软件开发人员在软件开发过程中使用 GitHub Copilot 工具的体验研究。

1st Mateusz Jaworski 2nd Dariusz Piotrkowski华沙理工大学 华沙理工大学华沙，波兰 华沙，波兰

arXiv:2301.04991v1 [cs.SE] 12 Jan 2023

摘要—**在软件开发中，人们始终面临着在不影响质量的前提下越来越快地编写代码的压力。为满足这一需求，各种支持开发人员的新工具应运而生。目前，新一代此类解决方案即将问世——由人工智能驱动的工具。2021年6月29日，GitHub Copilot 正式发布。它使用经过训练的模型，根据人类可理解的语言生成代码。本研究旨在探究软件开发人员对这一工具的看法。为此，我们准备了一份包含 18 个问题的问卷调查，并将其发给程序员。共收集到 42 份答复。研究结果表明，开发人员的意见不一。大多数受访者在参加问卷调查前就听说过 GitHub Copilot。受访者对该工具的态度总体积极，但愿意使用它的人并不多。对使用 GitHub Copilot 可能引发安全问题的担忧，是造成这一现象的原因。**

索引词—**GitHub Copilot、人工智能驱动工具、问卷调查、软件开发**

I. 引言

如今，软件开发人员可以使用提供自动代码补全和语法建议的工具。尽管这些工具很有帮助，能大幅加快开发人员的工作速度，但人们始终面临着越来越快地编写代码的压力。因此，新一代工具正在开发之中——基于机器学习的代码生成器。它们是经过训练的模型，能够根据某些定义生成代码，例如自然语言描述 [1,2]。

GitHub Copilot 于 2021年6月29日 发布，是一款人工智能工具，以扩展程序的形式为 Visual Studio Code 和 JetBrains IntelliJ 集成开发环境开发 [3]。GitHub Copilot 基于 Codex 语言模型，该模型使用大量源代码进行训练。它会根据函数名称和给定变量等信息，建议函数的源代码 [1]。程序员会收到所需代码部分的一些推荐实现方案，并可以选择最合适的方案，必要时还可进行调整。

本研究探讨软件开发人员使用 GitHub Copilot 的体验、他们对这一工具的看法，以及他们对其未来发展的态度。

A. 问题陈述

目前，我们正见证着开发人员支持工具的变革。先进的人工智能算法不仅能够建议或自动补全语法，还能生成完整的代码片段。迄今为止，还没有可比的工具能为开发人员提供如此大的帮助。首批如此先进的软件之一——新型基于人工智能的代码生成器 GitHub Copilot，于2021年6月29日发布。它比广泛使用的代码补全和语法建议工具先进得多。此类工具的发展可能会影响开发人员的日常工作以及他们在劳动力市场上的处境。

在此填写适用的资助机构。如无，请删除此内容。

B. 研究目标

本研究旨在了解开发人员使用 GitHub Copilot 的体验和态度。我们希望了解他们认为这一工具在日常工作中有何用处、哪些因素促使他们使用它，以及他们如何看待 GitHub Copilot 对信息技术行业就业的影响。

C. 贡献

本研究共考察了来自不同资历级别和专业领域的开发人员提交的 42 份答卷。问卷调查包含的问题涉及开发人员对 GitHub Copilot 的总体态度、他们对这类工具未来发展的预测，以及这类工具对软件工程和该解决方案安全性的影响。

II. 相关工作

GitHub Copilot 是一款不到一年前发布的新解决方案，因此针对它开展的研究并不多。不过，科学家多年来一直在研究人工智能在编程中的应用。Dominik Sobania、Martin Briesch 和 Franz Rothlau 在其《选择你的编程副驾驶》 [1] 一文中，将一种方法——遗传编程——与 GitHub Copilot 进行了比较。他们将 GitHub Copilot 应用于常见的程序合成基准问题，以研究这两种解决方案的性能。随后，他们将所得结果与相关文献中描述的遗传编程算法的结果进行了比较。研究表明，这两种工具在解决常见算法问题方面的性能相近。

人工智能在编程中的应用也是 Raphael's Jenni 的作品《面向编程语言的机器学习》[4]所探讨的主题。作者概述了

深度学习在辅助开发人员编写代码方面的应用。RaphaelJenni 将 GithubCopilot 列为运用这项技术的现代工具之一。

Stuart Fitzpatrick 在其著作《论人工智能代码副驾驶的本质》中也描述了 GitHub Copilot 的一些方面。他探讨了人工智能结对程序员是否构成训练数据的一种“编译”形式，类似于可执行文件；抑或它们更接近传统程序的源代码，并且能够通过进一步训练进行修改 [5]。Fitzpatrick 还提出了版权问题，即 GitHub Copilot 在搜索针对所描述问题的建议解决方案时所使用代码的版权归属。

Neil A. Ernst 和 Gabriele Bavota 在其著作《人工智能驱动的开发已至：你该担心吗？》中，将人工智能驱动的开发环境描述为软件工程领域的一场新革命[3]。他们将编程划分为三个时代：第一个时代，程序员必须独立编写全部代码；第二个时代，集成开发环境会提供语法建议和自动补全；第三个时代——目前正在到来——人工智能驱动的开发环境能够生成代码。作者以 GitHub Copilot 为这类环境的一个例子。这项研究简要介绍了该工具，描述了人们对这类环境的期望，以及它们面临的挑战。

谈到 GitHub Copilot，生成代码的安全问题引发了诸多担忧。该工具使用基于开源 GitHub 代码仓库训练的模型。这些代码仓库中的代码通常未经审查，因此模型可能是在有缺陷的代码上训练的。这是五位作者在《键盘前睡着了？评估 GitHub Copilot 代码贡献的安全性》一文 [2]中开展的研究主题。他们利用 GitHub Copilot 为安全敏感问题生成解决方案，并对结果进行了评估。在生成的 1689 个程序中，约有 40% 被判定为存在漏洞。

III. 研究设计与方法

为开展本研究，我们设计并编制了一份问卷。所有问题均由具有编程经验的软件开发人员拟定。随后对问题进行了审核，并将反馈意见纳入最终版本。问卷共包含 18 个问题，其中 14 个为封闭式问题，4 个为开放式问题。问卷分为 3 个部分。第一部分包含 2 个问题，询问受访者是否听说过或使用过 GitHub Copilot。接下来的 11 个问题考察受访者对该工具的态度。本部分还通过询问受访者的经验、专业领域和所用技术栈来了解其背景。问卷的第三部分涉及 GitHub Copilot 的安全性。

问卷调查使用 Google 表单创建，并在作者所在的环境中向软件开发人员发放，也通过 Facebook 和 LinkedIn 等社交媒体进行传播。所有封闭式问题均为必答题，开放式问题则为选答题。未发现虚假答案，因此所有答案均纳入考虑。共收集到 42 份答卷，收集时间为 2022 年 07 月 03 日至 2022 年 04 月 25 日。结果导出至 Excel，并使用现有工具制作图表。

IV. 结果

本节介绍问卷调查的 42 项结果。第一小节概述程序员是否听说过该工具以及他们对此的态度。接下来介绍有关程序员类型及其经验的数据。最后一小节讨论安全性，以及受访者阅读 GitHub Copilot 创作者提供的有关该工具安全性的信息后态度的变化。

A. 总体概述

第一个问题旨在了解开发人员是否听说过 GitHub Copilot。如图 1 所示，28 人（66.7%）听说过该工具，14 人（33.4%）没有。

更有意思的是，只有 6 人（14.3%）使用过该工具，而 36 人（85.7%）没有使用过。第二个问题的结果如图 2 所示。

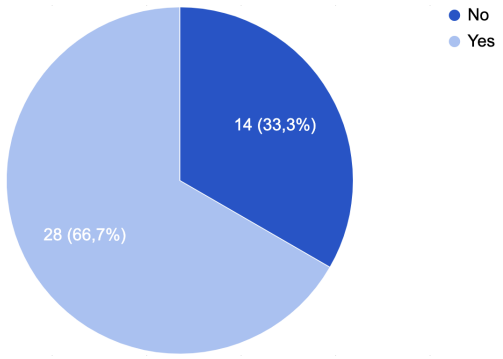


图 1. 听说过 GitHub Copilot

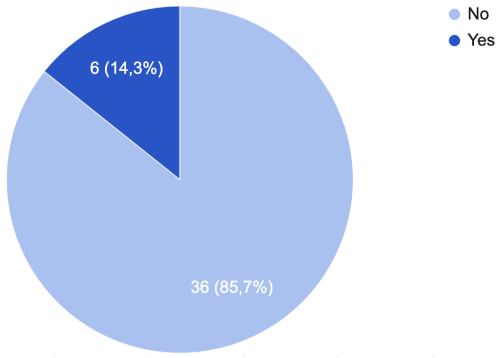


图 2. 受访者中 GitHub Copilot 的使用情况 ts

接下来的三个问题（第三、第四和第五个问题）是向正在使用 GitHub Copilot 的 6 个人提出的。

第三个问题询问了使用动机。受访者的回答如下：

- 编写代码的速度

- 工作
- 提升性能
- 创新解决方案
- 好奇心
- 协助编写重复的代码片段

接下来，在第四个问题中，他们被问及使用该工具已有多久。3 人（50%）使用了两周，另一半使用了超过两周。此外，一位受访者回答说：“几个月”。

图 3 展示了第五个多选题的结果，内容是使用 GitHub Copilot 的项目类型。5 名开发人员正在测试该工具，2 名开发人员将其用于商业项目、开源项目或作品集，另有 1 人用于学习。

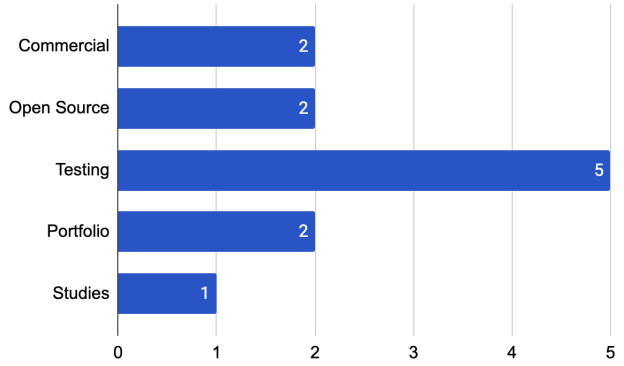


图 3. 使用该工具的项目类型

此后所有问题均面向全体受访者。

第六个问题询问了他们测试该工具的意愿。结果如图 4 所示：超过半数，即 26 人（ 61.9% ），不愿意测试该工具；16 人（38.1%）愿意。

第七个问题询问了开发人员对人工智能工具的态度。图 5 展示了他们的回答。14 人（33.4%）持积极态度，正在使用或希望使用此类工具；12 人（28.6%）态度相同，但不希望使用。6 名开发人员（14.3%）没有明确看法。6 名受访者（14.3%）持积极态度，但不希望使用此类工具。只有 4 人（9.5%）给出了否定回答，并且不愿意测试该工具。

B. 参与者特征

第二组问题旨在了解并描述程序员的类型及其经验水平。第八个问题询问不同群体的程序员未来就业是否会受到负面影响。图 6 展示了他们的回答。该图表可分为两组：按开发者类型和经验水平划分。结果按以下模式呈现：

[类型]（正面；不会产生影响；负面），显示开发人员的类型和数量。

(a) 经验水平

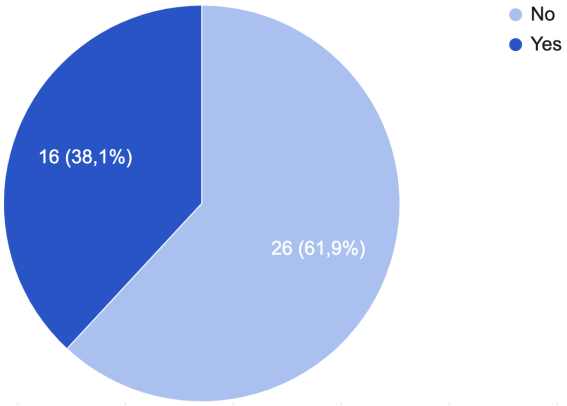


图 4. 测试该工具的意愿

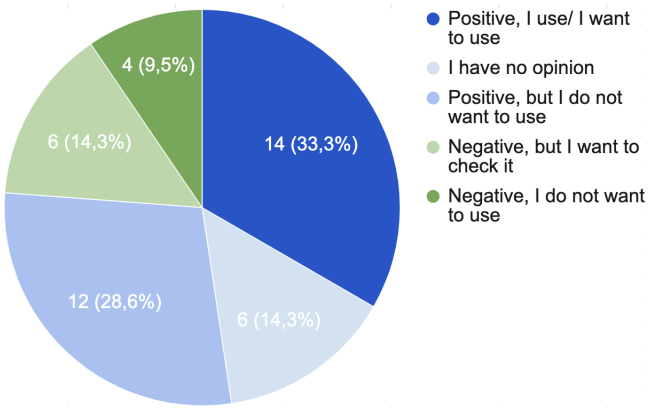


图 5. 对这类工具的态度

- 初级（8；12；22）
- 中级 (7; 32; 3)
- 高级 (10; 29;

33) (b) 开发者类型

- 后端（7；29；6）
- 前端（11；24；7）
- 全栈（11；27；4）
- 移动端（9；30；3）

下一个问题询问了他们的经验水平。如图7所示，初级开发人员的回答人数最多，为18人（42.9%）；其次是中级开发人员，为15人（35.7%）；高级开发人员人数最少，为9人（21.4%）。

本部分的第十个问题询问参与者是否认为该工具会有帮助。超过半数的人回答“是”，共24人（57.1%）；回答“肯定会”的有6人（14.3%）。有10名开发人员（23.8%）难以表达自己的意见，只有2人（4.8%）认为该工具不会有帮助。结果如图8所示。

在第十一个问题中，参与者被要求选择自己的专业领域。如图 9 所示，人数最多的是后端开发人员，共 24 人（57.1%）；其次是全栈，共 9 人（21.4%），以及前端，共 5 人（11.9%）。人数最少的群体是移动端，共 3 人（7.1%），其他类别有 2 人（2.5%）。

接下来，他们欣然写下了自己的技术栈。

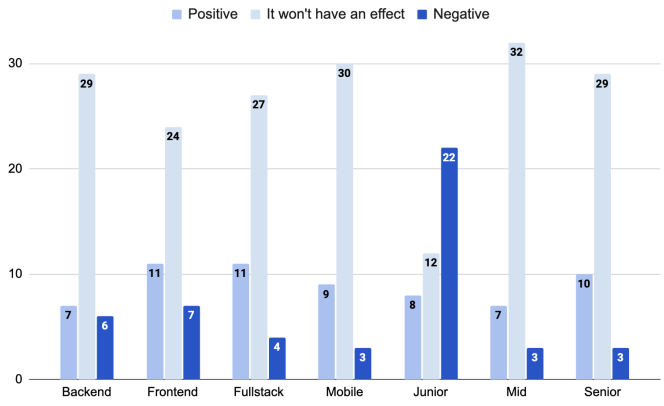


图6. 对程序员未来就业的影响

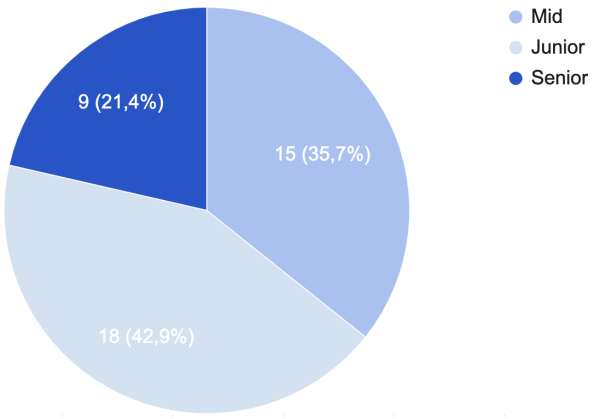


图7. 经验水平

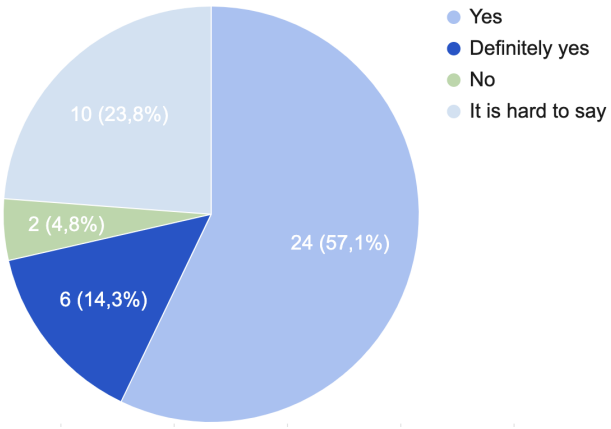


图8. GitHub Copilot 的帮助程度

Java 获得的回答最多，其次是 JavaScript 和 Python。也有人回答 C# 或 C++，但这类回答少得多。开发人员根据所用技术写下了不同的框架。

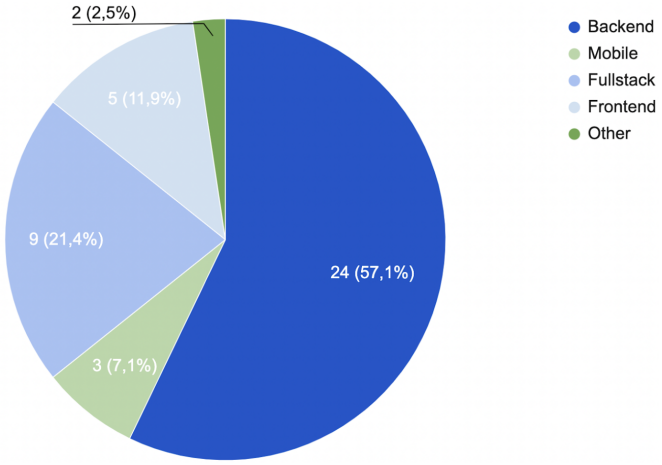


图9. 开发人员的专业领域

C. 安全性

使用 GitHub Copilot 时，代码建议会从互联网下载。首先，研究人员询问受访者对发送的数据（如漏洞利用）的担忧。随后，他们向受访者展示了作者提供的安全信息：

’所有数据均经过安全传输和存储。遥测数据的访问权限受到严格限制，仅向有必要知情的人员开放。’ [5]

随后，他们欣然再次作答。图 10 展示了阅读作者提供的信息前后的结果。在开发人员组中，近一半的人担心发送到项目中的数据：17 人（40,5%）；12 人（28,6%）表示不知道，13 人（31%）信任作者。阅读信息后，该组有 25 人（59,5%）表示担忧，17 人（40,5%）表示不担忧。

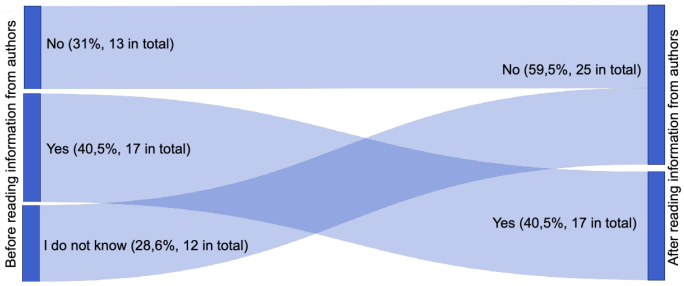


图 10. 对发送到项目中的数据的担忧

其次，他们被问及是否担心私人商业代码库中的数据泄露。随后，他们再次阅读作者提供的信息并作答：

“我们使用遥测数据（包括用户接受或拒绝哪些建议的信息）来改进模型。您的

私人代码不会作为建议代码提供给其他 *GitHub Copilot* 用户。”[5]

如图 11 所示，在阅读相关信息之前，超过半数的受访者（22 人，占 52.4%）担心项目会发生数据泄露，13 人（31%）不担心，只有 7 人（16.7%）表示不知道。了解相关信息后，回答“是”和“否”的人数相同，均为 21 人（50%）。

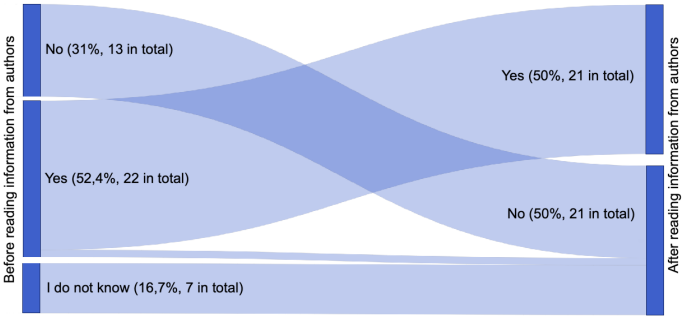


图 11. 对项目数据泄露的担忧

在最后一个开放式问题中，受访者很乐意表达自己的看法。分析所有回答后发现，受访者经常提到，当代码不太复杂时，该工具可以提供帮助并加快工作进度。但要借助它实现更复杂的组件则会比较困难。该工具对初级开发人员或学生的学习和知识积累过程可能很有帮助。许多受访者对数据泄露、版权和安全问题表示担忧。许多受访者认为该工具可能会有所帮助，但不会取代人类的工作，因为市场仍需要专业人才。

V. 讨论与未来工作

本研究的主要目标是探讨软件开发人员在软件开发过程中使用 Github Copilot 的体验和看法。尽管该工具相对较新，仍有三分之二的受访者听说过它。大多数开发人员出于测试和好奇而使用该工具。虽然 85.7% 的受访者没有使用过该工具，但在了解其功能和潜力后，超过半数的人愿意试用，并认为它似乎有助于提升工作效率。也有人对这类解决方案持否定态度，不愿使用。分析专业领域图表后可以发现，远远更多的人选择了“不会产生影响”。后端的情况更值得关注，因为“正面”和“负面”两种回答的比例相近。另一方面，在前端、全栈和移动端等输入更多的领域，认为它会产生正面影响的回答呈上升趋势。在按经验水平划分的情况下，中级和高级开发人员中，认为“不会产生影响”的回答数是其他回答的数倍。负面评价只有少数，但正面评价更多。不过，初级开发人员的图表尤为突出，因为认为它会产生负面影响的回答最多。尽管作者作出了保证，开发人员对”数据传输安全性或数据泄露方面的看法并不明确，大多数人对此有所顾虑。

许多程序员尚未使用过 GitHub Copilot，这可能是因为在「GitHub Copilot 等候名单」 [5] 上等待时间过长，也可能是因为安全性和版权方面的问题，导致他们没有获得工作许可。这似乎是一个很好的理由，可以对更多在较长时间内使用并测试过该工具的开发人员开展问卷调查。在这群人中，大多数受访者都使用了该工具数周。总体而言，大家认为这个工具很有趣，值得关注和测试。可以按照经验水平，考察该工具对生产力的影响。例如，可以将参与者分为两个较大的群体，其中初级开发人员在有无 GitHub Copilot 的情况下完成相同的任务。安全问题往往令人担忧，也让开发人员缺乏信心。为了确认将数据发送到项目中是否确实不存在风险（例如漏洞利用），一个值得开展的研究是让更多人长期使用该工具，并检查研究结果。

VI. 结论

本研究的主要目的是探讨软件开发人员对新型人工智能结对程序员 GitHub Copilot 的态度。问卷调查结果显示，开发人员对这类工具的看法各不相同。大多数人在参与问卷调查前就听说过 GitHub Copilot，但只有 14,3% 的人使用过它。就 GitHub Copilot 的使用情况而言，大多数应用都涉及非商业项目，例如测试该工具、制作作品集、学习或开源项目。

关于开发人员对 GitHub Copilot 态度的部分表明，他们对使用该工具的兴趣较低。只有 38,1% 的参与者愿意尝试这一工具，尽管大多数人对它持积极态度，并认为 GitHub Copilot 很有帮助。至于它对程序员未来就业的影响，大多数回答指出，它会对初级开发人员产生负面影响，而其他资历级别的开发人员则不会受到影响。

回答显示，大多数问卷调查参与者专攻后端编程。全栈和前端开发人员也占受访者的相当大一部分。只有 21,4% 的参与者是高级开发人员，人数较多的群体是初级开发人员和中级开发人员，分别占 42,9% 和 35,7%。

关于安全问题，程序员们意见不一。相当一部分受访者担心发送到项目中的数据以及从项目中泄露的数据。即使阅读了 GitHub Copilot 作者关于数据会被安全发送的保证，也没有多少参与者改变看法。

参考文献

[1] Dominik Sobania、Martin Briesch 和 Franz Rothlauf，《选择你的编程副驾驶》 [2] Hammond Pearce、Baleegh Ahmad、Benjamin Tan、Brendan Dolan- Gavitt、Ramesh Karri，《键盘前打盹？评估 GitHub Copilot 代码贡献的安全性》 [3] Neil A. Ernst 和 Gabriele Bavota，《人工智能驱动的开发已至：你该担心吗？》 [4] Raphael Jenni，《编程语言的机器学习》。 [5] https://copilot.github.com/