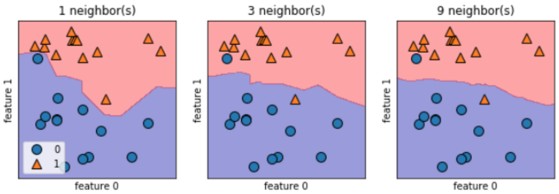
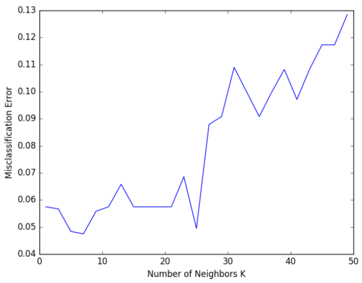
KNN-Implementation Issues

- Since the KNN algorithm is completely dependent on every point in the training data set it is vulnerable to noise. The presence of mislabeled data in the initial data set can lead to misclassification of new data points.
- KNN is completely dependent on every point in the training data set hence it has a linear time complexity.
- Incorrectly choosing the right value of ‘k’ increases the risk of wrongly predicting the output for new data. If the parameter is too small, the noisy or mislabeled data points may outweigh the correct ones in a region, and if the parameter I too large, we may end up getting rid of important details.



The figure above depicts that for a smaller value of ‘k’ noise will have a higher influence, and for a larger value of ‘k’ the classification boundary is much less distinct than required and makes computation too expensive.

- The KNN algorithm is known as the lazy algorithm since it only memorizes the training data set without actually learning it for future predictions.
- Since it is a lazy algorithm, KNN needs to remember the entire training data set which results in a larger space complexity.
- The KNN algorithm is sensitive to outliers. It works better if the outliers are removed before predictions are made.
- Using KNN for a large labelled data set is expensive since it takes a long time to find the K nearest neighbors.



- Finding the optimal value of ‘k’ is tedious since we experiment with different ‘k’ values until the one with the best accuracy is found.
- While using the KNN algorithm, finding ‘k’ becomes harder if the training data is not uniformly distributed.
- KNN works best for smaller data sets since it needs to parse through all the data points for each new classification.