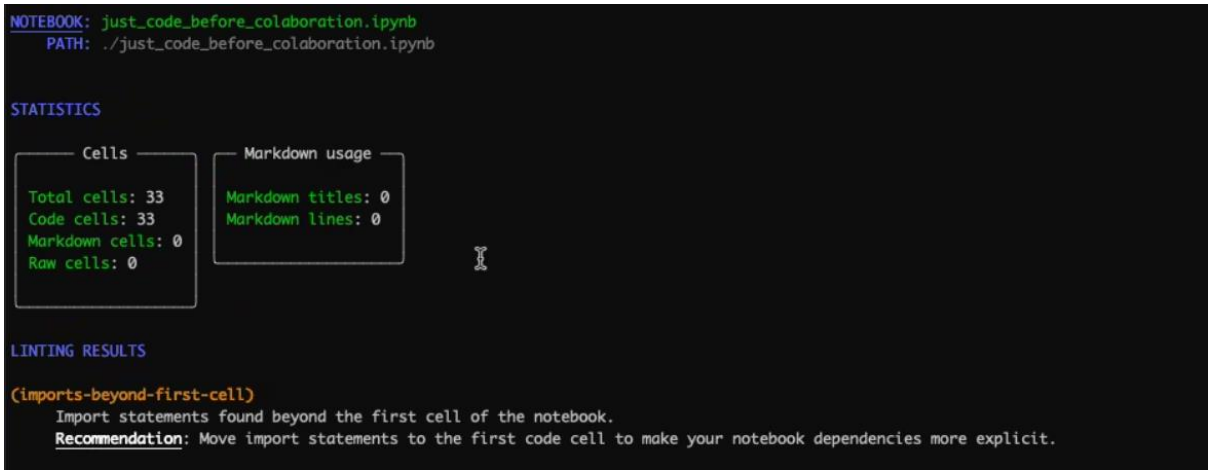
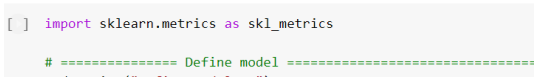
1. What did you learn, in your individual session, about static analysis for ML and the pynblint tool?

We checked my code (my sandbox for one of my WASP courses) for ECG classification. Then we discussed the output of the analysis.

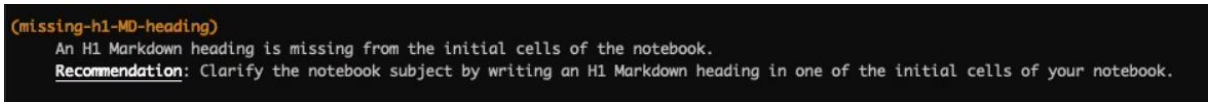
We got the following comments which I consider in my future coding in Jupiter notebook:



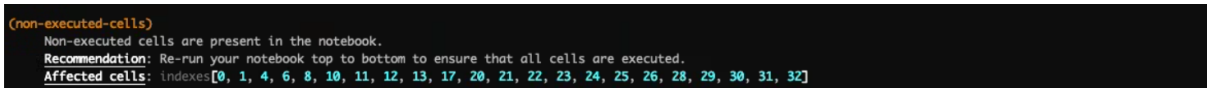
As it is raised by Pynblint there were some imports that exists in the middle of the code which were detected like the following:



Since the code was my sandbox there was no text to explain the code, but in the final delivery of the code, I will add them for my professor to explain the code.

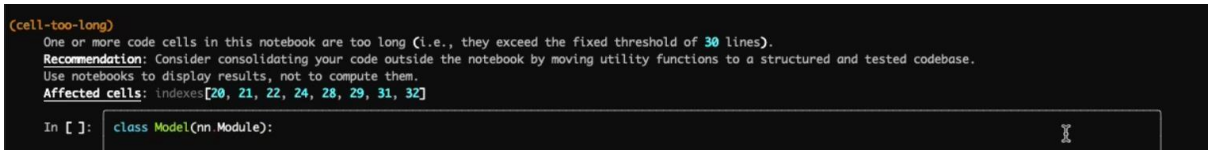


Since the code was exported from Colab then there was no sequence of execution in the export so the following was detected as an issue:



Some empty cells were detected which will be fixed in the final version of the code:

Some cells were longer than the recommended lines, which I can't change since the skeleton of the code is defined by the homework, but I will consider it in my future coding tasks:



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

Pynblint would be useful to remind us of the coding guidelines, especially in the Jupiter notebook which is quite new in comparison with other environments. If the code will be used in a collaboration with others, Pynblint would be useful to remind the team members to use the same style and do well-documented coding to ease the collaboration.

My Ph.D. is about the safety of Autonomous Driving functions, and the safety of ML-based SW components. This is why a well-documented and bug-free SW which will be used for the development of such a system would be an absolute necessity. So we discussed the effect of using such a tool for safety-related applications. As Luigi mentioned since the Jupiter notebook will be only used as a sandbox and not for the final SW development which will be used in production, then the tool has no direct safety-related effect, but still, by providing feedback during the trial and error phase help the team to work more efficiently and resolve the confusions, by having a cleaner, and well-documented code.

3. Ideas for how the tool could be improved?

Integrating the tool in the Jupiter Notebook would be a useful feature to provide feedback online when doing the coding.

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

One limit that was mentioned, is the analysis of the code itself to detect the bugs in the code.