



METRICS THEME

REPORTING WORKSHOP DECEMBER 2019

We set off to...

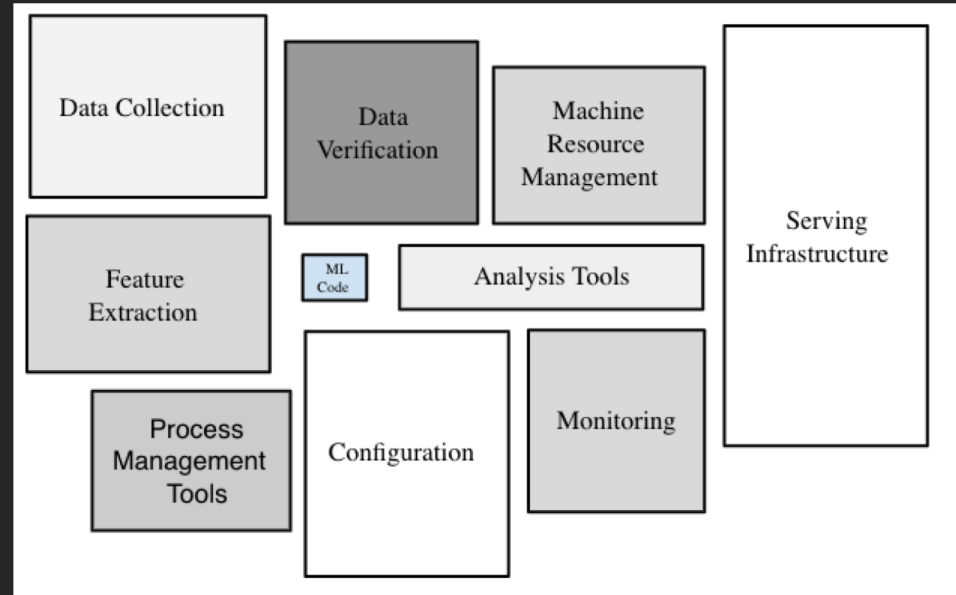
- Optimize test selection processes using machine learning
- Reduce effort in manual reviews
- Design measurement team assessment method

Action Research in Software Engineering

Theory and Applications

Machine learning / AI is just a small part of the whole pipeline

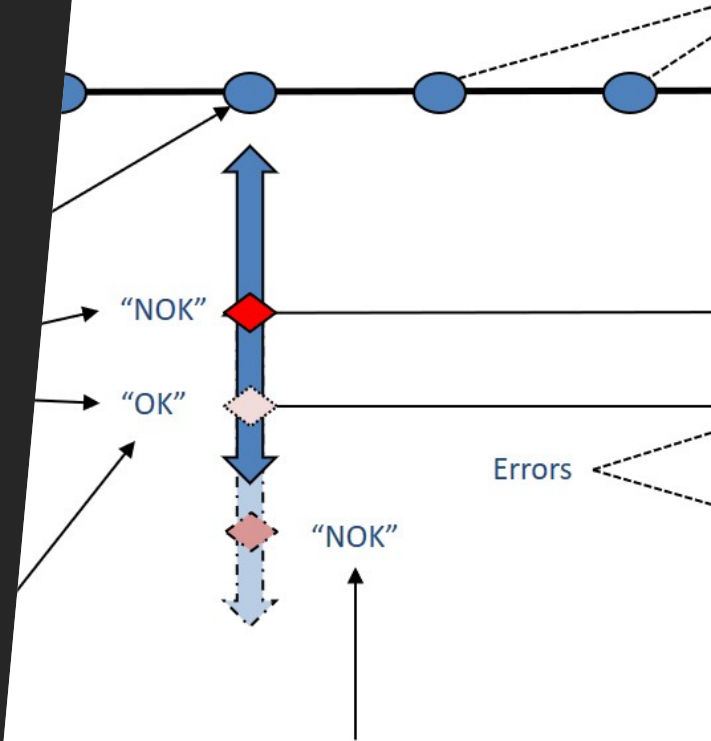
- Production ML systems needed for software engineering are still away
 - Lack of high quality, labelled data
 - Limited analysis capabilities due to non-obfuscated data sets
 - Non-standardized feature extraction
 - Manual configuration of data workflows



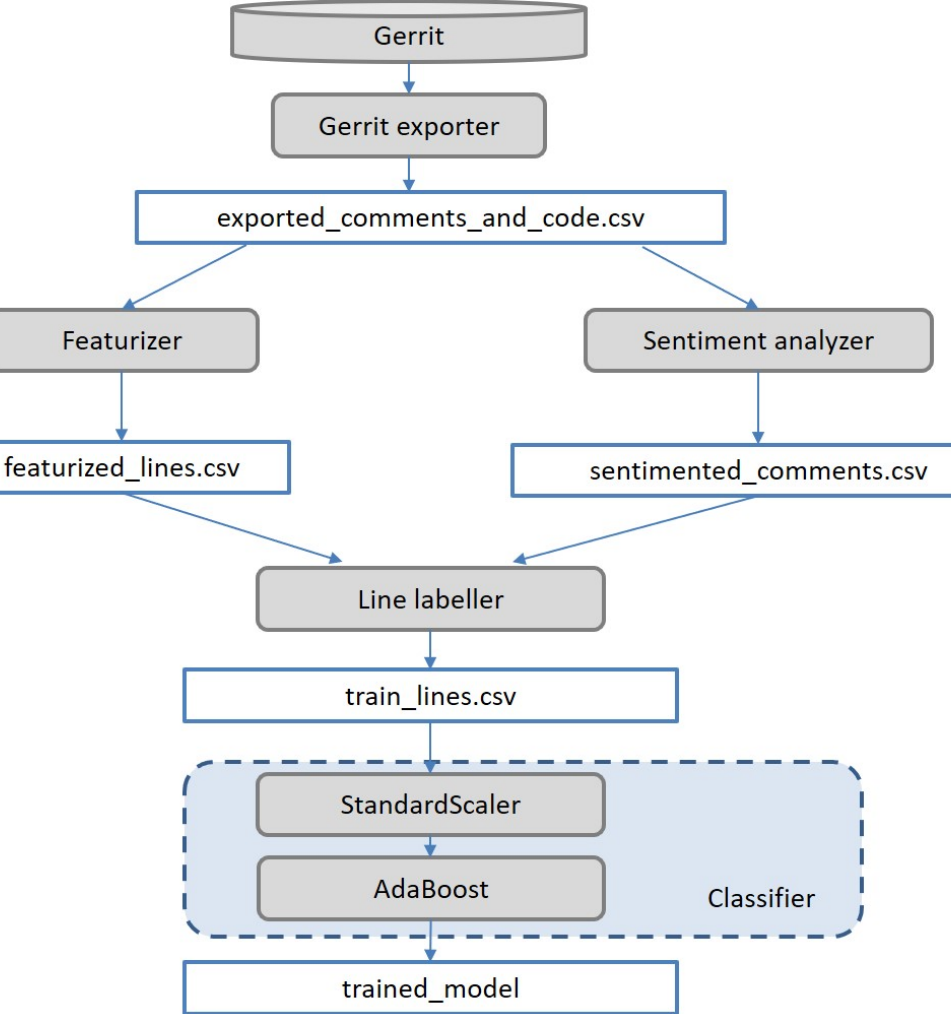
Source: <https://developers.google.com/machine-learning/crash-course/production-ml-systems>

Optimizing test selection

- New data curation algorithm
- Back-to-back evaluation with the CI tests
- New company evaluation started

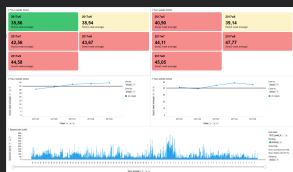


2. Expand the test suite with the tests that are predicted to fail



Reduce effort in manual reviews

- Exporting JSON data from code review repositories
- Using own feature extraction mechanism to characterize the lines
- Developing a project specific model for code quality



Measuring review speed

Speed measurement (duration)

Review speed measurement

Review data extraction

Rule definition

Code analysis

Feedback

Semantical

Multiline context

Uniline context

Design

Automated violation analysis

Automated violation extraction

Automated recommendation

Automated code reviews

Excellence levels

Technology support

Git extraction

Rule taxonomy

Gerrit extraction

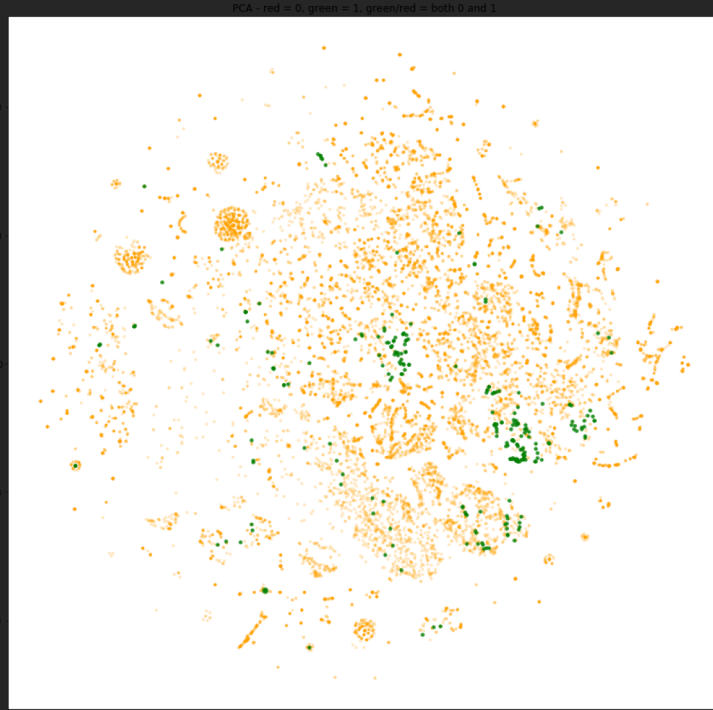
TBFeaturizer

Neural networks

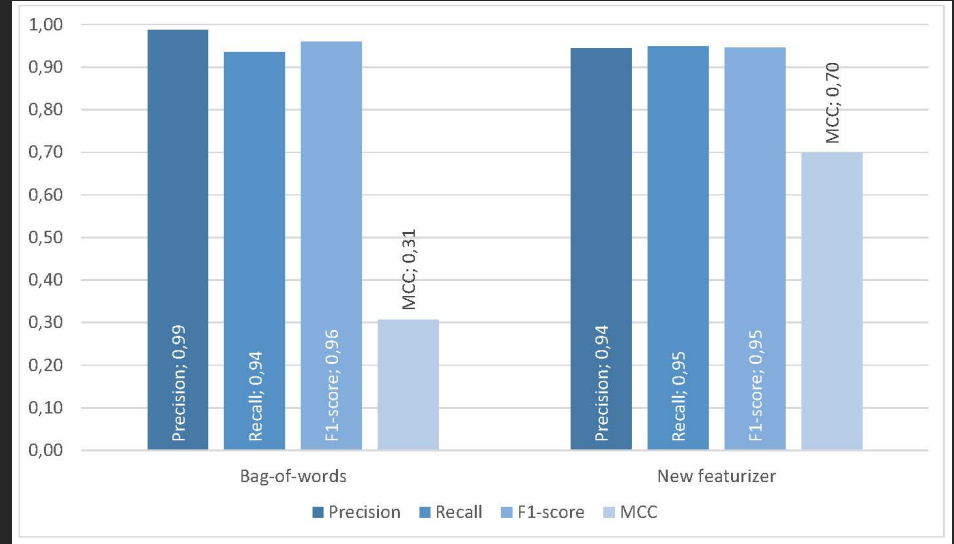
Clustering

Repair suggestions

Results from the code analysis



Classification based on sentiment analysis



Performance of classification algorithms

Code review: combining manual and automated reviews

- Training for one rule only
 - Gives control over results (easy to check if the classifiers are right/wrong)
 - Easy to understand if the results are correct/incorrect
 - Requires 1-7 iterations to get to the right set of example lines (our case, ca 500 LOC)
- Training from one commit – all files in one commit/pull request
 - Easier to export
 - Harder to understand what the reviewer means by a comment
 - Sometimes the commented lines do not match the comment → sometimes impossible to match
 - Takes more time to create the training file
 - Difficult to see whether the result is correct → needs to be taken as a recommendation
- Training from one file – all comments in all commits/pull requests
 - Easier to control what is exported
 - Comments need to be filtered, as they are sometimes about the semantics, e.g. "Should you really use XYZ?"
 - Sometimes the commented lines do not match the comment → sometimes impossible to match
 - Difficult to see whether the result is correct → need to be taken as recommendation



UNIVERSITY OF
GOTHENBURG

VISIT OUR WEBSITE
www.software-center.se