DEEP
LEARNING ACADEMY

Today we will implement the ’k nearest neighbors’ algorithm on the famous iris data set.
We will do this in six small steps.

1. **Handle the data:** write a function that will open the dataset and split it to training and testing. You can either:
   use the function ’*sklearn.datasets.import_iris()*’ which provides an object where *.data* and *.target* are the data and the labels
2. **Distance function:** write a function that can calculate the distance between two datasets.
3. **Nearest neighbors:** write a function that searches the whole dataset for the k nearest neighbors.
   Hint: you can sort the distances and find the k first elements using the function https://docs.scipy.org/doc/numpy/reference/generated/numpy.sort.html
   Or use python’s sort for list of tuples:
   https://stackoverflow.com/questions/3121979/how-to-sort-list-tuple-of-lists-tuples
4. **Predict from k nearest neighbors:** now that we have the k nearest neighbors we can calculate an average of them to predict the result, or for categorical data we can do voting i.e. finding the mode (השכיח) between these k nearest points.
5. **Calculate the accuracy on the test data:** calculate the prediction on every element of the test data and compare to the expected values. Calculate the percentage of the data sets that we calculated accurately.
6. **Main function:** write a main function that contains everything and calls all the functions that we have written.

Part 2: Classes (Optional part):

For tutorials refer to any of the internet tutorials on python classes, such as:
https://www.learnpython.org/en/Classes_and_Objects
https://en.wikibooks.org/wiki/A_Beginner%27s_Python_Tutorial/Classes
https://docs.python.org/3/tutorial/classes.html#classes

Rewrite your code to use class such that it contains:

1. A DataSet class
   a. It should be instantiated (__init___ function) with a dataset (with labels)
   b. It should contain a function which gets a percentage and returns the data, after randomly permuting it, split to train and test according to that percentage.
2. A KNN class
   a. Contains function which classifies test data
   b. the class should also contain all the relevant functions for the classification calculation.