# Submission Final Project AutoML Lecture WS 2022/2023

Yi Ren
mail: Yi.Ren@campus.lmu.de

Department of statistics, university of Munich

March 25, 2023

## Contents

# 1 Abstract

Imbalanced dataset is one big problem when applying ML methods on it since it can lead to biased accuracy value of the model. This work aims at finding a model that has a higher accuracy in term of imbalanced data while speeding up the tuning process. Different technologies and methods were applied during the optimization. Besides, several experiments on real-world benchmark datasets have been done to demonstrate the superior performance of the developed framework over the untuned random forest model.

# 2 Introduction

Binary-classification is one of the major tasks of ML, the existing ML methods for this kind of tasks assume that the target variable is balanced so that the accompanying measurement criterion used is fair. But in the real world it's often not the case, situations where only few instances for important (let's say positive) class are available, meanwhile for the less influential class many, happen a lot. Applying ML methods on such sort of datasets will result in a cheating good accuracy that discriminates the positive class. To fix this problem, the learned ML should bias towards the minority class, usual techniques[2] for this are: 1) algorithm-level, which requires special knowledge if the classifier. 2) data-level approach, which balances dataset through resampling. 3) cost-sensitive learning, which introduce different costs for misclassification and using them in learning process. 4) ensemble-based approach, which combines ensemble learning algorithm with one of the three approaches mentioned above. In this work, ensemble-based approach combined with data-level approach is used.

After taking a deeper look at the datasets from OpenML by using skimr package, it's convinced that a data cleaning/engineering is needed since missing values exist for both numeric and factorial variables, there are also constant variables for both arts of variables in datasets regarding "sd" and "n_unique" shown respectively from skimr results, which makes non-sense for data-analyze. The last point that those datasets are very imbalanced is already described in given table from project description. Therefore, to realize an optimized model for imbalanced dataset, firstly, constant variables are removed, then imputations are conducted for missing values with using classification model for factorial variable and histogram for numeric variables. After all these steps, a special technique for imbalanced data called "oversampling" is applied to the dataset to achieve fictional balance. Then, random forest is set as the applying model (because it can handle both factorial and numeric variables, and also for a better comparison with untuned random forest) whose hyperparameters are to be optimized using cross validation (3 folds) and bayesian optimization speed-up technology. To improve the accuracy of the learner, stacking is applied additionally to the whole processing graph as shown below in figure 1.

For benchmark experiment at last, a cross validation with 4 folds is used to compare the performance with tuned RF model and untuned RF model.

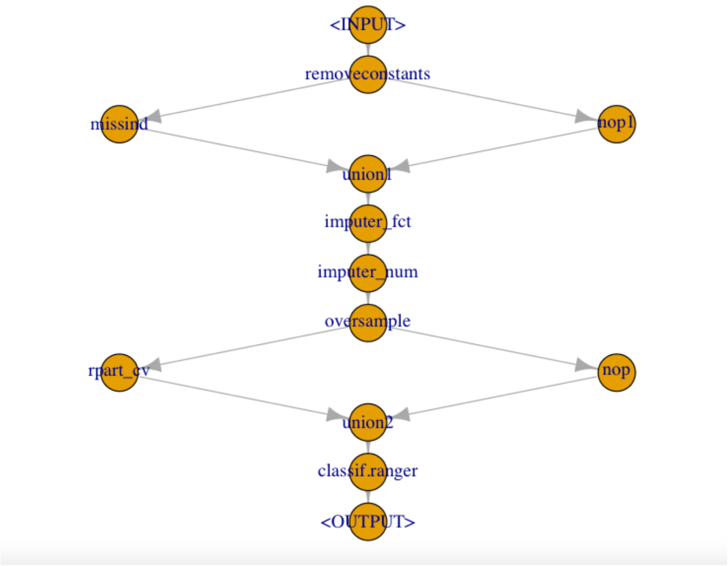


Figure 1: Path

# 3 Methods

## 3.1 Imputation

There are several ways for imputing a missing value in numerical and factorial column, for example impute with media, mode or mean. It's also possible to impute by sampling from non-missing data[1]. An alternative is to impute with a constant value, like setting a constant value out of range as imputation, which is especially sensible in the context of tree-base methods [3]. to keep the stability, in this work, imputation with histogram is used for numerical variables. for factorial columns, an imputation-learner which is trained on all other variables is set as the imputation-method.

## 3.2 Oversampling

As mentioned before, to get a less biased accuracy, data-level approach is chosen to be applied in this work. It consists basically of down-sampling and up-sampling of the original dataset whereby oversampling can be realized through repeating data points of the minor class or applying the Synthetic Minority Ovesampling TEchninque (SMOTE), which creates new synthetic instances. The former one is used for this work.

## 3.3 Emsemble Methods[6]

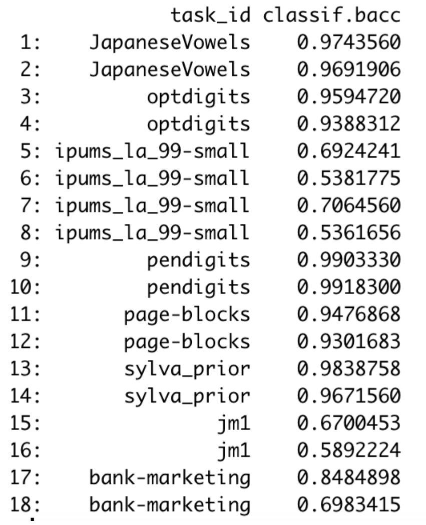
Ensemble Methods are basically a kind of technique that uses weak learners trained on dataset and then combine them in a way to obtain a strong performed model. There are three major kinds of algorithms that perform combination: bagging, boosting and stacking. For Bagging, homogeneous weak learners are usually trained independently on subsets of original dataset, then a deterministic averaging process will be conducted. This consideration aims at minimizing the variance of the prediction. Boosting, in another way, get the ensembled model through iteratively adapting homogeneous weak learners according to the changing data-status resulted from former learner. This algorithm aims at instead minimizing the bias. Stacking is a technique that both for minimizing bias and variance. It tries to address the problem of which models to choose among different skillful models. Therefore, it combines heterogeneous models in two levels, in level-0, base learners will be trained on whole dataset. In level-1, a meta model will be learned on the predictions from level-0 and the original dataset to automatically combine learners from level-0. For this work, a simplified stacking was implemented for using only one classification tree learner with default values in level-0.

## 3.4 Bayesian optimization[4]

Bayesian Optimization is an algorithm that's used for finding the optimal hyperparameter. It's an iterative algorithm that makes use of a continuously updated surrogate model built for the objective function. By optimizing a (comparably cheap to evaluate) acquisition function defined on the surrogate prediction, the next candidate is chosen for evaluation, resulting in good sample efficiency . The typically used surrogate models are gaussian Processes, random forests and Bayesian neutral networks in practice, especially, gaussian process has good performance for numerical spaces but is not easily applicable in discrete or conditional spaces. In the meanwhile, random forests can handle with conditional, categorial and discrete spaces easily with nature. Bayesian neutral network is flexible for all spaces but has also its weakness like the demand of large amount of data. Based on computational budget, there are cheap acquisitions like Probability of Improvement (PI), Expected Improvement (EI), lower/Upper confidence bounds and Thompson Sampling (TS) available, expensive acquisition functions are for example Knowledge Gradient (KG) and entropy search. For this work, Bayesian optimization is applied with default values: for numeric-only parameter space, a Kriging model with kernel “”matern3_2”” is created as surrogate model, for mixed numeric-categorial parameter space, a ranger regression forest with 500 trees is created. The corresponding default acquisition function is EI.

# 4 Performance experiments

To investigate the advantages of the designed path graph along with the tuning method, a benchmark experiment with an untuned random forest learner in terms of balanced accuracy is conducted for all the datasets from OpenML. Nested resampling method is applied to get an unbiased performance measurement of the estimated learner. As accuracy is a misleading metric for imbalanced data sets, balanced accuracy as one of the many metrics that give fair measurement for imbalanced datasets, is chosen instead to measure the performance. Basically, it normalizes true positive and true negative predictions by the number of positive and negative samples, respectively, and divides their sum by two [5]. The inner loop of nested resampling was set to 3 folds CV to facilitate the modelling process, while outer loop was set to 4 folds CV for a more accurate aggregated performance result. Reproducibility is realized by seed setting. dataset with id 41160 has exceeded the time-limitation of one hour in the benchmark experiment, therefore is not presented in the figure 2 shown below (for every task the first row refers to designed learner, notice that the third and fourth task have the same id name). Otherwise, a clear improvement in balanced accuracy from the side of our designed learner is proven except for task “pendigits”, this can be related to the very short tuning time of less than 15 minutes for this individual case or the correspondingly already well-performed learner without tuning. Note: as only in-built functions are used, no docstrings were written for extra. Hardware used: R version of 4.2.2 Cores of CPU: 4 RAM:



```
              task_id classif.bacc
 1:    JapaneseVowels    0.9743560
 2:    JapaneseVowels    0.9691906
 3:         optdigits    0.9594720
 4:         optdigits    0.9388312
 5: ipums_la_99-small    0.6924241
 6: ipums_la_99-small    0.5381775
 7: ipums_la_99-small    0.7064560
 8: ipums_la_99-small    0.5361656
 9:         pendigits    0.9903330
10:         pendigits    0.9918300
11:       page-blocks    0.9476868
12:       page-blocks    0.9301683
13:       sylva_prior    0.9838758
14:       sylva_prior    0.9671560
15:               jm1    0.6700453
16:               jm1    0.5892224
17:    bank-marketing    0.8484898
18:    bank-marketing    0.6983415
```

Figure 2: Results

8.59 GB

# References

[1] Bernd bischl et al. Pipeline operators. https://mlr-org.com/pipeops.html, 2022.

[2] Bengs Casalicchio. "advanced machine learning" course, chapter " imbalanced learning". university of Munich, LMU, S22.

[3] Yufeng Ding and Jeffrey S Simonoff. An investigation of missing data methods for classification trees applied to binary response data. *Journal of Machine Learning Research*, 11(1), 2010.

[4] Lennart Schneider Dr.Janek Thomas. "automated machine learning" course,chapter "bayesian optimization for hpo" and "speedup tecjniques for hyperparameter optimization". University of Munich, LMU, W22/23.

[5] Jeffrey P Mower. Prep-mt: predictive rna editor for plant mitochondrial genes. *BMC bioinformatics*, 6:1–15, 2005.

[6] Joseph Rocca. Ensemble methods: bagging, boosting and

stacking. https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205, 2019.