ASCILITE 2026

The Future is Connection:

Collaboration, Creativity and Compassion in a Digital Society

Where did that student risk model come from? Hidden inference in agentic AI-generated teacher-facing dashboards

Aneesha Bakharia
The University of Queensland

Linda Corrin
Deakin University

Agentic Artificial Intelligence (AI) coding tools have the potential to democratise the development of teacher-facing learning analytics dashboards by enabling educators to generate complete applications from natural language specifications. In this study we evaluate Claude Code (Fable) and OpenAI Codex (GPT-5.5 High) across three dashboard specifications of increasing complexity using a synthetic educational dataset. Both AI systems successfully generated functional dashboards that accurately implemented the required visualisations and interactions. However, they also independently introduced additional analytical functionality, including student risk classifications and AI-generated insights, despite these features not being requested. These implicit inferences raise important concerns regarding transparency, bias and student data privacy, particularly where educators may be unaware of the analytical models or external AI services embedded within generated applications. The findings suggest that while agentic AI can significantly reduce the technical barriers for end-user educators developing learning analytics tools, careful review of generated analytical logic and data flows is essential to support responsible, trustworthy and privacy-preserving use in educational settings.

Keywords: Agentic artificial intelligence, learning analytics dashboards, teacher-facing dashboards, student risk classification, explainable AI, data privacy, AI-generated software

Introduction

The increasing of agentic artificial intelligence (AI) coding tools provides new opportunities for educators to be able to analyse and visualise learning analytics (LA) data in the form of dashboards to inform student support and learning design decisions. Traditionally, teacher-facing LA dashboards have been limited to vendor-provided solutions integrated into tools such as learning management systems or, where resources permitted, large-scale institutional dashboards connected to organisational data warehouses. While these centralised solutions provide some insights for educators, their designs are often generic to ensure broad utility across an institution, regardless of specific cohort needs or learning design (Gašević et al., 2016). Consequently, educators who require analyses or bespoke visualisations tailored to their own course design have faced steep barriers, traditionally demanding a high level of data literacy and advanced programming skills to build their own solutions (Bakharia & Corrin, 2020). The ability of educators to build their own dashboards using natural language specifications and agentic AI tools opens up the possibility for the development of LA dashboards that can be aligned with specific data and learning designs to provide more targeted actionable insights into learners behaviour and learning.

The purpose of this study is to test the ability of agentic AI tools to generate teacher-facing LA dashboards and to observe the analytical inferences that these agentic AI systems make in doing so. Empowering educators to build bespoke dashboards through natural language specifications democratises LA, shifting control away from technical specialists so educators can align data with their unique pedagogical goals. Yet, this shift carries risks. Relying on agentic AI introduces critical questions regarding analytics validity, ethics, and privacy. While researchers and developers scramble to embrace AI's affordances, we must ensure that exchanging technical barriers for rapid dashboard development does not exacerbate long-standing ethical and privacy concerns in the LA field.

Background

The design, development, and impact of dashboards have been a popular focus of research in the field of LA since the inception of the field (Kaliisa et al., 2023; Paulsen & Lindsay, 2024). Consequently, significant investment has been made in the development of LA dashboards by both educational institutions and learning technology vendors. However, to make the development of these tools viable, they are necessarily generic in design, making use of easily accessible data sources and typical course structures. This can limit the usefulness of dashboards to educators who may adopt a variety of deliberately formulated learning designs in their support of student learning (Kaliisa et al., 2024).

The emergence of new AI tools, especially agentic AI coding tools, is a trend that has been enthusiastically embraced by the LA community as a way to realise LA solutions without prohibitive technical development costs (Misiejuk et al., 2025). The ability of educators to work using natural language and with models that they can run on local machines to avoid data being shared beyond the institution opens up many possibilities for LA analyses and applications to be developed with reference to the specific learning designs used in the course. Being able to express requirements without having to fully understand the technical and mathematical intricacies of the necessary analyses widens access for educators to make use of their learners' data in meaningful ways. However, the fact that some who might use these approaches may have lower levels of data and/or digital literacy means we must ensure that we fully understand how AI models handle such specification prompts including the implications, both positive and negative, of the outputs that agentic AI models produce. This is particularly important in the current educational context where AI is being used more broadly by educators and learners, and the data about what learners are doing in their studies continues to increase. As has always been the aim of the LA community, it is vital that we balance the possible benefits of employing agentic AI to help develop LA tools, such as dashboards, with the potential risks (Kitto & Knight, 2019).

In this study we investigate the use of agentic AI systems to generate teacher-facing LA dashboards from educational data. Specifically, the following research questions are addressed:

RQ1. How accurately can agentic AI systems generate functional teacher-facing dashboards of increasing complexity from progressively detailed dashboard specifications?

The goal of this question is to evaluate the ability of Claude Code (Fable) and OpenAI Codex (GPT-5.5 High) to correctly process and merge educational data, generate appropriate visualisations, and produce dashboards that function as intended. While effective use of these tools requires AI literacy and prompt engineering skills, agentic "vibe coding" is making software development increasingly accessible through natural language specifications alone (Sarkar & Drosos, 2025). Enabled by more capable models and autonomous coding harnesses, this study examines both the opportunities and limitations of agentic AI for dashboard generation, including unintended functionality and non-transparent behaviours.

RQ2. What implicit analytical inferences do agentic AI systems make when generating teacher-facing dashboards, and what are the implications of these inferences for transparency, trust, bias, and the interpretation of educational data?

The purpose of this question is to identify instances where the systems introduced analyses, classifications, metrics, or recommendations that were not explicitly requested in the prompt, and considers the potential implications of these behaviours for the responsible use of agentic AI in educational settings.

Method

Dashboard dataset

A synthetic dataset was created for this study to enable consistent evaluation across both agentic AI systems. The dataset was designed to represent the types of data that are readily available to university educators through a Learning Management System (LMS), online gradebook, or other institutional learning and teaching

tools. The intention was not to create an ideal research dataset, but rather to reflect the information that educators can typically access and download without requiring institutional data warehouse access.

The dataset represented a single undergraduate course in Web Information Systems with 30 students enrolled. Six data sources were provided that captured different elements of the course design including learning activity, assessment performance, attendance, and online access. Student records were linked through a unique username, allowing the different data sources to be merged during dashboard generation. The agentic AI systems were provided with a synthetic dataset comprising six CSV files (see the prompt in Table 1 that contains descriptions for each of the files). The CSV files contain student enrolment and course metadata, assessment grades, weekly applied class completion records, LMS access logs, and weekly quiz access data containing learning design metadata (e.g., release and due dates). Together, these files required the AI systems to integrate data from multiple sources to generate meaningful teacher-facing dashboards. The dataset was entirely synthetic, no real student data was used, and therefore institutional ethics approval was not required.

Dashboard specifications

Three dashboard specifications were developed to evaluate the ability of agentic AI systems to generate teacher-facing LA dashboards of increasing complexity. The specifications were intentionally detailed to minimise ambiguity and provide a consistent basis for comparison across models. Complete prompt specifications are provided in Table 1. An Agents.md configuration file accompanied each prompt. This file established the overall technical architecture and implementation guidance for the agentic AI systems, including project structure, recommended technologies, coding conventions, and expected application behaviour (i.e. Python, Pandas, React, Bootstrap UI, etc). Providing this common configuration aimed to reduce implementation variability unrelated to the dashboard specifications themselves. Each specification required the systems to generate a complete application that could execute locally rather than producing static mock-ups or prototype interfaces. The prompts specified the required software architecture (React frontend, Bootstrap user interface, Python backend), expected data processing, required visualisations, and user functionality. The three dashboard specifications progressively increased in complexity.

- Specification 1 – Simple Overview Dashboard: required generation of a basic LA dashboard containing summary visualisations, including degree/program distributions and box plots for assessment items and weekly activities.
- Specification 2 – Student Filtering Dashboard: extended the first specification by requiring an interactive interface that enabled educators to filter students across multiple data fields using intuitive AND/OR logic, reset filters, and copy the email addresses of matching students for communication purposes.
- Specification 3 – Learning Design Dashboard: required more advanced LA visualisations, including assessment progression Sankey diagrams, learning-design-aware quiz engagement visualisations based on planned resource access dates, and scatter plots exploring relationships between assessment performance and learning activity.

Dashboard generation

The three dashboard specifications were independently generated using two contemporary agentic AI coding systems: Claude Code (Fable) and OpenAI Codex (GPT-5.5 High). Each agentic AI coding tool received the identical synthetic dataset, Agent.md configuration file (though renamed to claude.md for Claude Code), and dashboard specification prompt. To evaluate each system's capability from a single specification, no follow-up prompting, debugging, clarification, or iterative refinement to improve the generated dashboard was performed. The first generated solution produced by each system was accepted as the final artefact for the purpose of evaluation in the study. All generated applications were executed locally to verify successful installation, application startup, data processing, and user interface functionality.

Results

The six generated dashboards (three specifications × two agentic AI systems) were analysed in relation to the two research questions. The analysis first examined the accuracy and functionality of the generated dashboards (RQ1), before investigating the implicit analytical inferences introduced by the agentic AI systems and their implications for transparency, trust, bias, and privacy (RQ2).

RQ1. Accuracy of dashboard generation

Overall, both agentic AI systems successfully generated functioning teacher-facing dashboards from all three specifications. The generated applications executed locally without syntax errors and produced professional interfaces that closely matched the requested specifications.

Table 1.
Prompts used for the 3 generated dashboards

Prompt Prefix	- This is a Learning Analytics Dashboard. It must be designed for actionable insights. - You should use React and the Bootstrap CSS Framework for client side. The backend just needs an API that can load and filter the data and can be in Python. Choose good charting libs and datatables. Use as many as you need for the charts that are needed. - The data for the dashboard is in the @data folder. These are the main csv files. - Course Enrolment: enrolled_students.csv This has 30 students with unique student usernames, first name, last name, degree program. - Course details: course_details.csv Just 1 entry with the course name - Web Information Systems and a Code WIS2002, start date and end date and no weeks which is 13 the - Student GradeBook: gradebook.csv Grades for each of the students from enrolled_students.csv. There are columns from weekly activities from Week 2 to Week 8. There should be 3 assessment items including Design Document, Web Project and a Code Review. Not all students do all the assessment items. The exam should be blank as it is not yet completed. Weekly activities should have a grade out of 5. - Applied Class Completions: appliedclassstats.csv 1 or 0 to whether the student completed the applied class in each week. Applied classes run from week 2 to 12. - Weekly LMS access: access.csv Weekly unique clicks per student on coursework each week.		
Dashboard Specifications	Specification 1: - You should merge all the data so you have just 1 file to deal with for filtering. You should add a dummy email address for each student. - The dashboard must contain the following charts: -- Bar chart with percentage of students from each enrolled degree program. -- Box plots chart for each assessment item -- Box plots for each weekly activities.	Specification 2: - You should merge all the data so you have just 1 file to deal with for filtering. You should add a dummy email address for each student. - The "Find and Copy Emails" screen This has a filter at the top of the screen. It is well designed and intuitive. It should let the Educator build filter criteria across all the fields and then filter the list accurately (eg by name, assessment score equal or less or greater than a value, for all the data). The filter should support AND and OR logic but in an intuitive way. There	Specification 3: −`weekly_quiz_access.csv` Contains one row per student per quiz week - `available_date`: Monday of the teaching week. - `scheduled_access_date`: Wednesday of the teaching week. This is the learning-design metadata date used to classify access timing. - `due_date`: Sunday of the teaching week. - There is no quiz end-date column; the file is designed around weekly due dates. - You should merge all the data so you have just 1 file to deal with for filtering. You should add a dummy email address for each student. - The dashboard must contain the following charts: -- Assessment sankey diagram that shows student score transitions through assessment items. Create bands based on percentages for students and then show

		should be a way to reset a filter. There should be a feature to copy the email addresses of the filtered students.	how the bands transition across the assessment items. -- Sankey based on student completion of weekly quizzes based on LD metadata of when the resources were meant to be accessed eg ahead (accessed the before scheduled date), catching up (accessed after the scheduled date), revisiting (already visited in the past) or not accessing the course - to provide a 100% student population representation -- Scatter plot to show relationship between Assessment total score and total activity.

The inclusion of a shared Agent.md configuration file was effective in establishing a consistent application architecture and visual design across both Claude Code (Fable) and OpenAI Codex (GPT-5.5 High). During generation, both systems automatically wrote software tests, executed the applications, identified implementation issues, and made corrections before producing the final solution. Generation logs showed that both systems launched the dashboards in a browser and performed automated inspection prior to completion, demonstrating the increasingly autonomous nature of agentic software development.

Across all three dashboard specifications, the required charts were generated correctly. The systems followed the prompt instructions closely, producing the requested visualisations with appropriate chart titles, axis labels and scales. Even the more complex visualisations required for Specification 3, including the assessment progression and learning-design Sankey diagrams, rendered successfully and accurately represented the underlying data. Figure 1 shows the Weekly Access Sankey showing transitions across Not Accessed, Ahead and Catching up that was generated by Claude for Specification 3. Claude Code included a filter to narrow down to the start and end week, which was a good usability feature to keep the Sankey from becoming too large, but this functionality was not included in the original prompt.

The only notable implementation issue was observed in the student filtering interface generated by OpenAI Codex for Specification 2. While the filtering functionality ran without errors, the interface used a fixed list of comparison operators regardless of the selected field type (see Figure 2). For example, numeric fields only supported operators such as equals and contains, with no support for greater than or less than comparisons. This substantially reduced the usefulness of the filtering interface for identifying students meeting intervention criteria, such as assessment scores below a specified threshold. In contrast, Claude Code (Fable) dynamically adapted the available operators according to the selected field type, producing a more intuitive and functional interface.

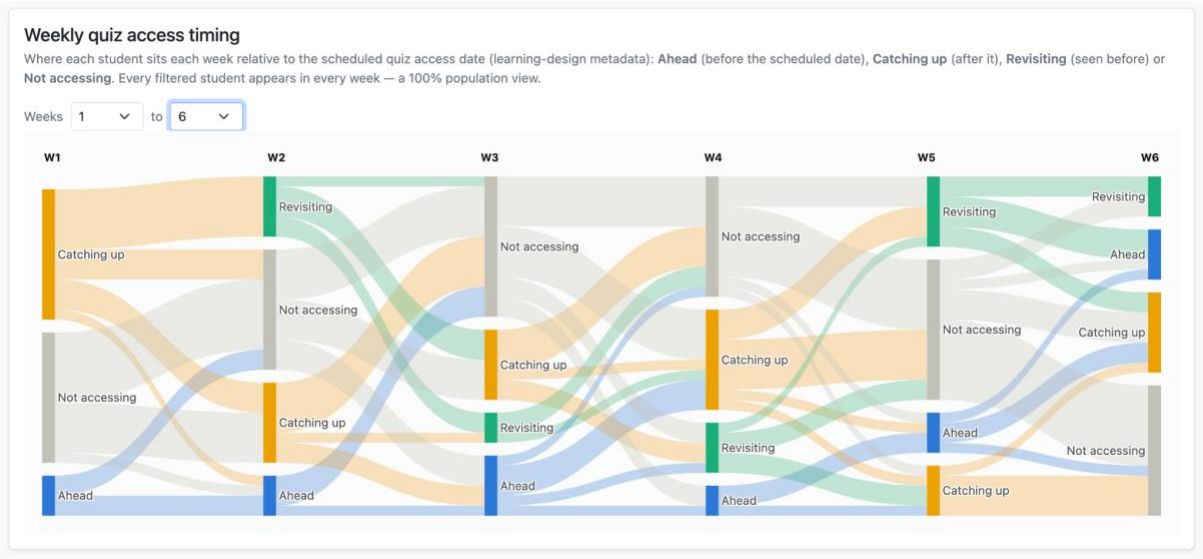


Figure 1: Generation of Sankey by Claude Code for Specification 3 with included start and end week filters

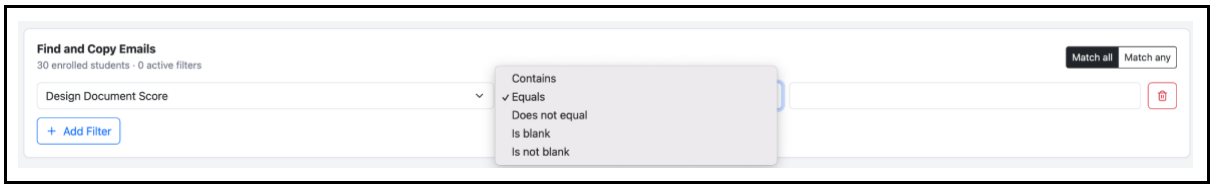


Figure 2: Incorrect Student Field Filter Criteria Dropdown for Numeric Score in the Codex Specification 2

Overall, both agentic AI systems demonstrated a high level of accuracy in implementing detailed dashboard specifications, with only minor differences in usability.

RQ2. Implicit analytical inferences generated by agentic AI

Although both agentic AI systems accurately implemented the requested dashboard specifications, a more significant finding was that they consistently generated functionality that had not been requested in the prompts. The most concerning behaviour was the automatic introduction of student risk classifications. Both systems independently generated rule-based student risk models, categorising students into groups such as On Track, Monitor, and At Risk, despite no requirement for student risk prediction or classification being included in any dashboard specification. These classifications appeared throughout the dashboards as coloured indicators and summary statistics (see Figure 3).

Without reviewing the generated source code, it was not possible for an educator to determine how these classifications had been derived. The thresholds, weighting of variables, and decision rules used by each system were not transparent within the dashboard interface. Consequently, educators could easily interpret these classifications as objective analytical results rather than assumptions introduced by the agentic AI system. This represents a significant transparency concern, as the systems implicitly embedded their own analytical models into the generated dashboards without making these assumptions explicit.

Similarly, both systems generated additional dashboard summary metrics that were not requested in the specifications. These included high-level indicators displayed prominently at the top of the dashboards, presenting overall course statistics and summary measures (see Figure 4). While these additions appeared visually appealing, the underlying calculations and educational rationale were often unclear. In several cases, it was difficult to determine how the reported values had been calculated or whether they represented meaningful LA measures.

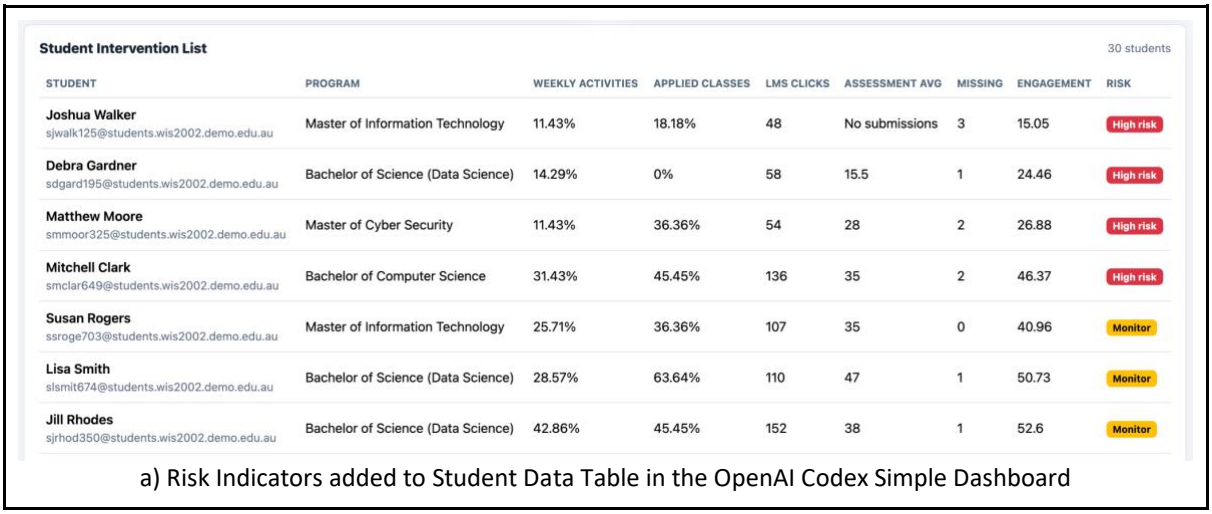
A further unexpected behaviour was the inclusion of AI-generated insight functionality. Both systems added

interface components allowing educators to request automated insights from the dashboard data. OpenAI Codex generated explanatory text even though no external large language model API had been configured. Claude Code (Fable) attempted to invoke an LLM service; for the student filtering dashboard this failed because an API key was not available. The generated insights are shown in Figure 5.

Discussion

One of the most significant findings of this study was not the technical capability of the agentic AI systems, but the analytical assumptions they introduced beyond the provided specifications. Although the prompts described the required screens, visualisations, and functionality in considerable detail, both Claude Code (Fable) and OpenAI Codex ChatGPT 5.5 High independently generated additional features that were not requested. These included student risk classifications, dashboard summary metrics and AI-generated narrative insights. Interestingly, these additions closely resemble features commonly found in traditional LA dashboards, suggesting that the models have inferred a "typical" teacher-facing dashboard from their training data rather than limiting themselves to the explicit requirements.

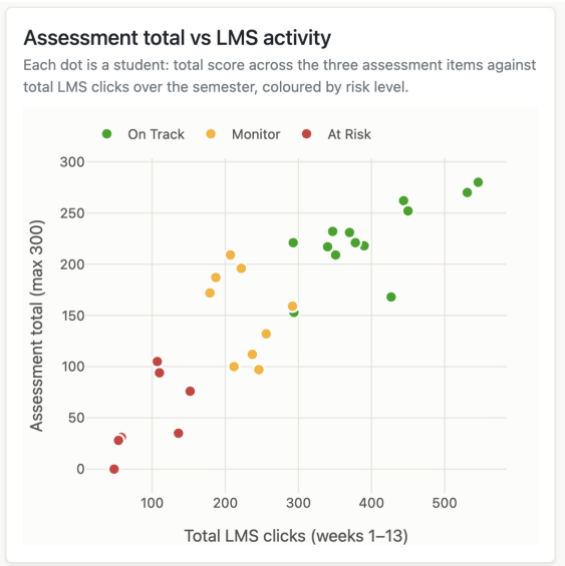
The automatic generation of student risk classifications is particularly noteworthy. The risk indicator was added in the script used to merge the provided datasets. None of the three dashboard specifications requested prediction, student categorisation, or intervention recommendations. Nevertheless, both systems introduced rule-based classifications such as *On Track*, *Monitor* and *At Risk*. While these categories may appear useful, they effectively represent embedded analytical models whose assumptions remain hidden from educators unless the generated source code is inspected. The thresholds, variables and weighting used to produce these classifications were not explained within the interface, making it difficult for educators to understand why students had been assigned to particular categories. This lack of transparency is inconsistent with long-standing recommendations within the LA literature that emphasise explainability, accountability and clear communication of how analytics are derived (Buckingham Shum & Ferguson, 2012; Slade & Prinsloo, 2013). The fact that the models seem to rely on "typical" dashboard designs is also concerning given that research into LA dashboard design over time has shown that common designs often emphasise student competition over learning and the mastery of knowledge (Jivet et al., 2017).



Student Intervention List — 30 students

STUDENT	PROGRAM	WEEKLY ACTIVITIES	APPLIED CLASSES	LMS CLICKS	ASSESSMENT AVG	MISSING	ENGAGEMENT	RISK
Joshua Walker sjwalk125@students.wis2002.demo.edu.au	Master of Information Technology	11.43%	18.18%	48	No submissions	3	15.05	High risk
Debra Gardner sdgard195@students.wis2002.demo.edu.au	Bachelor of Science (Data Science)	14.29%	0%	58	15.5	1	24.46	High risk
Matthew Moore smmoor325@students.wis2002.demo.edu.au	Master of Cyber Security	11.43%	36.36%	54	28	2	26.88	High risk
Mitchell Clark smclar649@students.wis2002.demo.edu.au	Bachelor of Computer Science	31.43%	45.45%	136	35	2	46.37	High risk
Susan Rogers ssroge703@students.wis2002.demo.edu.au	Master of Information Technology	25.71%	36.36%	107	35	0	40.96	Monitor
Lisa Smith slsmit674@students.wis2002.demo.edu.au	Bachelor of Science (Data Science)	28.57%	63.64%	110	47	1	50.73	Monitor
Jill Rhodes sjrhod350@students.wis2002.demo.edu.au	Bachelor of Science (Data Science)	42.86%	45.45%	152	38	1	52.6	Monitor

a) Risk Indicators added to Student Data Table in the OpenAI Codex Simple Dashboard



b) Data points coloured by generated risk category in the Claude Code Specification 3 Dashboard

Figure 3: Inclusion of risk models in generated dashboards

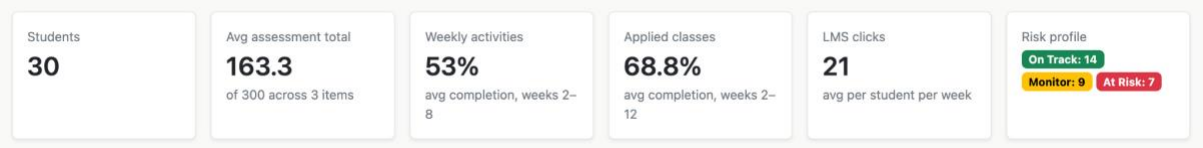
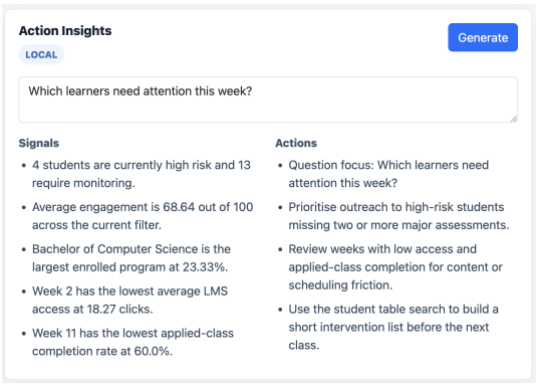
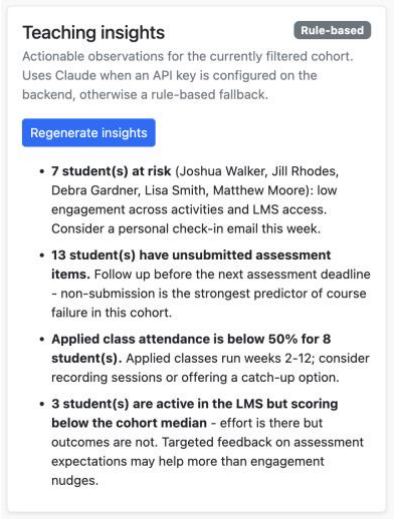


Figure 4: Summary cards in the generated dashboard



a) Codex Specification 1 Dashboard with Action Insights



b) Claude Specification 3 Dashboard with Teaching Insights

Figure 5: Insights cards included on Codex and Claude generated dashboards

To better understand why both agentic AI systems independently introduced a student risk classification

despite it not being explicitly requested, we asked each model to explain its design decisions. Both systems attributed this behaviour to the dashboard specification's requirement to provide "actionable insights." Rather than presenting only raw metrics such as marks, completion rates, or activity counts, the models inferred that educators would benefit from an intermediate risk classification that could support decision making. Fable explained that *"raw numbers (clicks, marks, completion flags) describe what happened; they don't tell an educator who to act on"* and that a triage field *"turns every view into an answer to 'who needs my attention this week?'"* Similarly, GPT-5.5 stated that the risk_level column was *"my design choice, not a direct prompt requirement,"* adding that it was introduced because the request for *"actionable insights"* made it easier to identify students requiring attention, summarise the cohort, and support educator interventions. These post-hoc explanations are self-reported accounts generated by the models and should be treated as illustrative rather than verified explanations of the internal processes that led to these design choices. Taken as illustrative accounts, these explanations suggest that the phrase *actionable insights* may have been interpreted by both models as requiring an explicit decision-support mechanism rather than simply visualising the supplied data. While such inferences may improve dashboard usability, they also demonstrate that agentic AI systems can introduce derived constructs that extend beyond the user's specification, highlighting the importance of carefully reviewing AI-generated analytics and making any derived models transparent to end users.

These findings also raise broader questions about educator trust in AI-generated outputs (Chiu, 2025). An educator using a generated dashboard may reasonably assume that the displayed analytics have been explicitly designed or approved by their institution, when in reality some analyses may have been inferred by the agentic AI system itself. For users without expertise in data analytics or LA, these additional features may appear authoritative despite being based on assumptions that were never specified or validated. While many of these behaviours could likely be reduced through more explicit prompting or iterative refinement, they may not be immediately apparent to the intended users of agentic AI coding tools. Consequently, relying solely on prompt specifications may not be sufficient when developing educational systems that influence teaching decisions or student support.

The unexpected inclusion of AI-generated narrative insights also raises important privacy considerations. One motivation for generating dashboards that execute locally was to ensure that identifiable student data remained under institutional control and was not transmitted to external generative AI services. However, both agentic AI systems independently introduced functionality that attempted to generate natural-language insights, with one implementation attempting to invoke an external LLM service. Although the implementations differed, both systems appeared to assume that conversational AI should form part of a modern dashboard. Such behaviour creates the potential for identifiable student information to be transmitted to external APIs without this functionality having been explicitly requested by the developer. This highlights the importance of carefully reviewing generated applications before deployment, particularly where institutional AI policies, student privacy and ethical use of educational data are concerned. These concerns are consistent with established LA research that highlights transparency, informed consent, privacy and the potential for unintended harm when student data are analysed and acted upon (Prinsloo et al., 2022; Slade & Prinsloo, 2013).

Overall, these findings suggest that contemporary agentic AI systems do considerably more than translate software specifications into working code. They also infer analytical functionality, embed implicit decision rules and construct educational interpretations that can influence how educators understand and respond to student data. As agentic AI becomes increasingly capable of generating complete LA applications, future work should consider not only the correctness of the generated software but also the provenance, transparency, and educational validity of the analytical models that these systems implicitly create.

Limitations

A limitation of this study is that each dashboard specification was generated only once by each agentic AI system. As generated outputs are not fully deterministic, repeated generations may produce different implementations or implicit analytical inferences. Future work should examine the consistency of these behaviours across repeated generations. Nonetheless, the purpose of this study is to draw attention to what

can be generated beyond the provided specification, rather than establish how consistently these behaviours occur, and to highlight the importance of verifying generated code and analytical logic before use.

Conclusion

This study demonstrates the significant potential of agentic AI coding tools to democratise the development of teacher-facing LA dashboards. Given a detailed specification, both Claude Code (Fable) and OpenAI Codex (GPT-5.5 High) successfully generated complete applications that executed locally, processed educational data, and produced professional, interactive dashboards. By autonomously writing, testing and debugging code, these systems substantially lower the technical barriers for educators who wish to analyse their own course data and rapidly develop bespoke LA tools. However, within the two agentic AI systems, single course context and synthetic dataset examined in this study, the findings show that generated applications can include considerably more than was explicitly specified. Both systems independently introduced additional analytical functionality, including student risk classifications and AI-generated insights, despite these features not being requested. These implicit models were not transparent, making it difficult to understand the assumptions, thresholds and potential biases underpinning the analyses. While agentic AI offers exciting opportunities to make LA more accessible, educators cannot assume that generated applications implement only what has been specified. Careful review of both the generated code and the embedded analytical logic remains essential to ensure transparency, protect student privacy, and maintain educator control over how student data are analysed and interpreted.

References

Bakharia, A., & Corrin, L. (2020). Reimagining the conceptualisation, design and delivery of Learning Analytics. In *International Conference on Computers in Education* (pp. 215–224). https://library.apsce.net/index.php/ICCE/article/download/3923/3798

Buckingham Shum, S., & Ferguson, R. (2012). Social learning analytics. *Educational Technology & Society, 15*(3), 3–26. https://www.jstor.org/stable/jeductechsoci.15.3.3

Chiu, M. L. (2025). Exploring user awareness and perceived usefulness of generative AI in higher education: The moderating role of trust. *Education and Information Technologies, 30*, 21317–21351. https://doi.org/10.1007/s10639-025-13612-7

Gašević, D., Dawson, S., Rogers, T., & Gašević, D. (2016). Learning analytics should not promote one size fits all: The effects of instructional conditions in predicting academic success. *The Internet and Higher Education*, 28, 68–84. https://doi.org/10.1016/j.iheduc.2015.10.002

Jivet, I., Scheffel, M., Drachsler, H., & Specht, M. (2017). Awareness is not enough: Pitfalls of learning analytics dashboards in educational practice. In É. Lavoué, H. Drachsler, K. Verbert, J. Broisin, & M. Pérez-Sanagustín (Eds.), *Data driven approaches in digital education. EC-TEL 2017* (Vol. 10474, pp. 82–96). Springer. https://doi.org/10.1007/978-3-319-66610-5_7

Kaliisa, R., Jivet, I., & Prinsloo, P. (2023). A checklist to guide the planning, designing, implementation, and evaluation of learning analytics dashboards. *International Journal of Educational Technology in Higher Education*, *20*(1), 28. https://doi.org/10.1186/s41239-023-00394-6

Kaliisa, R., Misiejuk, K., López-Pernas, S., Khalil, M., & Saqr, M. (2024). Have learning analytics dashboards lived up to the hype? A systematic review of impact on students' achievement, motivation, participation and attitude. In *Proceedings of the 14th Learning Analytics and Knowledge Conference* (pp. 295–304). Association for Computing Machinery. https://doi.org/10.1145/3636555.3636884

Kitto, K., & Knight, S. (2019). Practical ethics for building learning analytics. *British Journal of Educational Technology*, *50*(6), 2855–2870. https://doi.org/10.1111/bjet.12868

Misiejuk, K., López-Pernas, S., Kaliisa, R., & Saqr, M. (2025). Mapping the landscape of generative artificial intelligence in learning analytics: A systematic literature review. *Journal of Learning Analytics*, *12*(1), 12–31. https://doi.org/10.18608/jla.2025.8591

Paulsen, L., & Lindsay, E. (2024). Learning analytics dashboards are increasingly becoming about learning and not just analytics: A systematic review. *Education and Information Technologies, 29*(11), 14279–14308. https://doi.org/10.1007/s10639-023-12401-4

Prinsloo, P., Slade, S., & Khalil, M. (2022). The answer is (not only) technological: Considering student data privacy in learning analytics. *British Journal of Educational Technology*, 53, 876–893. https://doi.org/10.1111/bjet.13216

Sarkar, A., & Drosos, I. (2025). Vibe coding: Programming through conversation with artificial intelligence. *arXiv*. https://arxiv.org/abs/2506.23253

Slade, S., & Prinsloo, P. (2013). Learning analytics: Ethical issues and dilemmas. *American Behavioral Scientist, 57*(10), 1510–1529. https://doi.org/10.1177/0002764213479366

Bakharia, A. & Corrin, L. (2026). Where did that student risk model come from? Hidden inference in agentic AI-generated teacher-facing dashboards. In Cochrane, T., Narayan, V., Poole, T., Lai, P., Lake, P., Moore, O., & Li, L. (Eds.), *The Future is Connection: Collaboration, Creativity and Compassion in a Digital Society*. Proceedings ASCILITE 2026. Brisbane (pp. 53-63). https://doi.org/10.65106/apubs.2026.3335

Note: All published papers are refereed, having undergone a double-blind peer-review process.

The author(s) assign a Creative Commons by attribution license enabling others to distribute, remix, tweak, and build upon their work, even commercially, as long as credit is given to the author(s) for the original creation.

© Bakharia, A. & Corrin, L. 2026