

Slide set 6 assignments

NOTE : you can merge the subprojects analyses into the same document, zip can contain all the exercise codes, there is no need to do separate zip packages, just one.

NOTE2: The classifier/predictor codes are expected to have training and testing phases and thus the data must be divided (close to) 80% / 20% ratio for training and testing.

Exercise 1

Load 'HalvingRandomSearchCV_random_forest.py' from folder and test it. Improve the score accuracy by changing the parameter ranges in optimization procedure. Include of the score values in your analysis (+zipped code)

Max 2 p.

Exercise 2

Load 'BayesSearchCV_SVC.py' from folder and test it. Change the classifier to AdaBoost classifier:

<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html>

and include/optimize all the three parameters in Bayes optimization mechanism (exclude the 'random_state'). Can you improve the Adaboost result compared to optimized SVC classifier?

Include of the score values in your analysis (+zipped code)

Max 4
p.

Exercise 2

Look at the example in:

https://github.com/optuna/optuna-examples/blob/main/pytorch/pytorch_simple.py

Modify the code to test minimum of four different activation functions (for example nn.ReLU(), nn.Tanh(), nn.celu(), nn.Sigmoid()). Show the optimized architecture in your analysis (+zipped code)

Max 4 p.

EXTRA: Exercise 3

(Extremely time consuming) Look at the example in:

https://github.com/optuna/optuna-examples/blob/main/pytorch/pytorch_simple.py

and

https://github.com/optuna/optuna-examples/blob/main/pytorch/pytorch_lightning_simple.py

Modify the 'Exercise_train_test.py' (Look at the "Practical example code folder) architecture using optuna mechanism. Include the number of layers, layer sizes and selected activation functions (minimum of 10) in the optimization process. Show the test results (figure) and MSE for the original and optimized codes in your analysis (+zipped code).

Max 15 p.