1. **What did you learn, in your individual session, about static analysis for ML and the pynblint tool?**
   I was not very aware about the space of static analysis at all and I learned during the lectures more about its usefulness and limitations. In my individual session I learned in practice how it can be used.

2. **Will pynblint be useful to you in your WASP PhD project? Why or why not?**
   I don't think so. I barely use any notebooks since most of my compute is done on high performance clusters. I do think it could be useful in industry or similar where consistent usage of notebooks is important and can be easily achieved by adding a tool like pynblint to the git pipeline.

3. **Ideas for how the tool could be improved?**
   One problem that is very common is the tensor dimension mismatching problem. It would be very useful to be able to define the expected dimensions of the input and the output at different stages of a machine learning training procedure and be able to statically check whether it is correct with respect to a machine learning model in for example pytorch or similar.

4. **What do you see as the limits for static analysis tools in ML? For code, models, and for data?**
   The code at one part of the program may depend on the computed output from another step of the program. Thus, to statically analyze whether it will work or not, you need to perform the computation first which is not feasible in static analysis. Same may be true for models. A model may have been trained on data from some distribution, and you would maybe like to be able to check whether the data during inference is similar or out of distribution. This can not be done statically I believe, but you have to run all data through some computation.