Tuning Code Smell Prediction Models: A Replication Study

Nunes, Henrique; Santana, Amanda; Figueiredo, Eduardo; Costa, Heitor.

Introduction

Code smells are bad code design that compromise:

- Software evolution.
- Software maintenance;

How to detect code smells in projects with different sizes, contexts and strategies?

Cruz et. al.* study proposes to **use machine learning** models to predict code smells.

*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.

Objectives and Contributions

Replicate Cruz et. al. study using a dataset with **more modern systems**;

Provide a public dataset with 50k classes and 295k method instances, with ground truth for four types of code smell;

Insights about **feature selection, polynomial features, and resample data**.

Replicate Cruz et. al. Study

Replicate the base study with a **new dataset**.

Both studies evaluated the **same ML algorithms and code smells**.

The current study **extends the base study** with resample data, feature selection and polynomial features techniques.

Category	**Base study**	**Current**
Projects	*20*, Qualita Corpus[46]	*30*, GitHub
Datasets	*35,600 (classes)* *263,211 (methods)*	*50,765 (classes)* *295,832 (methods)*
Detection Tools	JDeodorant, JSpirit, Organic, PMD, *DECOR*	JDeodorant, JSpirit, Organic, PMD, *Designite*
Algorithms	DT, RF, NB, LR, KNN, MLP, GBM	
Features Selection	*30 manual*	*22 manual, 5 auto*
Resample Data	*No*	*Yes*
Measures	F1	F1, *ROC-AUC*
Models Selection	Randomized Search	
Feature Engineering	None	*Polynomial Features*

Selected Code Smells

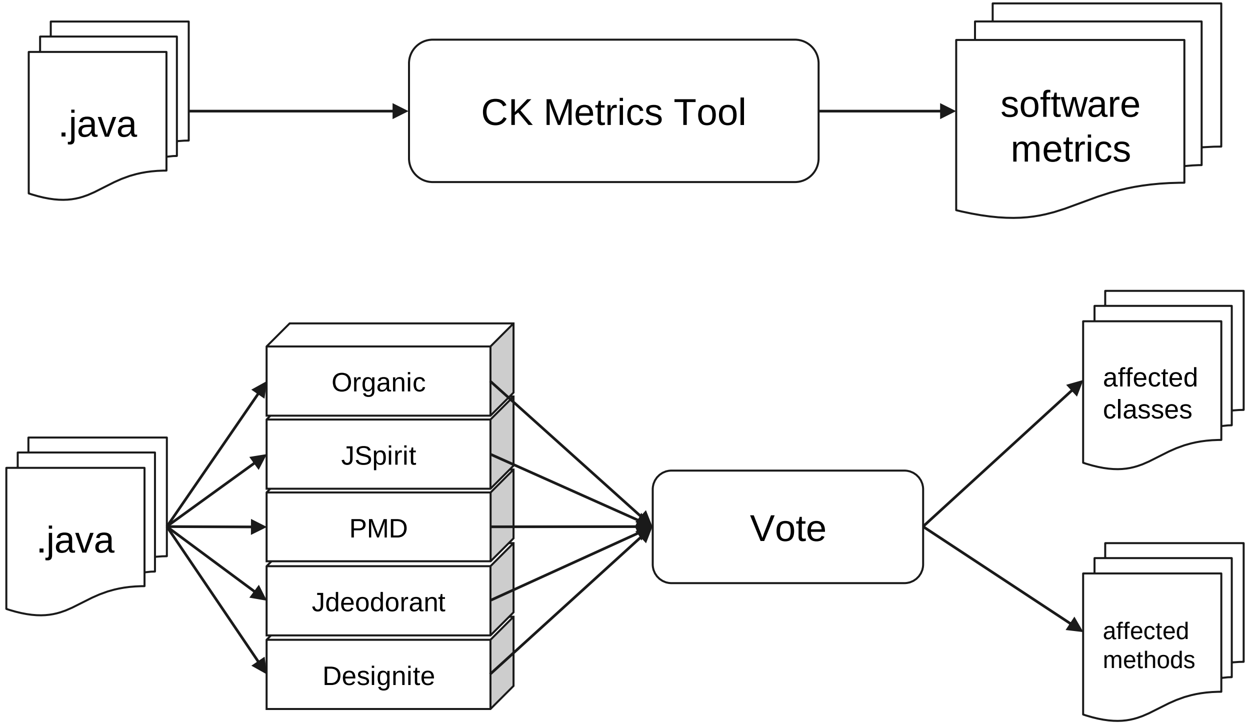
God Class: classes have **excessive responsibilities**, that strongly indicating design flaws.

Refused Bequest: a child class **does not fully support all the methods or data** that it inherits.

Feature Envy: a method that is **more interested in another class** other the one it is in.

Long Method: long and complex method, including many data and responsibilities.

Dataset: Features and Classes



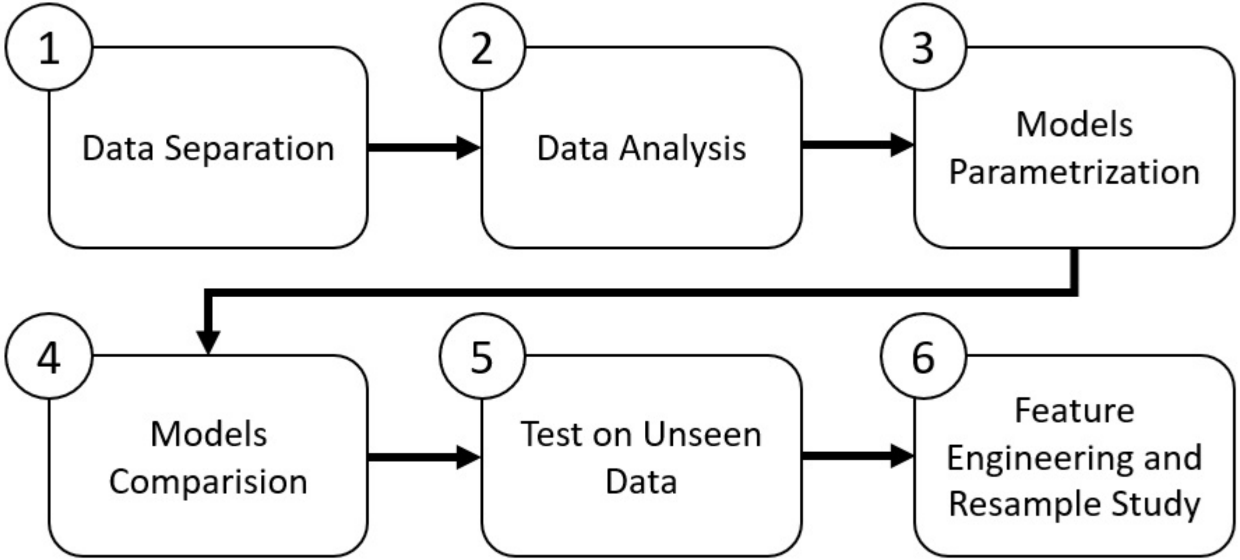
*Used 3 tools per smell

Dataset: Features and Classes (example)

fanin	**wmc**	**dit**		**God Class**
1	10	1	**...**	1
3	4	3		0
20	9	3		1

Experiment Execution

Steps 1 to 5 are the same of the base study, step 6 is an extension of the study:



Results: Base Study Replication

Code Smells	*Highest F1*	*Highest AUC*	*Chosen Algorithm*
GC	RF (.72), GBM (.72), DT (.71)	DT (.96), RF (.95), GBM (.95)	DT
RB	DT (.10), RF (.09), GBM (.09)	DT (.74), GBM (.70), NB (.64)	DT
FE	RF (.19), KNN (.16), NB (.11)	RF (.64), KNN (.57), NB (.57)	RF
LM	KNN (.41), RF (.33), LR (.33)	GBM (.90), DT (.88), NB (.88), LR (.88)	KNN

Results: Feature Engineering

Code Smells	*F1: FS*	*F1: FS + PF*	*AUC: FS*	*AUC: FS + PF*
GC	↓ .34 ↑ .73	↓ .34 ↑ .74	↓ .61 ↑ .89	↓ .61 ↑ .90
RB	↓ .00 ↑ .17	↓ .50 ↑ .54	N/A	N/A
FE	↓ .00 ↑ .07	↓ .00 ↑ .08	↓ .50 ↑ .51	↓ .50 ↑ .52
LM	↓ .63 ↑ .64	↓ .62 ↑ .64	↓ .75 ↑ .76	↓ .74 ↑ .76

Results: Resample

Code Smells	*F1: Oversample*	*AUC: Oversample*	*F1: Undersample*	*AUC: Undersample*
GC	↓ .72 ↑ .76	↓ .89 ↑ .96	↓ .38 ↑ .71	↓ .86 ↑ .97
RB	↓ .00 ↑ .12	↓ .50 ↑ .81	↓ .00 ↑ .09	↓ .50 ↑ .87
FE	↓ .00 ↑ .23	↓ .50 ↑ .73	↓ .00 ↑ .22	↓ .50 ↑ .85
LM	↓ .76 ↑ .78	↓ .88 ↑ .94	↓ .42 ↑ .72	↓ .80 ↑ .93

Discussion and Conclusion: Similarities

Both studies have **very imbalanced dataset**:

- Base Study: GC - 4.77%, RB - 8.96%, FE - 3.46%, and LM - 0.87.
- Current Study: GC - 2.31%, RB - 0.41%, FE - 1.39%, and LM - 0.39%.

In both studies, the **better results** are within the smells with **less imbalance**:

- Base Study: GC and RB.
- Current Study: GC and LM.

Discussion and Conclusion: Differences

F1 metric values were **higher in the base study**.

The **best prediction performances** in the base study were for the GC and RB code smells, while in our study were GC and LM code smells.

Discussion and Conclusion: Extensions

The **AUC metric performed well** in predicting smells.

Resample techniques were better than Feature Selection and Polynomial Features techniques to improve the performance of code smell prediction models.

Thank you for your attention!

I am available for questions or suggestions on the panel with speakers.