lab-soft
software engineering laboratory

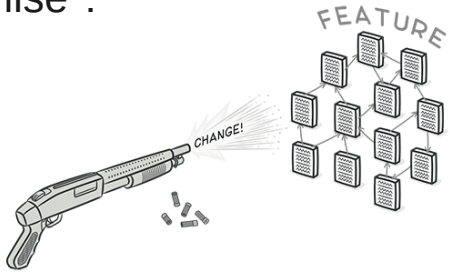
A reproducibility study of code smell predictions

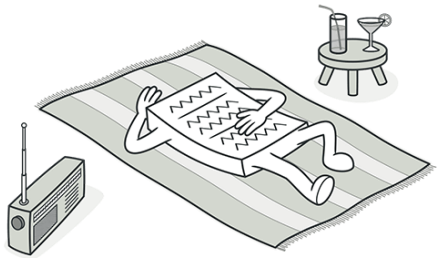
Henrique Gomes Nunes

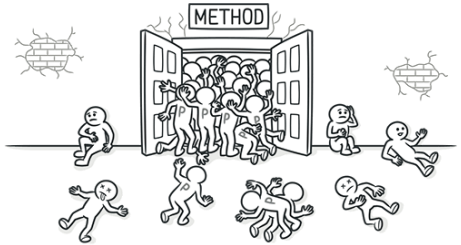
Whats is code smells ?

Codes substructures that compromise*:

- Software evolution;
- Software maintenance.







* Fowler, M. (2018). Refactoring. Addison-Wesley Professional.

Objective

Replicate the study:

> Cruz, Daniel, Amanda Santana, and Eduardo Figueiredo. *"Detecting bad smells with machine learning algorithms: an empirical study."* Proceedings of the 3rd International Conference on Technical Debt. 2020.

To extend and improve the insights.

Experiment Planning

Scope: Goal Question Metric

Analyse **code smells predictions**

for the purpose of **replicating a study**

with respect to the use a **different dataset**

from the viewpoint of **software engineering researchers**

in the context of **academic experiments**.

Scope: Research Questions

RQ1:
How **accurate** is the detection of bad smells using static software metrics and a machine learning algorithm?

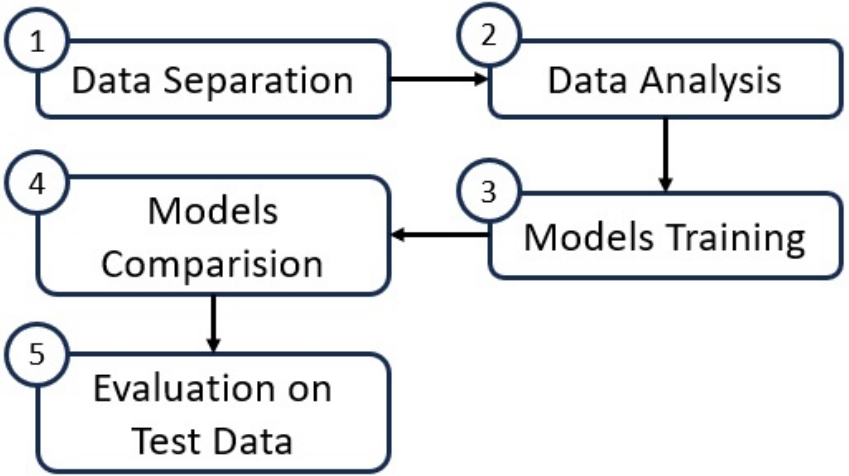
RQ2:
How does the accuracy of the detection **vary** across the use of different machine learning algorithms?

Context: Summary Comparison

	Cruz et. al. (2020)	Current Study
Systems	20 (2010)	30 (2020)
Datasets Size	267.000	346.597
Detection Tools (Statistical Sample Validation)	JDeodorant, JSpirit, Organic, PMD, DECOR	JDeodorant, JSpirit, Organic, PMD, Designite
Algorithms	DT, RF, NB, LR, KNN, MLP, GBM	DT, RF, NB, LR, KNN, MLP, GBM
Features Selection	30 (manual)	22 (manual and auto)
Resample Data?	No	Yes
Measures	F1	F1 and ROC-AUC
Model Selection	Randomized Search	Randomized Search

Experiment Design

Step-by-step


1
Data Separation
2
Data Analysis
3
Models Training
4
Models Comparision
5
Evaluation on Test Data

Pre-experiment

Resample Strategy: undersample and oversample variation between 0.0 to 1.0

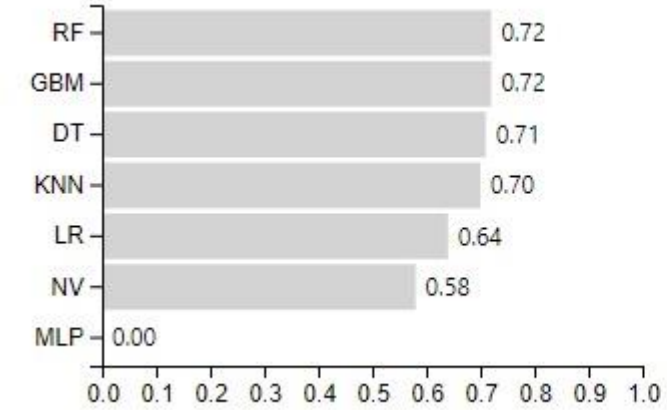
Feature Selection: auto feature selection variation between 1 to max features.

Polynomial Features: experiment with and without PolynomialFeatures:

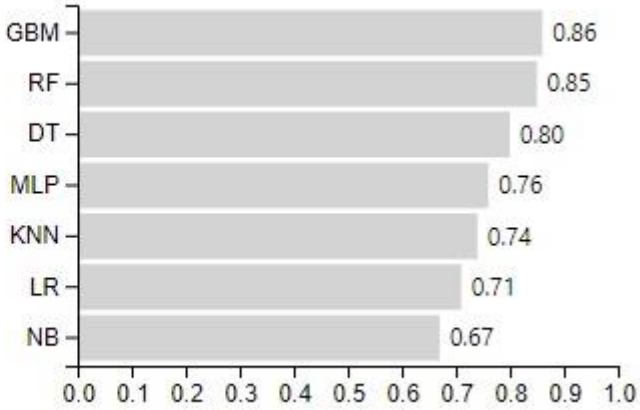
- [1, a, b, a^2, ab, b^2]

Results

God Class: F1 score results

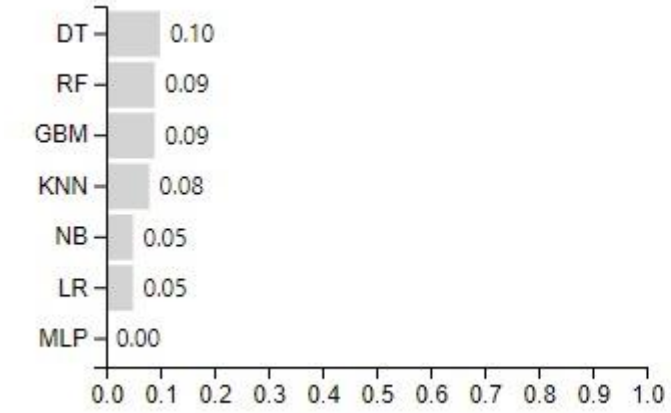


Current Study

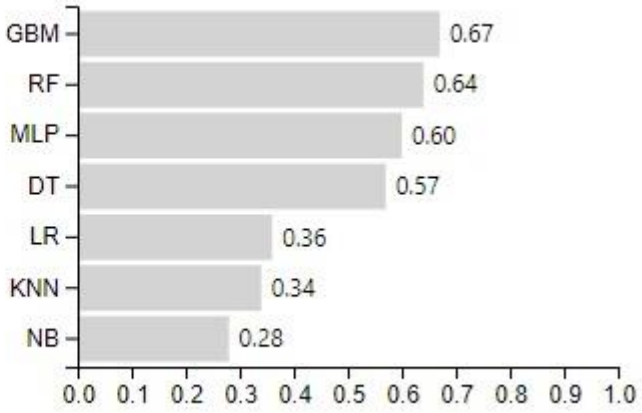


Previous Study

Refused Bequest: F1 score results

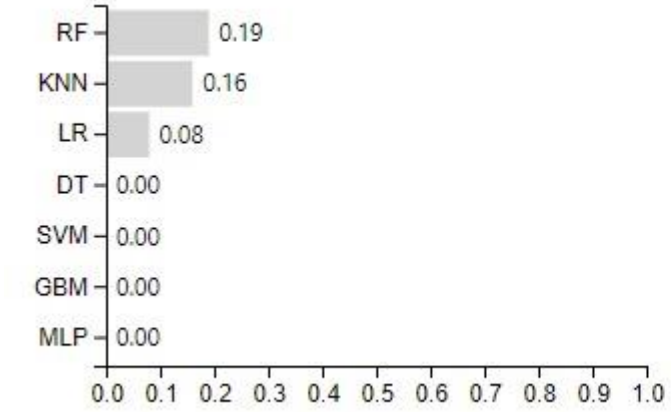


Current Study

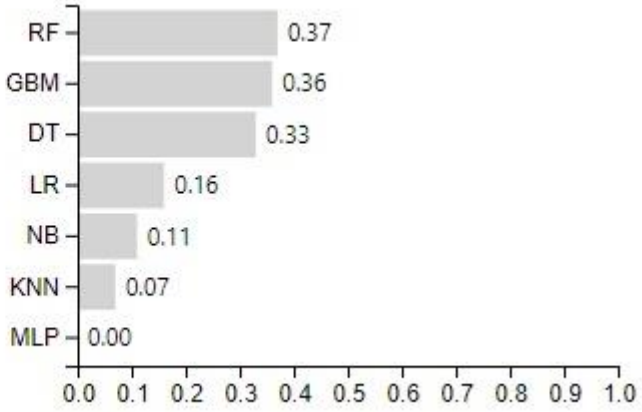


Previous Study

Feature Envy: F1 score results

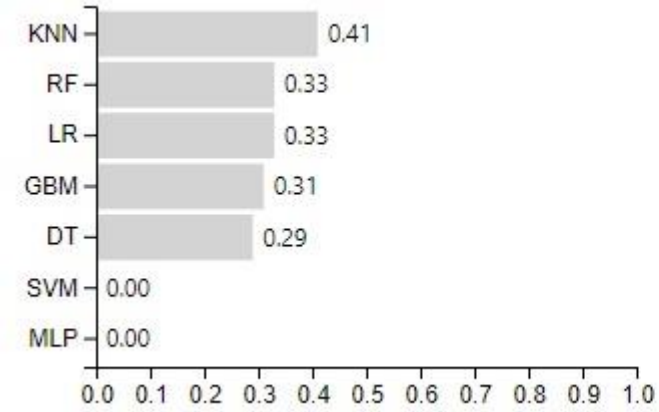


Current Study

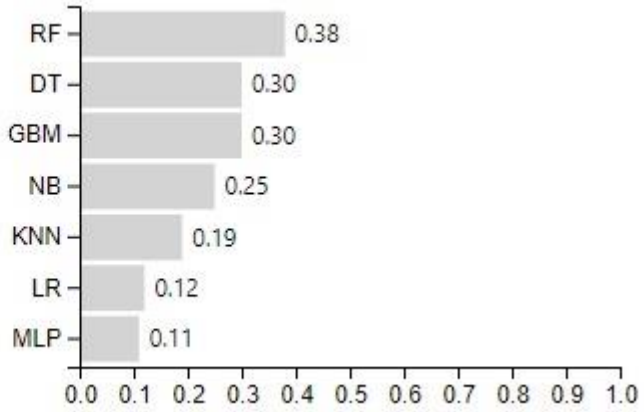


Previous Study

Long Method: F1 score results

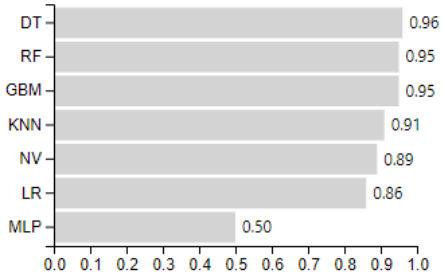


Current Study

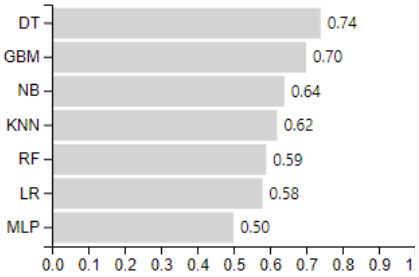


Previous Study

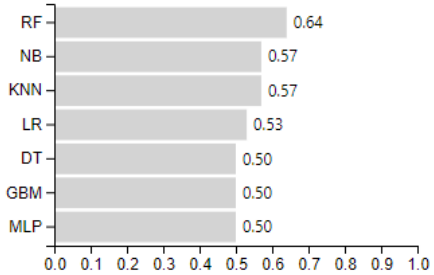
ROC-AUC Results



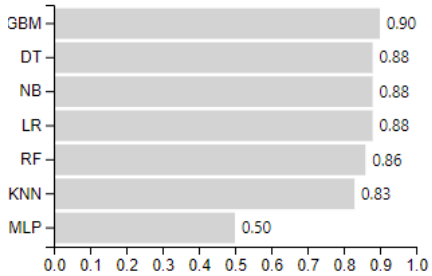
God Class



Refused Bequest



Feature Envy



Long Method

Answer for Research Question 1

RQ1: Previous Study
In this study, the models achieved a good F1 performance for the bad smells GC and RPB, and bad performance for the bad smells LM and FE.

RQ1: Current Study
Except for GC, all F1 performances are below 70%, in both studies. About ROC-AUC performances, except for FE, all scores for DT and GBM algorithms are above 70%.

Answer for Research Question 2

RQ2: Previous Study
The difference between classifiers, reported by F1 measure, can be of more than ten times. For example, for FE, MLP achieved 0.034 and RF 0.351. The bad smell with less divergence on detection performance was GC, with a difference between the maximum and the minimum of about 0.2 (NB: 0.655 / GBM: 0.861). We can observe that XGB and RF performed better on the detection of all smells.

RQ2: Current Study
The variations for the F1 score are very large in some cases. These variations seem to be more controlled using the ROC-AUC metric.

Next Steps

Metrics Evaluation: *“Which are the best software metrics to detect each bad smell?”*

Have a depth and breadth discussion about the results.

Thank you!