Imba: Configuration-free Imbalanced Learning

Maksim Aliev, Sergey Muravyov
National Research University ITMO
St.Petersburg, Russian Federation
imaxaliev@gmail.com

***Abstract*—Class imbalance of the target variable is a common feature of quite a few areas. Classic machine learning models are not the best solution in this case, since there will be a prediction bias towards the majority class. To solve this problem, various balancing techniques have been invented, grouped into: undersampling, oversampling and reweighting. However, their implementation requires manual research and configuration. In order to simplify the use of machine learning models, various methods and tools for automated machine learning are being developed. In this paper, the question of the applicability of existing methods to the problem of imbalanced classification was investigated. As it turned out, this problem is solved by them mainly by the same means as in a balanced classification setting. In this connection, Imba is announced - configuration-free imbalanced learning tool. The AutoML benchmark performance revealed worthy competition with the leading solution for automated machine learning model search and hyperparameter optimization - AutoGluon. But more importantly, these results were achieved with a search space of only three classifiers, resulting in significant reductions in computational costs and hence savings in operating time.**

I. Introduction

In the modern world, machine learning is an integral part of human life. It automates many processes such as text translation, web search, content recommendations, facial recognition, content generation, risk assessment, etc. Machine learning is already actively used in almost any branch of business. And business does not spare huge resources for the work in this direction. All this gives rise to the constant emergence of new non-standard problems, which are sometimes difficult to solve using classical machine learning methods. Finding a relevant machine learning model and optimizing its hyperparameters can take an unlimited amount of time and human effort. In connection with this, various methods and tools for automating machine learning have been actively developed (AutoML) [1].

One of the most common machine learning problems is the problem of imbalanced classification, in which the number of labels of one or more classes is significantly lower than the number of labels of other classes [2]. Examples of areas for which a significant predominance of labels of one of the classes is common: hardware fault detection, disease diagnosis, credit fraud detection, spam detection, churn prediction, etc [3, 4, 5, 6, 7]. However, this task has received unfairly little attention in the field of automatic machine learning. Therefore, an urgent task can be considered the development of a tool that can solve the problem of imbalanced classification no worse than advanced solutions in this area. The results obtained demonstrated the advantage of the proposed imba solution over the leading AutoML solution, and therefore the prospects for further development in this direction. Proposed solution will be soon available as an open-source library.

The rest of the article is organized as follows: Section II provides a review of relevant literature; Section III describes the proposed method and details of its implementation; Section IV provides a description of the experimental setup; section V contains statistical results obtained during the experiment; and finally, section VI summarizes the work done and describes the direction for further research and development.

II. Related work

A. Tools for automated machine learning

This section discusses various solutions for recommending machine learning models for binary classification problems that are publicly available and popular today.

One such solution is AutoGluon-Tabular [8]. Its model search and hyperparameter optimization strategy is based on the defaults setted adaptively. While the ensembling strategy is based on multi-layer stack ensembling and n-repeated k-fold bagging (Fig. 1).

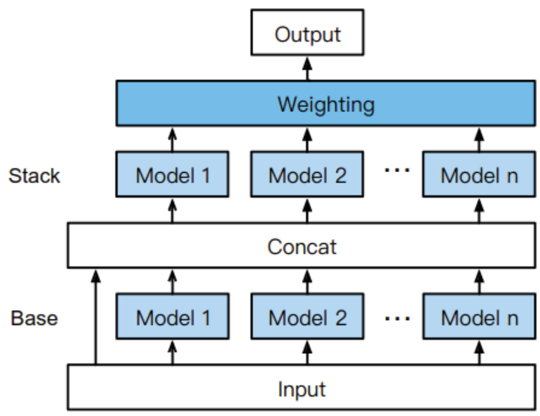

Fig.1 AutoGluon ensembling strategy [8, Fig. 2].

AutoGluon search space also consists of a huge number of deep learning models. Based on the results of various benchmarks, the advantage of AutoGluon over a number of advanced solutions in the field was demonstrated in terms of reliability and speed of finding a solution, and, no less important, the accuracy of the solution found. In addition, the potential of AutoML was further confirmed at the Kaggle competition, where it won 99% of data scientists.

Another well-known solution is auto-sklearn [9]. Within its framework, the search for a model and optimization of hyperparameters is carried out on the basis of the SMAC tool, which uses Bayesian optimization in combination with an aggressive racing mechanism [10]. The ensemble strategy in turn includes: bagging, boosting, stacking and voting. Another distinctive feature of auto-sklearn is the use of meta-learning technique, which is designed to reduce the time to find a solution by leveraging the experience of previous runs. In this case, these are simple meta-features of training datasets [11].

There is also such a solution as TPOT [12]. It is based on genetic programming for model search and hyperparameter optimization. This approach has worked particularly well for finding composite machine learning pipelines. Stacking is used as an ensemble strategy. An example of a pipeline is given on Fig.2.

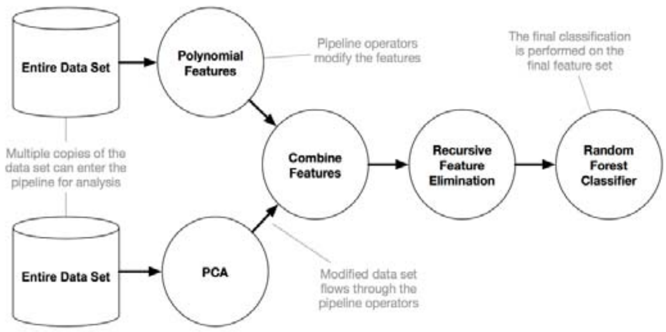

Fig. 2. TPOT pipeline [12, Fig. 1].

The solution, called AutoML-Zero, was developed with the goal of reducing human bias when designing the search space of machine learning algorithms through a generic approach. This solution searches for machine learning algorithms from scratch, that is, by combining mathematical formulas using evolutionary algorithms. It was able to discover two-layer neural networks trained by backpropagation. These neural networks can then be surpassed by evolving directly on tasks of interest [13].

Another solution is new and not yet popular, but deserves its attention - AutoIRAD [14]. It proposes to search for a machine learning model through meta-learning on dataset embeddings(generated in the form of images) by pre-trained convolutional neural networks. Thus, it is possible to achieve results comparable to leading solutions in the industry at much lower computational costs. This approach is called zero-shot learning, which means that the model is trained to predict the class of an object without seeing examples of this class previously.

The last approach that will be mentioned is ML-Plan [15]. It is based on the design of machine learning pipelines based on hierarchical planning (Fig. 3).

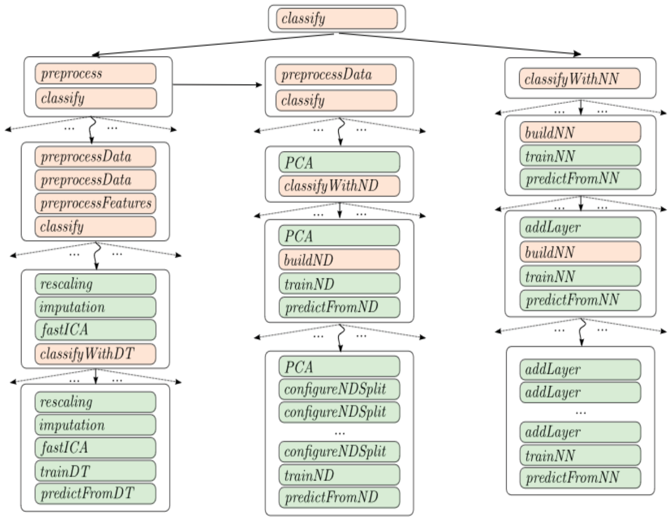

Fig. 3. Creation of pipelines with hierarchical planning [15, Fig. 2].

Of course, there are a number of other solutions with their own subtleties and nuances, but we will not dwell on them in the context of this article. The only thing to note is that based on the results of the analysis, not a single AutoML solution was found that uses ensemble machine learning algorithms, which are generally resistant to class imbalance.

B. Imbalanced learning techniques

We divide approaches for solving the problem of unbalanced classification into the following groups:

1) Oversampling. This approach involves generating new instances for the minority class. One of the most popular approaches is synthetic minority oversampling technique(SMOTE), in which generation is based on interpolation among nearby minority class instances [16].
2) Undersampling. In this case, it is assumed that majority class instances will be removed. One option is cluster centroids, in which candidates for removal are identified by generating centroids based on the chosen clustering method.
3) Reweighting. Modified ensemble methods, such as balanced random forest (BRF) and AdaUBoost [17, 18]. BRF is a variation of random forest with random removal of instances from each bootstrap subset of the data before training the decision stump. AdaUBoost is at the same time a variation of the AdaBoost algorithm, allowing you to optimize unequal loss, caused by class imbalance within successive boosting rounds.

For most real-world problems, oversampling and undersampling are used in combination with each other. At the same time, the analysis of the literature showed that more often than others, reweighting techniques(imbalanced ensemble algorithms) are a more accurate and reliable solution.

III. Method

This section describes a proposed approach to automate the imbalanced classification task called Imba. So, first of all, when creating an AutoML solution, you need to develop an optimization strategy. In this case, the tree-structured Parzen estimator (TPE) algorithm will be used, which is based on Bayesian reasoning to build a surrogate model, which allows you to use information about runs of previous configurations of machine learning models to select the next configuration [19].

After that the space of algorithms and their hyperparameters will be considered. For now, it is limited to ensemble solutions, namely: AdaUBoost, AdaCost and AsymBoost. Their configuration is identical. An example is shown in Fig.4.

```python
class AdaUBoostGenerator(MLModelGenerator):
    n_estimators = scope.int(hp.loguniform('n_estimators', np.log(10.5), np.log(500.5)))
    learning_rate = hp.lognormal('learning_rate', np.log(0.01), np.log(20.0))
    early_termination = True
```

Fig. 4. Python class, responsible for storage of *AdaUBoostClassifier* search space.

Configurations of hyperparameter *n_estimators* are uniformly drawn between low and high values. At the same time configurations of the hyperparameter *learning_rate* are normally distributed based on the mean and standard deviation. We enable *early_termination* by default so as not to wait too long.

Next, the objective function for the optimizer will be discussed (Fig. 5). Basically, it is a wrapper for balanced accuracy. This metric is a modification of the classic accuracy for the problem of imbalanced classification (formula 1).

```python
@staticmethod
def compute_balanced_accuracy_score(hyper_parameters, clf_class, data):
    X_train, y_train, X_test, y_test = data

    clf = clf_class(**hyper_parameters)
    clf.fit(X_train, y_train)
    clf_predictions = clf.predict(X_test)

    return {'loss': balanced_accuracy_score(y_test, clf_predictions), 'status': STATUS_OK}
```

Fig. 5. Implementation of the optimizer objective function.

$$Balanced\ accuracy = \frac{Sensitivity + Specificity}{2} \qquad (1)$$

$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

IV. Experimental setup

The benchmark suite from the OpenML platform with an id=271 will be used. This is one of the most popular benchmarks for evaluating AutoML performance for classification tasks. However, the number of data sets will be reduced. First of all, the number of classes for the target label will be limited to two. Next, a threshold for the maximum size of the data set will be specified, so that it will be possible to wait for the result of AutoML run. Also, most data sets are initially balanced, and therefore there are few options for assessing performance on an imbalanced classification task. In this connection, it was decided to mix them with modifications of the original balanced sets. This was achieved by calling the *make_imbalance* function from the imbalanced-learn package (Fig. 6)[20].

```python
if not is_dataset_initially_imbalanced:
    X_train, y_train = make_imbalance(X_train, y_train, sampling_strategy=class_belongings)
    logger.info(Counter(y_train))
```

Fig.6. Modification of a balanced data set.

In this case, the sampling strategy multiplies the proportion of the minority class by a factor proportional to the number of instances of the majority class. This ensures proportion to be not greater than 0.2 of the total number of instances. The level of imbalance can be: mild (20-40% of minority class instances), moderate (1-20%) and extreme (<1%). The size of the datasets: small(>100 instances and <1000 instances), small/average(<1000 and <2000), average(>2000 and <10000) and large(>10000).The description of data sets claimed in the experiment is presented in Table I.

TABLE I. Dataset metadata.

Dataset name	Dataset size	Imbalance level
Australian	Small	Moderate
numerai28.6	Large	Moderate
phoneme	Average	Mild
credit-g	Small/average	Mild
jasmine	Small/average	Moderate
ozone-level-8 hr	Average	Moderate
madeline	Small/average	Moderate
philippine	Average	Moderate
ada	Average	Mild
Satellite	Average	Extreme
kc1	Small/average	Moderate
Internet-Advertisements	Average	Moderate

gina	Small/average	Moderate
PhishingWebsites	Average	Moderate
sylvine	Average	Moderate

As additional metrics, precision, recall and f2 will be used. They can also be useful for assessing the quality of problem solving.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F_2 = \frac{(1 + 2*2)*(precision*recall)}{(2*2*precision+recall)}$$

And the last thing left to discuss: AutoGluon-tabular will be used as AutoML for comparison, since based on numerous benchmarks it is one of the best solutions. The preset will be set to *good_quality*. The AutoGluon authors report that this preset is quite enough to outperform other AutoML solutions.

V. Results

The calculated performance measures for the model found using AutoGluon are presented in Table II. The results obtained using imba are in turn presented in Table III.

TABLE II. Autogluon performance measures.

Dataset name	Balanced accuracy	Recall	Precision
Australian	0.75	0.51	0.963
numerai28.6	0.5	0	0
phoneme	**0.897**	0.857	0.849
credit-g	0.678	0.491	0.604
jasmine	0.56	0.149	0.852
ozone-level-8hr	**0.73**	0.975	0.964
madeline	0.516	0.987	0.489
philippine	0.621	0.265	0.923
ada	**0.802**	0.928	0.903
Satellite	0.86	0.997	0.995
kc1	0.62	0.288	0.528
Internet-Advertisements	**0.948**	0.993	0.98
gina	**0.841**	0.974	0.766
PhishingWebsites	0.938	0.996	0.916
sylvine	0.856	0.754	0.95

TABLE III. Imba(proposed) performance measures

Dataset name	Balanced accuracy	Recall	Precision
Australian	0.787	0.608	0.912
numerai28.6	0.509	0.226	0.514
phoneme	0.733	0.688	0.56
credit-g	**0.708**	0.814	0.461
jasmine	0.67	0.524	0.753
ozone-level-8hr	0.712	0.576	0.982
madeline	0.562	0.851	0.52
philippine	0.641	0.449	0.731
ada	0.688	0.946	0.78
Satellite	0.983	0.997	**0.908**
kc1	**0.685**	0.788	0.259
Internet-Advertisements	0.71	0.972	0.893
gina	0.73	0.808	0.695
PhishingWebsites	0.889	0.925	0.892
sylvine	0.898	0.983	0.848

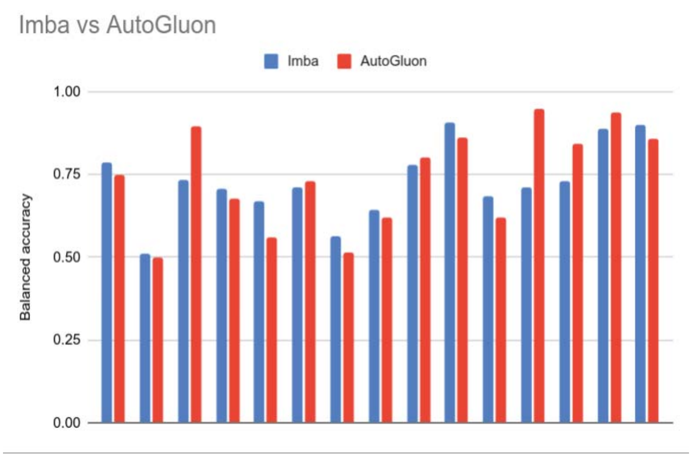

Fig. 7. Comparison of performance of Imba and AutoGluon on the validation metric.

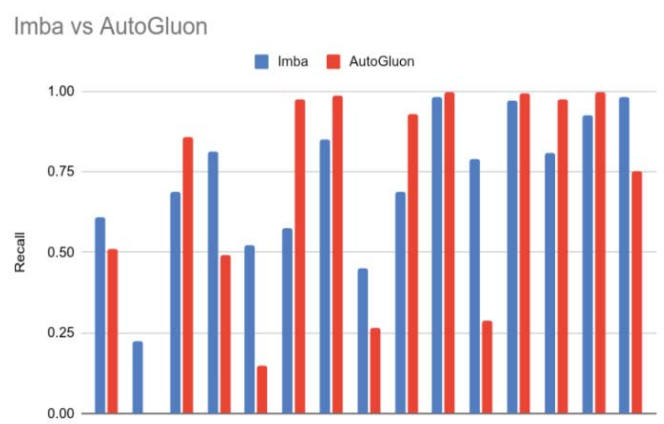

Fig. 8. Comparison of performance of Imba and AutoGluon on the recall metric.

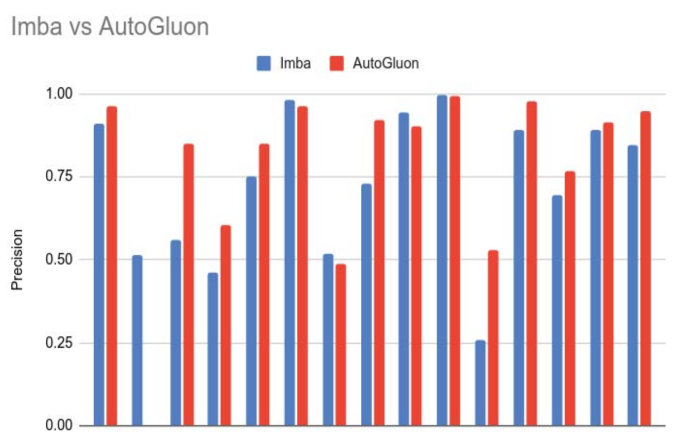

Fig. 9. Comparison of performance of Imba and AutoGluon on the precision metric.

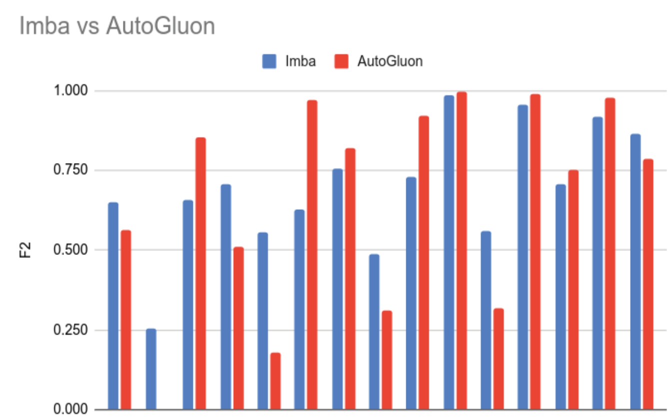

Fig. 10. Comparison of performance of Imba(proposed) and AutoGluon on the f2 measure.

Data sets that were taken from the benchmark in their original form are highlighted in bold, the rest were reduced to maintain a moderate level of imbalance. The advantage in score for each particular dataset is highlighted in red. The best validation metric scores for the solution found from each of the software tools are highlighted in red.

For a more visual representation, the results were placed on the charts in Fig. 7-10, covering: balanced accuracy, precision, recall and f2 measure scores.

To summarize, in most cases the difference in performance does not differ too much. However, the relative advantage of the proposed imba solution is stated on the majority of the selected data sets (8 out of 15) in terms of the validation metric(balanced accuracy). In addition, it should also be noted that the imba search space consists of only a few ensembles, which allows, among other things, to significantly reduce computational costs and, consequently, operating time. Other metrics (recall, precision and f2) are given for additional information.

VI. Conclusion

Automated machine learning model selection and hyperparameter optimization is a growing field of research and development, and using AutoML tools does not always lead to the best possible machine learning model. Like in the case of imbalanced learning. Based on the results of the work done it can be concluded that the leading AutoML solution(AutoGluon) does not use all possible means to combat the problem of class imbalance, which confirms the relevance of the development of the proposed solution Imba. It showed itself to be very worthy in different settings compared to one of the best solutions in the field of automated machine learning model selection and hyperparameter optimization (on average outperforming its competitor), while using a search space many times smaller than AutoGluon, not even considering deep learning. At the same time, most of the models found by AutoGluon are ensembles, consisting of deep learning models.

In the future, it is planned to expand the capabilities of the solution to the problem of multi-classification, as well to add the functionality of working with image data (imbalance learning of images is also a very big and interesting area of research and development). In addition, it is planned to introduce undersampling and oversampling techniques to significantly expand the search space of the machine learning model. Proposed solution Imba will be soon available as an open-source library. There are also other things that can be automated in imbalanced learning (or automated machine learning in general) and this can also be considered in the future research and development.

References

[1] F. Hutter, L. Kotthoff and J. Vanschoren, *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.

[2] G. Haixiang, G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue and G. Bing, "Learning from class-imbalanced data: Review of methods and applications", *Expert systems with applications*, vol.73, 2017, pp. 220-239.

[3] X. Du and L. Cong, "Predicting uncorrectable memory errors from the correctable error history: No free predictors in the field.", *Proceedings of the International Symposium on Memory Systems*, 2021.

[4] N. Liu, X. Li, E. Qi, M. Xu, L. Li and B. Gao, "A novel ensemble learning paradigm for medical diagnosis with imbalanced data.", *IEEE Access* 8, 2020.

[5] S. Makki, Z. Assaghir, Y. Taher, R. Haque, M. C. Hacid, and H. Zeineddine. "An experimental study with imbalanced classification approaches for credit card fraud detection.", *IEEE Access* 7, 2019.

[6] C. Zhao, Y. Xin, X. Li, Y. Yang and Y. Chen, "A heterogeneous ensemble learning framework for spam detection in social networks with imbalanced data.", *Applied Sciences* 10.3, 2020.

[7] B. Zhu, B. Baesens and S. vanden Broucke, "An empirical comparison of techniques for the class imbalance problem in churn prediction.", *Information sciences*, vol. 408, pp. 84-99.

[8] N. Erickson et al, "Autogluon-tabular: Robust and accurate automl for structured data." arXiv, 2020, https://arxiv.org/abs/2003.06505.

[9] M. Feurer, K. Eggensperger, S. Falkner, M. Lindauer and F. Hutter "Auto-Sklearn 2.0: Hands-free AutoML via Meta-Learning", arXiv, 2020, https://arxiv.org/abs/2007.04074.

[10] F. Hutter, H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration", *Proceedings of the Fifth International Conference on Learning and Intelligent Optimization*, Selected Papers 5, 2011, pp. 507–523.

[11] A. Rivolli, L.Garcia, C. Soares, J. Vanschoren and A. de Carvalho, "Meta-features for meta-learning", *Knowledge-Based Systems*, vol. 240, 2022.

[12] R. Olson, and J. Moore, "TPOT: A tree-based pipeline optimization tool for automating machine learning.", *Workshop on automatic machine learning*, 2016, pp. 66-74.

[13] E. Real, et al., "Automl-zero: Evolving machine learning algorithms from scratch.", *International conference on machine learning*, 2020.

[14] I. Dagan, et al., "Automated algorithm selection using meta-learning and pre-trained deep convolution neural networks.", *Information Fusion 105*. 2024.

[15] F. Mohr, N. Wever, and E. Hüllermeier, "ML-Plan: Automated machine learning via hierarchical planning.", *Machine Learning 107*, 2018, pp. 1495-1515.

[16] N. Chawla, A. Lazarevic, L. Hall, and K. Bowyer, "SMOTEBoost: Improving prediction of the minority class in boosting.", *European conference on principles of data mining and knowledge discovery*, 2003,pp. 107-119.

[17] C. Chen, A. Liaw, L. Breiman et al., "Using random forest to learn imbalanced data", vol. 110, 2004.

[18] G. Shawe-Taylor and G. Karakoulas, "Optimizing classifiers for imbalanced training sets", *Advances in neural information processing systems 11*, 1998.

[19] J. Bergstra, et al, "Algorithms for hyper-parameter optimization", *Advances in neural information processing systems* 24, 2011.

[20] G. LemaÃŽtre, F. Nogueira and C. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning.", *Journal of machine learning research*, vol.18, pp.1-5.