

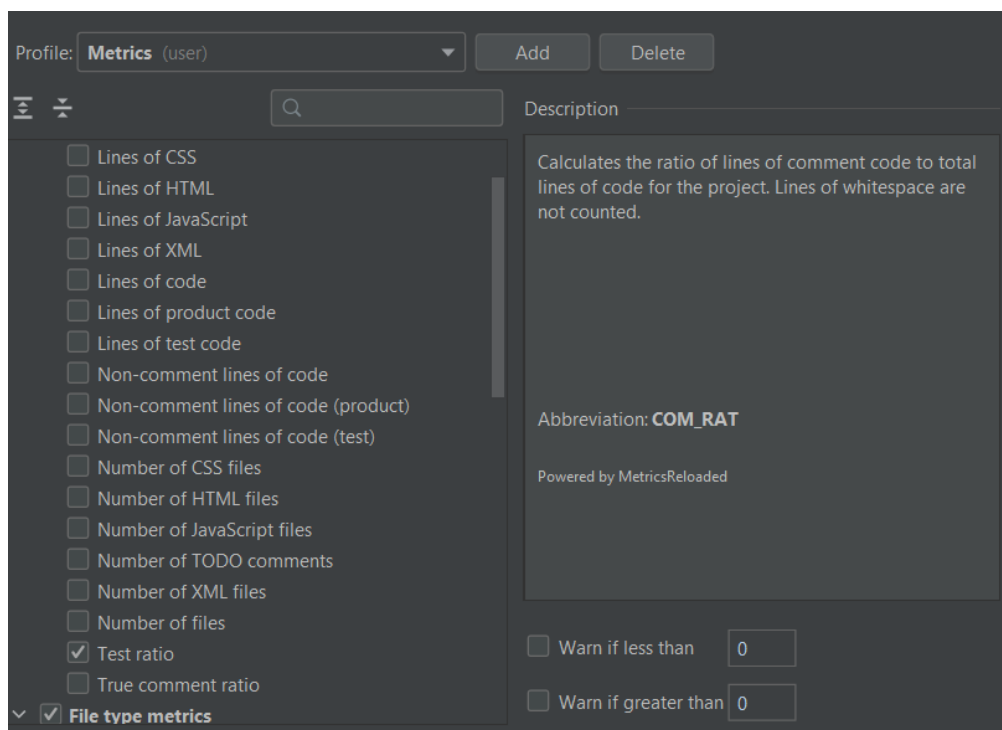
QA Report

Project used:

My bachelor's thesis application, which is a small cross-platform application written in python, can observe a subject's facial features from a continuous video-stream and detect its early signs of drowsiness. The targeted users are mainly drivers, the end goal is eliminating the risk of falling asleep while driving, therefore reducing the number of car crashes.

Software metrics tools used:

Since my project is written in python, my choice was **MetricsReloaded**, which is a plugin for all JetBrains IDEs, but after installing and running it for the first time, I understood that this was not what I needed. Apparently, the metrics that I was interested in were having serious issues in Pycharm, many people complaining about the lack of utility of the plugin. The image below describes what MetricsReloaded can offer:



two predefined profiles and the possibility of creating a third custom one with a wide range of properties, most of them being related to the number of lines of some type or percentages of comments, etc.

For some real results, I decided to go with **Multimetric**, which is a tool used for a wide range of programming languages, and it calculates metrics such as *comment to code percentage*, *cyclomatic complexity according to McCabe*, *Difficulty according to Halstead*, *Effort according to Halstead*, *Fan-Out*, *Lines of code*, *Maintainability index*, *Metric according to Pylint*, *Metric according to TIOBE*, *Number of delivered bugs according to Halstead*, *Time required to program according to Halstead*, *Volume according to Halstead*.

The output that I received from this tool is consistent, yet not too detailed, since it doesn't point out the specific places where improvements could be made. Also, it is not so user-friendly, because of its way of displaying the output in console or .json file, making it hard to navigate. Below you can find a significant sample of the output:

```
"overall": {
  "comment_ratio": 24.77916194790487,
  "cyclomatic_complexity": 14,
  "fanout_external": 11,
  "fanout_internal": 1,
  "halstead_bugprop": 3.5723852727715704,
  "halstead_difficulty": 48.729559748427675,
  "halstead_effort": 522242.284781776,
  "halstead_timerequired": 29013.460265654223,
  "halstead_volume": 10717.155818314712,
  "lang": ["Python"],
  "loc": 305,
  "maintainability_index": 26.85702358233307,
  ...
  "tiobe": 84.86937283570603,
  "tiobe_compiler": 100.0,
  "tiobe_complexity": 2.4624855713736054,
  "tiobe_coverage": 100.0,
  "tiobe_fanout": 90.0,
}
```