# Premessa

Se ho capito bene, un potenziale obiettivo del mio lavoro sarà quello di identificare quali sono le feature più interessanti per determinare la **code coverage** di un input consistente in source code + test. Non dico solo test perché come faccio a capire la bontà di un test senza avere a disposizione il codice da testare?

Nel dataset ho trovato due feature interessanti: **numCoveredLines** e **projectSourceLinesCovered.** La mia interpretazione di tali feature è la seguente: dato uno specifico test ad esempio “ch.qos.logback.classic.asyncappendertest.eventwaspreparedfordeferredprocessing” con id “9879”, solo considerando questo test abbiamo ottenuto un numero di **numCoveredLines** pari a: 17.0 mentre per quanto riguarda le **projectSourceLinesCovered** abbiamo: 46. Cosa cambia tra queste due features? Non sapendo né leggere né scrivere d’ora in poi farò riferimento a **numCoveredLines** dando per scontato che tali feature esprimano la medesima informazione

## Idea

Andando a partizionare il dataset iniziale in df1, df2.. (uno per progetto ad esempio df1 = logback) sommando **numCoveredLines** in tutte le entity di df1 avremo il numero totale di tutte le linee di codice raggiunte dai test. Mi chiedo se tali linee di codice siano uniche o ripetute: esistono due test che raggiungono le medesime linee di codice? Inoltre mi chiedo se fosse disponibile da qualche parte il numero totale di linee di codice appartenente ad ogni progetto

Il mio intento è quello di riuscire a determinare il rapporto tra LOC (lines of code) testate e LOC totali; se il mio obiettivo è quello di andare a predirre la code coverage associata ad un test case input, ritengo che tale informazione potrebbe essere interessante per lo sviluppo di un eventuale modello

## Analisi su **numCoveredLines**

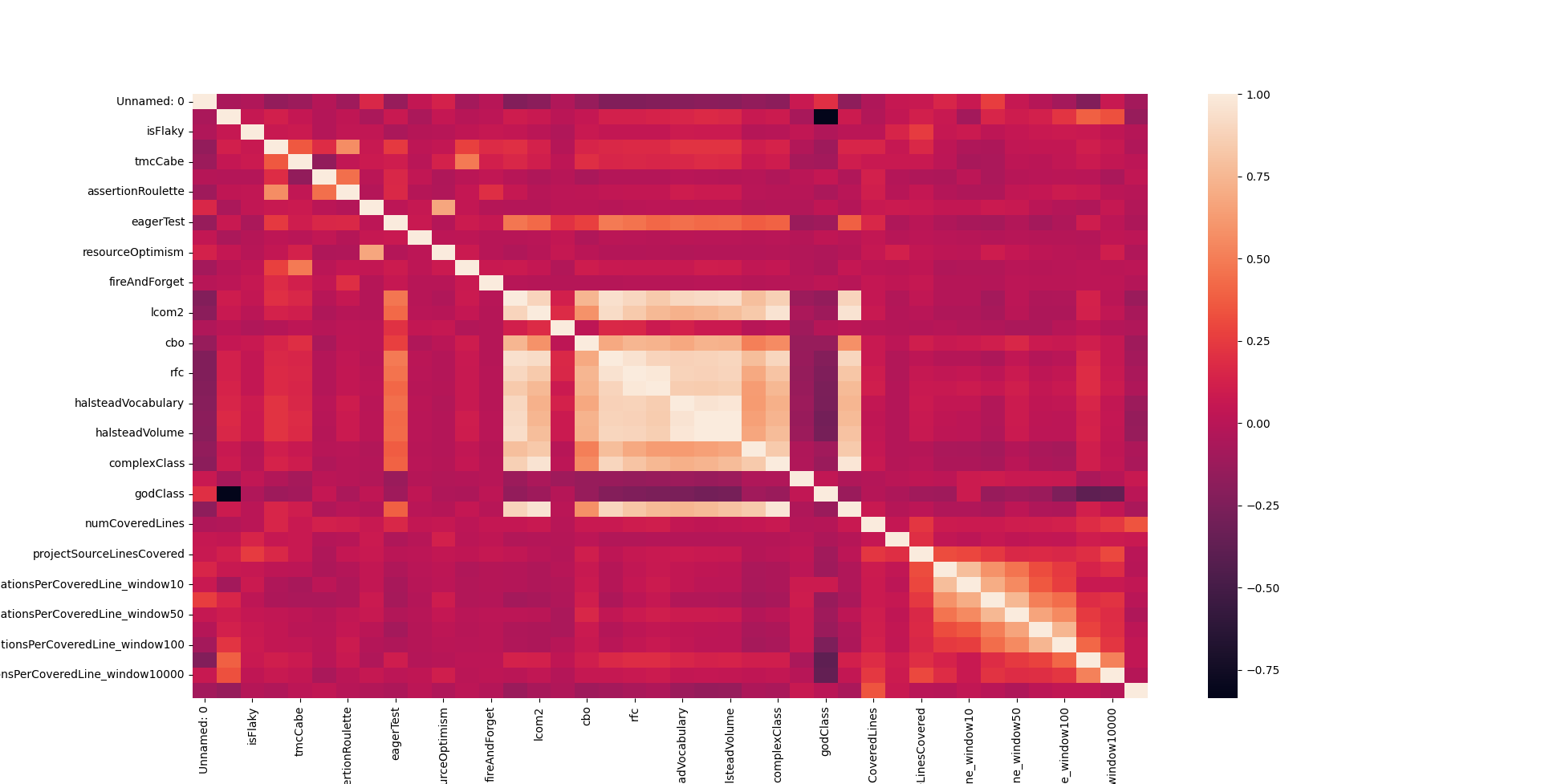
Calcolo il valore medio M della feature nel df, partiziono il df in due parti

underDF avrà i test case dal **numCoveredLines < M**

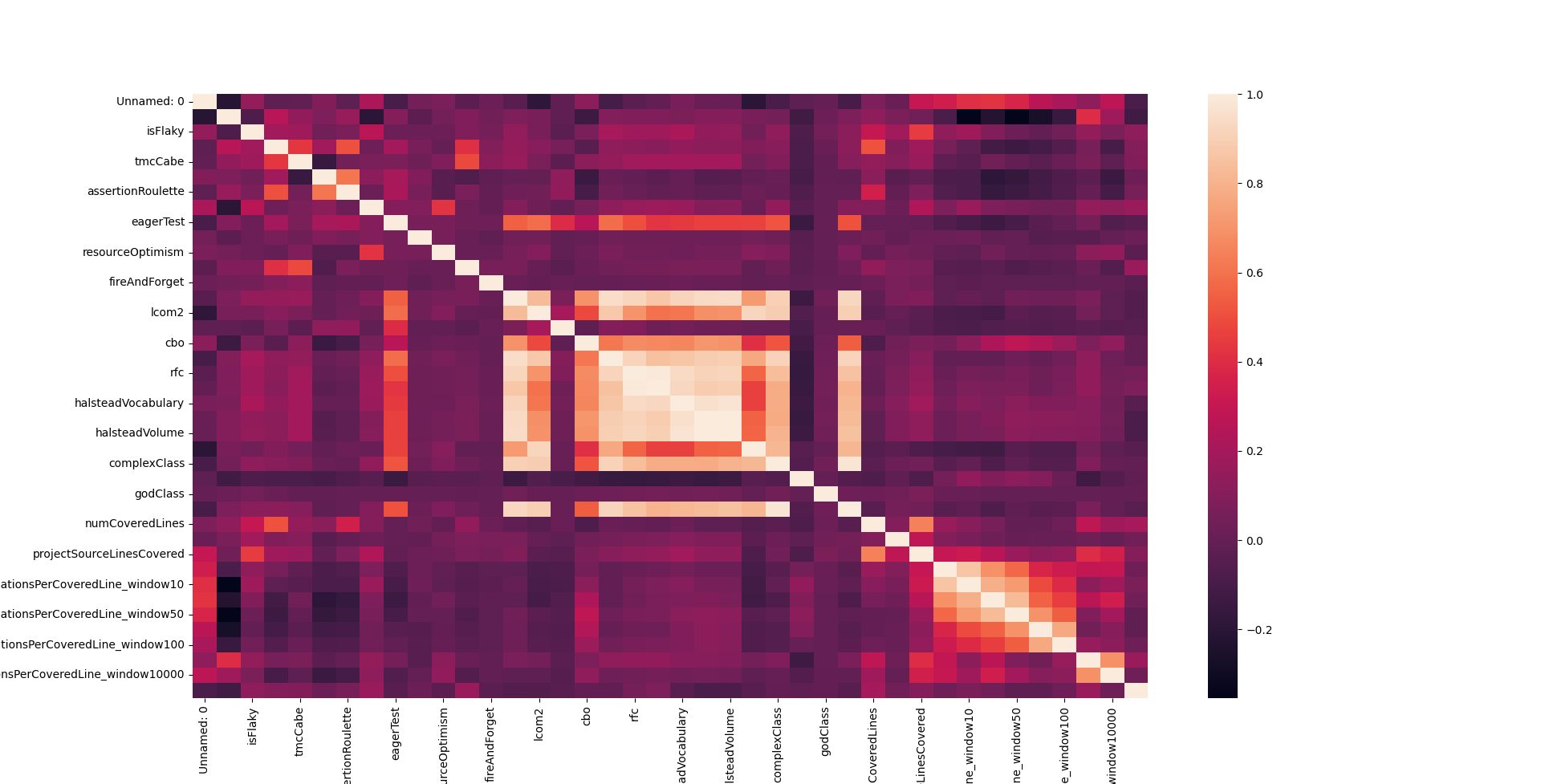
overDF avrà i test case dal **numCoveredLines > M**

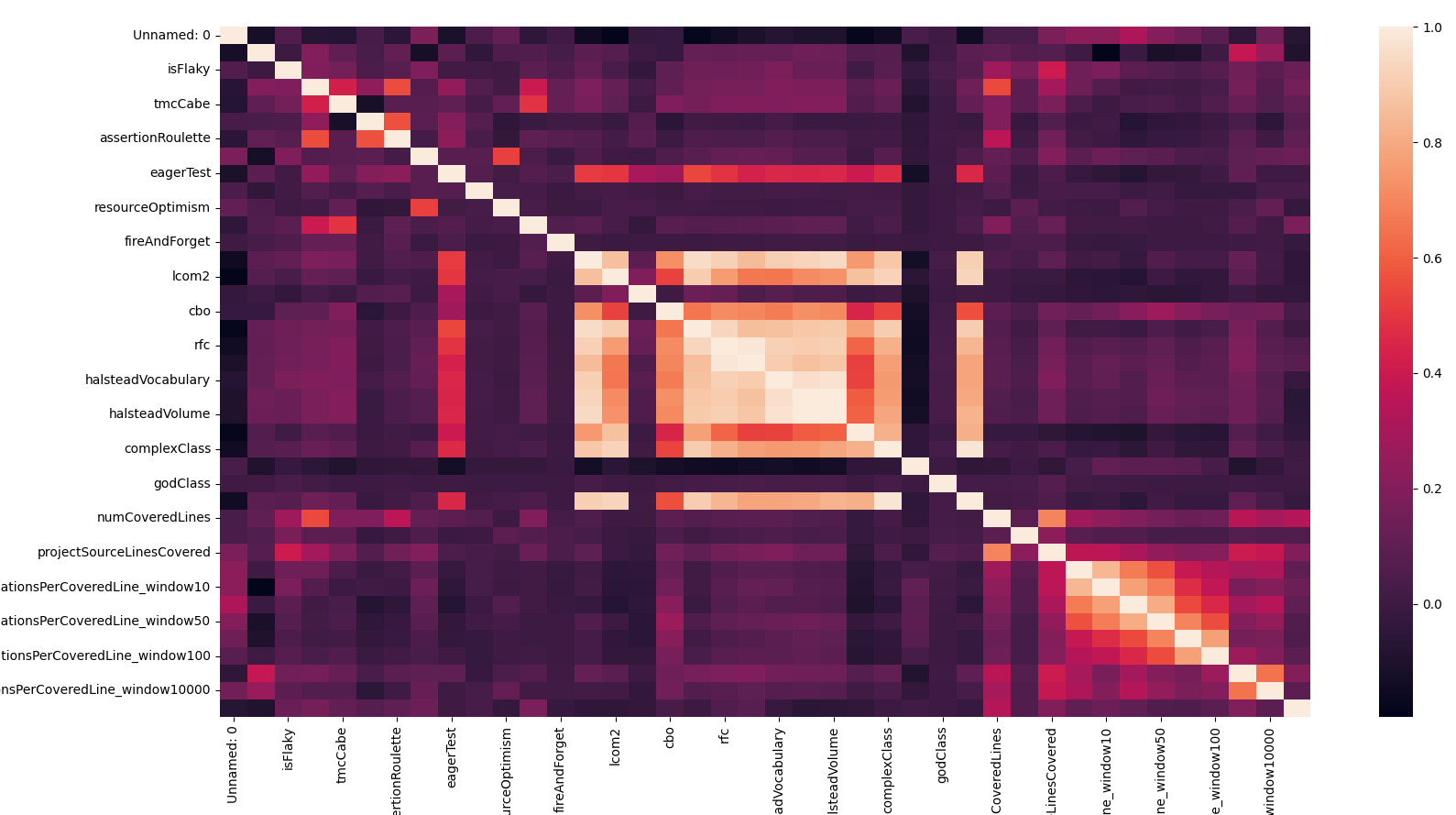
l’idea è trovare elementi NON in comune tra underDF e overDF, ottimisticamente parlando se troviamo una correlazione in overDF non presente in underDF, tale elemento potrebbe essere un indizio sulla bontà nella coverage di un ipotetico test.

Pearson su underDF (under.png su github)



Pearson su overDF



Analisi esplorativa su tutto il dataset

## Cos’ho notato

* La mia ipotesi **numCoveredLines = projectSourceLinesCovered** è errata siccome si le due variabili abbiano una buona correlazione, quest’ultima non è neanche troppo elevata pertanto tali feature esprimono informazioni diverse ma non so su cosa differiscono
* Esistono due ‘gruppi’ principali: una più marcata centrale ed una meno marcata nell’angolo inferiore a destra. Ciò implica una forte correlazione tra il seguente gruppo di features: **loc, lcom2, cbo, wmc, rfc, mpc, halsteadVocabulary, halsteadLength, halsteadVolume, classDataShouldBePrivate, complexClass, spaghetti code**

apparentemente **lcom5** resta esclusa dalla ‘cerchia’, sembra che tale features abbia uno sviluppo indipendente rispetto alle altre

da come ho capito esiste un algoritmo sviluppato dal SESALAB che preso in input uno snippet di codice riesce ad analizzarne alcune caratteristiche (le feature di sopra), esiste un glossario che definisce nel dettaglio l’entità di tali features?

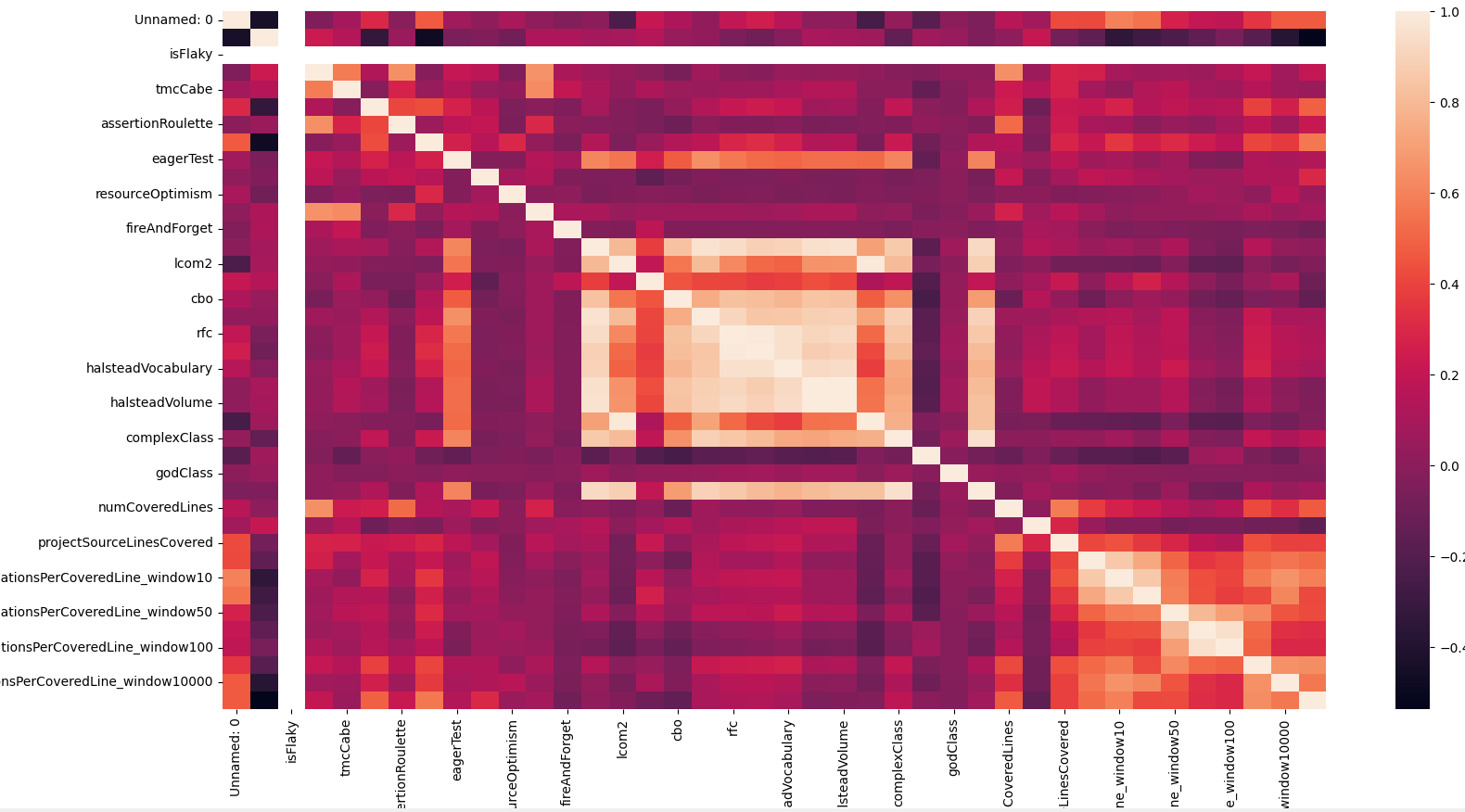
* Il secondo gruppo ha il seguente comportamento

l’informazione descritta è hIndexModificationsPerCoveredLine\_window (d’ora in poi HModX dove X è un numero tra 5 e 10000)

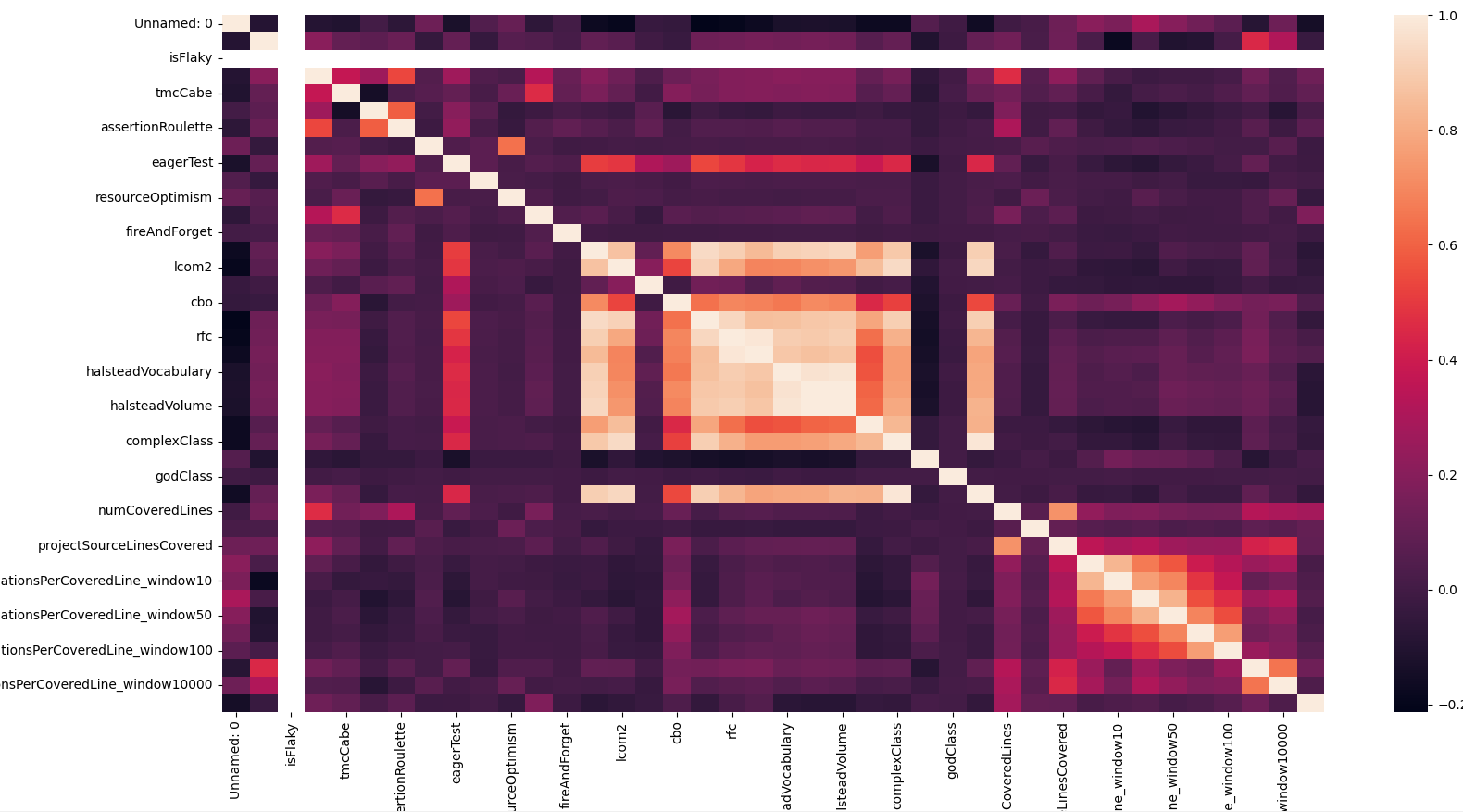
sono ignaro su cosa si faccia riferimento però ho notato che la correlazione tra le seguenti features è fortemente influenzata dalla vicinanza di X, ad esempio HMod5 ha l’apice nella similiarità rispetto a HMod10, tale similarità va a decrescere all’aumentare di X fino a diventare quasi estranea rispetto a HMod10000. Tale comportamento vale per tutte le feature appartenenti a questa categoria HModX, ad esempio HMod10000 è molto vicino (e solo) a HMod500.

* Ho partizionato il dataset in istanze flaky test = true e flaky test = false e riapplicato il pearson:

Correlation test sui flaky test



Correlation test sui test non flaky



Le correlazioni tra le feature sono più marcate nelle istanze flaky test = true