Industrial Software Development (ISDe) Course
Evaluation of Programming Skills

Name: ……………………………. **Surname:** ………………………… **Student ID:** …………

Programming Exercise: Implement the code as described below.

Load the MNIST digit data. Split the data to have 60% training samples and 40% test samples.

Create a package data_perturb that will contain three different classes of objects:

1. An abstract class CDataPerturb serving as interface, which requires implementing an abstract method named data_perturbation, which takes as input a flat vector x and returns a perturbed version xp. Moreover, this class should implement also a (concrete) method named perturb_dataset(X) which will iteratively apply data_perturbation to each row of the dataset X and return the perturbed version of the whole dataset Xp.
2. A child class inherited from CDataPerturb, named CDataPerturbRandom, which randomly changes K values in the input vector x, selecting such values **uniformly** in the range [min_value, max_value]. The constructor of the class will take as input parameters: min_value, max_value, and K, with default values respectively of 0, 255, 100. For all these parameters, setters and getters should be available.
3. Another child class inherited from CDataPerturb, named CDataPerturbGaussian, which randomly perturbs **all** values in the input vector x with Gaussian noise. The Gaussian noise must have zero mean and standard deviation parametrized by sigma. Hint: use sigma * np.random.randn(...) to rescale the values sampled from the standard normal with zero mean and unit variance. If the values in the perturbed image are below min_value or above max_value, they should be set to min_value and max_value respectively. The constructor will take min_value, max_value and sigma as input parameters, having default values of 0, 255, and 100.0. Setters/getters should be available for all these parameters.

Test both perturbation models on ten random images drawn from the MNIST dataset, and visually compare the results. *Hint:* you can use the function plot_ten_images().

Train the NMC classifier on the training data and test it on the test set. Compute the classification accuracy, that is, the fraction of correctly classified samples in the test set.

Perturb the digit images in the test set using the two perturbation models with the following parameter values: K=[0, 10, 20, 50, 100, 200, 500] and sigma=[10, 20, 200, 200, 500].
Compute the classification accuracy values against K.
Compute the classification accuracy values against sigma.

Create a plot with two subplots, plot accuracy vs K in the leftmost plot, and accuracy vs sigma in the rightmost plot.