STM JOURNALS

Journal of
Software Engineering Tools & Technology Trends

ISSN: 2394-7292
Volume 11, Issue 1, 2024
January—April
DOI (Journal): 10.37591/JoSETTT

https://journals.stmjournals.com/josettt

Review JoSETTT

Impact of AI Tools in Software Engineering: Boon or a Bane

Bijee Lakshman[1,*], Abhinav S.[2]

Abstract

Artificial Intelligence (AI) has become a transformative force, revolutionizing diverse sectors by integrating intelligent systems into everyday processes. Natural Language Processing (NLP) plays a crucial role, enabling machines to understand and produce human language, marking a significant advancement in technology. This innovation has various applications, including chatbots, language translation, and sentiment analysis, thereby improving interactions between humans and computers and facilitating information processing. Generative AI, a subset of AI, takes innovation to new heights by enabling machines to autonomously create content. This advancement is particularly evident in language models that exhibit context-aware content generation, revolutionizing creativity in various fields such as writing and art. The synergy between NLP and generative AI has paved the way for unprecedented advancements, showcasing the potential for machines to understand and generate contextually relevant content. In the realm of AI tools, a critical player is Codeium, an open-source code editor. Designed with software developers in mind, Codeium incorporates AI-powered functionalities such as smart code suggestions and syntax highlighting. This study describes how Codeium significantly enhances the efficiency of developers, facilitating smoother code writing, editing, and debugging processes. A comparative analysis with few other AI tools is made highlighting few demos of prompt engineering in Codeium. As the collective impact of AI, NLP, generative AI, and advanced tools like Codeium, continues to unfold, these technologies not only redefine the boundaries of what is achievable but also underscore their pervasive influence across industries. From language understanding to innovative content creation and streamlined software development, the multifaceted applications of these technologies underscore their significance in shaping the future of artificial intelligence.

Keywords: Artificial Intelligence (AI), Natural Language Processing (NLP), Generative AI (Gen AI), Codeium, prompt engineering

***Author for Correspondence**
Bijee Lakshman
E-mail: bijee@wcc.edu.in

[1]Assistant Professor, Department of Data Science, Women's Christian College, Chennai, Tamil Nadu, India
[2]Student, Computer Science and Business Systems Engineering, SRM Institute of Science and Technology, Ramapuram, Chennai, Tamil Nadu, India

Received Date: February 26, 2024
Accepted Date: March 21, 2024
Published Date: April 05, 2024

Citation: Bijee Lakshman, Abhinav S. Impact of AI Tools in Software Engineering: Boon or a Bane. Journal of Software Engineering Tools & Technology Trends. 2024; 11(1): 14–23p.

INTRODUCTION

The field of software development is experiencing substantial changes due to the revolutionary impact of artificial intelligence (AI) across different domains of our professional endeavors. Among the various domains experiencing substantial productivity enhancements, code writing stands out prominently. The integration of AI in the form of code assistants has emerged as a game-changer, facilitating developers in crafting code with increased efficiency and safety. This study aims to delve into the realm of AI coding assistants, highlighting the

most commendable options available and assisting you in selecting the most suitable one tailored to your specific needs.

An AI coding assistant is a software innovation that utilizes artificial intelligence to improve the efficiency and accuracy of code development for programmers. This tool operates by generating code in real-time based on prompts or offering suggestions for auto-completion as developers actively engage in coding tasks. The capabilities of AI code assistants extend to various functionalities, including:

- Integration with Popular Code Editors: AI code assistants seamlessly integrate with commonly used code editors, such as Visual Studio Code.
- This guarantees a seamless and unified experience for developers who depend on particular coding environments.
- Generation of Complete Code Snippets: These assistants excel in generating comprehensive code snippets based on prompts or code comments. This feature streamlines the coding process, providing developers with ready-made solutions and accelerating their workflow.
- Real-time Auto-completion: AI coding assistants are adept at auto-completing lines of code as developers actively write, contributing to increased coding speed and accuracy. This immediate assistance notably enhances the overall efficiency of the coding process.
- Error Detection and Bug Identification: The AI-driven capabilities extend beyond code creation to include the detection of errors, bugs, and security vulnerabilities within the code. By actively identifying and flagging potential issues, these assistants contribute to the development of more robust and reliable software.
- Assistance in Code Commenting: AI code assistants play a valuable role in helping developers comment their code effectively.
- This facilitates the creation of documentation that is clear and serves as a reference for developers, fostering collaboration and future code maintenance.

A sophisticated deep learning algorithm known as a Large Language Model (LLM) plays a pivotal role in the realm of natural language processing (NLP) tasks. These advanced models utilize transformer architectures and undergo extensive training on diverse datasets, enabling them to excel in a wide array of language-related tasks, including language translation, prediction, and content generation.

Referred to interchangeably as neural networks (NNs), large language models draw inspiration from the complex structure of the human brain. The neural networks consist of layered nodes, mirroring the interconnected neurons in the human brain, facilitating intricate information processing and pattern recognition.

Beyond their proficiency in mastering human languages, these models showcase versatility by being trainable for various applications. Examples include understanding protein structures, coding software, and other specialized tasks. The training process involves two key phases: pre-training and fine-tuning.

The capabilities of these models in problem-solving span various tasks like text classification, question answering, document summarization, and text generation. These models are utilized across industries such as healthcare, finance, and entertainment. They offer solutions in tasks ranging from translation services to chatbot interactions and AI assistants. By seamlessly integrating into different domains, large language models contribute to advancements in technology and the development of innovative solutions that enhance communication and problem-solving capabilities in the digital landscape.

AI coding assistants represent a transformative force in the realm of software development, offering multifaceted support to developers and significantly elevating the efficiency and quality of the coding process. Github Copilot, Tabnine Pro, Replit Ghostwriter, Codeium are few AI coding assistants. In this study, Codeium is taken for discussion and explains the advantages of using same comparing to the other mentioned tools.

CODEIUM: UNLEASHING THE POWER OF AI IN CODING

Elevate the coding experience with Codeium, the modern coding superpower designed to accelerate our development workflow: Harnessing cutting-edge AI technology, Codeium offers a comprehensive toolkit that transforms the coding process into a seamless and efficient journey. From autocomplete to chat and search functionalities, Codeium is a great solution for supercharging your coding endeavors.

Key Features

- *Autocomplete Excellence:* Replacing the tedious task of manually completing code snippets via this AI tool, Codeium provides unlimited single and multi-line code completions, ensuring that it can effortlessly translate our ideas into code. Coding at the speed of thought with lightning-fast suggestions and state-of-the-art quality can be experienced with this tool.
- *IDE-Integrated Chat:* Codeium takes collaboration to the next level with its integrated chat feature within Visual Studio Code. No need to switch between applications; communication can be done directly with ChatGPT without leaving the coding environment. Codeium has the convenience of suggestions like Refactor and Explain, enhancing the coding conversations.
- *Multilingual Mastery:* Codeium supports over 70 programming languages, covering a vast array of options including Javascript, Python, Typescript, PHP, Java, C, C++, Rust, Ruby, and more. Codeium is a versatile companion for any coding challenge.
- *Seamless Integration:* Setting up Codeium is effortless, requiring a straightforward procedure that can be completed in under 2 min. Easily integrate Codeium into Visual Studio Code, allowing to focus on becoming the best software developer without getting bogged down by the intricacies of installation.
- *Supportive Community:* Discord Community is available for dedicated support and collaboration.
- Developers can engage with peers, exchange ideas, and remain informed about the latest advancements within the Codeium community.

The introduction provides a clear overview of the research objectives, scope, and key advancements being presented. It references relevant findings from previous studies. Both theoretical and experimental methods sections offer adequate details about the conducted research. The results and discussion section outlines the obtained results and their potential significance. In the discussion, the researcher has emphasized the impact of their findings in comparison to recent studies [2].

LITERATURE REVIEW

Researchers have discovered several areas in the field of Artificial Intelligence and the way it behaves. Lv made a comprehensive review of key technologies in the utilization and metaverse of generative AI that delve into the significant role of the metaverse in various domains like governance, scientific research and industry [3]. The review mentions that Core technologies like artificial intelligence, AR/VR, the Internet of Things, and blockchain are integrated for empowering the intelligent economy's high quality development by the metaverse. Venkatesh did a research agenda grounded in UTAUT on adoption and use of AI Tools [4].

The Unified Theory of Acceptance and Use of Technology (UTAUT) serves as a foundational framework for analyzing the factors impacting the adoption of technology. This study outlines the groundwork for exploring various dimensions, such as individual traits, technological attributes, environmental factors, and possible interventions. Saari *et al.* underscored the importance of enhancing awareness and understanding of AI within university teaching, emphasizing educators' crucial role in guiding students on the responsible use of AI tools [5]. In a separate study, Zawacki-Richter *et al.* conducted a thorough examination of research on the applications of artificial intelligence in higher education [6].

In this study the need for a more comprehensive exploration of ethical considerations and educational approaches emerges as a key takeaway. The application of AIEd in higher education necessitates a

nuanced understanding of its ethical implications and demands an alignment with pedagogical principles to ensure a harmonious integration into the educational landscape. Khurana *et al*. published a research paper on state of art, current trends and challenges in natural language processing [7].

The study delved into the understanding of key terminologies in NLP and NLG (Natural Language Generation), as well as explored the historical background, applications, and recent advancements in the NLP field. The researcher explores datasets, methodologies, and assessment criteria in NLP. In a recent article, Ekin offers a comprehensive guide on mastering prompt engineering techniques for optimal outcomes with ChatGPT [8]. The study emphasizes the impact of effective prompt engineering on ChatGPT's performance, outlines future research directions, and underscores the significance of fostering creativity and collaboration within the ChatGPT community. Additionally, Naveed *et al*. [9] present a comprehensive overview of Large Language Models, while Peng *et al*. [2] delve into the opportunities and challenges of LLMs in medical research and healthcare. They introduce GatorTronGPT, a generative clinical LLM trained using a GPT-3 architecture with up to 20 billion parameters, evaluating its utility for biomedical NLP and healthcare text generation. Furthermore, Collobert *et al*. proposed a flexible architecture capable of learning task-relevant representations without extensive task-specific feature engineering [10]. This approach, based on vast amounts of mostly unlabeled training data, leads to the development of a highly performant tagging system with minimal computational requirements. Finally, Meskó provides an overview of prompt engineering research, offering practical recommendations for healthcare professionals to enhance their interactions with LLMs [1].

METHODOLOGY

The objective of this study is to bring out the awareness of AI coding assistants/tools like Codeium which could understand and process the queries in a quick way thereby reducing the time of a software developer to complete the coding process and increase the productivity.

Codeium stands out as the contemporary coding superpower, representing a code acceleration toolkit that leverages state-of-the-art AI technology. This dynamic tool encompasses two primary features: Autocomplete and Search, both designed to streamline and enhance the coding experience. Autocomplete is a standout feature of Codeium, offering intelligent code suggestions that significantly reduce the time spent on various coding tasks. From generating boilerplate code to assisting with unit tests, autocomplete empowers developers to code more efficiently and with greater precision. By predicting the code intended to type, Codeium enables a seamless and productive coding process, that allows to focus on the creative aspects of software development rather than getting bogged down in time-consuming and routine tasks.

In addition to Autocomplete, Codeium boasts an impressive Search functionality. This feature transforms the way developers interact with their repositories by allowing them to pose natural language questions. Instead of manually sifting through lines of code, specific functionalities or snippets can be simply asked in Codeium about, and it will intelligently navigate through the codebase to provide relevant and accurate results. Not only does this conserve time, but it also boosts the overall accessibility and usability of the code repository.

One of Codeium's strengths is its easy integration into popular code editors, ensuring a seamless and hassle-free experience for developers. By integrating Codeium into our preferred coding environment, its capabilities can be harnessed without disrupting the workflow. Its advanced AI-driven features, including Autocomplete and Search, make coding more intuitive, efficient, and enjoyable. With Codeium, the focus shifts from mundane coding tasks to unleashing our creativity and innovation, allowing us to truly excel in the world of software development.

Codeium stands to be one of the four leading AI code assistant (Githib Copilot, Tabnine, Replit Ghostwriter and Codeium) along multiple axes, each of which provides a different perspective of the word "best".

A small comparative study for the above-mentioned AI code assistants is done here in this study with respect to the price, functionality, supported IDEs, supported languages, stated security and privacy policies as shown in Table 1.

Codeium boasts exceptional industry-leading latency and code autocomplete quality, rivaling renowned tools like GitHub Copilot. What sets Codeium apart is not only its outstanding performance but also its accessibility, as it is freely available across a wider range of integrated development environments (IDEs). In addition to its autocomplete prowess, Codeium offers an extended array of functionalities, including the innovative Codeium Search feature.

CodeiumAI stands out by seamlessly integrating into a wide range of development workflows, showcasing its impressive adaptability across diverse coding styles [8].

Table 1. Comparative analysis of popular AI coding assistant tools (codeium.com- internet source).

Tool	Github Copilot	Tabnine Pro	Replit Ghostwriter	Codeium
Price	$10/month or $100/year (free for students/open source contributors)	$12/month	$10/month (cycles equivalent)	Free
Functionality	Single + multi line codegen	Single + multi line codegen	Single + multi line codegen, Code explanation	Single + multi line codegen, In-IDE integrated chat and search
Supported IDEs	VSCode, Visual Studio, Vim/Neovim, JetBrains	VSCode, JetBrains, Neovim, Eclipse, Sublime Text	Replit only	VSCode, JetBrains, Visual Studio, Jupyter/Colab/Deepnote/Databricks Notebooks, Vim/Neovim, Emacs, Eclipse, Sublime Text, VSCode Web IDEs (ex. Gitpod), Chrome Extension
Supported Languages	C, C++, C#, Go, Java, JS, PHP, Python, Ruby, Scala, TS, more with potentially lower quality	C, C++, C#, CSS, Dart, Go, Haskell, Java, JS, Kotlin, Perl, PHP, Python, Ruby, Rust, Scala, TS	Bash, C, C++, C#, CSS, Go, HTML, Java, JS, PHP, Perl, Python, R, Ruby, Rust, SQL	Assembly, C, C++, C#, Clojure, CMake, CoffeeScript, CSS, CUDA, Dart, Delphi, Dockerfile, Elixir, F#, Go, Groovy, Haskell, HCL, HTML, Java, JavaScript, Julia, JSON, Kotlin, LISP, Less, Lua, Makefile, MATLAB, Objective-C, pbtxt, PHP, Protobuf, Python, Perl, Powershell, R, Ruby, Rust, Sass, Scala, SCSS, shell, Solidity, SQL, Starlark, Swift, Typescript, TSX, VBA, Vue, YAML, more with potentially lower quality
Stated Security and Privacy Policies	Opt-out for code snippet telemetry, Filter to reduce public code matches	Never train generative model on private code (unless for enterprise), SOC 2 compliance, No training on non-permissively licensed code	Unclear	Opt-out for code snippet telemetry, Never train generative model on private code, SOC 2 compliance, No training on non-permissively licensed code

The unique attribute of CodeiumAI ensures that developers can seamlessly integrate it into their existing workflows, boosting productivity without causing disruption to established practices. What distinguishes CodeiumAI is its adaptability, enabling it to adapt to diverse coding conventions and making it a versatile tool for teams with varying coding styles and preferences.

Unlike typical plugins, CodeiumAI's integration capabilities surpass conventional limits, seamlessly aligning with widely used development environments and collaborative platforms. Whether a project follows agile methodologies, employs continuous integration, or adheres to specific coding standards, CodeiumAI seamlessly integrates itself as an essential element of the workflow. This level of adaptability plays a crucial role in maintaining productivity and cohesion within development teams that operate with diverse practices [3].

In summary, CodeiumAI's commitment to adaptability and integration is a testament to its role as a valuable tool that not only enhances efficiency but also harmonizes with the dynamic and diverse landscape of modern development practices. Some demos in at least three languages like java, python and html were tried, using Visual Studio environment in Codeium. Following are the screenshots of the demos and its outputs as shown in Figures 1–8.

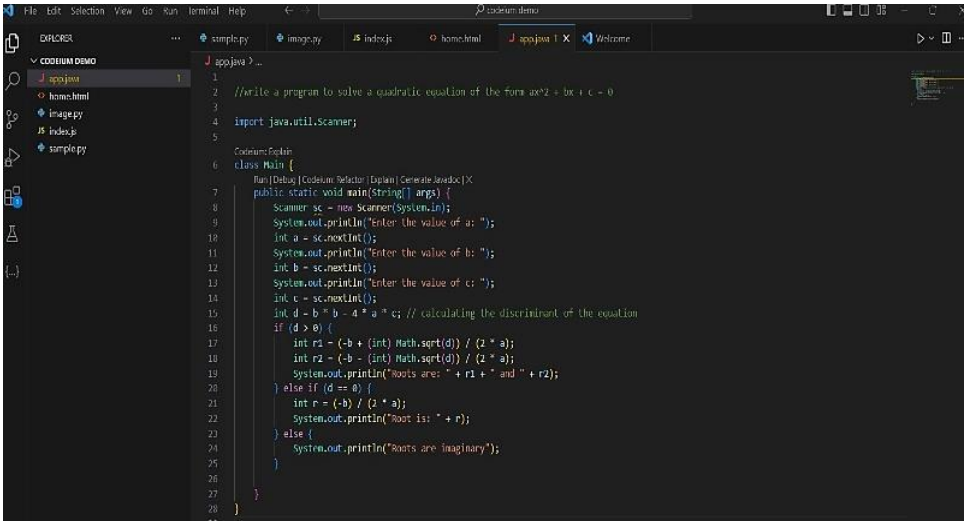

Figure 1. Demo 1-java program.

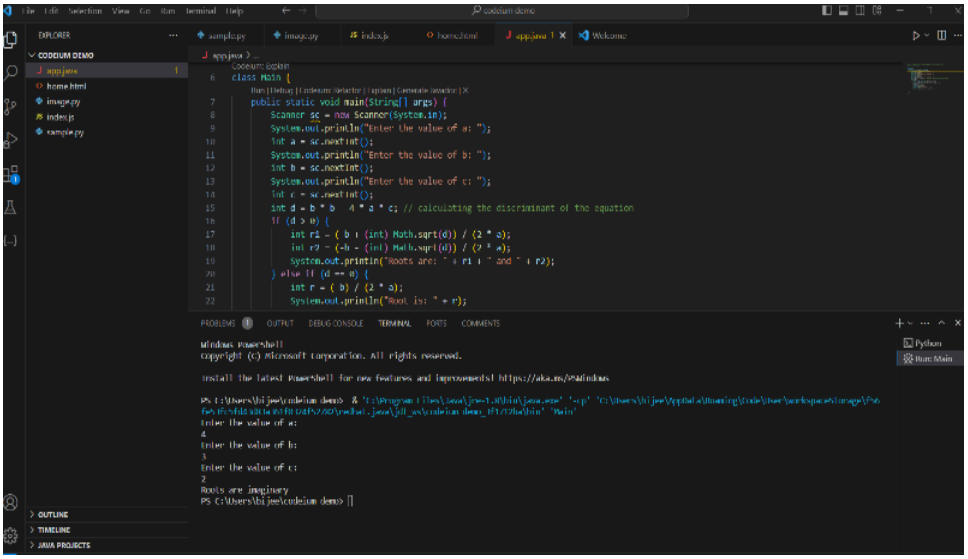

Figure 2. Output of demo 1.

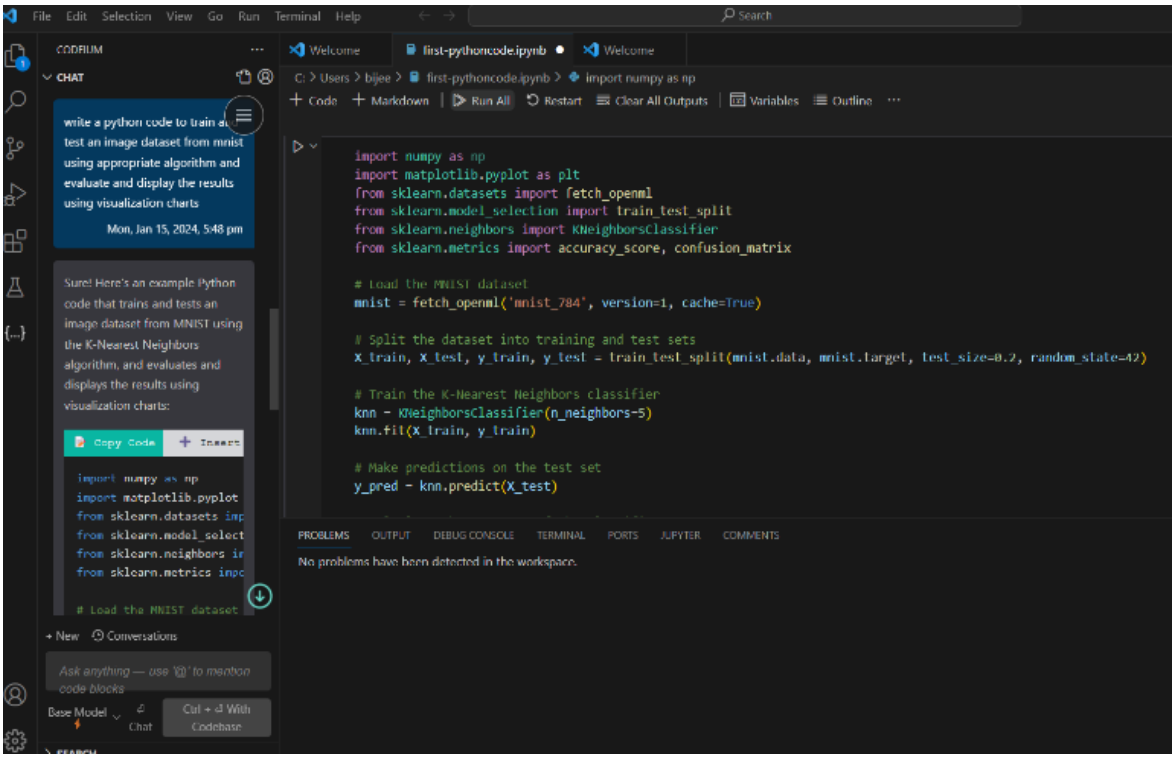

Figure 3. Demo 2: Python program.

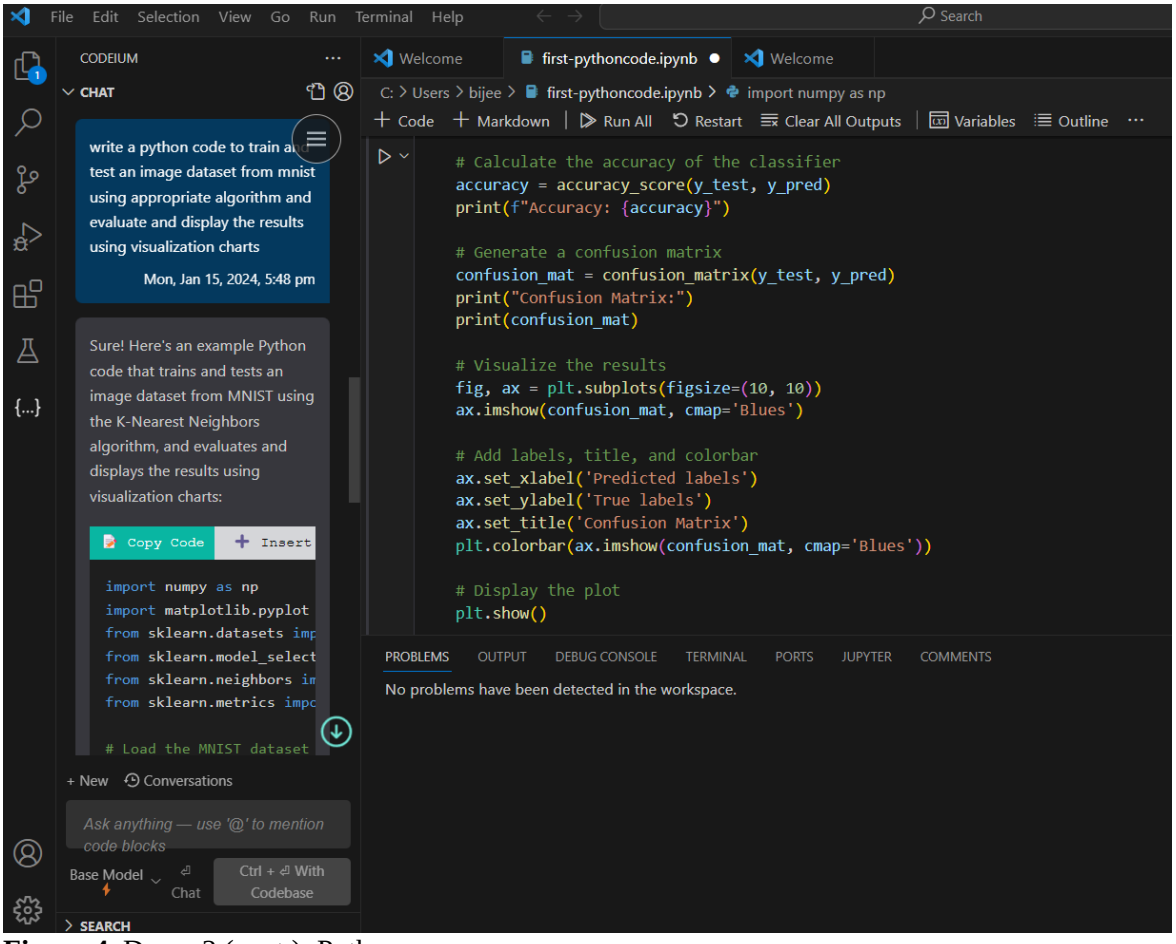

Figure 4. Demo 2 (cont.): Python program.

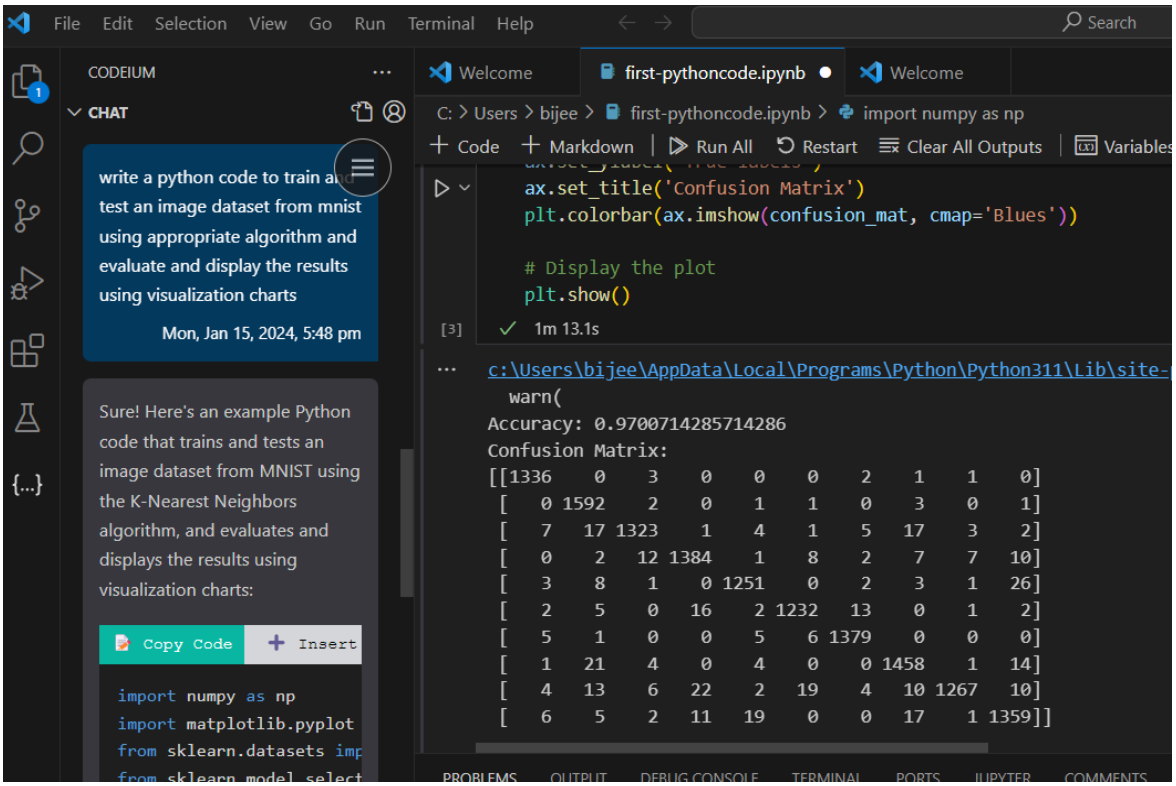

Figure 5. Output-1 of demo 2.

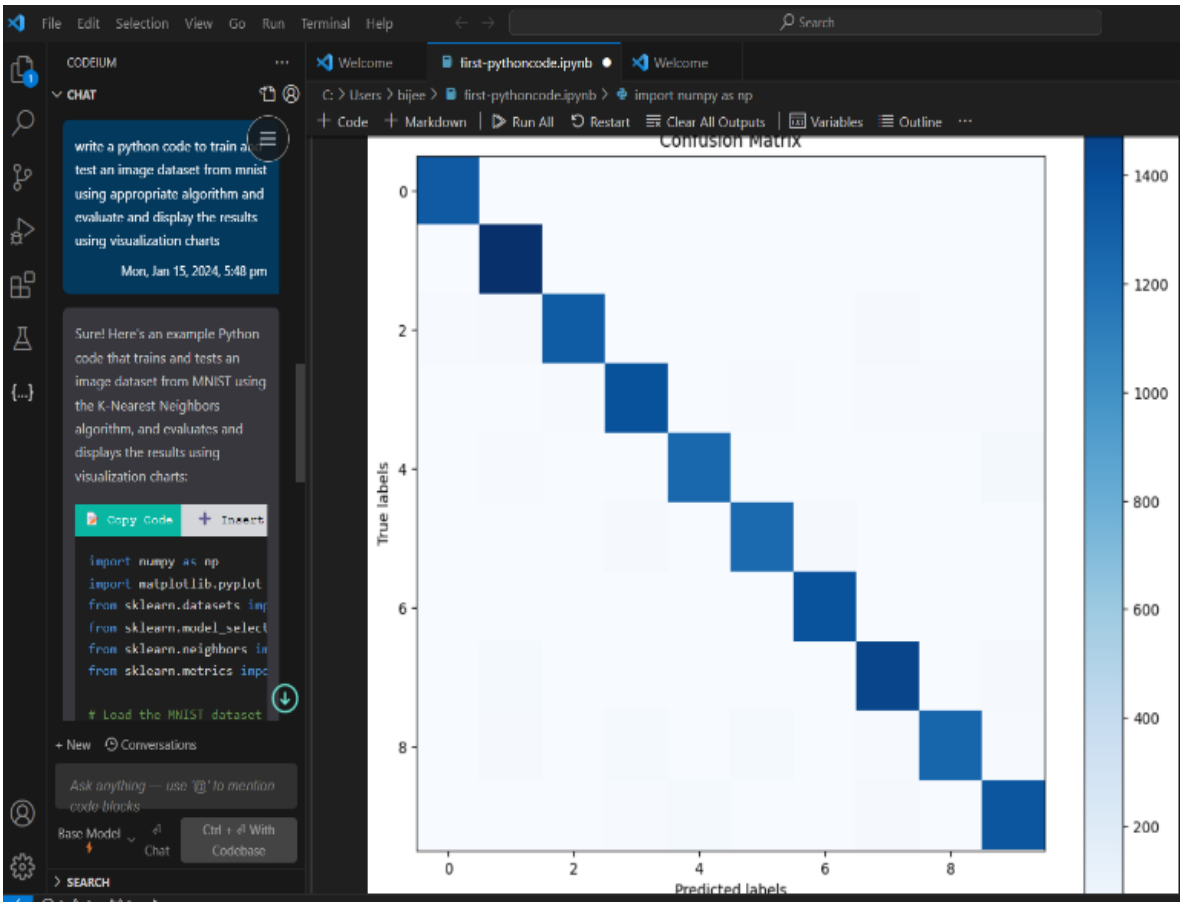

Figure 6. Output of demo 2.

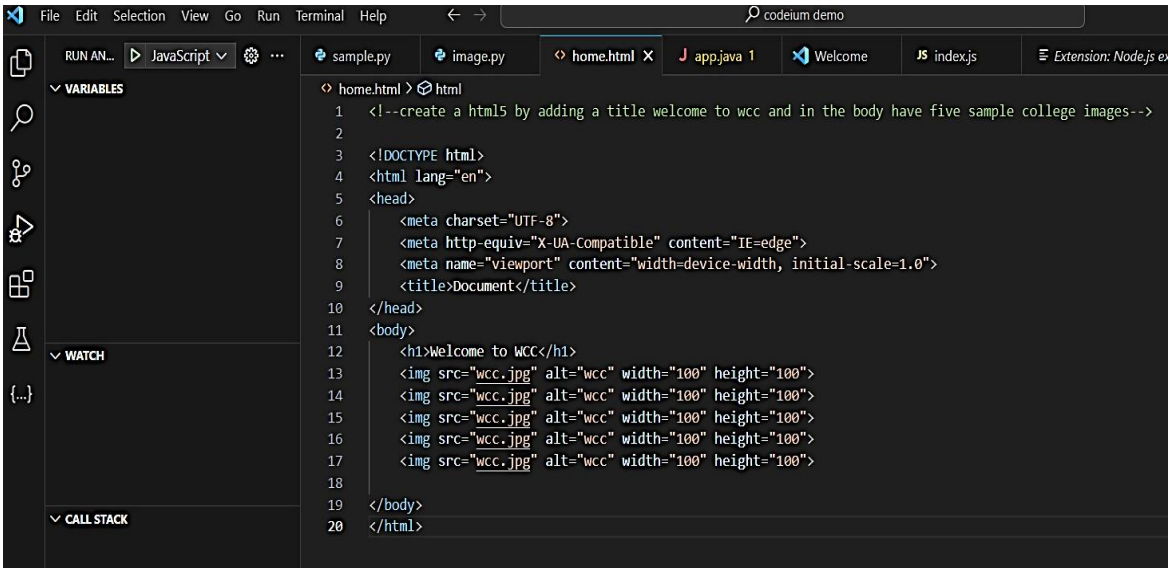

Figure 7. Demo 3 HTML program.

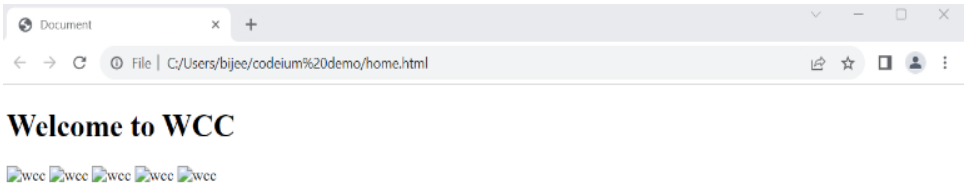

Figure 8. Output of demo 3.

CodeiumAI stands out for its remarkable flexibility, demonstrated by its ability to comprehend and offer pertinent suggestions irrespective of the coding style utilized. Developers, who may have distinct preferences for naming conventions, indentation styles, or commenting practices, benefit from CodeiumAI's tailored recommendations that seamlessly align with their individual choices. This adaptability not only guarantees a smooth integration into existing workflows but also underscores CodeiumAI's unwavering dedication to elevating the developer experience within diverse coding environments.

The tool's capability to understand and adapt to a variety of coding styles reflects a commitment to inclusivity and user-centric design. CodeiumAI aims to empower developers by providing a personalized and efficient coding experience that respects and accommodates their individual preferences. This emphasis on adaptability positions CodeiumAI as a versatile and indispensable companion for developers navigating the intricacies of different coding environments.

CONCLUSION

Our conviction in the transformative impact of Codeium extends beyond perceiving it as a mere tool; instead, we see it as a catalyst for change within the coding landscape. At the core of our mission lies a steadfast commitment to usability, functionality, and high quality, positioning Codeium as a beacon of excellence in the realm of code autocompletion. Codeium's transformative potential arises from its ability to redefine the coding experience.

We have carefully devised and constructed this tool to surpass industry norms, ensuring that it not only meets but surpasses expectations. Usability is a cornerstone of our approach, ensuring that developers, regardless of their expertise level, find Codeium intuitive and efficient to use.

Functionality is another pillar of our commitment. Codeium goes beyond conventional code autocompletion, offering a suite of features that empower developers to streamline their workflow, enhance productivity, and produce cleaner, more efficient code. We recognize the dynamic nature of coding requirements, and Codeium's multifaceted functionality is tailored to adapt and evolve with the ever-changing demands of the coding landscape.

High quality is the hallmark of Codeium. Our dedication to delivering a superior product is reflected in every aspect of its development, from the precision of its autocomplete suggestions, to the robustness of its underlying algorithms. It is understood that a tool is only as good as its ability to consistently meet and exceed user expectations, and Codeium embodies this philosophy by consistently delivering excellence.

In essence, Codeium is not just a tool; it is a testament to our commitment to revolutionize the coding experience. As a beacon of excellence, it stands poised to shape the future of code autocompletion by setting new standards and empowering developers to achieve greater heights in their coding endeavors.

In conclusion, while GitHub Copilot, TabNine, and Codeium share commonalities in providing advanced code autocompletion, each tool brings unique strengths to the table. Codeium distinguishes itself through its commitment to accessibility, extended functionality, and a philosophical approach that prioritizes both cutting-edge ML and developer feedback. The choice between these tools ultimately depends on individual preferences, coding requirements, and the specific features that align with the needs of developers and development teams. When relating to the title of this study, conclusion is that the impact of artificial intelligence tools is both boon and a bane as using such tools brings productivity for the organization (boon) but with less man power, as a result, there is going to be heavy retrenchment in almost every organization including educational institutions with the replacement of AI which is a bane. Hence, only people with updated AI skills can survive better in future.

REFERENCES

1. Bertalan Meskó. Prompt Engineering as an Important Emerging Skill for Medical Professionals: Tutorial. J Med Internet Res. 2023 Oct; 25: e50638.
2. Cheng Peng, Xi Yang, Aokun Chen, Smith Kaleb E, Nima Pour Nejatian, Costa Anthony B. A study of generative large language model for medical research and healthcare. NPJ Digit Med. 2023; 6(1): 210.
3. Zhihan Lv. Generative artificial intelligence in the metaverse era. Cognitive Robotics. 2023; 3: 208–217.
4. Venkatesh V. A Research Agenda grounded in UTAUT- Adoption and use of AI Tools. Ann Oper Res. 2022 Jan; 308(1): 641–652. https://doi.org/10.1007/s10479-020-03918-9.
5. Mika Saari, Petri Rantanen, Mikko Nurminen, Terhi Kilamo, Kari Systä, Pekka Abrahamsson. Survey of AI Tool Usage in Programming Course: Early Observations. Agile Processes in Software Engineering and Extreme Programming – Workshops. 2023 Dec; 182–191.
6. Olaf Zawacki-Richter, Marín Victoria I, Melissa Bond, Franziska Gouverneur. Systematic review of research on artificial intelligence applications in higher education – where are the educators? Int J Educ Technol High Educ. 2019; 16(1): 39.
7. Diksha Khurana, Aditya Koli, Kiran Khatter, Sukhdev Singh. Natural Language Processing- State of art, current trends and challenges. Multimed Tools Appl. 2023; 82(3): 3713–3744. doi: 10.1007/s11042-022-13428-. Epub 2022 Jul 14.
8. Sabit Ekin. Prompt Engineering For ChatGPT: A Quick Guide To Techniques, Tips, And Best Practices. TechRxiv. May 04, 2023. DOI: 10.36227/techrxiv.22683919.v2.
9. Naveed H, Khan AU, Qiu S, Saqib M, Anwar S, Usman M, Barnes N, Mian A. A comprehensive overview of large language models. arXiv preprint arXiv:2307.06435. 2023 Jul 12. doi: https://doi.org/10.48550/arXiv.2307.06435.
10. Collobert R, Weston J, Com J, Karlen M, Kavukcuoglu K, Kuksa P. Natural Language Processing (Almost) from Scratch. J Mach Learn Res. 2011;12:2493–2537.

