

## Coding Test

Consider the “Motion Capture Hand Postures” dataset, whose description is here <https://archive.ics.uci.edu/ml/datasets/Motion+Capture+Hand+Postures>. You can download the dataset from here <https://archive.ics.uci.edu/ml/machine-learning-databases/00405/>. After unzipping, you get a csv file with the data contents.

This dataset contains 38 attributes.  $X_0, Y_0, Z_0, \dots, X_{11}, Y_{11}, Z_{11}$  are continuous attributes. *User* is a discrete feature that refers to the user ID. Note that we **ONLY** care about users with IDs 0, 1, 2, 5, 6, 8, 9, 10, 11, 12, 13, 14. That is a total of 12 users that we care about. Finally, *Class* is a discrete attribute with values 1, 2, 3, 4, 5 (5 values in total).

You are asked to:

1. Build a shallow NN-classifier that takes as input attributes  $X_0, Y_0, Z_0, \dots, X_4, Y_4, Z_4$ , and returns the *Class* as the output.
2. Build a shallow NN-classifier that takes as inputs attributes  $X_0, Y_0, Z_0, \dots, X_4, Y_4, Z_4$  **AND** the *User* information, and returns the *Class* as the output.
3. Build a shallow NN classifier **ONLY** for *Class* 0. The classifier takes as inputs attributes  $X_0, Y_0, Z_0, \dots, X_4, Y_4, Z_4$ , and returns the *User* ID as the output.
4. Report your classification accuracy on each of the above tasks. You can randomly split the dataset into a training set (60%), a validation set (20%) and a testing set (20%). You can use any type of cross-validation you want to, but you need to explain your choice clearly.

Note that you need to take into account all aspects very carefully. For example:

- a) Your code must be well-structured, readable and respect good software-engineering principles.
- b) To read input files, you may use the pandas library or other libraries if you prefer.
- c) You need to normalize the continuous data.
- d) You need to treat categorical data properly.
- e) Your NN must be shallow, so 2 hidden layers should properly suffice. You can decide on the activation function of the hidden layers. The output layer must be decided by you based on the task at hand.
- f) You need to deal with missing data, for instance you can use mean imputation.
- g) You can use stochastic gradient descent or any other methods you prefer to train the network.
- h) You are free to optimize your neural network classifier as you want, as long as this helps you achieve better performance.

Please note that you must clearly explain your thinking in a separate document.