Machine learning and statistical learning

K-Nearest Neighbors

E. Gallic

1 Context

In this exercise, you will build a classifier using the K-Nearest Neighbors algorithm.

2 Lab setup: generating data

In a first step, you will generate data. To do so:

1. Draw $n = 50$ observations in a unit square. To do so:
    - randomly generate 50 observations from a Beta distribution with parameters $\alpha = \beta = 1$ and store the drawn values in an object you will call `x`.
    - Do the same procedure and store the draws in an object you will call `y`.
2. Create a vector you will call `true_label` of size $n = 50$ which will contain the true labels: "orange" or "blue".
    - "orange" if $x + y \geq 1$
    - "blue" otherwise.
3. Create a new point (x_0, y_0) at which you will try to assign a label, depending on the values of the nearest neighbors. For example: $(x_0 = 0.75, y_0 = 0.5)$.
4. Create a matrix with 3 columns: the x and y coordinates of your generated points, and the assigned label.
5. Plot your 50 observations on a scatter plot and add the new (x_0, y_0) observation using a different color/shape.

3 The algorithm

1. To know which are the K closests points of your new observation, you need to compute the distance between each point of your dataset and your new observation. To that end, create a function that computes the distances between two points:
    - this function will require four parameters: the two coordinates of a first point (x_A and y_A) and the two coordinates of a second point (x_B and y_B).
    - it will return the Euclidean distance between the two points whose coordinates are given as parameters.

2. Using a loop, apply this function to your new point (x_0, y_0) and each of the points in your dataset. In other words, at iteration i, store the Euclidean distance between your point (x_0, y_0) and the i-th point from your data, *i.e.*, (x_i, y_i). Once you have computed the distance from your point (x_0, y_0) to all points from your dataset, order your dataset by increasing distances to your new point.

3. Pick a value for K. For example, $K = 3$.

4. In a new object, copy the K first rows of your dataset that was previously ordered by ascending values of the distance to the new point: this allows you to keep the K nearest neighbors.

5. Plot the points of this dataset in a different color.

6. Based on that dataset with only the K nearest neighbors, compute the number of "blue" and the number of "orange", then provide an estimation of the probability for the new observation to be blue.

7. Based on that probability, assign a predicted class to your new observation.

8. Set a different value for K and look at how it may change your prediction.