My PhD project is about leveraging cloud computing to improve performance of control systems. I have practically never used notebooks and barely ever used Python.

- The main thing I learned was the problem of version control on large data sets (e.g. DVC). The session was mostly a demonstration since I don't use the tools, it therefore was like a repetition of the lecture.

- I will most likely not use pynblint for my research, mainly since I do not use Python and probably will not start using notebooks. For interactive use I prefer normal terminals or specific tools (such as a dedicated web interface for cloud setup).

- The most obvious improvement for me would be adding support for more languages, although that might be more suitable as a separate tool. Certain changes could be applied automatically, such as adding missing markdown sections with an appropriate template.

- For "glue code" (which from my limited experience is the use for Python in ML), static analysis should be more complex than for lower level languages such as Rust (known for compiler warnings and errors) or Ada (with the Spark subset formally verifying behaviour) due to all libraries being used compared to basic components of the language.

  For models I imagine one of the main limits, as was mentioned in the lecture, is the understanding of how models work. How can a static analysis be performed of something not fully understood? There is also the problem of multiple ML frameworks working differently, is it possible to make a modular analyser combining e.g. three cooperating analysers made specifically for Python, notebooks and a model framework?

  For data I guess accurately identifying outliers not caused by errors limits the usefulness.