SheppardMullin

Solving Open Source Problems With AI Code Generators - Legal issues and Solutions

PART 2 – SOLUTIONS

By: James Gatto

In Part 1 of this article, I addressed various legal issues that can arise with AI-based code generators. Many of these problems stem from using open source to train the models and the compliance obligations imposed by the applicable open source licenses. I also discussed how several companies have prohibited their developers from using AI-based code generators due to the legal risks.

Fortunately, there are some solutions to at least some of these problems. We are working with many companies to create corporate policies on employee use of AI. Employee use of AI code generators is just one of the issues to be addressed. Companies that want to permit their developers to use these tools may want to consider requiring the use of some or all of these solutions. This enables developers to leverage the advantages of these tools while mitigating legal risks from doing so.

Potential Solutions to Mitigate Open Source Legal Risks with AI Code Generators

- filters to prevent the output of problematic code
- code referencing tools to flag problematic output
- code scanning tools to assist developers with open source compliance.

In apparent recognition of the potential open source legal issues, some of the leading AI code generators have optional features. To mitigate legal risk, some companies are mandating, as part of their AI policies, that employees use these features.

Below, I summarize some AI code generator tools, how they describe the operation of their tool and features they offer to help companies manage certain open source issues.

Copilot

The Copilot website states the following about its operation:

> GitHub Copilot's suggestions[1] are all generated through AI. GitHub Copilot generates new code in a probabilistic way, and the probability that they produce the same code as a snippet that occurred in training is low. The models do not contain a database of code, and they do not 'look up' snippets. Our latest internal research shows that about 1% of the time, a suggestion may contain some code snippets longer than ~150 characters that matches the training set. Previous research showed that many of these cases happen when GitHub Copilot is unable to glean sufficient context from the code you are writing, or when there is a common, perhaps even universal, solution to the problem.

Copilot Solutions

Filter

Copilot offers a solution that includes a filter "to help detect and suppress GitHub Copilot suggestions which contain code that matches public code on GitHub." Use of this filter is optional for individuals accessing Copilot via an individual account[2]. But use of the filter can be mandated by an Enterprise administrator for access via a business account.[3]

When the filter is enabled, Copilot checks code suggestions against public open source code on GitHub. If there is a match, the suggestion is not shown. Assuming this works as intended, this filters out known open source code and prevents it from being shown to the developer user.

Reference Tool

Copilot has another tool [announced] called a "reference." This tool provides a "reference" for suggestions that resemble public code on GitHub so the developer user can make "a more informed decision about whether and how to use that code." This means, if the suggestion contains known open source code the developer (or counsel) can determine the relevant license and assess the potential risks of using that code. Most notably, as addressed in Part 1 of this article, this assessment can ensure there is no tainting of proprietary software and/or reveal any compliance obligations.

CodeWhisperer

According to its website:

> As a generative AI, CodeWhisperer creates new code based on what it has learned from the code that it was trained on and the context that you provided as prior code and comments. While CodeWhisperer is not designed to reproduce code that it was trained on, it is possible that on rare occasions it will generate code that closely matches particular code snippets in the training data.

---

[1] Copilot refers to the outputs of its AI code generator as "suggestions."

[2] Copilot for Individual users have the choice to enable that filter during setup on their individual accounts.

[3] For Copilot for Business users, the Enterprise administrator controls how the filter is applied. They can control suggestions for all organizations or defer control to individual organization administrators. These organization administrators can turn the filter on or off during setup (assuming their Enterprise administrator has deferred control) for the users in their organization.

Filter

According to its website, CodeWhisperer also can flag or filter code suggestions that resemble open-source training data and provide the associated open-source project's repository URL and license so developer users or counsel can more easily review and assess them.

Reference

CodeWhisperer also offers a reference tracker that detects whether a code suggestion is similar to particular CodeWhisperer open-source training data. If CodeWhisperer detects that its output matches particular open-source training data, the built-in reference tracker will notify you with a reference to the license type and a URL for the open-source project. The reference tracker can flag such suggestions or optionally filter them out.

In addition, all references can be logged for later review later (e.g., by counsel) to enable a developer to adopt the suggestion and keep coding without interruption.

The options for these tools are in the configuration setting for CodeWhisperer. For the free CodeWhisperer Individual Tier users, this setting is available in the IDE. With CodeWhisperer Professional, the AWS administrator can centrally configure this setting on an organization level from the AWS Management Console.

Independent Scan

Many companies with well-established open source policies and procedures routinely use an open source code scanning tool (e.g., Black Duck) to ensure that their code includes no problematic open source and so they can ensure compliance obligations are met. For companies that already do this, this scan can detect any open source code that may have been output as a suggestion from an AI code generator and adopted by the company's developer.

For companies without an established open source policy and/or do not do open source code scans before releasing their software, you should! For an overview of open source policies and why you need them see here [link]. This is true whether or not you permit your developers to use AI code generators. If you do permit your developers to use AI code generators this is just one more reason to develop an effective open source policy and do code scans.

Mandating Use of AI Code Generator Tools In Company AI Policies

When adopting a corporate AI use policy you should consider various issues regarding use of AI code generators. The following are some issues to consider:

- Do you want to permit employees to use AI code generators?
- If so, further consider:
    - o whitelisting an approved set of AI code generators based, at least, on whether they have adequate tools (e.g., filters, referencing and/or other tools) that prevent your employees from seeing known open source code as suggestions and/or reference the license if the suggestion is based on open source
    - o requiring the use of a business account where an administrator can configure the settings to mandate use of these tools
    - o ensuring that any referenced open source is vetted based on the companies existing open source policy and/or by legal counsel.
- And where the cost is justified, consider running your own open source code scan to further ensure there is no tainting or other legal issues and that you can understand and abide by any compliance obligations.

Besides adopting an AI use policy, it is advisable to update your open source policies to address AI code generator issues.

Conclusion

AI has great promise and can bring great efficiency to help you developers. But it can also raise several thorny legal issues. Fortunately, there are solutions to help manage the legal risks. It is prudent for companies to develop and/or update their AI use polices and open source policies to ensure responsible use by employees (and contractors).

The open source issues addressed in this article are just one set of issues that need to be addressed in an AI use policy. For examples of some other issues, see here.

Many companies just getting started on these policies often find that a custom in-house presentation on these topics is a great way to start and get their arms around the issues. We routinely do presentations to help companies understand the issues that should be addressed in employee AI use policies and various options for addressing the issues. We also help create these policies. If you have questions on developing AI use or open source policies reach out to me.

For more details on generative AI, please contact:

James Gatto
Blockchain and Fintech Team Co-leader
bio
202.747.1945
jgatto@sheppardmullin.com

Sheppard Mullin's Blockchain Technology and Fintech team helps clients develop innovative and comprehensive legal strategies to take advantage of what may be the most disruptive and transformative technology since the Internet. We focus on advising clients on how to meet their business objectives, without incurring unnecessary legal risk. Our team includes attorneys with diverse legal backgrounds who collectively understand the vast array of legal issues with and ramifications of blockchain technology and digital currencies. More Information

---