Predicting code defects using interpretable static measures

Wojciech Bogucki
Tomasz Makowski
Dominik Rafacz

Case Studies, WUT, summer semester 2019/2020

Agenda

1. Problem
2. Data
3. Methodology
4. Results

Introduction

basic measures

μ

N

G

n

McCabe and Halstead's measures

$e = 1 / D$

$V = N * \log2(\mu)$

$D = 1 / L$

$L = V / n * \mu$

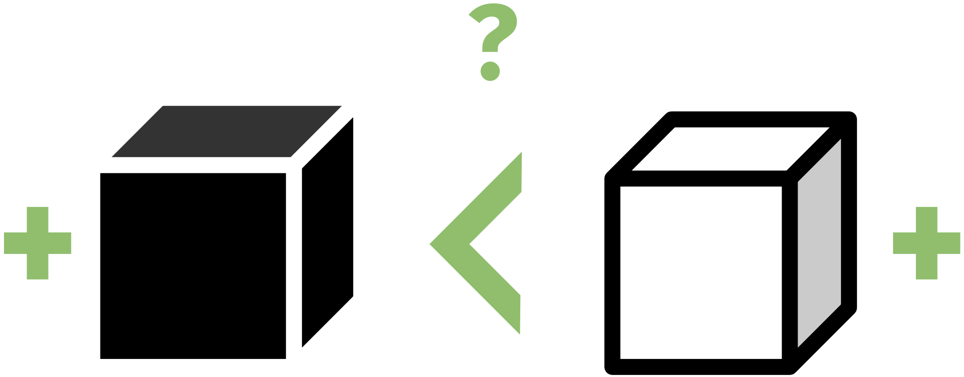
Introduction

μ

N

G

n



$e = 1 / D$

$V = N * \log2(\mu)$

$D = 1 / L$

$L = V / n * \mu$

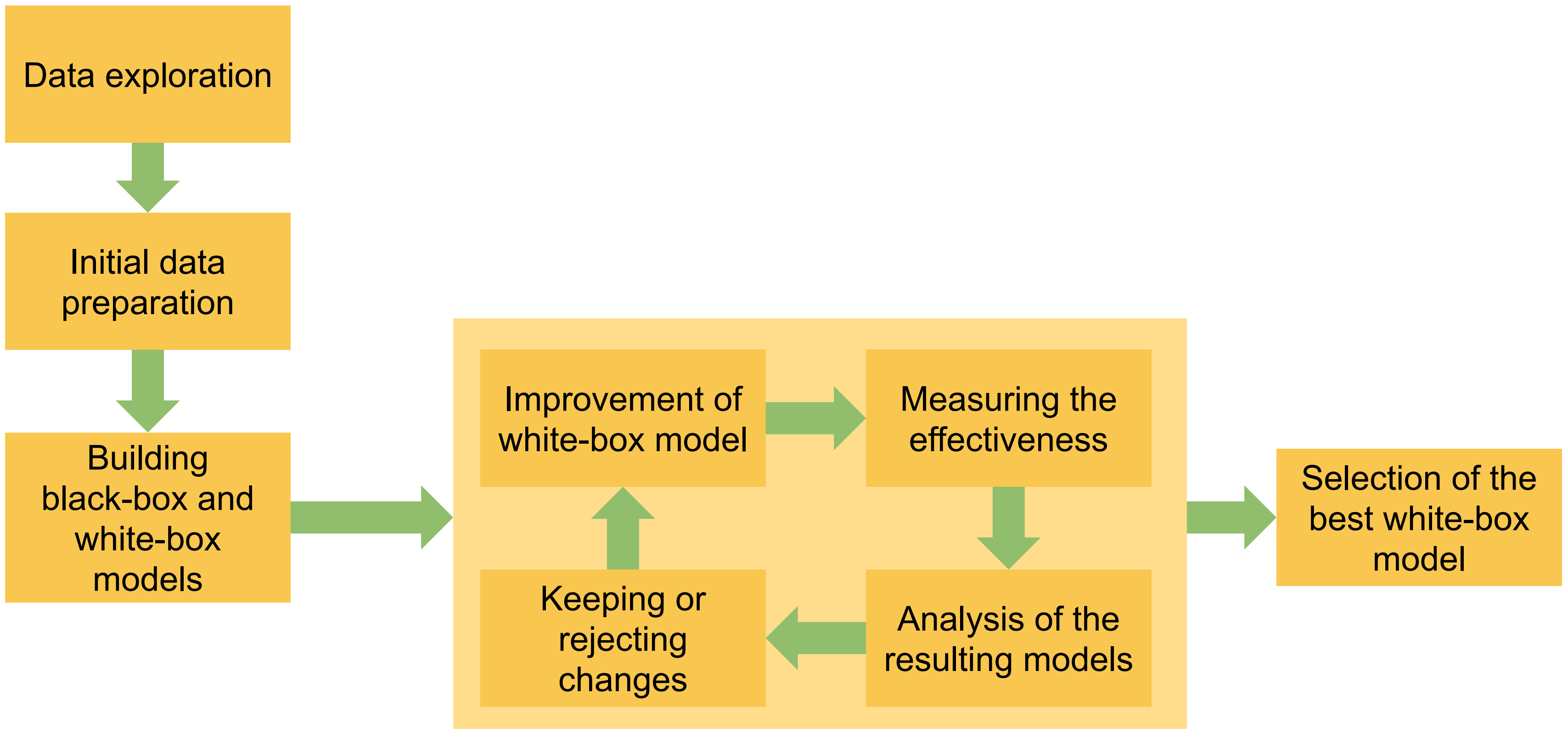
Data

23

15123

μ	N	G	D	b	...	defect
1.2	2	1.4	1.3	1.3	...	TRUE
1.0	1	1.0	1.0	1.0	...	FALSE
17.0	51	7.0	20.31	23029.10	...	FALSE
...	...	...	...	...	...	...

Methodology



Results

The main goal was to beat the ranger model AUC: 0.7916

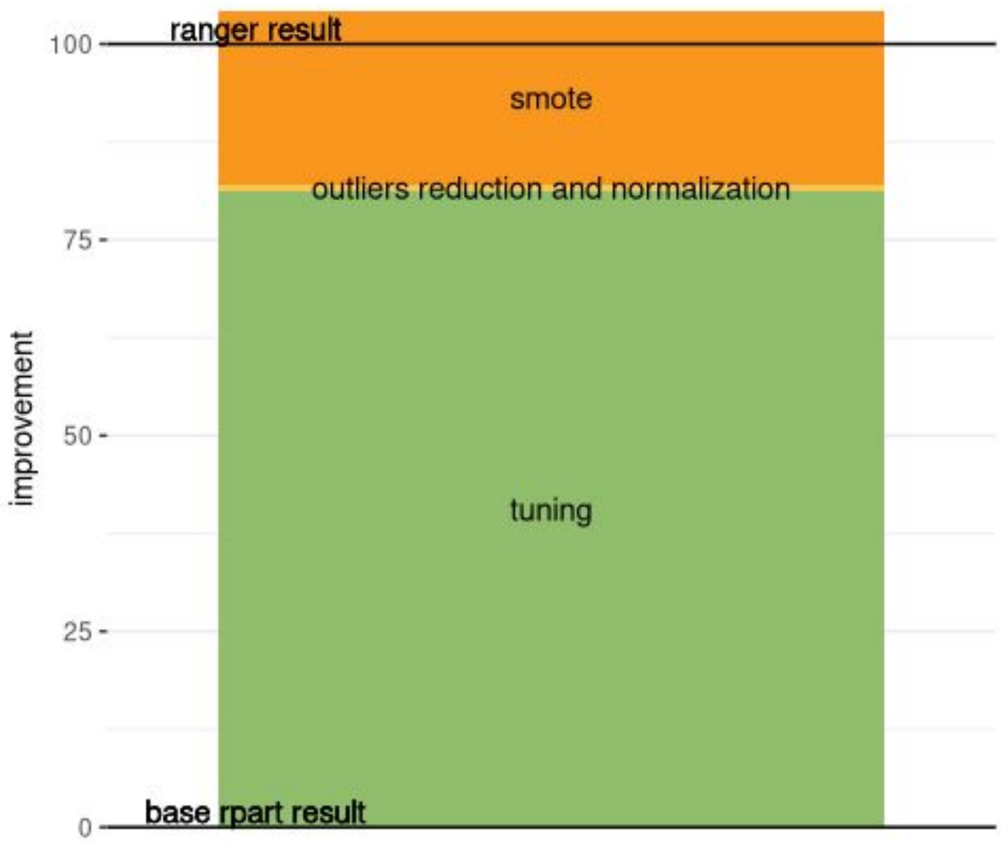
Results

The main goal was to beat the ranger model AUC: 0.7916

AUC results for white-box models

Operation	logreg	kknn	rpart
Base	0.7347	0.7275	0.5000
Rpart tuning	0.7347	0.7275	0.7369
Outlier reduction and normalization	0.7433	0.7320	0.7394
New features selected by ranger	0.7472	0.7288	0.7334
Smote without new features	0.745	0.736	0.804

AUC improvement relative to the base difference



References

- T.J. McCabe, *A Complexity Measure*, p. 308--320 at *IEEE Transactions on Software Engineering*, December 1976
- M.H. Halstead, *Elements of Software Science*, 1977
- P. Biecek, *DALEX: Explainers for Complex Predictive Models in R*, v. 19, p. 1-5 at Journal of Machine Learning Research, 2018
- F. Hu and H. Li, *A Novel Boundary Oversampling Algorithm Based on Neighborhood Rough Set Model: NRSBoundary-SMOTE* at *Mathematical Problems in Engineering*, November 2013
- A. Gosiewska and A. Gacek and P. Lubon and P. Biecek, *SAFE ML: Surrogate Assisted Feature Extraction for Model Learning*, 2019