Part 1: Classification

The dataset 'sekholopane.csv' has 100 variables and almost 2000 samples. We will assume that it has neither outliers nor missing values.

You have to perform the following tasks:

- (0) [0.25 points] **Fix a seed** for all random_state parameter. Use the sum of the digits of your DNI. Use this seed in all functions that have random_state parameter.

- (1) [0.25 points] **Split the data into train and test sets**, keeping two thirds of the data for training. Make sure you keep the ratio of all the classes in both parts.

- (2) [1.5 points] **Tune one of the SVC algorithms**. Optimize the two more relevant parameters of the algorithm you have chosen, by means of 10-fold cross validation. Make sure that you check at least 17 different parameters combinations.

- (3) [1 points] **Tune a neural network**, optimizing the layers topology with at least 5 different topologies. Keep the rest of parameters with their default values, except random_state (where you must use your seed) and max_iter (use 1000).

- (4) [1 points] **Tune a random forest algorithm**, optimizing n_estimators and max_depth with at least 4 different values for both parameters. Comparing (4) with (2) and (3), which is the overall winning model? (Note: by winning model, we mean the best paradigm + parameters. One example could be random forest with n_estimators=3 and max_depth=700).

- (5) [0.5 points] Would you say that the data is imbalanced? Looking at the available options for feature selection and for imbalance correction, how many combined (feature selection + imbalance correction) preprocessing schemes can you define?

- (6) [1 points] For the overall winning model, check whether performing any of the possible preprocessing schemes performs even better.

- (7) [0.5 points] For the overall best model you have found in (6), obtain the classification report, the area under the ROC curve, and the confusion matrix.

Part 2: Regression

The dataset 'superconducting.csv' has 81 variables and more than 21K samples. It is a real-world dataset about predicting the critical temperature of semiconducting materials.

- (1) [0.25 points] **Split the data into train, validation and test sets**, keeping 20% for both validation and test.

- (2) [1 points] Using the validation data and the mean squared error score in principal components regression (PCR), find the adequate number, *a*, of **principal components**, up to a maximum of 40. Focusing now on the cumulative variance, is the cumulative variance you are not capturing (because of not selecting the last 81−a PCs) higher than 1 per 1000?

- (3) [1 points] Now we will check the performance using **feature selection**, instead of the feature extraction by PCA done in (2). Try all suitable scoring options of the feature_selection_model function, fixing percentile=5. Use multiple linear regression as predictive model. Which compression has been higher?

- (4) [1 points] Repeat (2) and (3) using **ridge regression** instead of multiple linear regression. Set the best of all models seen so far, as your winning model.

- (5) [0.75 points] Check the performance of the **winning model** on the test set. Compare it with the performance on the original dataset (without preprocessing) using the same predictor as in the winning model (multiple linear regression or ridge regression). Was our preprocessing beneficial?