Nearest Neighbour Classification:

What is the Nearest Neighbour?

The Nearest Neighbour (NN) is probably one of the simplest algorithms when it comes to AI.
NN is an algorithm that allows us to solve classification problems. These are problems where each input is associated with an output class. For example, a classification problem could be: given some coordinates, tell me which continent I am in.
In this case, each coordinate is associated with a continent as the output class.
To work, NN requires that we are given a dataset to start from. That is, simply some data containing input → output class. The larger our dataset, the better NN will perform.

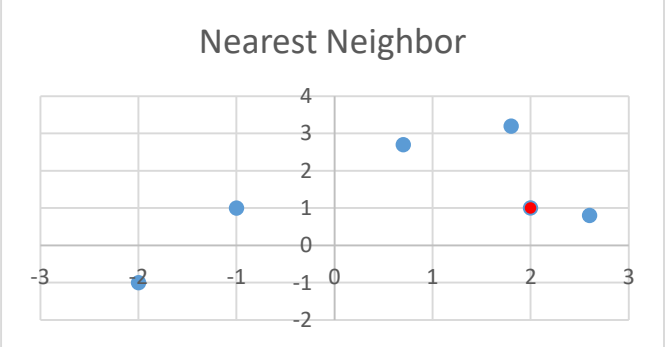
How does NN work?

As we know, within our model, every type of input is ultimately treated as a sequence of numbers, which represent the "features" of our input. Therefore, we can represent all the data in our dataset graphically. For simplicity, let's suppose the problem is: given a point as input, return the quadrant it belongs to (whether it belongs to the 1st, 2nd, 3rd, or 4th quadrant of the Cartesian plane).
Now, simply suppose we receive the point with coordinates (2,1) as input. We must then return the quadrant it belongs to using NN.

With NN, we simply need to go through our entire dataset and calculate the Euclidean distance between the given input point and the point contained in the dataset. Since we are in two dimensions, we can just use this formula: $\sqrt[2]{(x-2)^2+(y-1)^2}$. This for every point inside our dataset.

It is important, during all iterations, to keep track of the minimum distance found and the class of the point in the dataset associated with that minimum distance.
At the end of the loop, we will return as output the content of the variable dedicated to the class of the point closest to the input point.
In other words, it is very likely that the output will be the same as the output of the point in the dataset that is closest to the input point.
However, be careful—it is not guaranteed that the prediction will be correct. This is why the larger the dataset, the better. Especially if the data are evenly distributed across the space.

Graphical representation of NN:

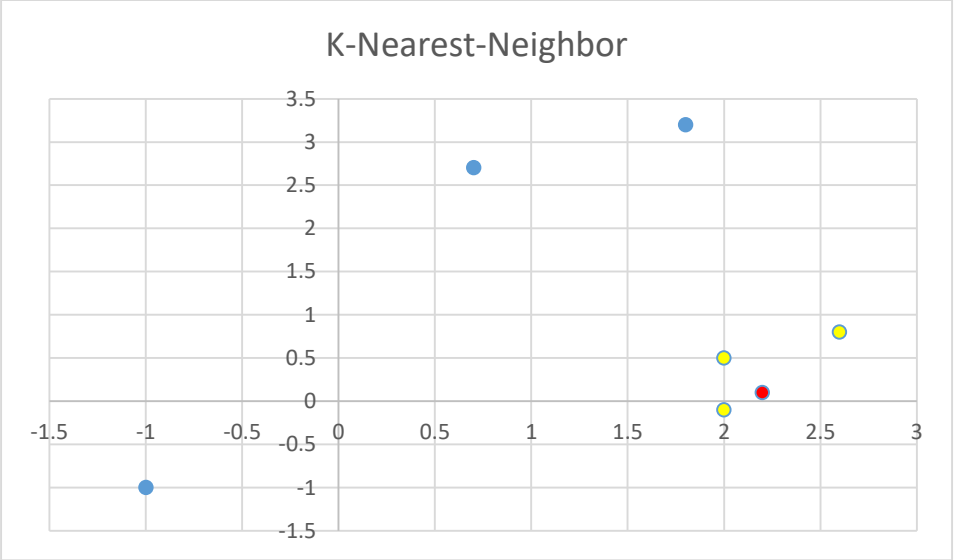


Let's suppose we receive as input the point defined earlier: P (2, 1), shown in red.
In this case, the point closest to it is the one immediately to the right, which is in the 1st quadrant. Therefore, the output for P will be 1.

Other versions of NN:

As mentioned earlier, NN does not always work correctly. However, there are modifications we can make to make it more efficient.
One of these is the K-Nearest Neighbour (K-NN). The functioning is exactly the same as NN, except that this time the output will be the most frequent output among the K points closest to the input point.

Graphical representation of K-NN:



In this case, we receive as input P (2.1, 0.1). If we used NN, the closest point would be (2, -0.1), so the output would be quadrant 4, which is incorrect. However, with K-NN, it works differently.

Suppose we have K equal to 3. In this case, the 3 closest points would be those highlighted in yellow. Once we have found the K nearest points, the output will be the most frequent output quadrant among these K points. In this case, we have:

- Quadrant 4 appears once
- Quadrant 1 appears twice

Therefore, the output will be quadrant 1, which is correct.