Università degli studi di salerno

Flaky Test Prediction

Angelo Afeltra, Antonio Trovato, Francesco Rastelli

26/07/2022

FLAKY TEST PREDICTION

Sviluppo di una pipeline di machine learning per prevedere i flaky test.

---

1 Costruzione del dataset

Senza dati non si può creare un intelligenza artificiale, essi sono fondamentali. Pertanto il primo passo da compiere per lo sviluppo di una pipeline di machine learning che permette di prevedere i test flaky è la raccolta di dati per la costruzione del dataset. Per affrontare tale problema occorre avere degli esempi di test flaky e non flaky, in modo da poter estrarre le caratteristiche necessarie che ci permettono di risolverlo.
Avere degli esempi di test flaky è abbastanza difficile, in quanto risultano complessi da identificare, tuttavia il sito web https://mir.cs.illinois.edu/flakytests/summary.html contiene una lista di 314 repository GitHub su cui sono stati identificati 3742 test flaky. Per ogni test flaky viene riportato:

- Project URL: Progetto GitHub su cui è stato identificato il test flaky.
- SHA Detected: SHA del commit su cui è stato identificato il test flaky. Può essere o non essere l'ultimo commit
- Module path: Path del modulo che contiene il test flaky
- Fully-Qualified Test Name: Nome del test flaky
- Category: Categoria del test flaky rilevato
- Status: Stato del test flaky, ad esempio fixato, rimosso ecc.
- PR Link: Link della pull request in cui e stato fixato il test flaky
- Days to Address PR: Numero di giorni trascrosi, dalla segnalazione del test flaky al fix del test.
- Notes: Informazioni aggiuntive

Si è deciso quindi di utilizzare tali dati per la creazione del nostro datasete. Essi possono essere esportati facilmente all'interno di un file CSV.
Prima di procedere alla creazione del dataset è stata eseguita un analisi dei dati esportati.
Con essa abbiamo notato che:

- Per alcuni progetti i test flaky risultano essere rilevati su più commit differenti, pertanto considerando i vari commit le repository passano da 314 a 400

- Alcuni test flaky risultano essere duplicati su commit differenti, passando cosi da 3742 a 3753
- I test flaky non si riferiscono soltano ad un singolo test case ma alcuni si riferiscono ad un intera test class.

Dopo aver concluso l'analisi, siamo passati alla clonazione delle repository, mentenendo traccia in due file differenti *RepositoryClonate.txt* e *RepositoryNonClonateCorrettamente.txt*, le seguenti informazioni:

- Nome Repository, nel caso in cui la repository possiede dei test flaky su piu commit viene riportato l SHA del commit tra le parentesi
- Path assoluto in cui è stata clonata la repository
- Url Git
- Lista test flaky

Tuttavia durante la fase di clonazione 40 respository non risultano essere clonate correttamente, causando la perdita di 514 test flaky.
Utilizzando il tool **metricsDetector**, per ogni repository clonata sono state estratte le seguenti metriche relative alle classi di test e alle loro rispettive classi di produzione:

Name	Description	Computed on ...
Production and Test Code Metrics		
CBO	Coupling Between Object, i.e., the number of dependencies a class has with other classes [16].	Production Class
Halstead Length	The total number of operator occurrences and the total number of operand occurrences.	Production Class
Halstead Vocabulary	The total number of distinct operators and operands in a function.	Production Class
Halstead Volume	Proportional to program size, represents the size, in bits, of space necessary for storing the program.	Production Class
LOC	Lines of Code, counting both source and comment lines.	Production Class
LCOM2	Lack of Cohesion of Methods version 2, i.e., the percentage of methods that do not access a specific attribute averaged over all attributes in the class.	Production Class
LCOM5	Lack of Cohesion of Methods version 5, i.e., the density of accesses to attributes by methods.	Production Class
McCabe	It uses to indicate the number of linearly independent paths through a program's source code [51].	Test Class
MPC	Message Passing Coupling, measures the numbers of messages passing among objects of the class.	Production Class
RFC	Response For a Class, i.e., the number of methods (including inherited ones) that can potentially be called by other classes [16].	Production Class
TLOC	Number of lines of code of the Test Suite.	Test Class
WMC	Weighted Methods per Class, i.e., the sum of the complexities (i.e., McCabe's Cyclomatic Complexity) of all the methods in a class [16]. Note that Chidamber and Kemerer [16] did not define a predefined complexity metric to consider for the computation of WMC. In our case, we opted for the McCabe metric to account for the individual complexity of methods.	Production Class
Code Smells		
Class Data Should Be Private	When a class exposes its attributes, violating the information hiding principle.	Production Class
Complex Class	When a class has a high cyclomatic complexity.	Production Class
Functional Decomposition	When in a class inheritance and polymorphism are poorly used.	Production Class
God Class	When a class has huge dimension and implementing different responsibilities.	Production Class
Spaghetti Code	When a class has no structure and declares long method without parameters.	Production Class
Test Smells		
Assertion Density	Percentage of assertion statements in the test code.	Test Class
Assertion Roulette	When a test method has multiple non-documented assertions.	Test Class
Conditional Test Logic	Conditional code within a test method negatively impacts the ease of comprehension by developers.	Test Class
Eager Test	When a test method invokes several methods of the production object.	Test Class
Fire and Forget	A test that is at risk of exiting prematurely because it does not properly wait for the results of external calls.	Test Class
Mystery Guest	When a test method utilizes external resources (e.g. files, database, etc.).	Test Class
Resource Optimism	When a test method makes an optimistic assumption that the external resource (e.g., File), utilized by the test method, exists.	Test Class
Sensitive Equal.	When the toString method is used within a test method.	Test Class

Figura 1: Metriche

Tuttavia anche durante questa fase si perdono alcuni test flaky, esattamente 1554, in quanto il tool non riesce ad ottenere la loro classe di produzione oppure non identifica la classe come una classe di test. **Statistiche**
I vari dataset creati sono stati uniti in un singolo dataset generale con 1972 test flaky e 320209 test non flaky:

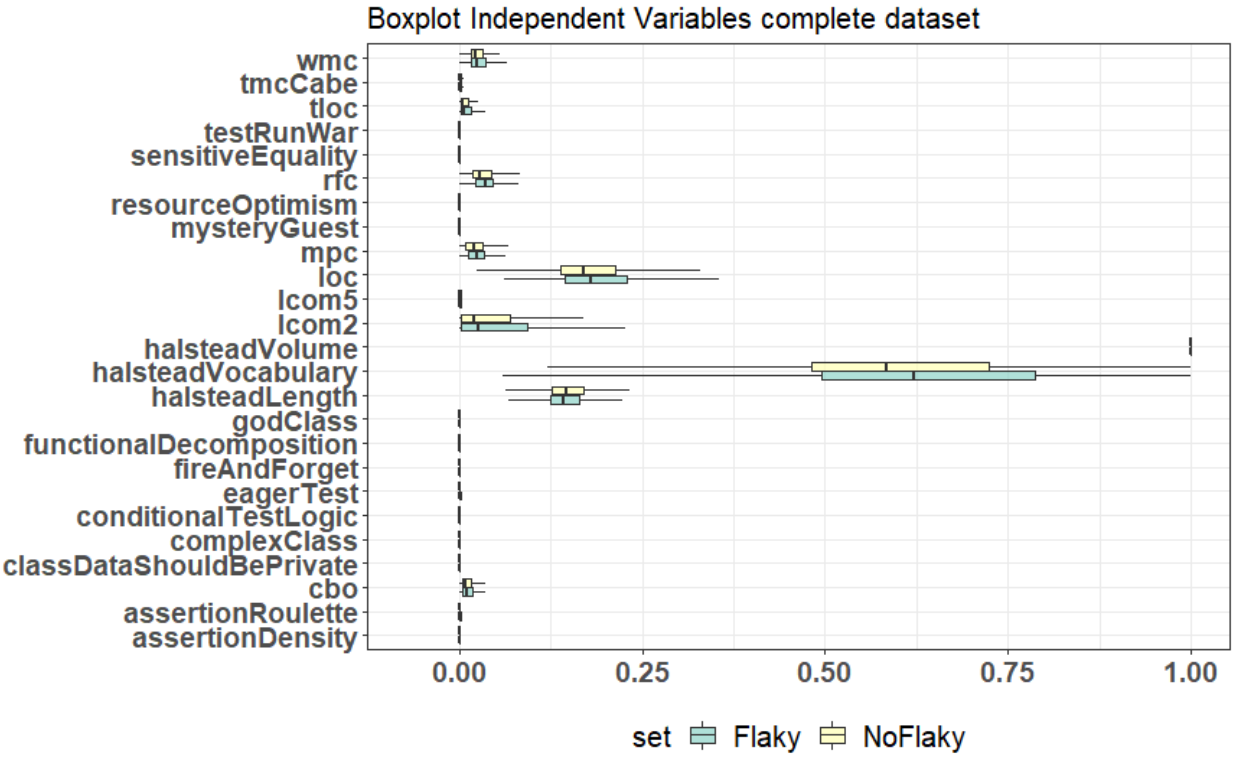

Figura 2: Dataset Completo Distribuzione Valori

Essendo il dataset generale fortemente sbilanciato, si è deciso di crare un ulteriore dataset soltanto utilizzado i seguenti criteri:

- Le repository per cui si perdono tutti i test flaky verrano escluse
- Per le repository con test flaky sparsi su più commit, sara utilizzato solo il commit con più test flaky
- I test di setup e teardown saranno esclusi

Tale dataset è composto solamente da 209 repository e possiede 223129 test non flaky e 1697 test flaky:

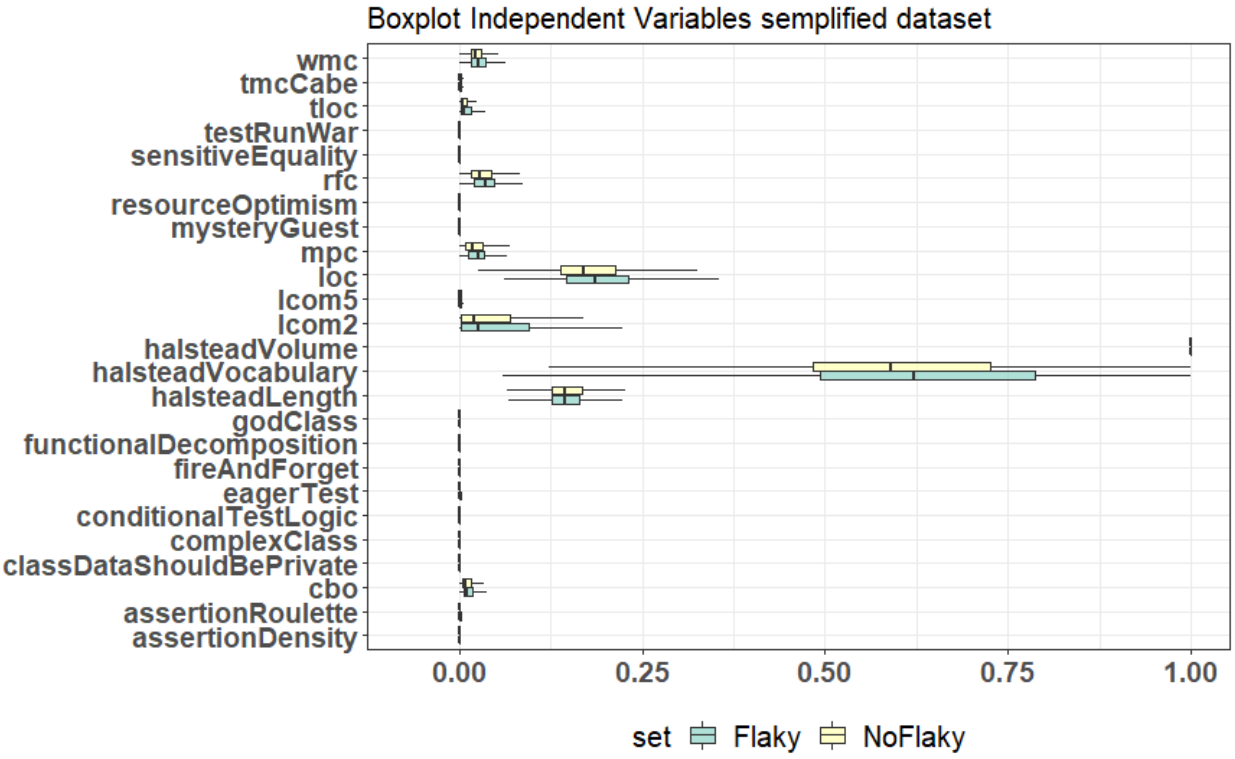

Figura 3: Dataset Semplificato Distribuzione Valori

2 Identificazione Machine Learning Pipeline

3 Conclusioni