A Reflection on Human-Notebook Experiences in the Era of AI

Yunchen Huang
Microsoft Azure Data
frank.huang@microsoft.com

Serena Hillman
Microsoft Azure Data
serena.hillman@microsoft.com

ABSTRACT
Computational notebooks provide an interactive way to work with data. They have been widely used by data professionals to write code, explore data, and generate visualizations, all in one document [11][19][22]. Previous research (e.g.,[5][27]) has revealed unique pain points around the user experience (UX) in computational notebooks. However, as artificial intelligence tools like ChatGPT or Copilot have emerged it is unclear whether these pain points have been reduced, changed, or new pain points have arisen. Due to the fast pace of AI technology, most of the development of new AI tools has been primarily driven by technology and not by user experience. In this paper, we summarize literature on how this new technology has impacted interaction and Human-Computer Interaction (HCI) paradigms, new challenges and user behavior around using AI assistants, and recent research on AI assistants in Computational Notebook scenarios. We outline gaps in existing literature and suggest a future focus on improving macro human-notebook experiences throughout a user's workflow, measuring and quantifying the value of AI systems, and establishing a set of standards and best practices for AI tools.

CCS CONCEPTS • Human-centered computing • Human computer interaction (HCI) • Interactive system and tools

Additional Keywords and Phrases: Computational Notebooks, ChatGPT, AI Assistants, Human-Notebook Experience

BACKGROUND

In this section, we delve into the current AI landscape and its impact on the data and analytics field. We begin by providing background information on computational notebooks, along with their associated challenges and then summarize the current AI Data Analytics landscape. This serves as a foundation for our exploration of Human-Notebook Interaction with AI assistants and the identification of gaps in current literature.

Computational Notebooks

Computational notebooks are versatile documents that blend code execution, text, visuals, and multimedia within one interface. They serve as a workspace for tasks like data analysis, exploration, visualization, and documentation. Users can write, edit, and run code alongside explanatory text, equations, images, and interactive visuals. This setup makes it easy for users to document their analysis step-by-step, aiming to make their work understandable, replicable, and extendable for others [22]. See Figure 1 for example Notebook Interface. This approach fosters a narrative-driven method for data analysis and engineering. Various professionals like researchers, data engineers, data scientists, and even citizen developers, such as journalists [24], utilize these notebooks.

Collaboration is integral to computational notebooks, allowing multiple users to work together with features such as version control, commenting, and real-time editing. Users can contribute code, notes, or visualizations, while also incorporating multimedia elements like images and videos to enhance communication and create engaging presentations.

However, despite their popularity, many computational notebooks used in the real-world often lack detailed explanations of reasoning and results, which along with the format, can lead to a perception of disorderliness [25]. Furthermore, there's a tension between the dual purposes of notebooks: exploration, and explanation, which is further complicated by the involvement of citizen developers who can often rely on explanation to leverage existing notebooks [25].

AI in Data Analytics

In today's rapidly changing technological landscape, the importance of artificial intelligence (AI) cannot be overstated. AI has emerged as a transformative force with profound implications for nearly every sector of society unlocking unprecedented opportunities for innovation, efficiency, and growth. Furthermore, AI is not merely a distant prospect but an immediate reality, as evidenced by visionary initiatives. It permeates the fabric of our daily lives. In the words of Satya Nadella, CEO of Microsoft, "AI is really in the air now" [16].

This recent progress in artificial intelligence (AI) highlights two key opportunities in the data and analytics field. The first opportunity is AI assistants, both reactive (e.g., Microsoft's Copilot [4]) and proactive (e.g., Clippy-like [1][15][23]). Reactive AI assistants respond to user inputs by generating responses based on the patterns it has learned during training. These assistants don't focus on proactive capabilities to anticipate user needs or provide assistance without explicit user requests or confirmation. Their primary function is to respond to user queries or prompts. Whereas proactive AI assistants attempt to anticipate user needs by offering direction or acting based on context, such as when they detects that a user might need help formatting a document or using a certain feature. While the implementation and effectiveness may vary, its design aimed to be proactive in providing assistance without waiting for explicit user requests. Later on in the paper, we will explore going beyond the binary mindset of proactive and reactive.

Secondly, there's emphasis on advancing the development and training of AI models, demonstrated by technologies such as

Reference Format:
Yunchen Huang and Serena Hillman. 2024. A Reflection on Human-Notebook Experience in the Era of AI. Microsoft Technical Report MSR-TR-2024-11. Microsoft Corporation.

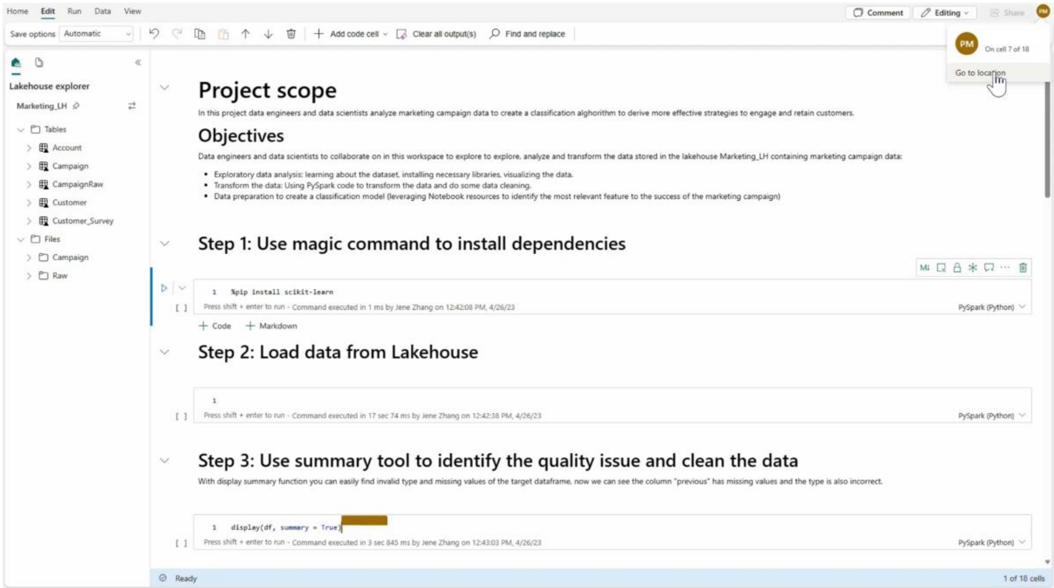

Figure 1. Microsoft Fabric's Notebook Interface with 3 code cells. Source: https://learn.microsoft.com/en-us/fabric/data-engineering/how-to-use-notebook, accessed February 26th, 2024.

Automated Machine Learning (AutoML), which assist users in creating new AI technologies (e.g.,[27]). These technologies look to automate parts of the machine learning pipelines or even the full process with a focus on low-code. The term AutoML was coined by Guyon et al. in 2015 [7]. Common AutoML tools include Google Cloud AutoML, Amazon SageMaker Autopilot, DataRobot and Azure Machine Learning (AutoML offerings). While AutoML is not the focus of our paper, instead, we focus on proactive and reactive AI Assistants, it is helpful to understand the breadth of conversation by calling out these two AI focal points within the data and analytics domain.

The heightened focus on AI assistants extends to various roles within the domain of data and analytics. Within this space, there is a growing anticipation for the potential of computational notebooks and AI assistants to revolutionize workflows and enhance productivity for data engineers, data analysts, data scientists and citizen developers [24]. These AI assistants aim to boost user efficiency by supporting various tasks like data analysis, management, querying, communication, collaboration and taking action from data, and aiding in helping the user understand the tools and conceptual capabilities of the system [2].

DISCUSSION

In this we start by reviewing the latest research findings by Jakob Nielsen on Human-Computer Interaction (HCI) paradigms [17], highlighting AI's importance as a new paradigm in user-computer interaction. We then explore Barkel et al.'s [3] investigation into Human-AI Interaction Paradigms, which extends beyond the reactive versus proactive interaction paradigm, aiming to represent the nuances surrounding AI system and levels of proactivity. Finally, we offer a detailed overview of ongoing research on AI assistants within computational notebooks, covering achievements, challenges, and potential opportunities.

DEFINING AND RE-DEFINING PARADIGMS

HCI Paradigms

The increasing prevalence of AI assistants in the HCI field is remarkable due to its profound influence on users and the introduction of novel methods for interacting with user interfaces (UI). This transition marks an important variation in HCI and User Experience (UX) practices, providing fresh opportunities for users to engage with technology and interfaces. Consequently, it has the potential to redefine conventional interaction patterns and user expectations.

According to Nielsen's UI paradigm framework, the evolution of HCI has been marked by three distinct paradigms [17]. The first paradigm, known as batch processing, emerged in 1945. In this model, users defined a complete workflow and submitted it to the data center for execution and was often processed overnight. This paradigm stems from the birth of computers and is aptly named for the lack of back and forth between the system and the user. Following this, around 1964, the second paradigm, command-based interaction design, came into prominence. This involved users interacting with computers through a turn-based system, issuing one command at a time. Despite its influence and longevity, the command-based approach will gradually yield ground to what we are experiencing now: the start of the third paradigm. Driven by advancements in generative AI technology, the third paradigm is currently in its infancy. Referred to as intent-

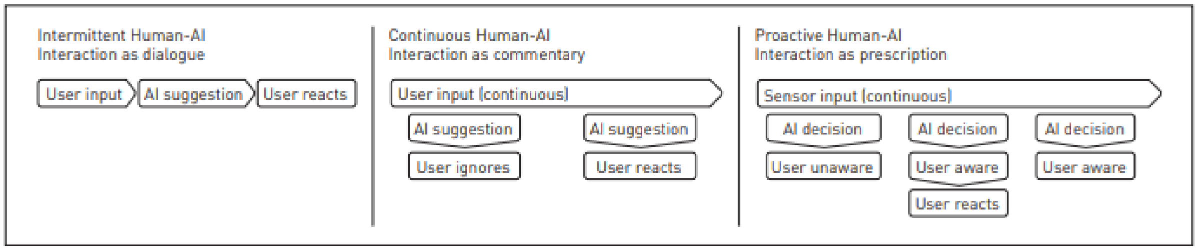

Figure 2. From Berkel et al.[3], a visual representation of Human-AI interaction across three paradigms of interaction.

based outcome specification, this represents a significant shift in HCI. Instead of providing specific instructions to the computer, users focus on specifying the desired outcome. Nielsen, a leading figure in usability engineering and UX, tells us that ChatGPT and Copilot-like applications are just the beginning of this third paradigm. He regards this AI-driven transition as revolutionary in reshaping how we engage with computers and expresses optimism about its potential for the future of HCI, recognizing its transformative impact [17].

Human-AI Paradigms

While understanding Nielsen's UI paradigm framework is crucial for comprehending AI's current impact on the evolution of HCI, it is also important to reflect on the categorization of Human-AI interaction types. In this paper, we refer to these as reactive and proactive interaction models. However, Berkel et al. [3] further break down these interaction models into a more granular three-category approach: intermittent, continuous, and proactive. This suggests that a binary classification of proactive and reactive is not adequate to describe current AI assistant interaction models. The intermittent paradigm involves a turn-taking process where the user initiates an action, and the system responds accordingly. Continuous interaction relies on a continuous stream of user input rather than discrete instructions. In the proactive paradigm, the system takes the initiative and autonomously performs tasks without explicit user input.

While traditional systems or tools (e.g. Siri, driver training simulator, smart thermostats, or smart lighting systems) often align with one or two of these interaction paradigms, computational notebooks use case scenarios span all three, each presenting their own set of unique challenges. Intermittent interaction necessitates explicit cues and inputs from the user, posing challenges as users must be highly articulate in their prompts to effectively interact with the AI. This challenge is compounded by the articulation barrier, where low literacy can hinder users' ability to use prompt-driven AI systems efficiently [18]. According to the Organization of Economic Co-operation and Development (OECD) data, half of the population in rich countries and more than 85% in some middle-income countries have low literacy levels [9], which means they cannot read or write complex texts. This limits their ability to use prompt-driven AI systems effectively. In addition, writing new descriptive prose is more challenging than reading and understanding prose already written by somebody else, adding to the articulation barriers users might face [18]. Nielsen recommends more qualitative research and design with users with different literacy levels using the AI systems. He also suggests that a hybrid interaction paradigm that combines intent-based and command-based paradigms might offer solutions to overcome the articulation barriers because UI presents to users what they can do rather than requiring them to articulate [18]. In computational notebooks, this challenge becomes apparent when users engage with tools like Copilot to generate code for specific tasks. In these scenarios, the barrier to articulation remains, compounded by the complexities of programming jargon and specialized terms, which can present additional challenges for users.

In the continuous interaction paradigm, users are typically focused on a task, and any AI suggestions that divert their attention from their primary objective may cause frustration. Balancing the timing and quantity of suggestions to avoid distracting users while enhancing their workflow poses a significant challenge [3]. A common scenario in a computational notebook occurs when AI offers code completion or auto-debugging based on the user's ongoing programming activity. The main challenge lies in balancing the degree of intrusiveness of these suggestions while ensuring that the user can remain focused on their programming tasks without distractions.

In the proactive interaction paradigm, where the system takes the lead in performing tasks, the interaction cycle shifts from action to reaction. While proactive AI systems aim to reduce cognitive load by automating tasks, poor interaction and inaccurate predictions may necessitate additional effort from the user to correct errors or adjust the system's behavior [3]. This challenge is prevalent across various scenarios in computational notebooks. For instance, if an AI system automatically generates code or visualizations for users, they must assess this auto-generated content. If the content is inaccurate, it only imposes additional effort on the users.

Common Behaviors with AI Assistants

Upon reviewing previous studies on AI assistants, it's clear that consistent patterns of user behavior are starting to emerge. These behaviors provide valuable insights into how users engage with AI assistants, revealing common practices, preferences, and challenges in human-AI interaction. Analyzing these patterns allows researchers to gain a deeper understanding of user needs and preferences, which can then inform the design and development of more effective AI assistant systems. In this section, we will review a few examples of research that have identified key user behavior patterns.

Gibbons et al. [6] identified two common behaviors when users interact with AI-generated text. The first, termed Accordion editing, involves users asking the AI to condense or expand its responses repeatedly to achieve a single goal. The second, referred to as Apple Picking, occurs when users reference previous AI responses in subsequent prompts to achieve their desired output. Both behaviors highlight challenges users face in interacting with AI and demonstrate how users adapt to current limitations. For instance, Accordion editing involves a lot of back-and-forth interactions, while Apple Picking requires extensive scrolling and strains users' working memory. The authors proposed three improvements to AI assistant systems based on these observations: Compartmentalization of AI responses, which allow users to make changes to a portion of a response directly rather than requiring them to write a prompt to produce a whole new response, direct editing, which allows user to edit the response text directly, and point to select, a GUI that allow user to interact with the dialogue with the AI system.

In a more focused study, Yang et al. [24] examined how AI systems could serve as decision support tools (DST) for clinicians. They found that AI-generated DSTs could integrate more effectively into clinical practice if their interactions were tailored to specific points in the decision-making workflow. This led to the insight that AI should not only be intelligent but also highly integrated into users' routines and workflow. Additionally, the authors discovered that certain levels of unobtrusiveness helped mitigate resistance among clinicians towards clinical DSTs. However, determining the optimal level of unobtrusiveness and striking a balance between enhancing decision-making and altering the nature of clinical decision-making remain open questions.

These insights, as revealed by studies such as those conducted by Gibbons et al. and Yang et al., offer a deeper understanding of user needs, preferences, and challenges. Whether it's addressing challenges like Accordion editing and Apple Picking identified by Gibbons et al. or optimizing AI systems for specific contexts like clinical decision-making as explored by Yang et al., the importance of understanding user behavior patterns cannot be overstated. These insights pave the way for more effective and user-centric AI assistants, ultimately enhancing human-AI interaction across various domains.

Computational Notebooks and AI Assistants

The industry is currently witnessing remarkable progress in the development of AI assistants tailored for Notebooks. Notably, Jupyter Notebooks has recently unveiled a suite of generative AI tools, which have garnered considerable attention. These tools are designed to explain code, fix errors, querying local files, and even generating entire notebooks [28]. These advancements are further enriched by research endeavors such as the investigation conducted by Wang et al., which delves into leveraging generative AI to support data scientists in the creation of documentation [25]. Wang et al.'s system harnesses deep-learning techniques to generate documentation, proactively prompts users to access API documentation, and reminds them to document their work diligently. Furthermore, Wang et al.'s research yields promising outcomes, as data scientists report heightened satisfaction and reduced documentation time.

In addition to these developments, McNutt et al. conducted two studies focused on understanding data scientists' expectations and opinions regarding code generation tools in notebooks [13]. This research complements the development of AI tools by outlining design implications categorized into politeness, notebook patterns, and code assistance patterns. Meurish et al. further explored user expectations of proactive AI systems and identified factors influencing the desired level of proactivity [14]. Their findings suggest that users are generally receptive to proactive support, but expectations vary based on use cases, personality traits, and context.

Barke et al. examined programmer interactions with Copilot and discerned user preferences between acceleration and exploration modes [2]. Acceleration entails the AI assistant's capacity to expedite tasks and processes, such as automation, decision support, and code creation. Exploration mode refers to the AI assistant's ability to aid in uncovering new information within the tool or system, as well as providing conceptual domain expertise. They recommend AI systems be aware of the current interaction mode and adjust behavior accordingly.

Sarkar et al. compared Copilot usage in programming with other actions and outlined key challenges [21]. These include communicating system capabilities, verifying generated code correctness, making AI-driven programming understandable, understanding the consequences of automation, and determining user preferences regarding direct answers versus solution processes.

While not surprising, it's worth mentioning that recent research is also uncovering common challenges with AI assistants also within a computational notebook environment. For instance, Jayagopal et al. explored how beginner programmers adapt to using code generation tools and the factors that affect their learning process [10]. Their results are consistent with Nielsen's idea of articulation barriers [18], which in this case, specification size (extra clarification that user crafted before successfully eliciting a synthesis output) negatively impact the user experience, especially in scenarios involving code creation.

CONCLUSION

The emergence of generative AI not only offers new methods for interacting with UIs but is also redefining conventional interaction patterns and user expectations. However, existing research in the field has primarily focused on intermittent interaction paradigms, characterized by sporadic engagement with AI systems. There has been relatively little exploration of continuous and proactive interaction paradigms, particularly within the intricate contexts of computational notebooks. This discrepancy highlights the need for further investigation to fully grasp the implications of mixed interaction paradigms, where elements of intermittent, continuous, and proactive engagement may coexist. By delving deeper into these mixed paradigms, researchers can gain valuable insights into how users interact with AI systems over extended periods and in various contexts. There is a pressing need to evaluate user experience against identified factors in order to inform the development of more effective AI systems tailored to the needs of users in computational notebook environments.

FUTURE DIRECTIONS

While existing research provides a foundation for the literature on AI in computational notebooks, notable gaps and unanswered questions remain, which is understandable given the paradigm's early stage of development. Here, we pose three questions to provide directions for future research:

How can we leverage AI capability and across the different interaction paradigms to improve the macro user experience throughout users' workflow?

Computational notebooks are just one component of the larger framework that comprises a data professional's workflow. Beyond simply using AI within these notebooks, there is a broader opportunity to leverage AI capabilities to enhance the overall user experience throughout the entirety of users' workflows. This involves considering how AI can streamline tasks and decision-making processes at every stage, from data acquisition and preprocessing to analysis, visualization, communication of results and action. By integrating AI seamlessly into the workflow, data professionals can not only improve efficiency and productivity but also enhance the overall satisfaction and effectiveness of their work.

How can we measure and quantify the user experience as well as the value from the AI system?

The measurement and quantification of user experience and the value derived from AI systems integrated into computational notebooks pose significant challenges. Traditional metrics or metric frameworks (e.g.,[8][20]) may not adequately capture the nuanced aspects of user interaction with AI tools, such as usability, efficiency, satisfaction, and impact on decision-making. Therefore, there is a need to develop comprehensive evaluation frameworks that encompass both quantitative and qualitative measures. These frameworks should consider factors such as user feedback, task performance metrics, user engagement, user trust, user confidence and business outcomes to provide a holistic assessment of the effectiveness and value of AI systems in computational notebooks.

How can we establish standardized guidelines, frameworks, and benchmarks for the design, implementation, and evaluation of AI tools in computational notebooks?

Establishing standards and best practices for AI tools in computational notebooks is essential to ensure their quality, consistency, and compatibility. This involves defining criteria for evaluating AI algorithms, models, and implementations, as well as guidelines for their integration into computational notebooks. Additionally, there is a need to develop frameworks for benchmarking AI tools against these standards and best practices to facilitate comparison and decision-making for data professionals. By establishing clear guidelines and benchmarks, stakeholders can make informed choices about the selection, deployment, and use of AI tools in computational notebooks, ultimately driving innovation and advancement in the field.

GENERATIVE AI DISCLOSURE

This paper utilized ChatGPT, an AI developed by OpenAI, to assist in modifying original content, such as shortening and copy editing. This disclosure ensures transparency regarding the use of AI tools in the research process and was informed by the ACM Publications Policy: Guidance for SIGCHI Venues [12].

REFERENCES

[1] Darren Allan. Forget Microsoft Copilot – an unofficial Clippy AI is available for Windows 11. techradar. June 28, 2023. Accessed February 22, 2024. https://www.techradar.com/computing/windows/forget-microsoft-copilot-an-unofficial-clippy-ai-is-available-for-windows-11

[2] Shraddha Barke, Michael B James, and Nadia Polikarpova. 2022. Grounded Copilot: How Programmers Interact with Code-Generating Models. arXiv preprint arXiv:2206.15000 (2022).

[3] Van Berkel, N., Skov, M.B. and Kjeldskov, J., 2021. Human-AI interaction: intermittent, continuous, and proactive. *Interactions*, *28*(6), pp.67-71.

[4] Christian Bird, Denae Ford, Thomas Zimmermann, Nicole Forsgren, Eirini Kalliamvakou, Travis Lowdermilk, and Idan Gazit. 2023. Taking Flight with Copilot: Early insights and opportunities of AI-powered pair-programming tools. Queue 20, 6, Pages 10 (November/December), 23 pages. https://doi.org/10.1145/3582083

[5] Souti Chattopadhyay, Ishita Prasad, Austin Z. Henley, Anita Sarma, and Titus Barik. 2020. What's Wrong with Computational Notebooks? Pain Points, Needs, and Design Opportunities. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20). Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376729

[6] Sarah Gibbons, Tarun Mugunthan and Jakob Nielsen: Accordion Editing and Apple Picking: Early Generative-AI User Behaviors. NN/g Nielsen Norman Group. Sept 24, 2023. Accessed Feb 28, 2024 Accordion Editing and Apple Picking: Early Generative-AI User Behaviors (nngroup.com)

[7] Isabelle Guyon, Kristin Bennett, Gavin Cawley, Hugo Jair Escalante, Sergio Escalera, Tin Kam Ho, Nuria Macia, Bisakha Ray, Mehreen Saeed, Alexandra Statnikov, Evelyne Viegas, "Design of the 2015 ChaLearn AutoML challenge," 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 2015, pp. 1-8, doi: 10.1109/IJCNN.2015.7280767

[8] Serena Hillman, Samira Jain, Vichita Jienjitlert, and Paula Bach. 2022. The BLUE Framework: Designing User-Centered In-Product Feedback for Large Scale Applications. In Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems (CHI EA '22). Association for Computing Machinery, New York, NY, USA, Article 21, 1–8. https://doi.org/10.1145/3491101.3503558

[9] Survey of Adult Skills Program for the International Assessment of Adult Competencies (PIAAC). (2012). Organization of Economic Co-operation and Development (OECD). Retrieved February 1, 2024, from https://www.oecd.org/skills/piaac/

[10] Dhanya Jayagopal, Justin Lubin, and Sarah E. Chasins. 2022. Exploring the Learnability of Program Synthesizers by Novice Programmers. In Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology (UIST '22). Association for Computing Machinery, New York, NY, USA, Article 64, 1–15. https://doi.org/10.1145/3526113.3545659

[11] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian E Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica B Hamrick, Jason Grout, Sylvain Corlay, and others. 2016. Jupyter Notebooks: A publishing format for reproducible computational workflows. In ELPUB. 87–90.

[12] Regan Mandryk. ACM Publications Policy: Guidance for SIGCHI Venues. ACM SIGCHI Official Blog Medium. https://medium.com/sigchi/acm-publications-policy-guidance-for-sigchi-venues-87332173aad1. Accessed February 14, 2024.

[13] Andrew M. Mcnutt, Chenglong Wang, Robert A Deline, and Steven M. Drucker. 2023. On the Design of AI-powered Code Assistants for Notebooks. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23). Association for Computing Machinery, New York, NY, USA, Article 434, 1–16. https://doi.org/10.1145/3544548.3580940

[14] Christian Meurisch, Cristina A. Mihale-Wilson, Adrian Hawlitschek, Florian Giger, Florian Müller, Oliver Hinz, and Max Mühlhäuser. 2020. Exploring User Expectations of Proactive AI Systems. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 4, 4, Article 146 (December 2020), 22 pages. https://doi.org/10.1145/3432193

[15] Allison Morrow. Microsoft could bring back Clippy, but make him, like, smart. CSS Business. January 25, 2023. Accessed February 22, 2024. https://www.cnn.com/2023/01/25/business/nightcap-chatgpt-microsoft/index.html

[16] Sunny Nagpaul, Microsoft CEO Satya Nadella says "AI is really in the air now" and is planning to train 2 million Gen Z in India with Tech Skills., Fortune.com. Accessed February 20, 2024. https://fortune.com/2024/02/07/microsoft-ai-india-2-million-training/

[17] Jakob Nielsen. AI: First New UI Paradigm in 60 Years. NN/g Nielsen Norman Group. June 18, 2023. Accessed February 22, 2024. https://www.nngroup.com/articles/ai-paradigm/

[18] Jakob Nielsen. The Articulation Barrier: Prompt-Driven AI UX Hurts Usability. UX Tigers. Jakob Nielsen's New UX Articles. May 24th, 2023. Accessed February 20th, 2024. https://www.uxtigers.com/post/ai-articulation-barrier

[19] M. Ragan-Kelley, F. Perez, B. Granger, T. Kluyver, P. Ivanov, J. Frederic, and M. Bussonnier. 2014. The Jupyter/IPython architecture: A unified view of computational research, from interactive exploration to communication and publication. AGU Fall Meeting Abstracts (Dec. 2014), H44D–07

[20] Kerry Rodden, Hilary Hutchinson, and Xin Fu. 2010. Measuring the user experience on a large scale: user-centered metrics for web applications. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10). Association for Computing Machinery, New York, NY, USA, 2395–2398. https://doi.org/10.1145/1753326.1753687

[21] Advait Sarkar, Andrew D. Gordon, Carina Negreanu, Christian Poelitz, Sruti Srinivasa Ragavan, and Ben Zorn. 2022. What is it like to program with artificial intelligence?. In Psychology of Programming Interest Group (PPIG 2022).

[22] Fernando Perez and Brian E Granger. 2015. Project Jupyter: Computational narratives as the engine of collaborative data science. Retrieved September 11, 207 (2015), 108.

[23] Jon Porter. Google's AI tools embrace the dream of Clippy. The Verge. May 14, 2023. Accessed February 22, 2024. https://www.theverge.com/2023/5/14/23721313/google-duet-ai-for-workspace-sidekick-clippy-generative-new-era-office-work

[24] Qian Yang, Aaron Steinfeld, and John Zimmerman. 2019. Unremarkable AI: Fitting Intelligent Decision Support into Critical, Clinical Decision-Making Processes. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19). Association for Computing Machinery, New York, NY, USA, Paper 238, 1–11. https://doi.org/10.1145/3290605.3300468

[25] Adam Rule, Aurelien Tabard, and James D. Hollan. Exploration and Explanation in Computational Notebooks. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. CHI18. https://doi.org/10.1145/3173574.3173606

[26] April Yi Wang, Dakuo Wang, Jaimie Drozdal, Michael Muller, Soya Park, Justin D. Weisz, Xuye Liu, Lingfei Wu, and Casey Dugan. 2022. Documentation Matters: Human-Centered AI System to Assist Data Science Code Documentation in Computational Notebooks. ACM Trans. Comput.-Hum. Interact. 29, 2, Article 17 (January 2022), 33 pages. https://doi.org/10.1145/348946

[27] Dakuo Wang, Justin D. Weisz, Michael Muller, Parikshit Ram, Werner Geyer, Casey Dugan, Yla Tausczik, Horst Samulowitz, and Alexander Gray. 2019. Human-AI Collaboration in Data Science: Exploring Data Scientists' Perceptions of Automated AI. Proc. ACM Hum.-Comput. Interact. 3, CSCW, Article 211 (November 2019), 24 pages. https://doi.org/10.1145/3359313

[28] Jason Weill. Generative AI in Jupyter. Jupyter Medium Blog. Accessed February 14th, 2024. https://blog.jupyter.org/generative-ai-in-jupyter-3f71