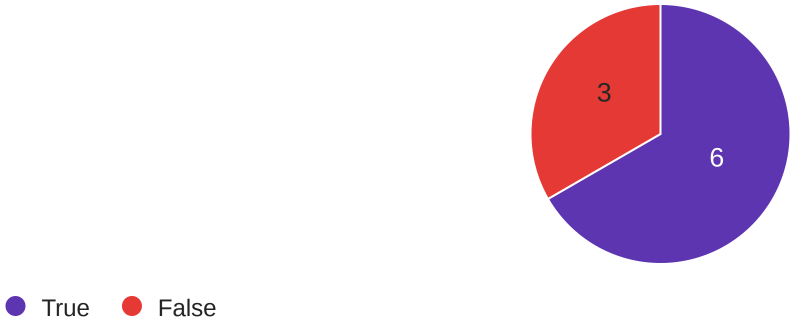
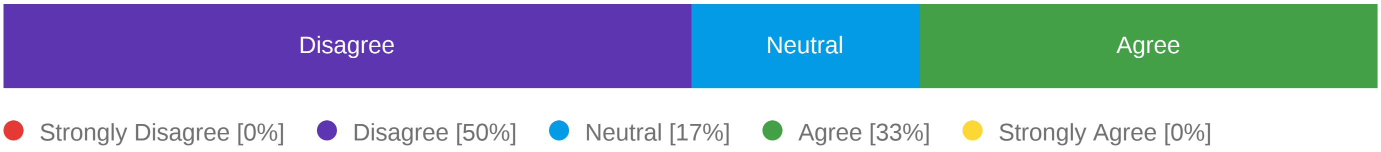
AST COMPLETE

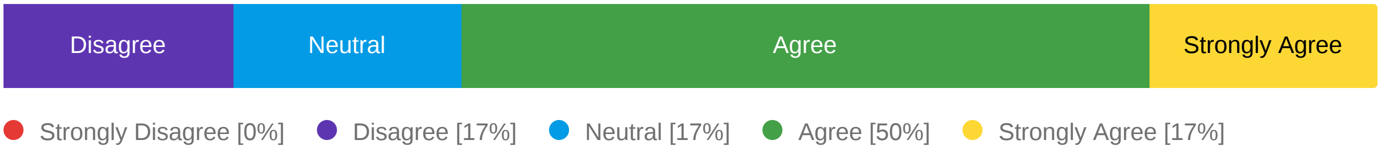


Visualization Usefulness

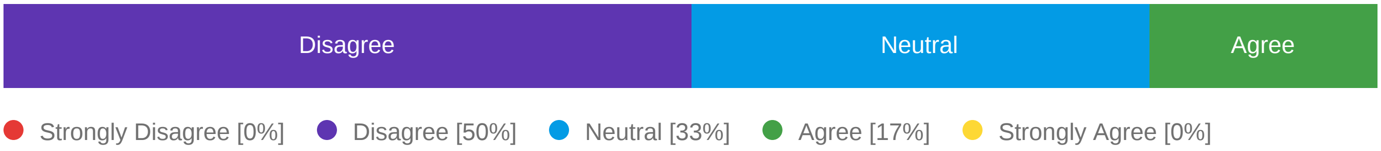
I think that I would like to use the visualization frequently.



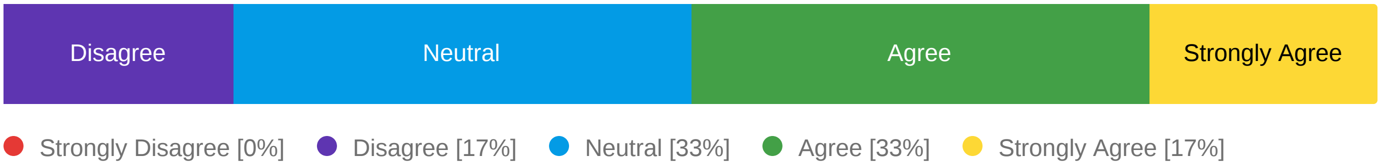
I found the visualization unnecessarily complex.



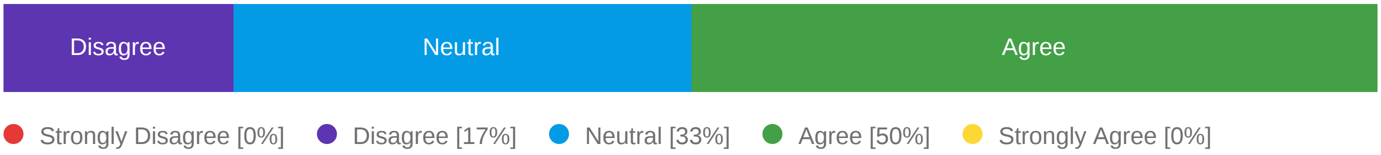
I thought the visualization was easy to read and use.



I thought the visualization was very cumbersome to read and use.

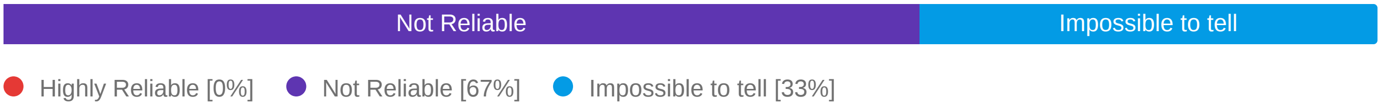


I thought the visualization was very useful for explaining the model's behavior.

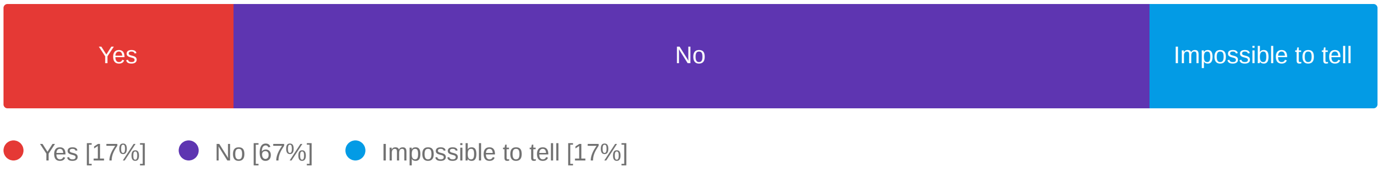


Model Performance

What is your perception of the reliability of the model in generating code?



Would you rely on the model to assist you in source code generation tasks?



Open Questions

What information from the visualization did you find useful in explaining the model's predictions?

8. What information from the visualization did you find useful in explaining the model's predictions?
The probability values. Also, the use of "error" to denote a token that produces an error in the code.
The given probabilities of error were useful
Visualization gives an idea of the general structure of the code. However, it might not always be necessary to comprehend the overall workflow of the code.

If with visualization you were referring to the AST, then the most useful was the ERROR in the first example, and the usage of different colors for different probabilities. It was also useful to have the previous code with any color in the AST.

But if with visualization you meant to the highlighted code, then that was useful to identify which code/errors were previously introduced and which ones were introduced by the model

The visualization was helpful as long as it was small; it could aid in understanding the code.

The probabilities were really helpful in determining model's behavior

What information from the visualization did you find useful in explaining the model's predictions?

9. What information from the visualization did you find useful in explaining the model's predictions?

The probability values. Also, the use of "error" to denote a token that produces an error in the code.

The given probabilities of error were useful

Visualization gives an idea of the general structure of the code. However, it might not always be necessary to comprehend the overall workflow of the code.

It is the same question as 8

The visualization was helpful as long as it was small; it could aid in understanding the code.

The probabilities were really helpful in determining model's behavior

What other information (if any) would you like to see in the visualization?

10. What other information (if any) would you like to see in the visualization?

Perhaps textual reasoning for the choice of a token. this may be difficult in practice though.

Information on how it was considering the input (the previous tokens), although this information may have been provided and I missed it, if so I'm sorry

Perhaps an example with a test case

A way to simplify the AST. With that I meant, a way to collapse branches that are not in my interest in a particular moment. It would be great to start with the complete tree, then once I made an idea of the general tree, it would be nice to be able to collapse branches. This way, I could focus only in the information that I need and not in the complete tree

I would like to see a higher-level visualization, not line by line. as an example,
1. the probability of having a loop VS method call after the incomplete code.
2. the probability of the condition being < 0 vs < = 0.

The probabilities here are very low for all the tokens, I would like to see other tokens that were candidates for incorrect predictions

What elements of the visualization did you like most?

11. What elements of the visualization did you like most?

the coloring was helpful in pinpointing the tokens with the least probability quickly. The error flag was useful in diagnosing bugs in the code, but this flag was not always present with tokens that produce errors.

The color coding was somewhat useful although it wasn't clear what the different colors meant (and it wasn't clear if colorblind users would be able to understand the color coding)

General structure

The differentiation between what was predicted and what was introduced previously. Also the colors (heatmap) helps with the probabilities.

color-coding

The predictions number right next to it

What elements of the visualization did you like least?

12. What elements of the visualization did you like least?

The complexity of the tree hierarchy. This made it difficult to locate certain parts of the code. Also, the visualization can not be read in a top-down approach similar to the code snippet.

The ASTs were a bit confusing to understanding in how child nodes followed from parents, but that's probably because I'm relatively new to ASTs.

Perhaps it is not necessary to mention different keywords such as identifiers, operators, etc.

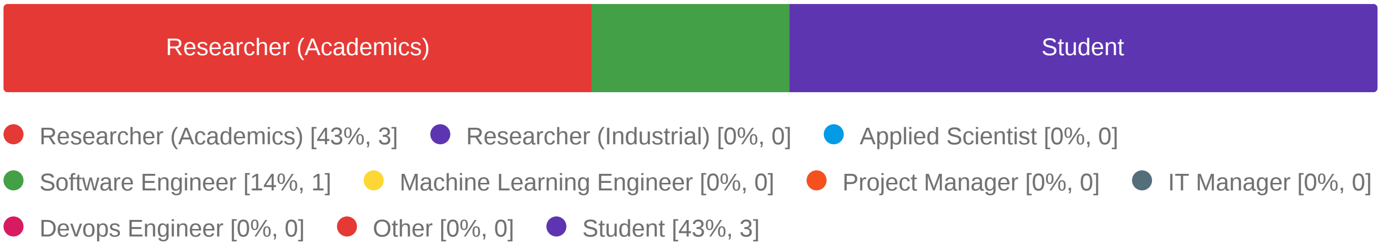
The amount of information presented as once.

the size of the big trees

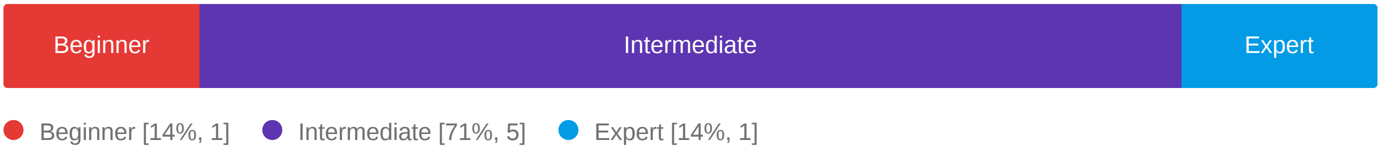
It is a little too much, the tree covers everything, perhaps focus on only the incorrect parts? or focus on weakness/highlight it maybe?

Demographics

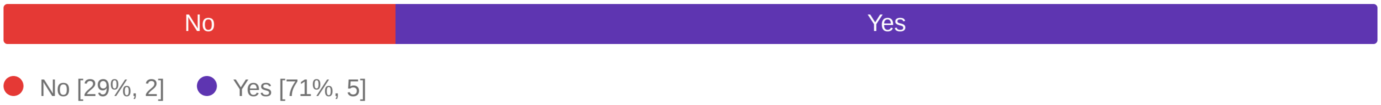
Q2 - Background



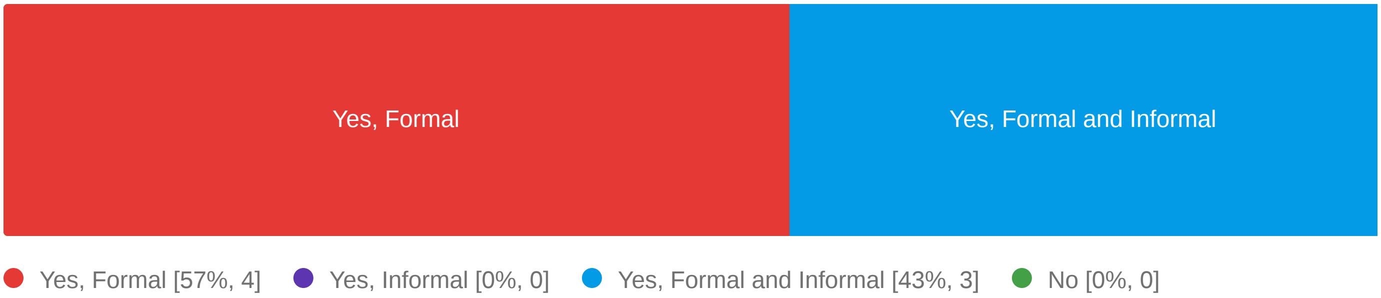
Q3 - 3. What is your level of expertise in python?



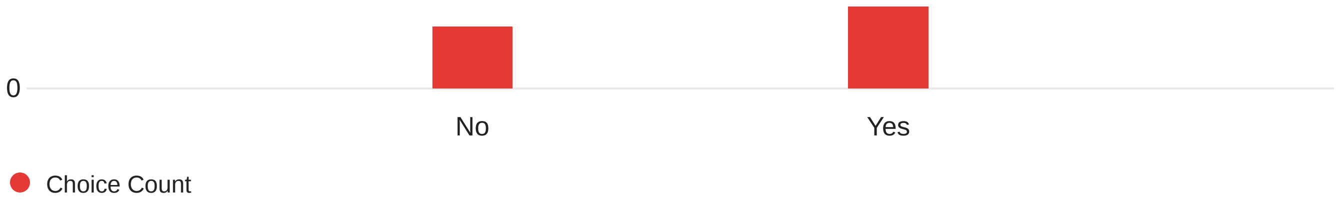
Q9 - 5. Are you familiar with Abstract Syntax Tree (AST) representations?



Q5 - 6. Do you have a background in machine learning? Formal (college classes, degree, certification) or informal (self-learning or other training)?



Q6 - 7. Have you used AI-assisted tools (e.g., Copilot or ChatGPT) or Large Language Models (e.g., CODEX or GPT) to aid in your programming?


No
Yes
● Choice Count