Fundamentos de Data Science
Lab N 2

In this lab, you will be implementing the KNN algorithm that we saw in class.

You will be using MATLAB's iris dataset to cluster flowers into classes. Each of the 150 flowers of the dataset is described by the following features:

- Sepal length in cm
- Sepal width in cm
- Petal length in cm
- Petal width in cm

Matlab provides a 150x4 matrix, where each column is a flower, and each row contains one of the features described above (in that order). It also for each of the flowers.

You will implement a program in Matlab to:

- Load the dataset.
  Load fisheriris.mat
- Choose two features to plot the points in 2D.
  - You can use any pair, but you can also use a combination of the features. (for example, you can choose the features to be the sepal length and width, the petal length and width, the area of the sepal and the area of the petal considering them to be rectangular, etc.).
  - Plot the points, make sure you:
    - Assign a colour to each class.
    - Paint the points with the color of the class they belong (use a circle to represent the points).
    - Label the axis with the name of the corresponding feature.
  - Do this for at least two combination of features, and see which one separates the class better in the plots.
- Compute the Euclidean distance between all the possible pairs of flowers in the dataset.
- Implement the k nearest neighbours algorithm. For a given point *x* you need to:
  - Find the *k* nearest neighbours of *x*.
  - Count how many times each class appears between the *k* neighbours of *x*.
  - Assign *x* to the class with the highest count of neighbours.

Today you should aim at finishing a simple script that plots the dataset, and implements the k-nearest neighbour algorithm.

Important points for today class:

- Write modular code (you should define functions for the different points above, and call them from the script).

- No for loops allowed!

During next week you should try to:

- Divide your dataset into training, testing and validation sets.
- Using the training and validation sets, test your defined distances and different values of *k* nearest neighbours.
- Make a plot for the training and testing sets:
    - Plot the training points using a circle, and paint them with the colour of the class they belong.
    - Plot the points on the testing set using an *X* and paint them with the colour of the class they were assigned by the KNN algorithm.
    - Add labels, title.
- Measure the accuracy of the model on the validation and testing dataset, remember that:
    - The accuracy can be measured for each class, as well as the overall accuracy.
    - A prediction is considered to be correct for a given sample if the predicted class is the same as the true class of the sample.
    - The overall accuracy can be defined as the number of correct predictions, divided by the total amount of samples of the test set.
    - The class accuracy can be defined as the number of correct predictions for that class, divided by the total amount of samples of the test set belonging to that class.
- Use *subplot* to make bar plots for the overall accuracy and the accuracy for each class.