Human to Document, AI to Code: Comparing GenAI for Notebook Competitions

Tasha Settewong*, Youmei Fan*, Raula Gaikovina Kula†, Kenichi Matsumoto*
*Nara Institute of Science and Technology, Japan
{tasha.settewong.ts1, fan.youmei.fs2, matumoto}@is.naist.jp
†The University of Osaka, Japan
raula-k@ist.osaka-u.ac.jp

Abstract—Computational notebooks have become the preferred tool of choice for data scientists and practitioners to perform analyses and share results. Notebooks uniquely combine scripts with documentation. With the emergence of generative AI (GenAI) technologies, it is increasingly important, especially in competitive settings, to distinguish the characteristics of human-written versus GenAI. Our new idea is to explore the strengths of both humans and GenAI through the coding and documenting activities in notebooks. We first characterize differences between 25 code and documentation features in human-written, medal-winning Kaggle notebooks. We find that gold medalists are primarily distinguished by longer and more detailed documentation. Second, we analyze the distinctions between human-written and GenAI notebooks. Our results show that while GenAI notebooks tend to achieve higher code quality (as measured by metrics like code smells and technical debt), human-written notebooks display greater structural diversity, complexity, and innovative approaches to problem-solving. Based on these early results, we highlight four agendas to further investigate how GenAI could be utilized in notebooks that maximize the potential collaboration between human and GenAI tech.

Index Terms—Empirical Study, Notebooks, GenAI Code

I. INTRODUCTION

Computational notebooks have rapidly become an important tool for data scientists and researchers. According to Jupyter, a notebook is a shareable document that combines code, text, data, and rich visualizations, offering an interactive environment for prototyping, data exploration, and sharing ideas, so that users can learn coding and documentation[1]. Platforms like Kaggle have further accelerated the adoption of notebooks, offering a space for machine learning competitions that attract both industry practitioners and academic researchers [14]. As Kaggle[2] competitions become more popular and prestigious, participants face increasingly tougher competition to create the best, most insightful notebooks.

The landscape of coding has been transformed by the emergence of generative AI (GenAI) technologies such as ChatGPT[3], Gemini[4] Claude[5] and open models from Meta Ollama[6] and Huggingface[7]. These tools have the potential to assist practitioners by generating code, writing explanations, and even producing entire notebooks with minimal human intervention. However, this technological shift raises important questions: How can we distinguish between notebooks created by humans and those generated by GenAI? More importantly, what can we learn from each to advance the state of the art in computational notebooks? While prior work has examined code quality [5], reproducibility [15], and collaboration [21] in notebooks, the specific contributions and limitations of GenAI notebooks remain underexplored. Competitions provide an ideal experimental setting, as participants are incentivized to submit their highest-quality work for rigorous community assessment and ranking.

To investigate this phenomena, in this paper, we perform experiments on three case studies to compare human-written notebooks and those generated by leading large language models (LLMs). The case studies is across three major Kaggle competitions (i.e., 1. Santander Customer Transaction Prediction, 2. Home Credit Default Risk, and 3. IEEE-CIS Fraud Detection). To differentiate between human-written notebooks, we use the heuristic of gold medals (i.e., notebooks voted as being high quality by the community) to identify quality notebooks[8]. By extracting and analyzing 25 features related to documentation and code quality from a curated dataset of 465 human-authored notebooks and 9 GenAI notebooks, we answer two research questions:

RQ1: <u>What are differences between documentation and code for human-written notebook?</u>
The motivation behind RQ1 is to characterize the fundamental code and documentation patterns in human-written notebooks, revealing essential programming behaviors, documentation styles, and structural approaches that differentiate gold-medal human notebooks from the rest, establishing a crucial baseline.

RQ2: <u>What are the key differences between GenAI and human-written data science notebooks?</u>
The motivation behind RQ2 is to identify the distinguishing factors that separate human and GenAI approaches in competitive data science, focusing on the impact of code quality metrics versus documentation on medal-worthy outcomes.

Our findings show that GenAI notebooks achieve higher code quality, with significantly fewer code smells, technical

---

[1] https://docs.jupyter.org/en/latest/

[2] https://www.kaggle.com/

[3] https://chatgpt.com/

[4] https://gemini.google.com/

[5] https://claude.ai/

[6] https://ollama.com/

[7] https://huggingface.co/

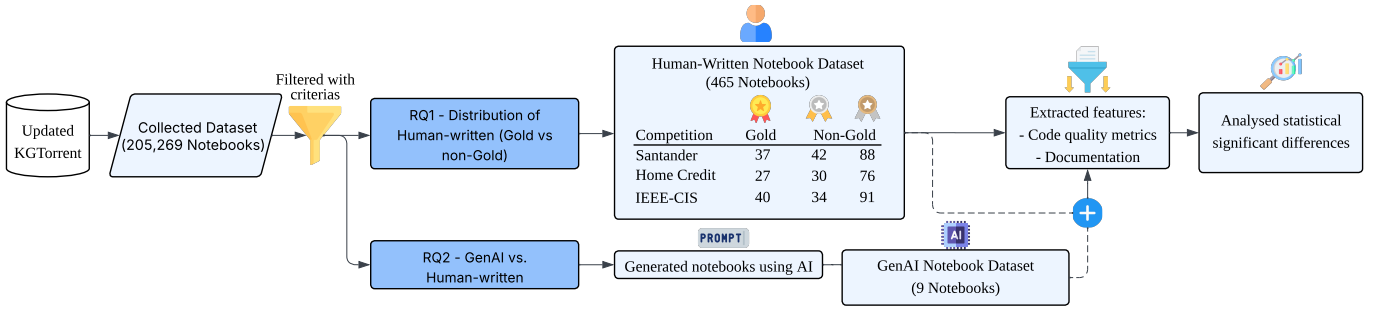[8] https://www.kaggle.com/progression/

Fig. 1: Overview of Data Collection

debt, and coding violations. However, human-written notebooks stand out for their more comprehensive and accessible documentation. In particular, gold medal-winning notebooks feature more than twice as much markdown content and narrative explanation as the non-gold group. Moreover, while GenAI documentation tends to require a higher reading level, human-written explanations are generally clearer and more approachable.

II. THREE CASE STUDIES OF NOTEBOOK COMPETITIONS

Three closed competitions were selected for this study: Santander Customer Transaction Prediction[9], Home Credit Default Risk[10], and IEEE-CIS Fraud Detection[11]. The Santander Customer Transaction Prediction competition contributed 167 medal-winning notebooks to the dataset used in this analysis. From the Home Credit Default Risk competition, 133 notebooks were analyzed. The IEEE-CIS Fraud Detection competition provided the final 165 notebooks for the study.

To utilize the KGTorrent dataset effectively for GenAI notebook generation, we conducted a comprehensive investigation of the data structure. This analysis revealed that GenAI models require well-defined problem contexts to generate meaningful notebooks, necessitating the selection of specific Kaggle competitions that provide clear problem statements and structured datasets.

The selection of these three competitions was based on the criterion that those with the highest number of participating teams were prioritized to maximize the sample size and ensure a robust statistical analysis.

III. STUDY DESIGN

Figure 1 presents the workflow of the data collection process. We utilized the KGTorrent [17] tool to retrieve Jupyter notebooks. KGTorrent constitutes a comprehensive dataset encompassing computational notebooks (code kernels) and their associated metadata sourced from the Kaggle platform.

To obtain an updated version of the KGTorrent dataset, Kaggle metadata[12] was downloaded on May 25th, 2025, and

[9]https://www.kaggle.com/competitions/santander-customer-transaction-prediction
[10]https://www.kaggle.com/competitions/home-credit-default-risk
[11]https://www.kaggle.com/competitions/ieee-fraud-detection
[12]https://www.kaggle.com/datasets/kaggle/meta-kaggle

TABLE I: Medal Distribution Across Competitions

Medal Type	Home Credit	Santander	IEEE-CIS
Bronze	76	88	91
Silver	30	42	34
Gold	27	37	40
Total	**133**	**167**	**165**

Jupyter notebooks were subsequently retrieved through API requests using the KGTorrent tool. This process yielded a total of 205,269 notebooks. The data collection process consists of four main stages: first, we identify target competitions for analysis; second, we apply data filtering criteria to select relevant notebooks; third, we generate corresponding GenAI notebooks using GenAI models; and finally, we extract features from both human-written and GenAI notebooks for comparative analysis.

A. Data Collection and Filtering

The filtering process yielded a total of 465 notebooks, with the distribution of medal achievements across the three competitions presented in Table I. In this study, the notebook medals (i.e., Bronze, Silver, Gold) serve as proxies for computational notebook quality, as medal progression and achievement represent community recognition and appreciation of the work's merit. Following competition selection, a systematic filtering process was implemented to ensure dataset integrity and completeness using the following criteria:

1) Both the notebook's metadata and corresponding notebook files must be present.
2) Contributor information for each notebook creator must be available.
3) The notebook must have been submitted to one of the three selected competitions.
4) The notebook must have received a medal recognition.

B. Generating notebook using GenAI

To facilitate comparison between human-written and GenAI notebooks, three large language models (LLMs) were selected: (1) GPT-4.1, (2) Llama 4 Maverick, and (3) Gemini 2.5 Pro Review. These models were chosen from different companies (OpenAI, Meta, and Google) to minimize potential bias and ensure diverse representation. A single, non-iterative prompt

TABLE II: Statistical Comparisons of Gold and Non-Gold Human-Written Notebooks

Feature	Home-Credit		Santander		IEEE-CIS	
	p-value	η^2	p-value	η^2	p-value	η^2
#Markdown Char	0.0005***	0.085	0.0004***	0.070	–	–
#Markdown Line	0.001**	0.074	0.0004***	0.071	–	–
#Sentences	0.001**	0.073	0.0004***	0.071	–	–
#Markdown Cell	0.001**	0.070	0.001***	0.056	–	–
Avg Sentence	0.013*	0.040	–	–	–	–
Duplicated Lines	0.021*	0.033	0.020*	0.026	0.004**	0.046
Duplicated Blocks	–	–	0.009**	0.035	0.004**	0.045
#Code Char	–	–	–	–	0.011*	0.033
#Visual	–	–	–	–	0.019*	0.027

* p-value < 0.05; ** p-value < 0.01; and *** p-value < 0.001. The effect sizes with thresholds are highlighted in Negligible   Small   Medium   Large .

TABLE III: Gold vs Non-Gold Feature Statistics Comparisons

Feature	Medal Type	Mean	Median	Std	Min	Max
#Markdown Char	Non-Gold	2688.10	934.0	6526.72	0	94377
	Gold	5369.35	2261.0	8071.63	0	45281
#Markdown Line	Non-Gold	35.02	16.0	67.15	0	748
	Gold	64.98	35.5	79.04	0	364
#Sentences	Non-Gold	27.39	12.0	44.14	0	308
	Gold	56.46	26.5	75.81	0	364
#Markdown Cell	Non-Gold	11.71	6.0	18.57	0	184
	Gold	21.12	12.0	27.64	0	160

was used to establish a controlled baseline for the models' raw generative capabilities. This approach avoids introducing prompt engineering as a variable, which could create bias as different GenAI respond optimally to different prompting strategies.

> *"generate .ipynb file in JSON format for Kaggle competition the overview is {comepetition_overview}. {evaluation_method}. Using these csv files; {competition_dataset}"*
>
> – *Used prompt*

After .ipynb files in JSON format were generated by the GenAI models, the JSON format files were converted to standard Jupyter Notebook format and manually checked for file integrity and structural validity to ensure proper formatting. As part of a post-review validation to assess practical runnability, a basic execution is performed to check on all nine GenAI notebooks.

C. Extracting features

Individual features were extracted, resulting in a total of 25 features:

(1) Documentation-related attributes (#Markdown Char, #Markdown Line, #Markdown Cell, #Sentence, Avg Sentence, Gunning Fog, #Visual, and Comment Lines*).

(2) Code quality-related attributes (#Code Char, #Code Line, #Code Cell, Cyclomatic Complexity*, Cognitive Complexity*, Functions*, Statements*, Duplicated Blocks*, Duplicated Lines*, Bugs*, Violations*, Code Smells*, Technical Debts*, Maintainability Rating*, Reliability Rating*, and Security Rating*)

This extraction process utilized the SonarQube[13] tool, an open-source static analysis platform designed for continuous code quality monitoring that integrates analysis and reporting throughout the software development process [2], [9], [13] and is used to extract some features(*) through API.

IV. EMPIRICAL STUDY

We now discuss the analysis method and results of the study.

A. Answering RQ1

To address RQ1, we assessed data normality through the Shapiro-Wilk test [19] and analyzed statistically significant differences between notebook features and medal achievement using the Kruskal-Wallis H test [8] with effect size calculations using the human-written notebooks dataset illustrated in Figure 1.

Table II shows that gold medalists primarily differentiate themselves through documentation practices rather than coding

[13]https://sonarcloud.io/explore/projects

metrics. Documentation features demonstrated the strongest discriminating power between gold and non-gold medalists, with markdown characters, lines, and sentence counts showing highly significant differences (p-value ≤ 0.001) and medium effect sizes (0.056-0.085).

The distribution analyses in Table III confirm that gold medalists consistently produce substantially longer documentation than non-gold medalists. In contrast, code-related metrics showed weaker differentiation patterns. The IEEE-CIS competition displayed significance only in specific elements like duplicated lines and code character quantity, but with small effect sizes (0.033-0.046), indicating limited practical impact on distinguishing performance levels. These findings indicate that documentation, rather than code quality, primarily separates gold medalists from other competitors. Gold medalists systematically generate notebooks with more extensive markdown content.

> ### RQ1 Summary
>
> Higher quality human notebooks tend to have longer documentation. Gold medal worthy notebooks are distinguishable by having longer documentation (i.e., markdown characters, lines, and sentences).

B. Answering RQ2

To investigate distinguishing characteristics between GenAI and human-written data science notebooks, we conducted statistical analyses using the Kruskal-Wallis H test [8] across the human-written notebooks dataset and the GenAI notebook dataset, as illustrated in Figure 1, focusing on medium (≥ 0.06) to large (≥ 0.14) effect sizes that indicate practically meaningful differences.

Documentation Accessibility: Table V reveals that human-written notebooks consistently produced more accessible documentation despite GenAI's code quality advantages. The statistical comparison shows GenAI explanatory text in the IEEE-CIS competition consistently operated at senior-college-level complexity (Gunning Fog score: 16.07), compared to human documentation which operated at high-school-senior level (Gunning Fog score: 10.32). Statistical analysis in Table IV confirmed these readability differences across multiple competitions, with medium effect sizes observed for both gold and silver medal comparisons in IEEE-CIS (p-value ≤ 0.05, $\eta^2 \geq 0.088$), while Santander showed similar patterns (p-value = 0.017, $\eta^2 = 0.124$). Furthermore, GenAI notebooks showed significantly fewer comment lines compared to human-written notebooks in the Home-Credit competition (p-value = 0.035, $\eta^2 = 0.122$, medium effect size), indicating reduced inline documentation coverage alongside the increased complexity of explanatory text.

Code Quality Features: Conversely, Table IV shows that GenAI notebooks demonstrated superior performance in technical code quality measures. In the IEEE-CIS competition, GenAI notebooks significantly outperformed human-written notebooks across several code quality indicators with large

effect sizes (≥ 0.191): code smells (p-value = 0.004, η^2 = 0.209), technical debt (p-value = 0.004, η^2 = 0.209), and violations (p-value = 0.006, η^2 = 0.191). Similarly, the Home-Credit competition showed GenAI notebooks exhibiting reduced structural complexity compared to medal-winning submissions, with large effect sizes observed for functions (GenAI mean: 4.00 vs Human mean: 5.71, p-value = 0.011, $\eta^2 = 0.197$) and statements (GenAI mean: 144.33 vs Human mean: 194.38, p-value = 0.025, $\eta^2 = 0.143$).Additionally, medium effect sizes were found for the number of code cells (GenAI mean: 13.00 vs Human mean: 33.65, p-value = 0.049, $\eta^2 = 0.103$) and cyclomatic complexity (GenAI mean: 16.00 vs Human mean: 18.53, p-value ≤ 0.050, $\eta^2 \geq 0.102$), indicating GenAI generates more streamlined code organization with lower algorithmic complexity.

Execution and Reproducibility: Of the nine notebooks executed, there were three that failed to complete the execution. These execution outcomes confirm that static analysis alone is insufficient, as it misses critical data-handling and logic errors that limit the notebooks' practical utility. The llama_home-credit and llama_IEEE notebooks failed due to runtime errors from passing non-numeric columns to the model. The GPT_IEEE notebook failed because column names were not consistently unified, leading to key mismatches during data processing.

The analysis demonstrates that GenAI and human notebooks exhibit distinct but complementary strengths: GenAI excels in producing clean, standardized code with superior adherence to best practices, while humans demonstrate greater structural innovation and complexity. Despite GenAI's advantages in code quality metrics, including lower technical debt, fewer code smells, and reduced violations, medal-worthy performance appears to depend more on human strengths such as creative problem-solving, domain expertise, and algorithmic innovation rather than code cleanliness alone.

> ### RQ2 Summary
>
> Human and GenAI notebooks have statistical differences. We identify and summarize key differences below:
>
> - *Documentation features* - GenAI notebooks are harder to read (higher reading levels), while the human written documentation is easier to read (gunning fog)
> - *Code features* - GenAI notebooks demonstrate higher code quality with fewer code smells, lower technical debt, and reduced violations. Meanwhile, human-written code is more complex, with greater structural complexity and higher function counts and statements.

V. DISCUSSION

In this section, we present a discussion of the implications and the research agenda from the three case studies.

TABLE IV: Statistical Comparisons of GenAI and Each Medal Notebooks

Feature	Medal	Home-Credit		Santander		IEEE-CIS	
		p-value	η^2	p-value	η^2	p-value	η^2
Functions	Gold	**0.011***	0.197	–	–	**0.041***	0.078
	Silver	**0.009****	0.191	–	–	–	–
	Bronze	**0.023***	0.054	–	–	–	–
Statements	Gold	**0.025***	0.143	–	–	–	–
	Silver	**0.018***	0.148	–	–	–	–
	Bronze	–	–	–	–	–	–
Comment Lines	Gold	**0.035***	0.122	–	–	–	–
	Silver	–	–	–	–	–	–
	Bronze	–	–	–	–	–	–
#Code Cell	Gold	**0.049***	0.103	–	–	**0.025***	0.098
	Silver	–	–	–	–	–	–
	Bronze	–	–	–	–	–	–
Cyclomatic Complexity	Gold	**0.050***	0.102	–	–	–	–
	Silver	**0.028***	0.124	–	–	–	–
	Bronze	–	–	–	–	**0.043***	0.034
Gunning Fog	Gold	–	–	**0.017***	0.124	**0.032***	0.088
	Silver	–	–	–	–	**0.030***	0.107
	Bronze	–	–	**0.024***	0.046	**0.049***	0.031
Code Smells	Gold	–	–	–	–	–	–
	Silver	–	–	–	–	**0.004****	0.209
	Bronze	–	–	–	–	–	–
Technical Debts	Gold	–	–	–	–	–	–
	Silver	–	–	–	–	**0.004****	0.209
	Bronze	–	–	–	–	–	–
Violations	Gold	–	–	–	–	–	–
	Silver	–	–	–	–	**0.006****	0.191
	Bronze	–	–	–	–	–	–

A. Implications

This study provides a comparative analysis of human-written and the raw output of GenAI without human intervention data science notebooks in competitive environments for an understanding of the baseline capabilities and inherent strengths and weaknesses of both approaches. Our findings reveal distinct complementary strengths between Human and GenAI approaches that suggest strategic applications in data science practice. As a proof of concept, there are differences, and we believe that a more comprehensive study will highlight more differences between the Human and AI. Although preliminary, the result from RQ2 demonstrates that GenAI notebooks achieved significantly superior code quality metrics features (code smells, technical debt, and violations). These results indicate that GenAI tools could serve as valuable automated code review and standardization tools.

The results from RQ2 also reveal a limitation: GenAI documentation operated at senior-college reading levels (Gunning Fog: 16.07) compared to human high-school-senior levels (10.32). Combined with our RQ1 findings showing that documentation features had the strongest discriminating power between gold and non-gold medalists, this suggests that current GenAI limitations in accessible documentation may undermine competitive performance despite superior code quality. In this study, we use the gold medal, but other heuristics of quality could be employed in future studies. Hybrid approaches, where GenAI assists in routine code generation and quality assurance while humans focus on documentation and complex problem-solving, may offer the best results. For immediate research directions, we will need to experiment with a larger and more diverse dataset that needs to be collected. In this study, we only provided 3 competitions, so we need more competitions. We would also need to interview developer and prototype different tools that simulate the human to GenAI collaboration. Would a competitor or the competition host be able to tell the difference between a human and GenAI solution.

The implications from the study at this stage are threefold. The first is that for hosts of the competitions, there are

TABLE V: Human vs GenAI Feature Statistics Comparison in IEEE-CIS Competition

Feature	Source	Mean	Median	Min	Max
Functions	Human	3.16	2.0	0	46
	AI	0.67	0.0	0	2
Statements	Human	121.49	91.5	10	586
	AI	94.33	35.0	25	223
Comment Lines	Human	40.93	23.5	0	252
	AI	32.67	18.0	13	67
#Code Cell	Human	31.36	22.0	2	205
	AI	9.00	8.0	7	12
Cyclomatic Complexity	Human	14.30	9.5	0	126
	AI	14.67	0.0	0	44
Gunning Fog	Human	**10.32**	9.91	0	64.61
	AI	16.07	16.13	11.55	20.53
Code Smells	Human	29.25	0.0	0	3166
	AI	**3.00**	3.0	3	3
Technical Debts	Human	44.39	0.0	0	3228
	AI	**15.00**	15.0	15	15
Violations	Human	29.59	0.0	0	3167
	AI	**3.00**	3.0	3	3

Color coding indicates superior performance: **AI outperforms** vs. **Human outperforms**.

distinguishable features of GenAI usage, thus could be used to detect when GenAI could be used when not permitted. For competitors, these results provide insights into how to best use the GenAI-assistants in their toolbox. For researchers, the results may imply what are the skills necessary and cannot be amplified by use of GenAI.

This study provides a foundation for further exploration of the intersections between GenAI-assisted development, human creativity, and competitive data science performance. RQ1 and RQ2 highlight the need to balance code quality with documentation comprehensiveness in evaluating excellence. To build on these insights, we develop the following agenda as a roadmap:

- **Agenda One. Explore notebooks as an IDE for Human-GenAI documentation and code collaboration** Investigate whether computational notebooks are truly the optimal environment for human–GenAI collaboration, or if alternative development contexts might better support creative and effective hybrid work.
- **Agenda Two. Conduct a comprehensive study** Expand analysis to larger and more diverse datasets, encompassing different competitions and platforms beyond Kaggle, to validate whether findings such as the centrality of documentation quality hold across domains.
- **Agenda Three. Expand from Notebooks to Software Projects** Extend the study to traditional software artifacts, including source code repositories and developer documentation, to determine whether the trade-offs between code quality and documentation clarity persist in broader

software engineering settings. Additional questions that could arise as secondary research topics could explore how different AI documentation is from human documentation in code comments, documentation artifacts and in discussions with other team members.
- **Agenda Four. Explore challenges and risks of using GenAI** Examine the ethical and procedural challenges introduced by GenAI, particularly the risks of GenAI-augmented cheating and violations of competition rules. Future research should develop guidelines and detection mechanisms to ensure fairness and maintain trust in both competitive and collaborative environments.

VI. THREATS TO VALIDITY

This section examines potential threats to the validity and measures implemented.

A. Internal Validity

Three primary threats to internal validity require consideration.

- First, selection bias affects both RQ1 and RQ2 as our dataset comprises only medal-winning notebooks from three Kaggle competitions (Santander Customer Transaction Prediction, Home Credit Default Risk, and IEEE-CIS Fraud Detection), which may not represent broader data science practices. This could bias our Kruskal-Wallis H test results comparing gold vs non-gold medalists and our human vs GenAI statistical comparisons. We address this by analyzing 465 notebooks across multiple competition domains.

- Second, the potential contamination of our human-written dataset with AI-generated content in both RQ1 and RQ2, as notebooks for older competitions may have been created recently. While the purity of this dataset cannot be guaranteed. However, this concern is mitigated by the large, observed gap in code quality between the human and GenAI groups, which suggests the comparison remains meaningful.
- Three, confounding variables like participant experience may simultaneously influence documentation practices and competition success in RQ1, while in RQ2, human notebooks represent experienced medal winners compared against GenAI lacking domain expertise. This could potentially confound our comparisons of code quality metrics and readability measures like Gunning Fog scores. The use of objective, quantifiable metrics helps to mitigate this concern.

B. External Validity

Three threats to external validity merit discussion.

- Generalizability beyond Kaggle competitions may be limited. Our RQ1 findings that documentation features are the strongest discriminators between gold and non-gold medalists and RQ2 results showing human superiority in documentation accessibility versus GenAI superiority in code quality may not apply to industry contexts where performance metrics differ. However, these patterns likely extend beyond competitions as they reflect fundamental differences in how current GenAI systems generate code versus explanatory text.
- Temporal validity affects our GenAI methodology. The models used in RQ2 represent current capabilities but could be outdated in the future. We mitigate this by focusing on fundamental trade-offs between code standardization and human-readable explanation revealed across all three competitions.
- We acknowledge that the sample size of 9 GenAI-generated notebooks constitutes a limitation of this study. This number was determined by the exploratory nature of the research and the computational resources required for generation and analysis. While the findings provide initial insights, a larger and more diverse sample of GenAI notebooks is necessary to ensure the generalizability of the results. Future work should aim to expand this dataset significantly.

C. Construct Validity

One threat to construct validity requires acknowledgment.

- Medal-worthy performance definition: Our first construct, notebook quality, is proxied by Kaggle's medal system (i.e., Bronze, Silver, Gold). This is a potential threat, as competition rankings may not capture all dimensions of exceptional data science work, such as long-term maintainability, real-world business impact, or deployment efficiency outside the competition environment. However, we mitigate this by using the established Kaggle medal

criteria, which represent community-validated standards for excellence in competitive data science. This provides a reliable and holistic benchmark for notebook quality within the competitive data science context.
- Practical Usability of Code Quality Metrics: Our initial analysis defined code quality using static metrics like code smells and technical debt, which was a valid concern regarding construct validity as it does not capture runnability. To address this and clearly bound our claims, we conducted a post-review execution validation. As reported in Section IV-B, this check confirmed that three of GenAI notebooks failed to run, despite their high static quality. This finding reinforces the limitation and shows that our 'code quality' construct does not fully map to 'practical usability'.

VII. RELATED WORK

A. Studies on Computational Notebooks

Prior research has established computational notebooks as a principal paradigm for data scientists, valued for their support of two distinct roles: private exploration and public explanation [18]. As literate programming environments, they are designed to integrate executable code, narrative documentation, and visualizations within a single document [3], [7]. However, achieving narrative coherence is not an automatic outcome. Kery et al. [7] underscore the significant human effort involved, observing that data scientists find it a non-trivial task to clean up and curate their messy, exploratory code into coherent "stories" for presentation or sharing .

In collaborative settings, a well-constructed narrative is indispensable for communicating results and knowledge dissemination. This dynamic is particularly evident in competitive platforms like Kaggle. Wang et al. [22] demonstrated that community-defined quality (i.e., highly-voted notebooks) correlates more strongly with extensive documentation and high readability than with the notebook's objective performance score on the leaderboard . Their analysis of these well-documented notebooks revealed nine distinct categories of documentation, including process descriptions, headlines for navigation, and explanations for analytical reasoning . This finding underscores the role of notebooks as vital communication artifacts, not merely as code containers.

Yet, this ideal of a clean, well-documented notebook is often at odds with the reality of data science workflows. The transition from exploration to explanation is fraught with pain points. Chattopadhyay et al. [3] cataloged numerous challenges that disrupt data scientists, including difficulties in setting up environments, managing dependencies, versioning exploratory code, and deploying notebooks to production. This difficulty contributes to the well-documented notebook-to-production gap [16].

A critical aspect of this gap is the widespread lack of reproducibility. A large-scale study of notebooks from biomedical publications by Samuel et al. found that the "large majority" of notebooks could not be automatically executed. The primary cause was "issues with the documentation of dependencies" ,

with most notebooks failing due to **ModuleNotFoundError** or **ImportError** . That same study noted a key correlation: notebooks with a low ratio of markdown-to-code cells were more likely to have exceptions, directly linking a lack of documentation to poor reproducibility . This challenge is precisely what Quaranta et al. [16] identified as the key blocker: deficiencies in code quality, structure, and maintainability prevent notebooks from being reliably integrated into production workflows.

B. Human and GenAI Complementarity

This emphasis on human-driven documentation, narrative, and high-level strategy aligns with findings on human-GenAI complementarity. Research indicates that humans demonstrate superior performance in areas requiring deep domain expertise, complex logical reasoning, and creative solutions. For example, Licorish et al. [10] found that humans performed better on tasks requiring in-depth domain knowledge, such as quantum optimization algorithms or debugging complex logic. GenAI, in contrast, specializes in the rapid generation of structured content and the management of repetitive, boilerplate code [1]. How this complementary relationship manifests specifically within computational notebooks remains underexplored.

The introduction of GenAI is prompting a re-conceptualization of the developer's role. Rather than authoring code line-by-line, the human is repositioned as a "curator" [4] or as a "system orchestrator" responsible for providing high-level intent [6], [11], [20]. In this paradigm, the human directs the overall strategy and validates the AI's output, while the AI manages the low-level implementation. Consequently, human validation emerges as a critical function.

Our research builds upon this concept by quantitatively comparing how this role differentiation manifests in a competitive environment. We examine the balance between documentation (a human strength) and technical code quality (a GenAI strength). Our work is supported by findings from Molison et al. [12], which found that code generated by LLMs generally contains fewer bugs and requires less remediation effort. They also observed that fine-tuning, while effective at reducing high-severity blocker and critical bugs by shifting them to lower-severity categories, simultaneously degraded the model's overall performance. Moreover, for complex, competition-level tasks, LLMs were found to introduce structural problems not present in human-authored code. Similarly, Licorish et al. [10] found that GPT-4 passed a higher percentage of functional test cases than human-written code. Cotroneo et al. [4] add to this, noting that AI code is generally simpler and more repetitive.

VIII. CONCLUSION

This study conducted a comparative analysis of human-written and GenAI-generated notebooks within the high-stakes context of competitive data science. Our goal was to explore the distinct, and often complementary, strengths and weaknesses inherent in both human and AI approaches. The findings reveal a clear dichotomy: GenAI excels at producing code with high static technical quality, consistently featuring significantly fewer code smells and less technical debt than human-written counterparts. This suggests a strong capability for adhering to programming standards and generating clean, standardized code.

However, this technical superiority comes with a critical caveat. Our execution checks revealed that high static quality does not translate to practical runnability, as several GenAI notebooks failed during execution due to fundamental data-handling and logic errors. In contrast, human-written notebooks, particularly the medal-winning entries, were distinguished by their comprehensive, accessible, and insightful documentation. This aligns with our RQ1 findings, confirming that narrative communication and clear explanations are a hallmark of high-quality human work in this domain.

The results strongly suggest that while GenAI can achieve superior code cleanliness, top-tier performance in these competitions currently depends more on human-centric strengths. These include creative problem-solving, domain-specific algorithmic innovation, and the ability to craft insightful documentation that communicates a clear analytical narrative—skills that code quality metrics alone fail to capture.

The implications of these findings are threefold and significant for the data science community:

- **For competition hosts:** The identifiable features of GenAI-generated code (e.g., high static quality but low readability scores) may allow for the development of new heuristics to detect GenAI usage, which is crucial for upholding competition integrity and rules.
- **For competitors:** These results provide a strategic guide for using GenAI assistants. Competitors can leverage these tools to handle routine tasks like code standardization and boilerplate generation, freeing up their own time to focus on high-impact, human-centric tasks such as feature engineering, model innovation, and narrative-building.
- **For researchers and educators:** Our findings help identify which critical data science skills, such as narrative reasoning, complex problem decomposition, and practical validation, are not easily amplified or replicated by current AI tools, guiding future research and curriculum development.

Ultimately, this work provides a foundation for exploring the intersections of GenAI development and human creativity. It points toward a future of hybrid approaches where GenAI handles routine code generation and quality assurance, while humans drive the innovative problem-solving and narrative communication that define exceptional data science.

IX. ACKNOWLEDGEMENT

X. DATA AVAILABILITY

To facilitate replication studies, our scripts and dataset are publicly available online, which can be found at

REFERENCES

[1] M. M. Bangerl, K. Stefan, and V. Pammer-Schindler, "Explorations in human vs. generative ai creative performances: A study on human-ai creative potential," TREW 2024: Trust and Reliance in Evolving Human-AI Workflows, at CHI 2024, May 11, 2024.

[2] G. A. Campbell and P. P. Papapetrou, SonarQube in action. Manning Publications Co., 2013.

[3] S. Chattopadhyay, I. Prasad, A. Z. Henley, A. Sarma, and T. Barik, "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, ser. CHI '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1–12. [Online]. Available: https://doi.org/10.1145/3313831.3376729

[4] D. Cotroneo, C. Improta, and P. Liguori, "Human-written vs. ai-generated code: A large-scale study of defects, vulnerabilities, and complexity," 2025. [Online]. Available: https://arxiv.org/abs/2508.21634

[5] K. Grotov, S. Titov, V. Sotnikov, Y. Golubev, and T. Bryksin, "A large-scale comparison of python code in jupyter notebooks and scripts," in Proc. MSR, 2022, pp. 353–364.

[6] A. E. Hassan, G. A. Oliva, D. Lin, B. Chen, Z. Ming et al., "Towards ai-native software engineering (se 3.0): A vision and a challenge roadmap," arXiv preprint arXiv:2410.06107, 2024.

[7] M. B. Kery, M. Radensky, M. Arya, B. E. John, and B. A. Myers, "The story in the notebook: Exploratory data science using a literate programming tool," in Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, ser. CHI '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1–11. [Online]. Available: https://doi.org/10.1145/3173574.3173748

[8] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," Journal of the American statistical Association, vol. 47, no. 260, pp. 583–621, 1952.

[9] V. Lenarduzzi, A. Sillitti, and D. Taibi, "A survey on code analysis tools for software maintenance prediction," in Proceedings of SEDA. Springer, 2020, pp. 165–175.

[10] S. A. Licorish, A. Bajpai, C. Arora, F. Wang, and K. Tantithamthavorn, "Comparing human and llm generated code: The jury is still out!" 2025. [Online]. Available: https://arxiv.org/abs/2501.16857

[11] M. Marron, "A new generation of intelligent development environments," in Proceedings of the 1st ACM/IEEE Workshop on Integrated Development Environments, 2024, pp. 43–46.

[12] A. S. Molison, M. Moraes, G. Melo, F. Santos, and W. K. G. Assuncao, "Is llm-generated code more maintainable & reliable than human-written code?" 2025. [Online]. Available: https://arxiv.org/abs/2508.00700

[13] M. I. Murillo and M. Jenkins, "Technical debt measurement during software development using sonarqube: Literature review and a case study," in 2021 IEEE JoCICI, 2021, pp. 1–6.

[14] J. M. Perkel, "Why jupyter is data scientists' computational notebook of choice," Nature, vol. 563, no. 7732, pp. 145–147, 2018.

[15] J. F. Pimentel, L. Murta, V. Braganholo, and J. Freire, "Understanding and improving the quality and reproducibility of jupyter notebooks," Empirical Software Engineering, vol. 26, no. 4, p. 65, 2021.

[16] L. Quaranta, "Assessing the quality of computational notebooks for a frictionless transition from exploration to production," in 2022 IEEE/ACM 44th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), 2022, pp. 256–260.

[17] L. Quaranta, F. Calefato, and F. Lanubile, "Kgtorrent: A dataset of python jupyter notebooks from kaggle," in 2021 IEEE/ACM MSR. IEEE, 2021, pp. 550–554.

[18] A. Rule, A. Tabard, and J. D. Hollan, "Exploration and explanation in computational notebooks," in Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, ser. CHI '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1–12. [Online]. Available: https://doi-org.ejournal.mahidol.ac.th/10.1145/3173574.3173606

[19] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," Biometrika, vol. 52, no. 3-4, pp. 591–611, 1965.

[20] C. Treude and M.-A. Storey, "Generative ai and empirical software engineering: A paradigm shift," arXiv preprint arXiv:2502.08108, 2025.

[21] A. Y. Wang, A. Mittal, C. Brooks, and S. Oney, "How data scientists use computational notebooks for real-time collaboration," Proceedings of the ACM on Human-Computer Interaction, vol. 3, no. CSCW, pp. 1–30, 2019.

[22] A. Y. Wang, D. Wang, J. Drozdal, X. Liu, S. Park, S. Oney, and C. Brooks, "What makes a well-documented notebook? a case study of data scientists' documentation practices in kaggle," in Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, ser. CHI EA '21. New York, NY, USA: Association for Computing Machinery, 2021. [Online]. Available: https://doi.org/10.1145/3411763.3451617