K-nearest neighbors (KNN)

The K-nearest neighbors (KNN) algorithm is a type of supervised machine learning algorithm. KNN is extremely easy to implement in its most basic form, and yet performs quite complex classification tasks. It is a lazy learning algorithm since it doesn't have a specialized training phase. Rather, it uses all of the data for training while classifying a new data point or instance. KNN is a non-parametric learning algorithm, which means that it doesn't assume anything about the underlying data. This is an extremely useful feature since most of the real world data doesn't really follow any theoretical assumption e.g. linear-separability, uniform distribution, etc.

The intuition behind the KNN algorithm is one of the simplest of all the supervised machine learning algorithms. It simply calculates the distance of a new data point to all other training data points. The distance can be of any type e.g Euclidean or Manhattan etc. Once the distances of the new data points (to be classified) wrt to all the other data points are calculated, they are tabulated and then sorted in an ascending order. Then an input of the K value by a user is used to select that K number of nearest data points and a majority vote is considered and depending on the distance of the new data points from the existing ones, they are classified accordingly.
This K value must be chosen wisely as a small value of k means that noise will have a higher influence on the result and a large value make it computationally expensive. The best way to choose the value of K is:

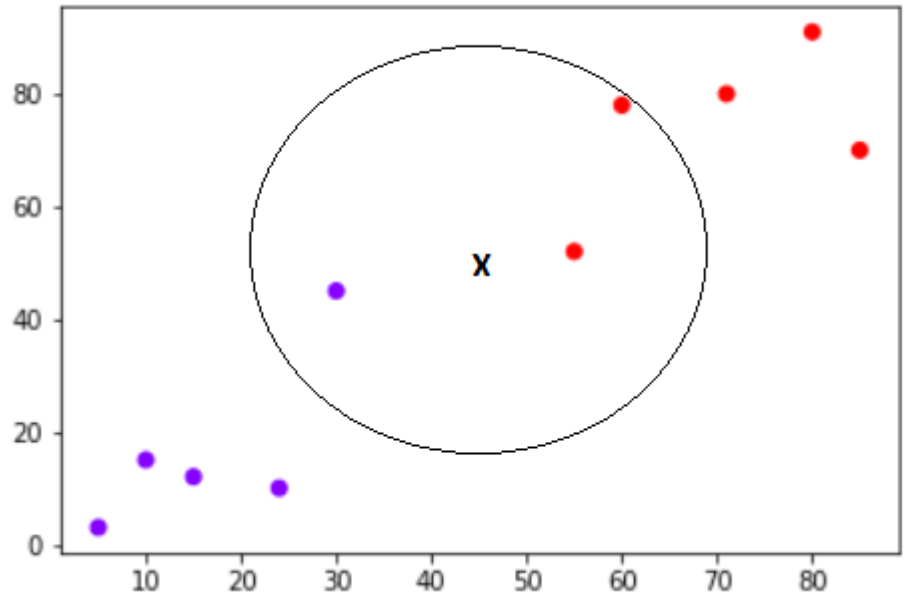
- Square root of the total number of data points
- (Square root of the total number of data points) +/- 1 to avoid confusion between two classes of data

But there are other traditional methods also like K-Fold Cross Validation (KFCV) to decide the value of K. You can use KFCV for testing performance of KNN with some various quantities of neighbours and it will be useful.

Example of KNN classification:

The task is to classify a new data point with 'X' into "Blue" class or "Red" class. The coordinate values of the data point are x=45 and y=50. Suppose the value of K is 3. The KNN algorithm starts by calculating the distance of point X from all the points. It then finds the 3 nearest points with least

distance to point X. This is shown in the figure below. The three nearest points have been encircled.



The final step of the KNN algorithm is to assign new point to the class to which majority of the three nearest points belong. From the figure above we can see that the two of the three nearest points belong to the class "Red" while one belongs to the class "Blue". Therefore the new data point will be classified as "Red".

Pros of KNN:

1. It is extremely easy to implement
2. It is lazy learning algorithm and therefore requires no training prior to making real time predictions. This makes the KNN algorithm much faster than other algorithms that require training e.g SVM, linear regression, etc.
3. Since the algorithm requires no training before making predictions, new data can be added seamlessly.
4. There are only two parameters required to implement KNN i.e. the value of K and the distance function (e.g. Euclidean or Manhattan etc.)