Software Engineering and Cloud Computing
Reflection on `pynblint`

Julian Alfredo Mendez
Umeå University

2022

In this reflective report, I first introduce some background about my research and opinions. In my research, I propose techniques and tools to formalize ethical requirements for artificial intelligence systems. I am an advocate of very clear and well tested source code, not only to reduce costs and improve efficiency, but also for the ethical implications that it has regarding the human-computer interaction.

Connected to that, I have observed how Jupyter Notebook is a powerful tool with high acceptance in the machine learning community. On the one hand, its interface and the possibility of prototyping makes it a valuable tool when finding quick results. On the other hand, this flexibility hinders very relevant features like stability, durability, reliability, and reproduceability. This is why I support the use of tools like `pynblint` which help to close the gap between the two different styles.

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

In the individual session I learned about the metrics that `pynblint` has. We used the tool in Jupyter Notebook files that were written by me. The tool helped me to learn some of my strengths and my weaknesses when writing Jupyter Notebook code. I also noticed how some online services that provide remote execution of Jupyter Notebook code can have some limitations that worsen the code quality.

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

At the moment I am not including Jupyter Notebook code as part of my research, and therefore I am not planning to use `pynblint`. Nevertheless, this tool provides a good running example of how to use static analysis to deal with code that needs to be improved.

Ideas for how the tool could be improved?

The tool itself is already very powerful. I believe that if the tool could generate a machine-readable report, for example a JSON file, it would be very useful, especially to integrate it to other tools.

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

Static analysis tools are especially useful to detect and prevent possible problems in the source code. Using static analysis could be less effective when it is applied to models or data, but it could be used on customized tools that test models and data.