KNN (K Nearest Neighbors)

KNN is a supervised machine learning algorithm that categorizes an input by using its k nearest neighbors. k-nearest neighbors can be used in classification or regression machine learning tasks.

KNN is a model that classifies data points based on the points that are most similar to it. It uses test data to make an “educated guess” on what an unclassified point should be classified as.

KNN is an algorithm that is considered both non-parametric and an example of lazy learning i.e. it does not make any assumptions on the underlying data and also there is no explicit training phase or it is very minimal.

Non-parametric means that it makes no assumptions. The model is made up entirely from the data given to it rather than assuming its structure is normal.

Lazy learning means that the algorithm makes no generalizations. This means that there is little training involved when using this method. Because of this all of the training data is also used in testing when using KNN.

KNN essentially relies only on the most basic assumption underlying all prediction i.e. that observations with similar characteristics will tend to have similar outcomes.

KNN is all about finding the next data point(s) which is at the minimum distance from the current data point and to club all into one class where k is no of nearest data points to consider.

KNN performs better when features are scaled

KNN is often used in search applications where you are looking for similar items i.e. when your task is some form of find items similar to some item. KNN is often used in simple recommendation systems and decision-making models. It is the algorithm companies like Netflix or Amazon use in order to recommend different movies to watch or books to buy.

How KNN works?

The algorithm of KNN is

-Computes the distance between a data point and all other data points.

-The model picks k entries in the database which are closest to the new data point.

-For classification it does the majority vote among the k neighbors i.e. the most common class/label among those K neighbors will be the class of the new data point. For regression it takes the average of the k neighbors i.e. the average of the k neighbors will be the value of the new data point.

How to select K in KNN?

K in KNN is the number of instances that we take into account for determination of affinity with classes. The general rule of thumb says

k = sqrt(N)/2

where N stands for the number of samples in your training dataset.

Higher value of k has lesser chance of error

The best value of k for KNN is highly data-dependent.

Historically, the optimal K for most datasets has been between 3-10.

Techniques like grid search can be used to find the value of k where prediction errors are the least.

Distances

In the instance of continuous variables any of the following can be used

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

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

Minkowski $\left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

In the instance of categorical variables, the Hamming distance must be used.

Hamming Distance

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

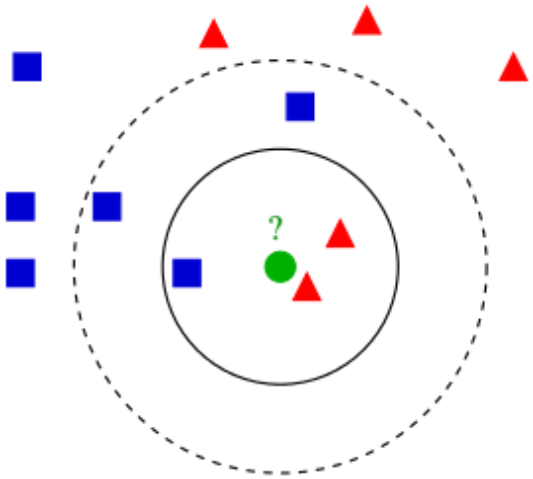
$$x = y \Rightarrow D = 0$$

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

X	Y	Distance
Male	Male	0
Male	Female	1

Examples

For KNN classification an input is classified by a majority vote of its neighbors. That is, the algorithm obtains the class membership of its k neighbors and outputs the class that represents a majority of the k neighbors.



Suppose we are trying to classify the green circle. Let us begin with k=3 (the solid line). In this case, the algorithm would return a red triangle, since it constitutes a majority of the 3 neighbors. Likewise, with k=5 (the dotted line), the algorithm would return a blue square.

In case of KNN regression, the value returned is the average value of the input's k neighbors.