then it always predicts the majority class as the class label for a given query point. In this scenario, the model just gives the predictions blindly.

In overfitting, the decision surface does its maximum job to classify all the points accurately and hence it becomes non-smooth and more complex. In the case above where K=1, the exceptional points may be noise/outliers, but still the decision surface tries to classify them as accurately as possible. Hence the decision surface is working extremely hard. In such a case, a small change in the ‘K’ value will affect the decision surface severely.

In underfitting, the decision surface is underworking and blindly gives the majority class label as the prediction. Hence even if you make a small change in the value of ‘K’, you do not find much difference in the predictions.

In case of a well-fit or best-fit, the decision surface is smoother and tries to classify the maximum number of points correctly (except the noise/outliers). Well-fit (or) Best-fit creates a balance between overfitting and underfitting.

Note:
If ‘K’ is small (say K=1), then the KNN model overfits
If ‘K’ is large (say K=n), then the KNN model underfits.
If ‘K’ is an intermediate value, then the KNN model fits perfectly.

	Train Accuracy	**Test Accuracy**	**Variance**	**Bias**
Overfitting	High	Low	High	Low
Best/Well Fit	Moderate/High	Moderate/High	Moderate	Moderate
Underfitting	Low	Low	Low	High

Note: In overfitting, if those exceptional points are noise/outliers, then the model makes a lot of errors because it has taken noise as the data, thereby creating a decision surface that is far from the truth. This decision surface is more prone(easily affected) to the outliers/noise.

In case of a well-fit model, those exceptional points are considered as noise and are ignored while building the decision surface. This decision surface is less prone to the outliers/noise when compared to the decision surface obtained with overfitting.

Q) When the model in overfitting is working more and harder to give perfect results, why don’t we consider that?
Ans) It is because in overfitting, the model will work hard to give perfect results on the training data, but not on the test data. We actually care for the performance on the test data, rather than the performance on the training data. When the model overfits, it predicts well on the training data, but not on the test data.

When we have noise/outliers in our training data, if we consider overfitting, the model fits very well to all the points in the training data including the noise/outliers, but does not fit well on the test data points. Hence we ignore the models that overfit.