

# Progress Report 2

**1. Changes:** Main goal is to evaluate value delivered to a general python developer by AI models in production and the leading academic models. Reiterating: I'm focusing on next token prediction which is basically autocomplete integrated into IDE's.

**2. Research Questions:** *Formulate some explicit research questions that your final report will answer. Some of these may have already been answered in the progress of building and testing your system - is it possible to do xxx by combining yyy and zzz? Others may be answered by your experiments/studies - how does my system compare on the aaa, bbb and ccc metrics against the state of the art? or how do users rate my system on ddd, eee and fff Likert scales compared to the state of the practice? If you are doing any user studies, make sure to record the relevant demographics of each subject; if all your subjects are (or are expected to be) students or alumni of Columbia or your previous institution, family members or personal friends, say so.*

How do the academic models compare to the available products? Which are more accurate? Which are more helpful with developing thought and logic? Last Q if I can get some friends to participate: How do other python developers view these predictions in helping with thought/logic development?

**3. Challenges:** *Tell us about any challenges you have run into and how you overcame or worked around those challenges. If everything is going fine and you do not anticipate any difficulties finishing on time, say so. If you are floundering or your project is otherwise in trouble, say so. In that case, also contact the instructor and TA on piazza asap!*

Main Challenges: Did a quick pivot before the last checkpoint to make this more manageable. Main Issue I have right now... The academic models weren't made to be highly usable so preprocessing my new code to check on is a bit annoying and it is made to compute next token for every token in the file, so it takes some manual work to find the predictions. This should be fine but just time consuming. Getting all my evaluation code and getting the products to work was pretty simple (as it was designed to be). I'm not the greatest at organization, so the record keeping by hand aspect is a bit tedious but by no means a major issue.

My biggest issue right now is the fact that I can't really figure out how to get multiple predictions out of the model. It only predicts one token, but I'm going to spend some time with the engineering to see if I can get the top N so it's more easily comparable to the products. However, if I can't do that I'll be sitting on top 1 prediction comparison for all models, which may be fine but is less complete than I'd like.

**4. Demo:** *Plan a 5 minute demo for the class. This should describe what you expect to be able to show during the last week of classes (the last two class sessions). Do not plan a 'presentation', there won't be time, just a demo with maybe 1 minute elevator pitch.*

I think my demo is pretty simple and straightforward. Going to come prepared with some of the most interesting examples of predicted tokens and the source code it was based on. I will try to create a hypothesis for the performance and note some of the intriguing outputs.

**Is there anything else I need to include?** *This part might be copied from your first progress report, but make sure to highlight any changes.*

*Tell us how you plan to deliver the code, documentation and other software artifacts for your project; in most cases this should be a github repository. If your project involves data sets or models too large for github, those can be posted on zenodo. You should not submit any materials via courseworks other than your prose assignments.*

*Tell us about any open-source software you plan to extend, incorporate or reuse. Include the download URL(s).*

Plan to deliver the code on Github with my evaluation code inside that. Will link to the standard py150 set that was used for evaluating the academic models that I looked at to understand them.

Majority of my code will be based on the Benchmark implementation by CodeXGLUE by Microsoft (<https://github.com/microsoft/CodeXGLUE/tree/main/Code-Code/CodeCompletion-token>) which enables me to use the state-of-the-art CodeGPT module for prediction. Going to try to change it for the above issues.

Going to utilize products from Tabnine, Kite, and Google Colab as the best free, python enabled, autocomplete software available.