MLCB 2024

Assignment 2: Some common mistakes to avoid

Be careful about what features you use!

The diabetes dataset used for assignment #2 includes a column labeled "ID," representing the patient identification number. Many submissions incorrectly used this column as a feature, resulting in non-functional models. Always carefully inspect the information carried by the features you use. Columns that serve as entry identifiers should be set as the data-frame index.

- You can achieve this when loading the data file with pd.read_csv("a/file/path.csv", index_col=[numerical index of the ID column]).
- Alternatively, you can set the data-frame index using dataframe.set_index("column_to_set", inplace=True, drop=True).

Random states are crucial when using repeated nested cross-validation

Another common mistake is related to the initialization of the StratifiedKFold objects used for splitting the data into train/test sets in each iteration of the outer and inner loops of the Nested cross-validation. These objects often either had no seed or the same seed for their random state parameter across all nCV rounds. In the first case, it becomes impossible to reproduce the results, which is essential for any experiment. In the second case, the situation is even worse because we end up performing essentially the same trial with the same data splits in every nCV round, wasting computational resources. The same results could have been obtained with a single nCV round! The optimal approach is to generate an array of distinct seeds, one for each round, and use a different seed for each round.

A final CV loop is needed after nCV to build the final model

Nested cross-validation offers a reliable estimation of how various classifier families (e.g., Logistic Regression, Naive Bayes, Random Forest) are expected to perform on unseen data. It can be used to compare classifiers and help us decide the most suitable (called “best”) classifier algorithm for the task at hand. However, it does not provide the optimal hyperparameters combination (optimal design) for the chosen classifier to deploy in the field. To find the best design (hyperparameters combination) for the selected classifier after nCV, it is necessary to run an additional simple cross-validation loop using only the selected classifier algorithm using now the entire dataset to fine-tune its hyperparameters. This will give rise to a “final model” to deploy. Its expected generalization performance metrics are those that were found using the same classifier at the end of nCV. So you really do not need an extra hold-out set to evaluate this final model!
So nCV is to compare classifiers and if you wish declare a “winner” algorithm
And the last CV is to fine-tune the hyperparameters of this “winner” algorithm.