Technical Guide

We have implemented this project as a python notebook using Google Colab platform. Therefore, we had some pre installed packages that their installation is not part of our code. The four additional packages that need to be installed are ‘deslib’, ‘scikit-posthocs’, ‘xgboost’ and ‘shap’. There is a cell in the notebook that is responsible for these installations.

In order to run the notebook, follow these steps:

1. Load the notebook to google colab or any other suitable platform.
2. Create a new directory, and upload all of the classification datasets to it.
3. In the beginning of cell number 9, there are two parameters (this is also documented as python docstrings in the notebook). The first is ‘input_path’, set its value to the directory with the datasets you created in step 2. The second parameter is ‘output_path’, set its value to a **different** folder than the input_path, and all output of the project will be created there.
4. Optional step: Since runtime of section c (150 datasets) is significantly long, we have also submitted a file called ‘evaluation.csv’ with the results of that run. To save time, you can place this csv into the directory you chose as ‘output_path’ in step 3. Our code will read this csv and continue from there. Make sure to comment out the outer loop of cell number 9 to avoid the long run.
5. Upload ‘ClassificationAllMetaFeatures.csv’. Set ‘meta_input_path’ in cell number 15 to the path of the csv, including the csv name (e.g: ‘/mydir/ClassificationAllMetaFeatures.csv’). Important: use the csv we submitted and **not** the one provided to us, since the one we were given has some typos in the datasets names. We have corrected them and submitted a new fixed file.
6. This is it, you can now run the notebook from the first cell.

Note: Since we perform a random search over hyper-parameters, in some cases there may be a conflict between the chosen values and the dataset (mostly when the dataset is too small for a certain parameter value). In such cases there will be an error, and the run should be restarted. This is another reason why we recommend using our ‘evaluation.csv’ instead of running the entire evaluation again.

Hyper Parameters Description

We have tried many combinations of hyper parameters during the making of this project. We have discovered that most of the hyper parameters of the package we use (deslib), are depending on each other, and some combinations of them do not work well. Therefore we chose the following two hyper parameters for the three algorithms we tested:

- k (int): Number of neighbors used to estimate the competence of the base classifiers. As described in the final report, these algorithms are using the neighbors of a classifier in order to choose whether or not it will be used in classification. This parameter is controlling the amount of neighbors consulted. The range we allowed for this parameter is between 2 (the minimum) and 9 (more than that usually led to errors). This parameter originally defaults to 7.

- DSEL_perc (float): Percentage of the input data used to fit DSEL. The remaining data will be used to fit the pool of classifiers. Possible values for this parameter are 0.1, 0.2 … 0.9 , and defaults to 0.5 .

For the baseline algorithm we chose (AdaBoost), we performed the search over all of the possible hyper parameters:

- n_estimators (int): The maximum number of estimators at which boosting is terminated. In case of perfect fit, the learning procedure is stopped early. The range used here is between 10 to 100. Defaults to 50.

- learning_rate (float): Learning rate shrinks the contribution of each classifier by learning_rate. We have used a range of 500 evenly distributed values between 0.05 to 1. The default is 1.