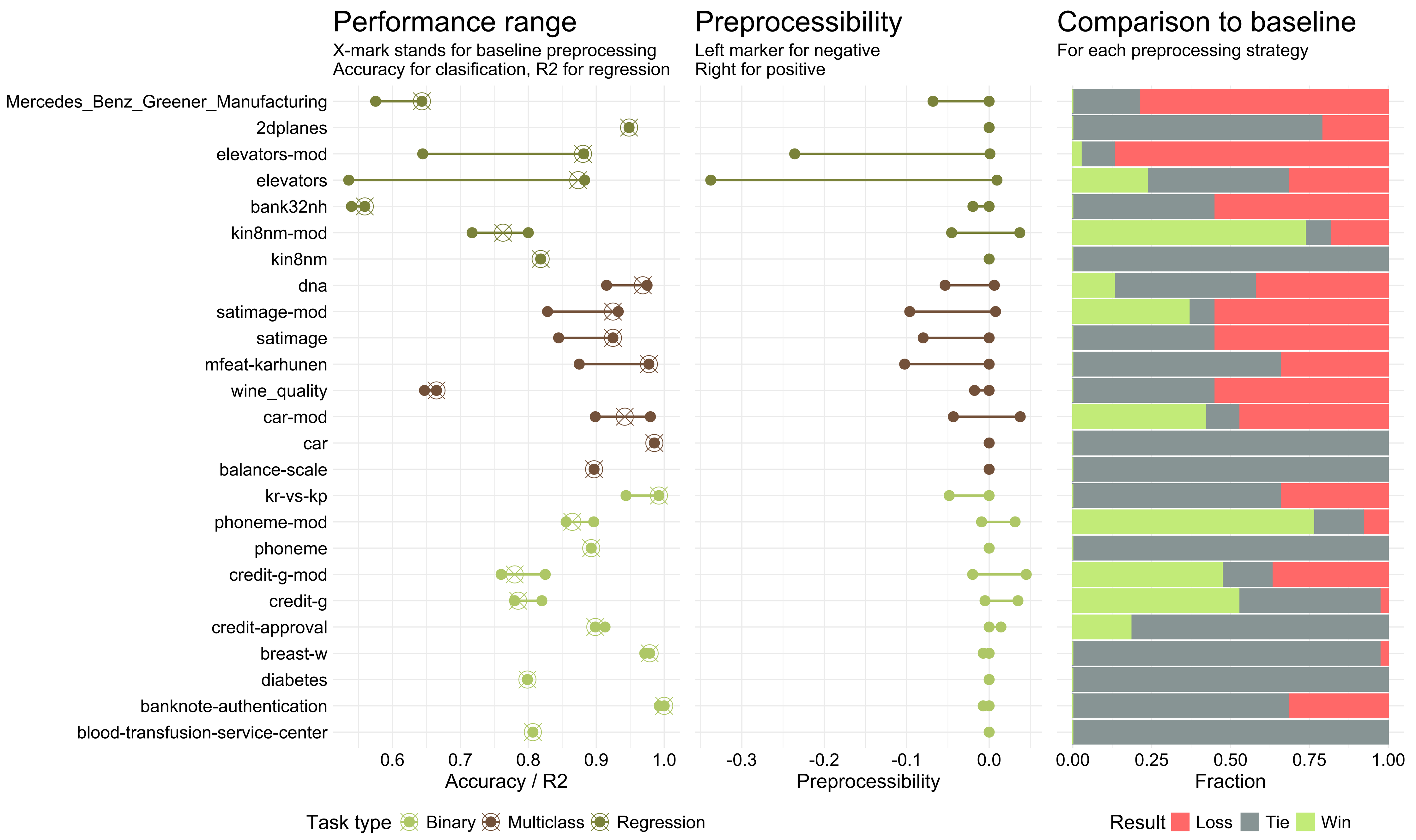
Do Tree-based Models Need Data Preprocessing?

Hubert Ruczyński, Anna Kozak

Warsaw University of Technology



Introduction

The number of machine learning solutions, and algorithms grows rapidly each year. Due to the information overload data scientists trust their own experience or common knowledge more and more. The goal of our study is to evaluate one of such well-known statements, which says that *'tree-based models do not require data preprocessing'*.

Study description

In order to conduct such study, we modified, and used the *forester* package [Kozak and Ruczyński, 2023], being a tree-based model oriented AutoML tool written in R. We used **25 datasets**, where 19 of them come from OpenML (mostly CC-18 benchmark), and 6 are modified version with artificially lowered quality. With *foresters* custom preprocessing module, we prepared **38 preprocessing strategies**, which focused on 3 major data preparation areas, being heuristic removals, data imputation, and feature selection. Eventually, for each scenario we train **105 tree-based models**, based on random-search algorithm, which results in **999 750 solutions**.

Statistic	All strategies	Is FS Used?		Feature Selection Methods				Removal Strategy			Imputation Method			
		No	Yes	Boruta	MCFS	MI	VI	Minimal	Medium	Maximal	MICE	Median-frequency	Median-frequency	KNN
Wins [%]	15.5%	12.3%	17.3%	22.5%	8.0%	14.4%	22.7%	10.0%	13.3%	13.0%	29.2%	29.2%	27.5%	58.3%
Ties [%]	56.6%	70.6%	48.5%	53.0%	76.0%	40.4%	36.0%	85.0%	71.4%	55.0%	12.5%	41.6%	40.0%	25.0%
Loses [%]	27.9%	17.1%	34.2%	24.5%	16.0%	45.2%	41.3%	5.0%	15.3%	32.0%	58.3%	29.2%	32.5%	16.7%
Mean positive preprocessibility	0.009	0.006	0.009	0.008	0.001	0.008	0.003	0.005	0.005	0.005	0.005	0.004	0.002	0.017
Mean negative preprocessibility	-0.048	-0.013	-0.047	-0.013	-0.010	-0.044	-0.021	-0.002	-0.003	-0.011	-0.021	-0.019	-0.016	-0.011

Preprocessibility

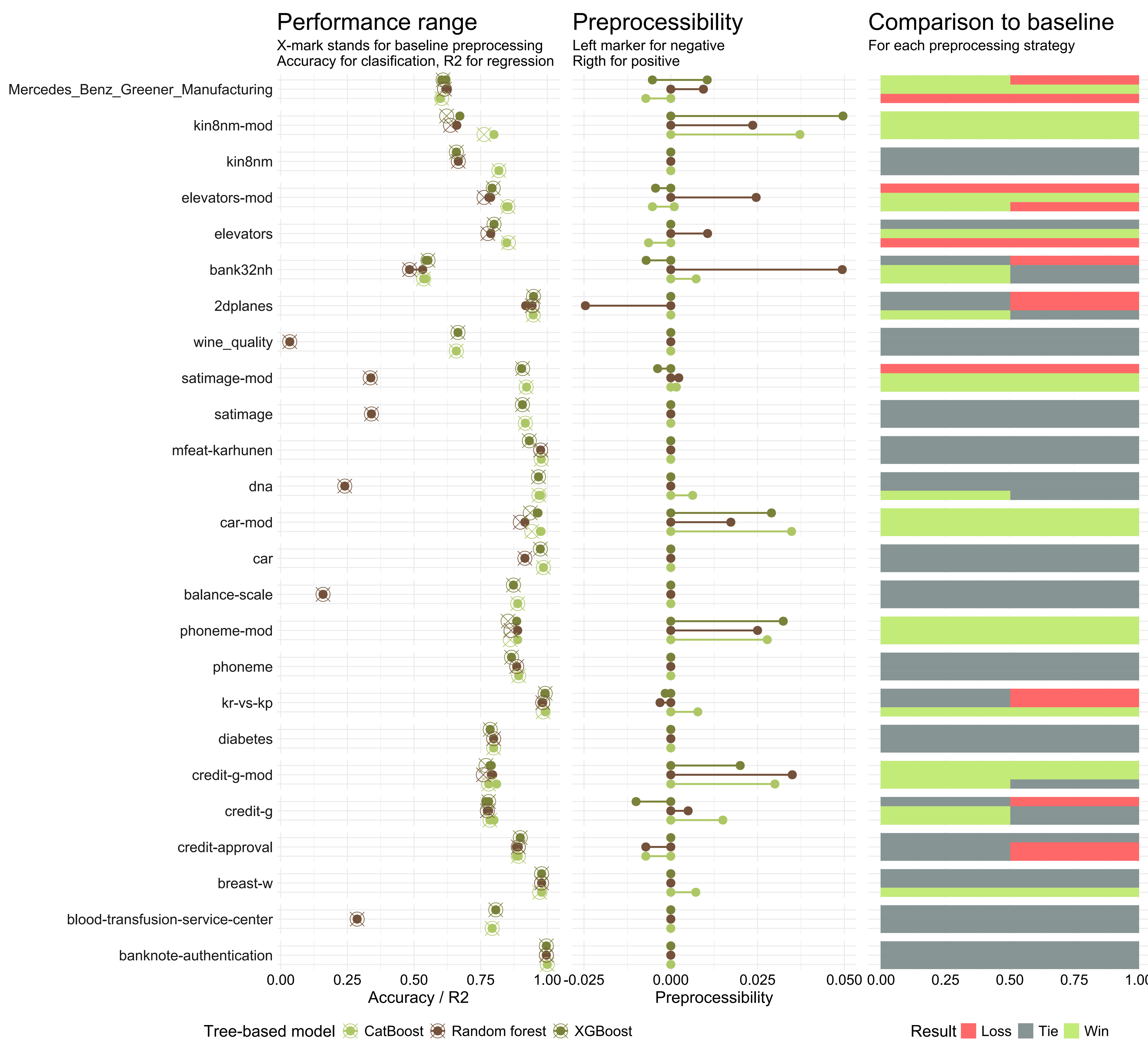
We introduce **preprocessibility** measure, based on tunability from [Probst et al., 2019], which describes how much performance can we gain for a dataset by using various preprocessing strategies. Equation (1) describes positive preprocessibility, and (2) the negative one.

$$P^{+}(D) = \max_{d_i \in D}\big(\max_{m_j(d_i)}(\theta(m_j))\big) - \max_{m_j(B)}(\theta(m_j)), \qquad (1)$$

$$P^{-}(D) = \min_{d_i \in D}\big(\min_{m_j(d_i)}(\theta(m_j))\big) - \min_{m_j(B)}(\theta(m_j)), \qquad (2)$$

where D is a set preprocessed datasets, $d_i \in D$ is a dataset from D, θ is the performance measurement metric which values have probabilistic interpretation (Accuracy / R^2), $m_j(d_i)$ is the model trained on d_i dataset, and B is a baseline dataset for D, prepared with a strategy where we use minimal removal, median-other imputation, and no feature selection.

Statistic	All preprocessing strategies					Best preprocessing strategies			
	Decision tree	Random forest	XGBoost	LightGBM	CatBoost	XGBoost	CatBoost	Random forest	All
Wins [%]	13.7%	18.2%	17.0%	10.7%	22.0%	18.0%	36.0%	36.0%	30.0%
Ties [%]	60.6%	52.5%	55.5%	62.8%	50.5%	64.0%	52.0%	58.0%	58.0%
Loses [%]	25.7%	29.3%	27.5%	26.5%	27.5%	18.0%	12.0%	6.0%	12.0%
Mean positive preprocessibility	0.007	0.014	0.010	0.004	0.010	0.006	0.007	0.008	0.007
Mean negative preprocessibility	-0.063	-0.039	-0.043	-0.066	-0.070	-0.001	-0.001	-0.001	-0.001
Mean maximal score (Accuracy/R^2)	0.752	0.694	0.845	0.795	0.866	0.840	0.862	0.688	0.797



Conclusions

1. In most cases (56,5%) preprocessing does not have impact on tree-based models performance.
2. **Performance reductions** happen more often than **the improvements**, and their mean absolute values are higher.
3. **Feature selection** yields the biggest impact on tree-based models performance.
4. Different feature selection methods yield various impact, where **Boruta** works best with this model family.
5. We should not remove **highly correlated columns** for tree-based models.
6. The **KNN** algorithm is the strongest imputation method.
7. The representants of **tree-based models family** react differently for preprocessing, where **XGBoost** and **CatBoost** benefit the most.
8. Employment of the outcoming best practices (medium removal, KNN, Boruta) leads to visible performance improvements.

References

Kozak, A. and Ruczyński, H. (2023). forester: A Novel Approach to Accessible and Interpretable AutoML for Tree-Based Modeling. In *AutoML Conference 2023 (Workshop Papers)*.

Probst, P., Bischl, B., and Boulesteix, A.-L. (2019). Tunability: Importance of Hyperparameters of Machine Learning Algorithms. *Journal of Machine Learning Research*.