Reflection on pynblint

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

   I had no idea that this kind of tools exist. I learnt that pynblint is important tool to make your notebook more readable, comprehensive and share worthy. It helps to achieve real purpose of creating a notebook.

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

   Yes, I think it can be useful in my research. Overall aim of project is data driven decision making. I need to make efficient decisions based on historical data. However, historical data does not contain counterfactual outcomes (outcome against the unconsidered decisions for that particular entity) which are essential for accurate Optimization Problem solution, inference of counterfactuals from historical data is a daunting task due to presence of confoundedness and selection bias. My research is more towards making data ready to take optimal decisions, after that I plan to use traditional machine learning for counterfactual outcomes. Currently, pynblint can play a handy role to make my notebooks more collaborative, reproducible, trustworthy and publication ready. Moreover, if it adds features of data linting (representation, scaling, and outlier detection of features; cleaning of data by handling missing and duplicate data) and code linting (suggestion about best model for my data and optimal code) in future then it would become super useful for my research.

3. Ideas for how the tool could be improved.

   At the moment, pynblint only analyses the style of coding (more towards the format of code) but it does not say much about coding itself. For instance, It does not rectify the code itself (like duplication of code, dead code etc.). If it executes the code and gives recommendation about model and specially about possible manipulation about data. Addition of these features in future then it can become more useful.

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

   These does not execute the code and therefore it does not know about the code, data, or models. That is what I think is the limitation of static analysis