Assignment 1, WASP Software Engineering Course
David Hasselquist
2022-06-30

Reflection on pynblint

What did you learn, in your individual session, about static analysis for ML and the pynblint tool?

As I have not used notebooks much previously, I learned more about notebooks in general, their purpose, as well as their strengths and weaknesses. By looking at some example warnings/recommendations, I learned more about the general issues with notebooks, such as cells executed in a non-linear order. We did not discuss the ML aspects during the session.

Will pynblint be useful to you in your WASP PhD project? Why or why not?

As I am currently not using notebooks in my research, this tool is currently not very useful. However, in the future, if I would start using notebooks more, then I think that the pynblint tool would be very useful.

Ideas for how the tool could be improved?

Luigi and I had interesting discussions about pynblint and its limitations. Some of the discussions include:

1. Pointers and recommendations to Git LFS for large data files.

2. Integration into jupyter notebook and/or visual studio code (for visual studio code, could perhaps work in a similar way as black for python).

3. More linting rules. However, one must be careful when adding more linting rules as one may overwhelm the user. Therefore, we also discussed the possibility of having different linting levels or severity, with the user being able to customize which severity of the linting rules to use.

What do you see as the limits for static analysis tools in ML? For code, models, and for data?

In general, static analysis tools do not analyze the code when being runned, and might miss some complex aspects. This might apply especially in the ML context, where it can be difficult to predict or approximate the behavior by just statically analyzing the code.