Analysing the Overfit of the auto-sklearn Automated Machine Learning Tool

Fabio Fabris and Alex A Freitas

School of Computing, University of Kent, Kent, CT2 7NF, UK
{F.Fabris,A.A.Freitas}@kent.ac.uk

Abstract. With the ever-increasing number of pre-processing and classification algorithms, manually selecting the best algorithm and their best hyper-parameter settings (i.e. the best classification workflow) is a daunting task. Automated Machine Learning (Auto-ML) methods have been recently proposed to tackle this issue. Auto-ML tools aim to automatically choose the best classification workflow for a given dataset. In this work we analyse the predictive accuracy and overfit of the state-of-the-art auto-sklearn tool, which iteratively builds a classification ensemble optimised for the user's dataset. This work has 3 contributions. First, we measure 3 types of auto-sklearn's overfit, involving the differences of predictive accuracies measured on different data subsets: two parts of the training set (for learning and internal validation of the model) and the hold-out test set used for final evaluation. Second, we analyse the distribution of types of classification models selected by auto-sklearn across all 17 datasets. Third, we measure correlations between predictive accuracies on different data subsets and different types of overfitting. Overall, substantial degrees of overfitting were found in several datasets, and decision tree ensembles were the most frequently selected types of models.

Keywords: Automated machine learning · Overfit · Classification.

1 Introduction

With the growing popularity and number of Machine Learning (ML) techniques, it is increasingly difficult for users to find the 'best' classification workflow (the combination of pre-processing methods, classification algorithms, and their hyper-parameter settings) to be applied to their data. This task becomes even more difficult when one considers the use of ensemble techniques, which may combine several classification workflows to make the final prediction.

Automated Machine Learning (Auto-ML) techniques were devised to solve the problem of how to automatically choose the best classification workflow for a given user's dataset. Typically, Auto-ML methods perform a search that works by using a dataset with instances with known class labels (the training dataset) and returning a fully-parameterised model to be used to predict the class labels of new unlabelled instances.

The state-of-the-art method for Auto-ML is the auto-sklearn tool [3, 8], which was the overall winner of the first ChaLearn Auto-ML challenge [4], and a variant of auto-sklearn also won the second, latest AutoML challenge [1]. Auto-sklearn uses meta-learning and the Sequential Model-based Algorithm Configuration (SMAC) method to build an ensemble of classification workflows given a training dataset. Meta-learning is used to initialise the SMAC search, suggesting a 'reasonable' classification model for the user's dataset, given the estimated predictive accuracy of the model in other datasets. Next, the iterative SMAC search uses a Bayesian approach to explore the huge space of possible classification workflows, training several candidate models per iteration and returning the best model found by the search.

SMAC methods normally return only the best classification workflow found by the search procedure as the final model. However, auto-sklearn exploits the fact the SMAC search procedure produces several 'good' candidate classification workflows that would normally be discarded. These classification workflows are used to build an ensemble instead of being discarded. By default, this ensemble contains at most 50 classification workflows at each iteration. Each workflow has a weight which is proportional to the workflow's relevance for the final prediction.

By default, auto-sklearn works by randomly dividing the training set into two disjoint sets, a learning set and a validation set. The learning set is used during the SMAC search to build the classification workflows. The validation set is used to estimate the accuracy of the workflows. One key aspect of Auto-ML tools like auto-sklearn, which has been to a large extent neglected in the literature thus far, is the degree of overfitting resulting from repeatedly using a fixed validation set across the search. Even though the training set has been properly divided into learning and validation sets, the fact that there are several iterations, each using the accuracy estimated in the validation set to guide the search, may lead to a high degree of overfitting to the validation set. That is, the search may select algorithms and their settings that classify the instances in the validation set very well (since it had several iterations to fine-tune its parameters) but fail to classify the test instances properly (due to model overfitting to the validation set). Besides the just defined overfit (between the validation and test sets), we also analyse two other types of overfit: 1) between the learning and validation sets and 2) between the learning and test sets.

This work has three contributions, all related to experimental analyses of auto-sklearn, as follows. First, we estimate the degree of overfitting of the tool using 3 measures of overfit. This analysis can be useful to ascertain to what extent SMAC's iterative learning procedure is actually hindering predictive accuracy due to overfit. Second, we identify the base classification algorithms most frequently selected by the SMAC method. This can be useful to find classification algorithms that are 'good' across several application domains and could be used as a principled 'first approach' to tackle classification problems. Third, we measure the correlations between several experimental results, aiming to uncover non-obvious relationships between the experimental variables that may lead to further insights about auto-sklearn.

We know of only one AutoML work [6] that measures and briefly analyses overfit, however, there is no work performing a comprehensive overfit analysis (considering 3 types of overfit) comparable to the one presented here. Actually, a very recent and comprehensive survey of Auto-ML studies [9] does not even mention the issue of overfitting.

The rest of this paper is organised as follows: Section 2 presents our experimental methodology. Section 3 presents the analysis of our results. Section 4 presents our conclusions and directions for future work.

2 Experimental Methodology

To measure the predictive accuracy of auto-sklearn we used 17 datasets which are pre-divided into training and test sets, taken from [7]. The training sets are further divided into a 'learning' set (which the SMAC method will use to build the ensemble) and a 'validation' set, which will be used to estimate the predictive accuracy of the models in each iteration of the SMAC search. The test set is never shown to the SMAC method, being reserved to estimate the predictive accuracy of the ensemble classifier created by each iteration of auto-sklearn. Note that, in a normal experimental scenario, the test set would be used only to evaluate the predictive accuracy of the ensemble returned after the last iteration of the SMAC search. However, since this study is interested in the overfit behaviour across iterations, we report results where the test set is also used to evaluate the ensemble at each iteration of the SMAC search. We emphasise that this procedure does not influence the SMAC search in any way.

Table 1 shows basic characteristics of the used datasets [7]. These datasets are very diverse in terms of application domain and dataset characteristics, varying from small datasets with 6 features and 1210 instances (car) to relatively large datasets with 3072 features (CIFAR-10-Small) or 43,500 instances (shuttle).

Auto-sklearn was run for 30 hours on each dataset, using default settings for the other parameters, except that it optimized the AUROC measure (Section 2.1). We ran Auto-sklearn in a computing cluster comprising 20 8-core Intel Haswell machines, with a clock speed of 2.6 GHz and 16 Gb of RAM memory.

2.1 Predictive Accuracy Estimation

We use the popular Area Under the Receiver Operating Characteristic curve (AUROC) to measure the predictive accuracy of auto-sklearn [5]. An AUROC of 1.0 indicates that the model correctly ranked all positive instances after the negative ones. An AUROC of 0.5 indicates that the classifier achieved an accuracy equivalent to randomly ranking the instances. For datasets with more than two class labels, the AUROC is calculated individually per class label and then averaged, weighted by the number of instances annotated with each class label.

2.2 Estimation of Three Types of Overfitting

We measure three types of overfitting, which can be used to analyse different aspects of auto-sklearn's training procedure, as follows.

Table 1. Dataset statistics.

Dataset name	Number of features	Number of training instances	Number of test instances	Number of class labels
gisette	5000	4900	2100	2
shuttle	9	43500	14500	7
kr-vs-kp	36	2238	958	2
car	6	1210	518	4
semeion	256	1116	477	10
abalone	8	2924	1253	28
amazon	10000	1050	450	50
convex	784	8000	50000	2
madelon	500	1820	780	2
waveform	40	3500	1500	3
CIFAR-10-Small	3072	10000	10000	10
dexter	20000	420	180	2
winequalitywhite	11	3425	1468	7
yeast	8	1034	445	9
german_credit	20	700	300	2
dorothea	100000	805	345	2
secom	590	1097	470	2

1. The *learning-validation* overfit – defined as the difference between the predictive accuracy in the learning and validation sets. This overfit can measure if auto-sklearn is successfully controlling the overfit of its training procedure by using the accuracy estimated in the validation set.
2. The *learning-test* overfit – defined as the difference between the predictive accuracy in the learning and test sets. This overfit measures the conventional overfit in standard classification, i.e., the difference between the accuracy in the learning set versus the expectedly smaller accuracy in the test set.
3. The *validation-test* overfit – defined as the difference between the predictive accuracy in the validation and test sets. This overfit can be interpreted as a measure of the effectiveness of using an internal validation set (part of the training set) to estimate the predictive accuracy on the test set (not used during training). That is, if the predictive accuracy in the validation set reflects the expected accuracy in the test set, this overfit should be close to zero. Note that even though the instances in the validation set are not directly used to train the models, the accuracy of the classification models is repeatedly estimated across iterations using the validation set. Therefore, the model choice can overfit the validation set across iterations and the ensemble can perform badly in the final test set while achieving good accuracy in the validation set. Arguably, this is the most interesting type of overfitting from an Auto-ML perspective, and it is not normally investigated in the literature.

2.3 Analysis of the Selected Classification Models

To analyse the classification models present at the final iteration of auto-sklearn we measure the frequency each classification algorithm is chosen and the total relevance weights associated with each classification workflow. Note that a classification workflow may be selected several times to be present in the ensemble, but its total weight may be lower than a workflow that is selected only once.

3 Results

3.1 Predictive Accuracy and Overfit Results

Table 2 shows the main experimental results of our analysis, ordered by increasing degree of validation-test overfit. The columns show, respectively: the dataset name; the final learning set AUROC (the AUROC on the learning set at the last iteration of the SMAC search); the final validation set AUROC; the final test set AUROC; the learning-validation overfit (the final learning set AUROC minus the final validation AUROC); the learning-test overfit (the final learning set AUROC minus the final test AUROC); the training-test overfit (the final training set AUROC minus the final test AUROC); and the total number of iterations. The last row of this table shows the mean overfits across datasets. Note that we do not average the AUROCs as they are not directly comparable, easier problems will naturally have greater AUROCs than harder ones.

Table 2. Results ordered by increasing degree of validation-test overfit.

Dataset	Learning AUROC	Val. AUROC	Test AUROC	Learning-Val. overfit	Val.-Test overfit	Learning-Test overfit	Its.
gisette	1.000	0.998	0.998	0.003	-0.001	0.002	210
shuttle	1.000	1.000	1.000	0.000	0.000	0.000	8
kr-vs-kp	1.000	0.999	1.000	0.001	0.000	0.000	34
car	1.000	0.999	0.999	0.001	0.000	0.001	324
semeion	1.000	0.999	0.998	0.001	0.001	0.003	174
abalone	0.883	0.788	0.785	0.096	0.003	0.099	161
amazon	1.000	0.995	0.991	0.005	0.003	0.009	207
convex	1.000	0.933	0.927	0.067	0.006	0.073	253
madelon	1.000	0.966	0.959	0.034	0.007	0.041	296
waveform	0.986	0.979	0.971	0.007	0.008	0.015	160
CIFAR-10-Small	1.000	0.869	0.861	0.131	0.008	0.139	193
dexter	1.000	0.994	0.982	0.006	0.012	0.018	428
winequalitywhite	1.000	0.851	0.829	0.149	0.023	0.171	186
yeast	0.997	0.868	0.840	0.129	0.028	0.157	74
german_credit	1.000	0.840	0.765	0.160	0.075	0.236	206
dorothea	0.979	0.966	0.878	0.013	0.088	0.101	241
secom	0.974	0.877	0.702	0.097	0.174	0.272	332
Mean overfit				0.053	0.026	0.079	

We can see in Table 2 that the learning AUROC (second column) in almost all datasets is 1.0 or very close to 1.0. Just the dataset "abalone" had an AUROC smaller than 0.97. This shows that the SMAC method is building models with high predictive accuracy in the learning set, as expected.

By analysing the column "Learning-Val. overfit" (fifth column) we can see that almost all datasets (except shuttle) exhibit this kind of overfit, the validation AUROC is almost always smaller than the learning AUROC. This is also expected, as SMAC did not have access to the validation instances during the training of each model. The degree of learning-validation overfit was smaller than 1% in 8 of the 17 datasets. However, a large degree of learning-validation overfit was observed in 6 datasets: 0.160 in german_credit, 0.149 in winequalitywhite, 0.131 in CIFAR-10-Small, 0.129 in yeast, 0.097 in secom, and 0.096 in abalone.

Also, by analysing the column "Val.-Test overfit" (sixth column) we can see that, with the exception of the first 4 datasets, the validation set AUROC is always over-optimistically estimated when compared to the test set AUROC, suggesting that the models are indeed overfitting in the validation set.

Finally, by analysing the column "Learning-Test overfit" (seventh column), we can see that, overall, it presents the largest overfit values across datasets. This is expected, as the SMAC method never had access to the test set to train the ensemble, but it did have direct access to the learning set (to train the classification models) and indirect access to the validation dataset (for predictive accuracy estimation).

Analysing the mean overfits (last row) we can see that the average learning-validation overfit is smaller than the learning-test overfit. This is expected, as the AUROC in the test set is usually smaller than in the validation set while the learning AUROC is the same for these two measures of overfitting. Also, the average validation-test overfit is smaller than the learning-test overfit, this is also expected, as the validation AUROC is naturally smaller than the learning AUROC, which drives the validation-test overfit value down.

Figures 1 and 2 show the evolution of the accuracy of the models across iterations by calculating the AUROC in the learning, validation and test sets. The datasets in the figures are ordered in the same sequence as in Table 2. To save space, we do not show figures associated with the first 5 datasets in Table 2, as those results are trivial (a horizontal line at AUROC=1.0). Note that several plots appear to show only two lines, this is because the test and validation lines are overlapping.

Figure 1 and the first three plots in Figure 2 show that the validation AUROC and the test AUROC are tracking very closely. Hence, for these datasets, the validation AUROC is a good estimator for the test AUROC, although there is little improvement in the validation and test AUROC along the search iterations for some datasets.

The last three plots in Figure 2, however, show that auto-sklearn is clearly overfitting to the validation set. Note that, in these plots, the validation AUROC increases with the iterations, while the test AUROC decreases. The results for the dorothea dataset are especially interesting, since there was a test AUROC *decrease* between the first and last iterations of the SMAC algorithm while the validation AUROC increases. We attribute this behaviour, which is also observ-

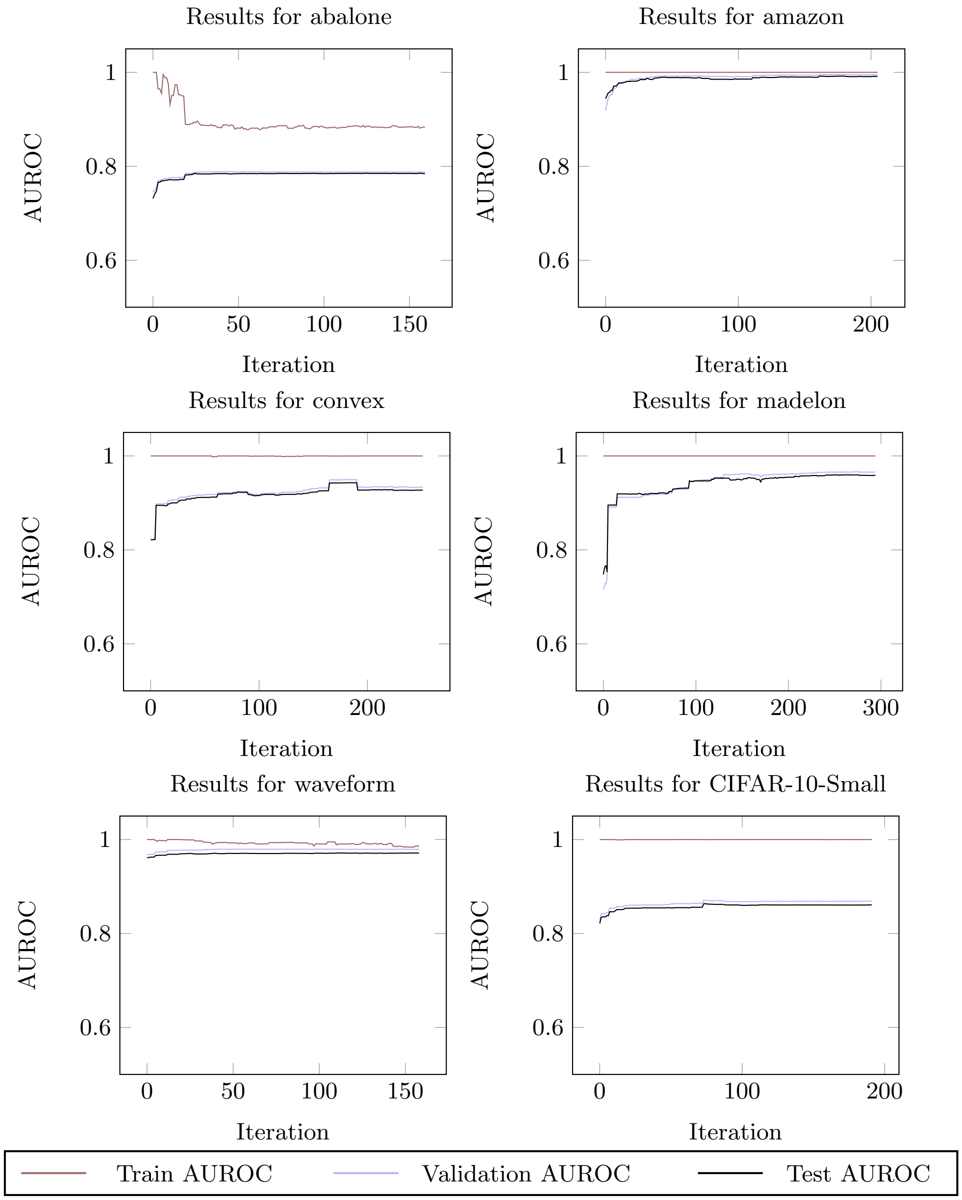

Fig. 1. Training, validation, and test AUROC variation across iterations. Overall, these plots show a 'good' convergence profile: the test AUROC increases with the iteration number.

able to a lesser extent in datasets german_credit and secom, to the fact that the validation set, which is never directly used to build the model, is being constantly queried to estimate the performance of the models, which leads the selected models to overfit the validation set. That is, since the performance in

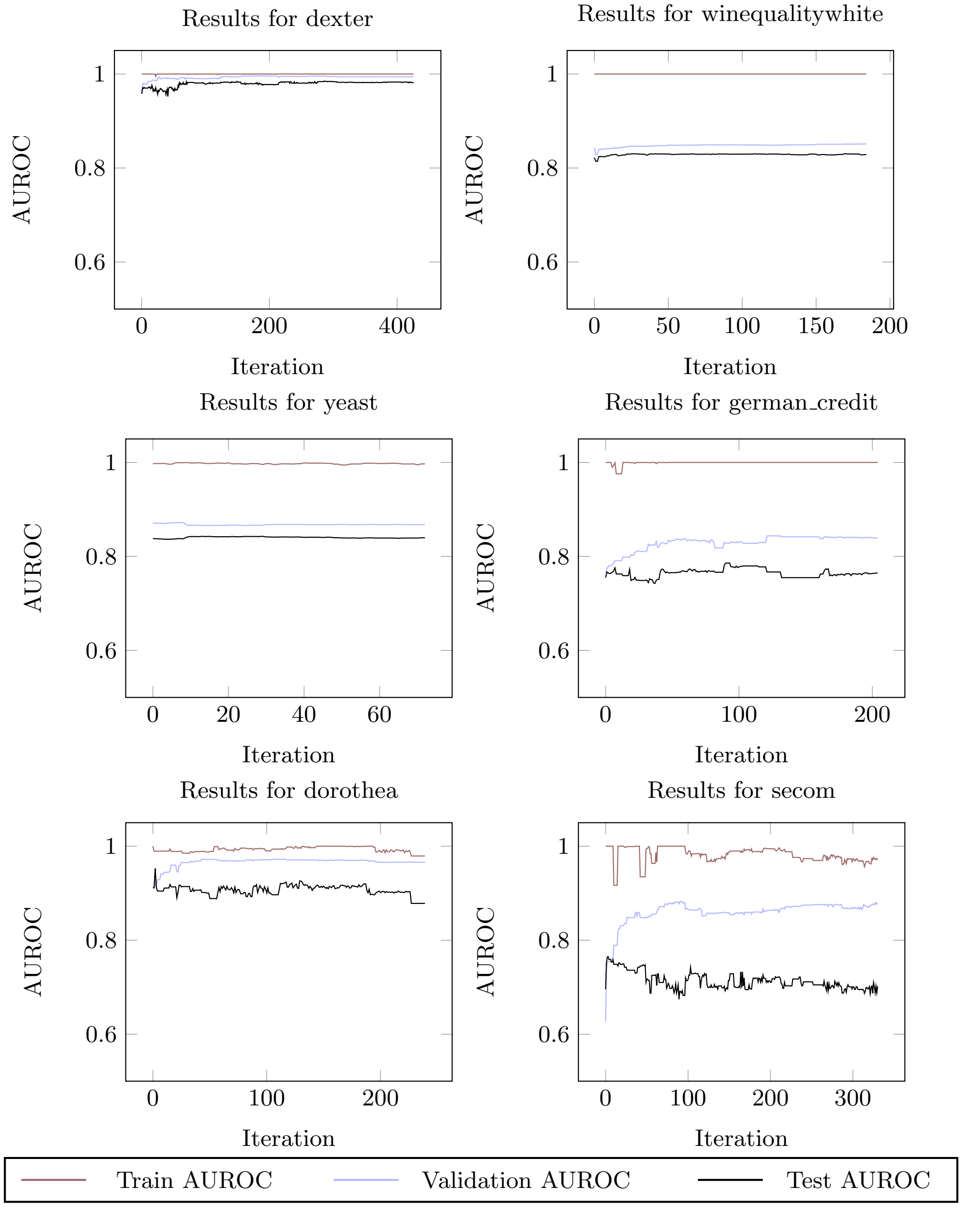

Fig. 2. Training, validation, and test AUROC values across iterations. The plots in this figure show a poor convergence profile: as the iteration number increases, the test AUROC remains close to its initial value, with no sign of clear improvement.

the same validation set is being repeatedly used to guide the SMAC algorithm's search, SMAC is selecting parameter settings and algorithms that are, by some degree of random chance, good at classifying instances in the validation set, but have poor performance in the testing set.

On another note, 11 of the 17 datasets evaluated in this work had a difference between the final test AUROC and the test AUROC after one iteration of just 0.01 or less. Out of those 11 datasets, the first five (gisette, shuttle, kr-vs-kp, car and semeion) are probably too “easy” for classification algorithms, having a test AUROC after just one iteration already close to 1.

3.2 Selected models

Table 3 shows the classification algorithms selected to be part of the final ensemble, with their total weight and the number of times each algorithm was selected to be part of the final iteration’s ensemble. The summation of these frequency numbers is the number of members in the final iteration’s ensemble.

Note that the algorithm frequency varies from just one final ensemble member in the shuttle dataset to 28 members in the dexter dataset; which is much smaller than the maximum allowed number of 50. Also, a high algorithm frequency in the final ensemble does not necessarily imply high importance. For instance, for the kr-vs-kp dataset, gradient boosting was selected 10 times, many more than the other algorithms. However, its weight (0.50) is similar to the weight associated with the extra_trees algorithm (0.46), which was selected only twice.

Table 4 shows a summary of Table 3, presenting the average weights and selection frequencies of each classification algorithm across all datasets.

Tables 3 and 4 contain 8 unique classification algorithms: 1) extra_trees and 2) random_forests, both ensembles of decision trees that randomly sample instances and a feature subset from which a feature is selected for each data split. Note that extra_trees introduce extra randomization to the decision tree ensemble by also randomizing the cutoff point of each split. 3) passive_agressive is an online version of SVM. 4) gradient_boosting induces a regression tree based on the negative gradient of the loss function. 5) lda is a Linear Discriminant Analysis classifier. 6) libsvm_svc and 7) liblinear_scv are both versions of a SVM classifier. 8) qda is the Quadratic Discriminant Analysis classifier.

Interestingly, the two classification algorithms with the highest average weight and average frequency across datasets are based on decision tree ensembles.

3.3 Statistical Analysis

The statistical analyses presented below are based on the Pearson’s correlation coefficient (r) under the null hypothesis of $r = 0$ using a t-test. We have tested in total 12 hypotheses, one hypothesis for each unordered column pair in Table 2, excluding the hypotheses involving correlations between the variable pairs that are deterministically correlated (e.g.: ‘Test AUROC’ and ‘Learning-test overfit’, since Learning-test overfit is equal to Learning AUROC minus Test AUROC). We used the well-known Bonferroni correction for multiple hypotheses testing [5], so the adjusted threshold of statistical significance (the α value) we consider below is $\alpha = 0.05/12 \approx 0.004$. We show the results of these analyses in Table 5.

Table 3. Distribution of selected classification algorithms per dataset. The second column shows the final composition of the SMAC-generated ensemble, displaying the name of the base classification algorithm, followed by two numbers in parenthesis: the total weight of the algorithm in the interval $[0, 1]$ and the number of times the algorithm was selected to be in the final ensemble for a given dataset.

Dataset	Selected algorithm (weight, frequency)
gisette	extra_trees (0.18, 1), gradient_boosting (0.30, 8), passive_aggressive (0.52, 1)
shuttle	random_forest (1.00, 1)
kr-vs-kp	liblinear_svc (0.04, 1), extra_trees (0.46, 2), gradient_boosting (0.50, 10)
car	passive_aggressive (1.00, 16)
semeion	passive_aggressive (0.42, 8), extra_trees (0.58, 5)
abalone	random_forest (0.42, 7), liblinear_svc (0.58, 16)
amazon	extra_tree (0.12, 4), random_forest (0.30, 12), passive_aggressive (0.58, 3)
convex	gradient_boosting (1.00, 15)
madelon	extra_trees (0.36, 7), libsvm_svc (0.64, 6)
waveform	liblinear_svc (0.02, 1), random_forest (0.22, 6), passive_aggressive (0.24, 3), lda (0.52, 5)
CIFAR-10-Small	random_forest (0.40, 10), qda (0.60, 1)
dexter	lda (1.00, 28)
winequalitywhite	extra_trees (1.00, 18)
yeast	random_forest (0.26, 1) extra_trees (0.74, 15)
german_credit	libsvm_svc (0.02, 1), random_forest (0.18, 3), extra_trees (0.80, 12)
dorothea	random_forest (1.00, 1)
secom	extra_trees (1.00, 12)

Table 4. Average weight and selection frequency of each algorithm across all datasets.

Classification algorithm	Avg. weight across datasets	Avg. freq. across datasets
extra_trees	0.31	4.47
random_forest	0.22	2.41
passive_aggressive	0.16	1.82
gradient_boosting	0.11	1.94
lda	0.09	1.94
libsvm_svc	0.04	0.41
liblinear_svc	0.04	1.06
qda	0.04	0.06

As expected, there is a strong statistically significant correlation between the validation AUROC and the test AUROC ($r = 0.90$). Hence, the model's AUROC in the validation set is a good predictor for the AUROC in the test set.

There is a strong, highly statistically significant, negative correlation between the test AUROC and the learning-validation overfit ($r = -0.84$). That is, the greater the learning-validation overfit, the lower the test AUROC. This is expected, as models with a high overfit tend to perform worst in the testing set.

Table 5. Correlations between pairs of measures in Table 2. The top three rows show the statistically significant correlations ($\alpha = 0.004$) among the measures in Table 2 using Pearson's correlation coefficient (r), ordered by absolute r value. The remaining rows show the non-statistically significant correlations, also ordered by absolute r value. The first and second columns show the measures being tested, the third column shows the r value and the last column shows the p-value associated with that r value.

Measure 1	Measure 2	r	p-value
Validation AUROC	Test AUROC	0.90	1.21×10^{-6}
Test AUROC	Learning-validation overfit	-0.84	2.30×10^{-5}
Validation AUROC	Learning-test overfit	-0.81	7.61×10^{-5}
Learning AUROC	Validation AUROC	0.55	2.23×10^{-2}
Learning AUROC	Test AUROC	0.45	6.81×10^{-2}
Validation-test overfit	Iterations	0.31	2.33×10^{-1}
Learning-test overfit	Iterations	0.13	6.30×10^{-1}
Learning AUROC	Validation-test overfit	-0.11	6.80×10^{-1}
Test AUROC	Iterations	-0.10	6.97×10^{-1}
Validation AUROC	Iterations	0.06	8.14×10^{-1}
Learn.val.overfit	Iterations	-0.05	8.42×10^{-1}
Learning AUROC	Iterations	0.04	8.68×10^{-1}

Similarly, there is a strong, highly statistically significant correlation between the validation AUROC and the learning-test overfit ($r = -0.81$).

Somewhat surprisingly, there is no statistically significant correlation between the test AUROC and the learning AUROC nor between the validation AUROC and the learning AUROC. This reinforces the need for using validation sets to properly estimate the accuracy of the SMAC method.

Also, unexpectedly, there is no statistically significant correlation between the number of SMAC iterations and any measure of predictive accuracy or overfit. We were expecting that the more the validation set is used to estimate SMAC's performance, the greater would be the potential for overfit, but our analysis did not support this notion.

4 Conclusions and Future Work

In this work we have analysed the following two important aspects of the auto-sklearn tool using 17 datasets: 1) the degree of overfitting of the tool in terms of 3 types of overfitting, and 2) the diversity of the base classification algorithms most selected by the tool. The three overfits are defined as follows. The *learning-validation* overfit is the difference between the predictive accuracy in the learning and validation sets. The *learning-test* overfit is the difference between the predictive accuracy in the learning and test sets The *validation-test* overfit is the difference between the predictive accuracy in the validation and test sets.

We have concluded that there is a strong statistically significant correlation between the AUROC in the validation and testing sets, which suggests that, overall, the AUROC in the validation set is a useful proxy for the AUROC in the test set. We have also detected a strong significant negative correlation between the test AUROC and the learning-validation overfit, which suggests that reducing learning-validation overfit could be an effective approach to increase test AUROC. This is an intuitive conclusion since overfitting to the validations set (part of the training set) should reduce the AUROC on the test set. This conclusion is also actionable, since approaches can be developed to control learning-validation overfit during training, such as re-sampling the learning and validation sets or using cross-validation across SMACs iterations [2]. Finally, we have also detected a statistically significant negative correlation between the validation AUROC and the learning-test overfit, which suggests that improving the validation AUROC (which is accessible during training) can lead to reduced learning-test overfit.

Regarding the base classification algorithms selected by auto-sklearn across all 17 datasets, the 2 most selected algorithms (with higher average weights and average selection frequency) were ensembles of decision trees.

Future work includes comparing the results obtained using auto-sklearn with other AutoML tools (such as Auto-Weka [7]), as well as investigating the scalability of Auto-sklearn to much larger datasets.

References

1. Feurer, M., Eggensperger, K., Falkner, S., Lindauer, M., Hutter, F.: Practical automated machine learning for the automl challenge 2018. In: International Workshop on Automatic Machine Learning at ICML-2018. pp. 1–12 (2018)
2. Feurer, M., Hutter, F.: Towards Further Automation in AutoML. In: ICML AutoML workshop. p. 13 (2018)
3. Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., Hutter, F.: Efficient and robust automated machine learning. In: Advances in Neural Information Processing Systems 28, pp. 2962–2970. Curran Associates, Inc. (2015)
4. Guyon, I., Chaabane, I., Escalante, H.J., Escalera, S., Jajetic, D., Lloyd, J.R., Macià, N., Ray, B., Romaszko, L., Sebag, M., et al.: A brief review of the chalearn automl challenge: any-time any-dataset learning without human intervention. In: Workshop on Automatic Machine Learning. pp. 21–30 (2016)
5. Japkowicz, N., Shah, M.: Evaluating Learning Algorithms A Classification Perspective. Cambridge University Press, Cambridge, UK (2011)
6. Kordík, P., Černỳ, J., Frỳda, T.: Discovering predictive ensembles for transfer learning and meta-learning. Machine Learning **107**(1), 177–207 (2018)
7. Kotthoff, L., Thornton, C., Hoos, H.H., Hutter, F., Leyton-Brown, K.: Auto-weka 2.0: Automatic model selection and hyperparameter optimization in weka. The Journal of Machine Learning Research **18**(1), 826–830 (2017)
8. Mohr, F., Wever, M., Hüllermeier, E.: Ml-plan: Automated machine learning via hierarchical planning. Machine Learning **107**(8-10), 1495–1515 (2018)
9. Yao, Q., Wang, M., Escalante, H.J., Guyon, I., Hu, Y., Li, Y., Tu, W., Yang, Q., Yu, Y.: Taking human out of learning applications: A survey on automated machine learning. CoRR **abs/1810.13306** (2018)