AutoML 2023 Homework 1

Mikołaj Piórczyński

1 Introduction

The following document is a report on experiments conducted as part of Homework 1 from the AutoML course at the Faculty of Mathematics and Information Sciences, Warsaw University of Technology, during the fall semester of 2023. The goal of the homework is to analyze the tunability of hyperparameters, as defined in [PBB19], for three selected machine learning algorithms across at least four datasets, utilizing two different methods of sampling points from hyperparameter spaces. Given that English is the language of science and translating a vast number of ML-related terms is challenging, we opted to compose this document in English instead of the course language, i.e. Polish.

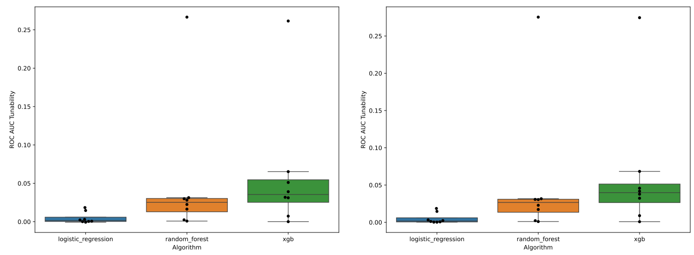
2 Experiments

2.1 Setup

We chose to explore the tunability of commonly employed supervised classification algorithms: logistic regression [Has+09], random forest [Bre01], and XGBoost [CG16]. In our experiments, we used a selected subset of 8 binary classification datasets from the OpenML-CC18 Curated Classification benchmark [Bis+17]. Additional details about the data are available in the appendix. Based on previous literature [PBB19] and the author's experience, we defined hyperparameter ranges for individual models, as presented in Table 1. In our experiments, we measured the mean ROC AUC score obtained through 5-fold cross-validation. For hyperparameter optimization (HPO), we utilized random search [BB12] and Bayesian optimization [SLA12] with Gaussian Processes prior, implemented in the `skopt` package [Hea+17] with 100 iterations for each method. For models' training we used `scikit-learn` package [Ped+11].

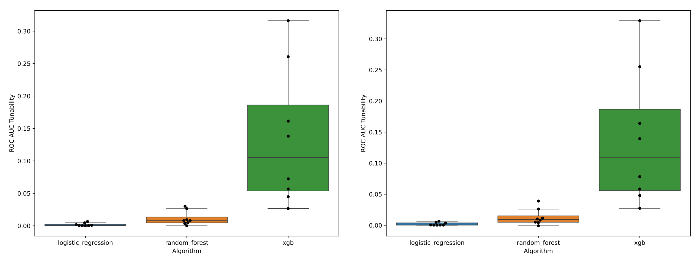
2.2 Algorithms Tunability

To comprehensively assess the algorithm's tunability, we examined it in the context of both package and found optimal default hyperparameters (as defined in [PBB19]). The reason is that we expected that package defaults should offer more robust hyperparameters, considering that computed optimal defaults are based on a relatively small number of experiments (both datasets and HPO iterations). The distribution of tunability of algorithms per dataset is presented in Figure 1 and Figure 2. We can observe that XGBoost is much more tunable than the logistic regression and random forest. Also worth noting is the expected higher tunability values, especially for XGBoost, in the case of optimal default hyperparameters. For precise numerical values, refer to Table 5 in the appendix.



(a) random search (b) bayesian optimization

Figure 1: ROC AUC tunability with respect to the package defaults



(a) random search (b) bayesian optimization

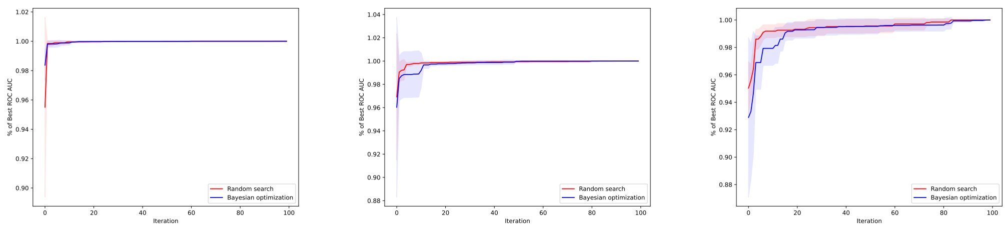
Figure 2: ROC AUC tunability with respect to the found optimal defaults

2.3 Sampling Bias

We also wanted to examine the impact of the hyperparameters sampling technique on conclusions drawn about algorithms' tunability. For this purpose, we employed Welch's t-test with a null hypothesis asserting the equality of means for both methods' results. We found that there is a sampling bias at the 0.05 significance level between random search and Bayesian optimization, i.e. the exact values of tunability significantly differ between these two methods. However, as can we see, both methods yield a consistent order of mean tunability of algorithms.

2.4 Optimization Dynamics

For each method, we investigated the number of iterations needed to obtain stable optimization results. For this purpose, we calculated the cumulative ROC AUC score of the best-found hyperparameters up to each iteration relative to the overall best score obtained during the entire procedure. Averaged across all datasets results are presented in the Figure 3. We can see that logistic regression converges to stable results for both methods after 20 iterations, random forest after 20 iterations for a random search, and 40 iterations for bayesian optimization, whereas XGBoost requires about 80 iterations for both methods. Specific details regarding the best ROC AUC scores attained through both optimization methods can be found in Table 5. It is noteworthy that Bayesian optimization generally yields similar or superior results, albeit at the cost of increased time and a more challenging parallelization process.



(a) logistic regression (b) random forest (c) XGBoost

Figure 3: Cumulative ROC AUC scores relative to the best score achieved throughout the entire procedure for both random and Bayesian search methods

3 Conclusion and Discussion

In conclusion, our findings did not replicate the results from [PBB19], where all investigated algorithms exhibited more similar mean tunability, with logistic regression identified as the most tunable. Several factors may contribute to this discrepancy, including the possibility of an insufficient number of datasets, iterations of HPO algorithms, or poorly specified hyperparameter ranges. It's noteworthy that [PBB19] employed surrogate models, not included in our work. What we discovered is that, despite a statistically significant difference between both optimization methods, they consistently yield very similar solutions, with Bayesian optimization occasionally achieving slightly superior results.

References

[PBB19] Philipp Probst, Anne-Laure Boulesteix, and Bernd Bischl. "Tunability: Importance of hyperparameters of machine learning algorithms". In: *The Journal of Machine Learning Research* 20.1 (2019), pp. 1934–1965.

[Has+09] Trevor Hastie et al. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer, 2009.

[Bre01] Leo Breiman. "Random forests". In: *Machine learning* 45 (2001), pp. 5–32.

[CG16] Tianqi Chen and Carlos Guestrin. "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, pp. 785–794.

[Bis+17] Bernd Bischl et al. "Openml benchmarking suites". In: *arXiv preprint arXiv:1708.03731* (2017).

[BB12] James Bergstra and Yoshua Bengio. "Random search for hyper-parameter optimization." In: *Journal of machine learning research* 13.2 (2012).

[SLA12] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. "Practical bayesian optimization of machine learning algorithms". In: *Advances in neural information processing systems* 25 (2012).

[Hea+17] Tim Head et al. *scikit-optimize: Sequential model-based optimization in Python.* 2017.

[Ped+11] Fabian Pedregosa et al. "Scikit-learn: Machine learning in Python". In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.

4 Appendix

4.1 Hyperparameters Ranges

Table 1: Hyperparameters of the algorithms.

Algorithm	Hyperparameter	Lower	Upper	Distribution
logistic regression	C	10^{-3}	10^{3}	log-uniform
	l1_ratio	0	1	uniform
random forest	n_estimators	10	1000	uniform
	max_depth	1	100	uniform
	min_samples_split	2	10	uniform
	criterion	gini	entropy	–
	min_samples_leaf	1	10	uniform
	max_samples	0.1	1.0	uniform
XGBoost	n_estimators	10	1000	uniform
	max_depth	1	15	uniform
	min_child_weight	1	10	uniform
	subsample	0.1	1.0	uniform
	colsample_bytree	0.1	1.0	uniform
	learning_rate	10^{-3}	1	log-uniform
	reg_alpha	10^{-4}	10^{4}	log-uniform
	reg_lambda	10^{-4}	10^{4}	log-uniform

4.2 Datasets Details

For handling missing values in the `sick` dataset, we applied simple imputation, using the mean for numerical features and the most frequent value for categorical features. All datasets are available on the OpenML service website (`https://www.openml.org`).

Table 2: Details of the datasets used in the experiments

Name	Id	Instances	Features	Numeric features	Symbolic features
blood-transfusion-service-center	10101	748	5	4	1
diabetes	37	768	9	8	1
credit-g	31	1000	21	7	14
qsar-biodeg	9957	1055	42	41	1
kc1	3917	2109	22	21	1
ozone-level-8hr	9978	2534	73	72	1
sick	3021	3772	30	7	23
churn	167141	5000	21	16	5

4.3 Tunability Details

Table 3: Mean tunability with the package defaults

Algorithm	Random search tunability	Bayesian search tunability
logistic regression	0.0048 ± 0.0074	0.0051 ± 0.0073
random forest	0.0497 ± 0.0884	0.0514 ± 0.0913
XGBoost	0.0609 ± 0.0838	0.0638 ± 0.0878

Table 4: Mean tunability with the optimal defaults

Algorithm	Random search tunability	Bayesian search tunability
logistic regression	0.0018 ± 0.0024	0.0021 ± 0.0025
random forest	0.0113 ± 0.0109	0.0129 ± 0.0131
XGBoost	0.1346 ± 0.1064	0.1374 ± 0.1077

4.4 HPO Results

Table 5: Detailed hyperparameters optimization results

Algorithm	Dataset	Random search	Bayesian optimization
logistic regression	blood-transfusion-service-center	0.8853	0.8870
	credit-g	0.6187	0.6185
	diabetes	0.8333	0.8334
	churn	0.6764	0.6764
	ozone-level-8hr	0.9063	0.9065
	sick	0.9497	0.9499
	qsar-biodeg	0.9188	0.9189
	kc1	0.7957	0.7959
random forest	blood-transfusion-service-center	0.8051	0.8139
	credit-g	0.6644	0.6679
	diabetes	0.8400	0.8409
	churn	0.7793	0.7796
	ozone-level-8hr	0.8860	0.8852
	sick	0.9950	0.9952
	qsar-biodeg	0.9227	0.9225
	kc1	0.7989	0.7997
XGBoost	blood-transfusion-service-center	0.8160	0.8292
	credit-g	0.6614	0.6640
	diabetes	0.8446	0.8392
	churn	0.7796	0.7855
	ozone-level-8hr	0.9029	0.9040
	sick	0.9940	0.9948
	qsar-biodeg	0.9204	0.9221
	kc1	0.7959	0.7989