K Nearest Neighbors

K Nearest Neighbors (KNN) is a simple algorithm that stores all available samples and classifies new samples or predict numerical targets based on a similarity measurement (e.g., distance functions). It is a nonparametric algorithm, meaning it does not make any underlying assumptions about the distribution of data. KNN is widely used in applications such as credit ratings, bank loan assignment, handwriting detection, image recognition and even video recognition.

KNN can be summarized as follows:

- **1) Store the training samples in an array of data points**
- **2) Specify a positive integer K**

In general, a large K value is more precise as it reduces the overall noise but there is no guarantee. Historically, the optimal K for most datasets has been between 3-10.

- **3) Calculate the distance of the new sample to all training samples, select the k entries which are closest to the new sample**

For continuous variables, Eulidean distance defined as follows is used:

Euclidean $\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

For categorical variables, the Hamming distance must be used:

Hamming Distance

$$D_H = \sum_{i=1}^{k} |x_i - y_i|$$

$$x = y \Rightarrow D = 0$$

$$x \neq y \Rightarrow D = 1$$

When variables have different measurement scales or there is a mixture of numerical and categorical variables, the distance can be poorly defined. One solution is to standardize the training set. However, results from the standardized training set may be different from the initial dataset.

4) Return the majority labels of the K entries as the label of the new sample

KNN have a few other features:

1) It can be used for both classification and regression.
2) It stores the entire training dataset which it uses as its representation, therefore it requires high memory and is computationally expensive.
3) It does not learn any model.
4) It is sensitive to irrelevant features and the scale of the data.
5) It is insensitive to outliers.

6) It makes no assumption about data.

KNN is often confused with K-means algorithm. These two algorithms are compared in the following table:

K-Means	KNN
It is an **Unsupervised** learning technique	It is a **Supervised** learning technique
It is used for **Clustering**	It is used mostly for **Classification**, and sometimes even for **Regression**
'K' in K-Means is the number of clusters the algorithm is trying to identify/learn from the data. The clusters are often unknown since this is used with Unsupervised learning.	'K' in KNN is the number of nearest neighbours used to classify or (predict in case of continuous variable/regression) a test sample
It is typically used for scenarios like understanding the population demomgraphics, market segmentation, social media trends, anomaly detection, etc. where the clusters are unknown to begin with.	It is used for classification and regression of known data where usually the target attribute/variable is known before hand.
In training phase of K-Means, K observations are arbitrarily selected (known as centroids). Each point in the vector space is assigned to a cluster represented by nearest (euclidean distance) centroid. Once the clusters are formed, for each cluster the centroid is updated to the mean of all cluster members. And the cluster formation restarts with new centroids. This repeats until the centroids themselves become mean of clusters, i.e., when updating centroids to mean doesn't change them. The prediction of a test observation is done based on nearest centroid.	K-NN doesn't have a training phase as such. But the prediction of a test observation is done based on the K-Nearest (often euclidean distance) Neighbours (observations) based on weighted averages/votes.

References:

1, https://www.geeksforgeeks.org/k-nearest-neighbours/

2, http://abhijitannaldas.com/kmeans-vs-knn-in-machine-learning.html