K NEAREST NEIGHBOURS

An Intuition to K-NN Classification Algorithm

What is k-NN?

K-Nearest Neighbor algorithm is a simple yet most used classification algorithm. It can also be used for regression.

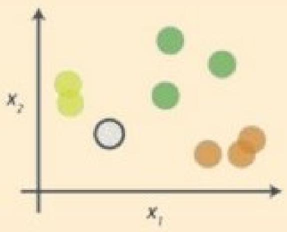
KNN is non-parametric (means that it does not make any assumptions on the underlying data distribution), instance-based (means that our algorithm doesnt explicitly learn a model. Instead, it chooses to memorize the training instances.) and used in a supervised learning setting.

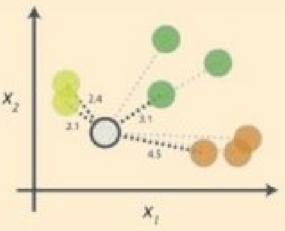
k-NN is also called a lazy algorithm because it is instance based.

How Does k-NN Algorithm work?

k-NN when used used for classification — the output is a class membership (predicts a class — a discrete value).
There are three key elements of this approach: a set of labeled objects, e.g., a set of stored records, a distance between objects, and the value of k, the number of nearest neighbors.

We want to classify the grey point into one of the three classes light green, green and red

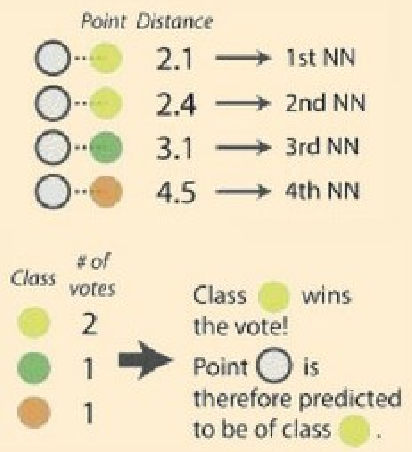




Start by calculating the distance between the grey point and k -nearest points

Making Predections

To classify an unlabeled object, the distance of this object to the labeled objects is computed, its k-nearest neighbors are identified, and the class label of the majority of nearest neighbors is then used to determine the class label of the object. For real-valued input variables, the most popular distance measure is Euclidean distance.



Point	Distance	
○··●	2.1	→ 1st NN
○··●	2.4	→ 2nd NN
○··●	3.1	→ 3rd NN
○··●	4.5	→ 4th NN

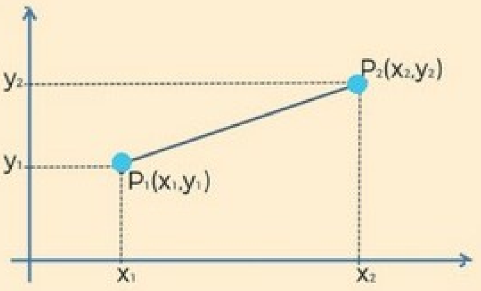
Class	# of votes
●	2
●	1
●	1

Class ● wins the vote! Point ○ is therefore predicted to be of class ●.

The Distance

Euclidean distance is calculated as the square root of the sum of the squared differences between a new point and an existing point across all input attributes .
Other popular distance measures include:

- Hamming Distance
- Manhattan Distance
- Minkowski Distance



Euclidean Distance between P_1 and $P_2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

Value of k

Finding the value of k is not easy. A small value of k means that noise will have a higher influence on the result and a large value make it computationally expensive. It depend a lot on your individual cases, sometimes it is best to run through each possible value for k and decide for yourself.