The training examples are vectors in a multidimensional feature space, each with a class label. The training phase of the algorithm consists only of storing the [feature vectors](https://en.wikipedia.org/wiki/Feature_vector) and class labels of the training samples.

In the classification phase, *k* is a user-defined constant, and an unlabeled vector (a query or test point) is classified by assigning the label which is most frequent among the *k* training samples nearest to that query point.

A commonly used distance metric for [continuous variables](https://en.wikipedia.org/wiki/Continuous_variable) is [Euclidean distance](https://en.wikipedia.org/wiki/Euclidean_distance). For discrete variables, such as for text classification, another metric can be used, such as the **overlap metric** (or [Hamming distance](https://en.wikipedia.org/wiki/Hamming_distance)). In the context of gene expression microarray data, for example, *k*-NN has been employed with correlation coefficients, such as Pearson and Spearman, as a metric.[[6]](https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm#cite_note-6) Often, the classification accuracy of *k*-NN can be improved significantly if the distance metric is learned with specialized algorithms such as [Large Margin Nearest Neighbor](https://en.wikipedia.org/wiki/Large_Margin_Nearest_Neighbor) or Neighborhood.

A drawback of the basic "majority voting" classification occurs when the class distribution is skewed. That is, examples of a more frequent class tend to dominate the prediction of the new example, because they tend to be common among the *k* nearest neighbors due to their large number.[[7]](https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm#cite_note-Coomans_Massart1982-7) One way to overcome this problem is to weight the classification, taking into account the distance from the test point to each of its *k* nearest neighbors. The class (or value, in regression problems) of each of the *k* nearest points is multiplied by a weight proportional to the inverse of the distance from that point to the test point. Another way to overcome skew is by abstraction in data representation. For example, in a [self-organizing map](https://en.wikipedia.org/wiki/Self-organizing_map) (SOM), each node is a representative (a center) of a cluster of similar points, regardless of their density in the original training data. *K*-NN can then be applied to the SOM.