SCAMs Project Update

KHALIMAT A. MURTAZALIEVA | BOSTONGENE

II

Goals

- Add error plot ("average the different repetitions and visualize mean value + standard deviation")
- Run pipeline with different parameters (experiment with models, sampling, splitting and datasets, etc.)

Error Plot

Iteration (int(N*0.88)

N - the number of samples
in the training ds

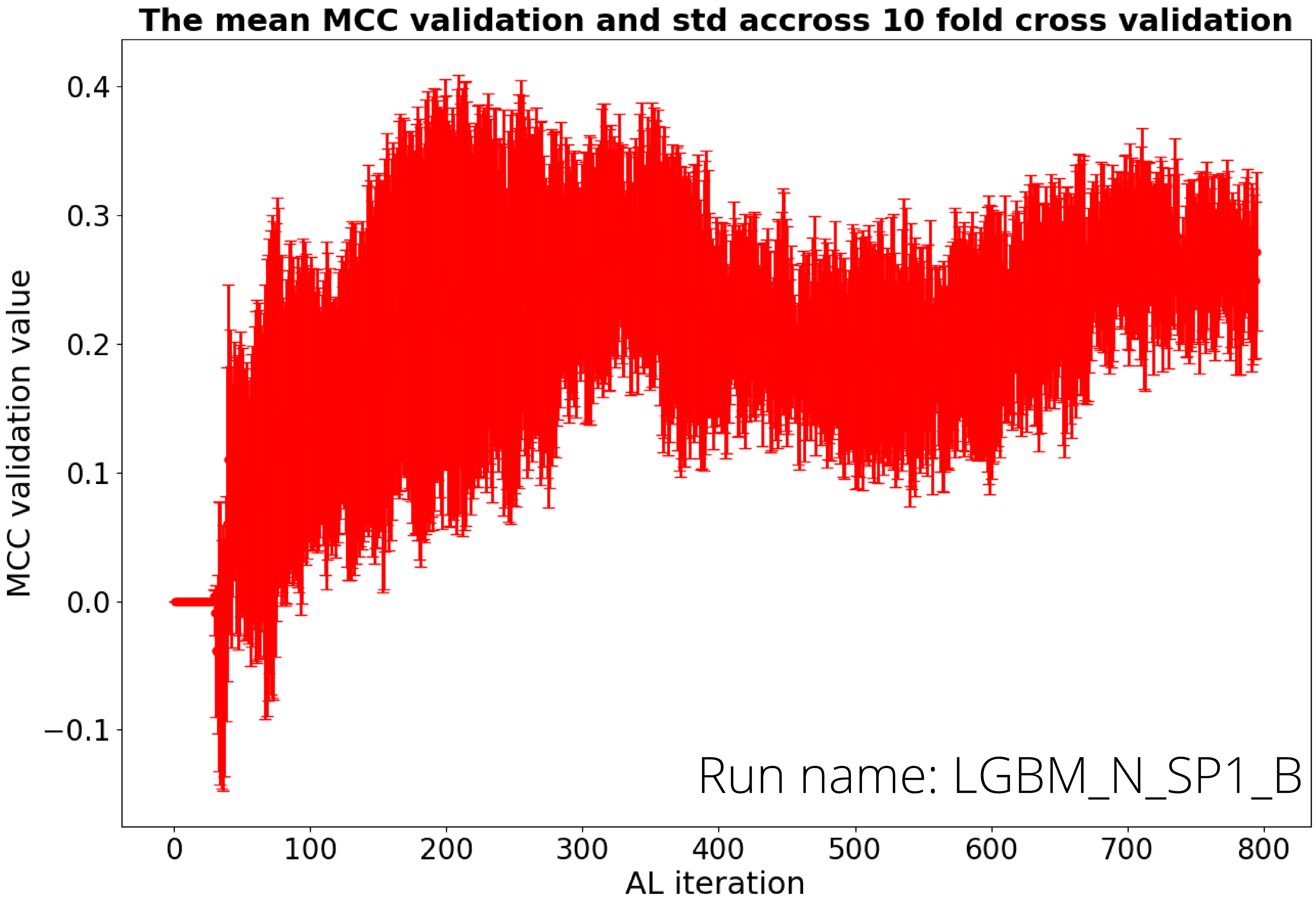
X - np.array,
X.shape=(10, int(N*0.88)

Fold

For the arrays (X) with calculated metrics (accuracy, F1, MCC, AUC LB, AUC, AUC UB) on test and validation datasets, I calculated mean - X.mean(axis=0) and std - X.std(axis=0) and visualized using matplotlib error plot

Error plot, example

IV



Covered Runs

V

I only managed to experiment with LightGBM, but from previous runs (please, see the folder early_runs), RF yielded lower performance. For example, I calculated the mean AL validation MCC value for RF (0.279) and LGBM (0.405). Do I need to further run the pipeline using RF?

SF

Classical (without sampling and TTS) - 1 run
Only with up/down sampling - 4 runs
Only with splitting (butina or scaffold_splitter) - 2
With sampling and splitting - 8 runs

15 runs

SP1

Classical (without sampling and TTS) - 1 run
Only with splitting (butina or scaffold_splitter) - 2

We do not run sampling with the balanced dataset

3 runs

SP2

Classical (without sampling and TTS) - 1 run
Only with up/down sampling - 4 runs
Only with splitting (butina or scaffold_splitter) - 2
With sampling and splitting - 8 runs

15 runs

Results

It could be too time-consuming to check every folder, therefore I decided to make a table with a summary.

I use run names as the table index and metrics as a column and the following notations:
1) AL_T - the mean AL value is higher and the difference is statistically significant
2) AL_F - the mean AL value is higher and the difference is not statistically significant
3) N_AL_T - the mean non-AL value is higher and the difference is statistically significant
4) N_AL_F - the mean AL value is higher and the difference is not statistically significant
5) E - the mean AL and non-AL values are equal

Please, see the full table on <u>GitLab</u>. Also, I a generated <u>table</u> with value_counts(), so it could be easier to drawn conclusions from .

	AUC_LB_test	AUC_test	AUC_UB_test	Accuracy_test	F1_test	MCC_test	AUC_LB_validation	AUC_validation	AUC_UB_validation	Accuracy_validation	F1_validation	MCC_validation
AL_F	16.0	15.0	15.0	4.0	13.0	5.0	6.0	6.0	6.0	2	2	3.0
AL_T	17.0	18.0	18.0	29.0	20.0	27.0	27.0	27.0	27.0	28	25	29.0
N_AL_F	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2	4	0.0
N_AL_T	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1	2	1.0

We can observe that **AL outperforms non-AL strategy** with high probability (385/12*33=**97%**) for runs with LGBM

Futher plans

- Make code more elegant and UML diagram
- Add Tropsha's and DeepSCMAs models
- Add Bayesian AL module

I think it could be interesting to add the following modules and make a UML diagram after, but looking forward to receiving feedback from you.