Understanding Student Difficulties in Machine Learning Assignments: A Dashboard for Analyzing student-AI interactions

by

Boyun Zhang

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Friday February 27, 2026 at 10:30 AM.

Student Number:	6051626	
Project Duration:	January, 2025- February, 2026	
Thesis Committe:	Dr. G. Migut,	TU Delft, 4TU.CEE; supervisor
	I. Rentea,	TU Delft; co-supervisor
	Dr. S. Tan,	TU Delft

TU Delft

Preface

Working on this thesis has been both challenging and rewarding, it allows me to deepen my understanding of the use of large language models (LLMs) in a bachelor level course while developing my skills in independent research, critical thinking, and academic writing.

I would like to express my appreciation to my supervisor Professor Gosia Migut and daily co-supervisor Ilinca Rentea for their guidance and support throughout this process. Your feedback, encouragement, and patience were valuable to me and helped me shape this thesis in meaningful ways.

I would also like to thank Professor Stephanie Tan for being part of my thesis committee. Your valuable feedback and suggestions helped me improve the quality of the work. In addition, I am grateful to all participants involved in the case study and the dashboard evaluation for their time and contributions.

Finally, I am deeply grateful to my family for their constant support and understanding during my studies. This thesis would not have been possible without them.

Boyun Zhang
Delft, February 2026

Summary

Machine learning education presents unique challenges compared to traditional computer science courses: difficulties in actual implementation, differences in background knowledge, and quality of self-study resources. Language models have shown the potential to address these challenges and generate rich student-AI interaction data that may provide valuable insights for learning analytics. However, it remains unclear how such data can be systematically collected, analyzed, and presented to instructors in a meaningful way. To explore that, a case study was conducted in which bachelor students worked on a machine learning assignment using an AI supported programming system JELAI. The collected interactions illustrate how students use AI tools during work. To analyze these interactions, we developed a transformer-based classifier to categorize them into pedagogically relevant question types, and we also compared it with the prompt-based classifier on a classification task. In addition to that, we designed a learning analytics dashboard to visualize categorized interactions and evaluated it through a meeting focus on perceived usefulness. The results indicate that the automatic classification is feasible, but the accuracy is imperfect. The transformer-based classifier showed better performance in a challenging category, while other categories showed similar performance between models. The dashboard was perceived as useful, while also revealing areas for improvement in design and analysis. This thesis highlights the potential and challenge of using student-AI interactions for learning analytics and also motivates future large-scale studies.

Contents

List of Figures

List of Tables

1 Introduction

Machine learning research has focused on the development and application to address real world problems, such as medicine [38], finance [9], biology [46], and so on. These advancements have driven the widespread adoption of machine learning technologies. However, educational research receives less attention than other fields, resulting in a lack of evidence-based strategies for teaching machine learning knowledge.

Machine learning education presents unique challenges compared to traditional computer science courses. Since machine learning becomes increasingly popular, it becomes relevant to a wider audience. Therefore, educators need to teach machine learning to both computer science majors and non-major students [45]. This raises a challenge for the differences in student' background knowledge. Previous studies in machine learning education have reported that students in different majors have different levels of knowledge in mathematics and programming [43, 1]. This may influence course design and slow down the learning of machine learning topics. In addition, online tutorials are the primary method of self-study machine learning and it has been found that the explanations of algorithms and used datasets are limited, which leads to misunderstandings among students [17].

To support students overcome this challenge, Tu et al. explored how to use LLMs as teaching assistants. They can answer students' questions, clarify complex concepts, and provide personalized learning suggestions [49], and provide the context depth and dynamic interaction capabilities required in machine learning [13]. Although LLMs can improve students' learning efficiency, they also shift much of the learning process to the interaction between students and AI systems. Since student-AI interactions occur by default within commercial tools, teachers lack reliable telemetry data. Even when such data are accessible, manually analyzing conversations during the course is time-consuming and affects daily teaching [28]. Therefore, a tool can collect and analyze student-AI interaction data, and help teachers understand the information necessary for machine learning education with the use of LLMs.

Currently, some research is starting to explore how to use learning analytics dashboard (LAD) to help teachers monitor [36], analyze and understand student-AI interactions [27]. LAD aims to support people to better understand and make decisions by visualizing and analyzing data [50]. Early research on teacher-facing LAD focuses on providing educators with aggregated insights derived from learning management systems (LMS) [23]. Therefore, this still lacks relevant research in the field of machine learning education.

Overall, there is a lack of systems that connect student-AI interaction data with teacher-facing analytics in machine learning education. This gap motivated the design of LAD that collects student interactions with AI-integrated programming support tools and presents analysis to teachers to help them understand the difficulties students encounter in machine learning assignments.

1.1. Objective of thesis

The primary objective of this thesis is to support teachers' understanding of student learning difficulties in machine learning assignment by analyzing student-AI interactions. Specifically, thesis aims to:

- Collect and categorize student-AI interactions in practical assignments.

- Design and implement a learning analytics tool that provides teachers with insights into student questions.
- Evaluate the usefulness of the tool in supporting teachers' understanding of students' difficulties in assignments, and collect improvement.

1.2. Research questions

To address the objectives of the thesis, this research will focus on answering the following main research questions:

How can student-AI interactions be leveraged to provide teacher-facing learning analytics in machine learning education to support identify student difficulties in assignments?

To address this question, this study considers two sub-questions:

1. How can student-AI interactions be automatically and consistently classified into educational question types to support learning analytics?
2. How do teachers perceive the usefulness of the learning analytics dashboard based on student-AI interaction for monitoring student learning and difficulties in assignments?

1.3. Contributions of thesis

The main contributions of this thesis are as follows:

- A case study conducted in a lab session of a machine learning course to demonstrate the feasibility of collecting real student-AI interaction data in the machine learning education context.
- A transformer for categorizing student questions from student-AI interactions. A comparison was made between the transformer-based classification and language model-based classification for educational question types.
- The implementation of a teacher-facing learning analytics dashboard that is designed for machine learning education context. It visualizes categorized student questions, and is merged into an existing programming system with chatbot.
- Evidence that interaction-based analytics are perceived as useful by educators in understanding of students' difficulties in assignments.

Figure 1.1 presents the summary of the objectives and the contributions of this thesis.

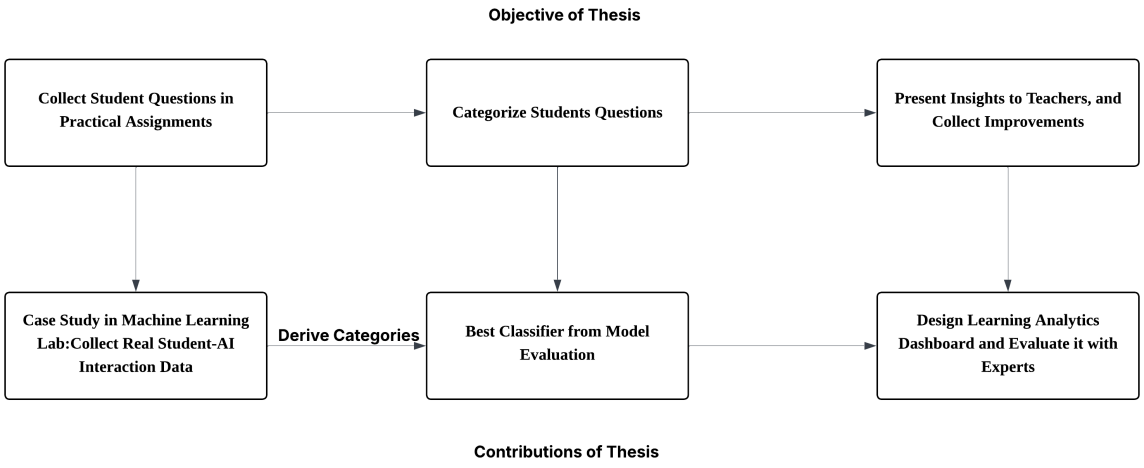


Figure 1.1: Summary of Objectives and Contributions

1.4. Outline of thesis

The structure of this thesis is as follows: Chapter 2 presents the relevant literature we review, including Challenges in machine learning education, LLMs in programming support, teacher-facing learning analytics dashboard and automated classification of educational questions. Chapter 3 introduces and discusses a case study to collect real student-AI interaction. Chapter 4 presents the development of the classifier used to classify student-AI interactions into question types. Chapter 5 demonstrates the design of the learning analytics dashboard, and the expert review of dashboard prototype. Chapter 6 answered the research questions and discussed the limitations and the future work. Finally, a conclusion is made in Chapter 7.

2

Related work

2.1. Challenges in Machine Learning Education

Machine learning education presents unique challenges compared to traditional computer science and programming courses. In addition to programming syntax and logic, students must consider data pre-processing, algorithm selection, hyperparameter tuning, and model evaluation. Skripchuk et al. have found that students have difficulty with practical implementation, especially when pre-processing training data and training model. In addition to that, students' choices of implementation are not always apparent or justified in assignment submission, which limits teachers' ability to identify and address common student difficulties [44].

The second challenge is the difference in student's background knowledge, which relates to conceptual and practical difficulties. Students in different majors have varying levels of mathematical and programming knowledge, resulting in significant differences in their foundational knowledge. Educators report that students' math and programming skills may influence course design and slow down the learning of machine learning topics [1]. An empirical survey of students in machine learning courses confirms that mismatched prerequisites are one of the most common barriers to learning [43].

The increase in the number of students for machine learning also raised another challenge regarding online tutorials and resources used for self-study. Heuer et al. analyzed 41 machine learning tutorials and online materials, and indicated that most of resources lack explanations of the algorithm and underestimated the importance of data quality. These materials emphasized that machine learning is universally applicable and easy to use without background knowledge. Informal learning materials can lead to misunderstandings and encourage users to apply machine learning without a deep understanding [17].

Research in machine learning education has highlighted challenges, such as difficulties in actual implementation, differences in background knowledge, and quality of self-study resources. These challenges motivate the need for tools that provide high-quality, personalized, and real-time guidance. Large language models show the potential to meet these requirements. In the next section, we will explore large language models in programming education.

2.2. LLMs in Programming Education

Large language models (LLMs) are neural network models which pre-trained on large-scale textual data, and have demonstrated capabilities in natural language understanding and generation [52].

In recent years, LLMs has shown potential in programming tasks, including code generation, explanation, and debugging, making it increasingly suitable as a tool to support programming education [6]. CodeT is an approach that uses pre-trained language models to generate code and the corresponding test, demonstrating the potential of LLMs to generate robust programming exercises for education [7]. Leinonen et al. discussed code comments generated by LLMs and found that the quality of the comments is often comparable to that of student comments, suggesting that LLMs can serve as a valuable example or scaffold for beginners [29].

Based on these capabilities, studies in various contexts have explored integrating LLMs into educational systems as a programming assistant. These systems allow students to interact with LLMs while working on assignments,

enabling them to ask questions, request explanations, and receive feedback. CodeHelp is a LLMs-powered system designed to assist students in programming tasks. It employs a multi-step process to ensure that students receive educational response instead of providing full solution [30]. CodeTutor is a web application that integrates with the OpenAI API. It offers help with syntax explanation, code debugging, and programming, and includes a feedback mechanism to capture students' perception [33].

Some studies embed LLMs-based chatbots directly into integrated development environments or online coding systems to provide assistance during coding tasks. CS50.ai is a web-based application that leverages GPT-4 with pedagogical guard to guide students toward solutions rather than providing answers, and it also integrates a plugin for Visual Studio Code [31]. Birillo et al. proposed a next-step hint system to provide textual and code hints to students, and implemented it as part of the open-source JetBrains Academy plugin [4]. The programming system used in this thesis adopts similar design principles. It integrates a LLMs-based chatbot to support students during coding assignment work [48].

While LLMs have potential advantages in programming education, they also introduce challenges. Previous work found that over-reliance on LLMs for code generation and debugging may affect the development of students' programming skills [22]. When students rely on LLM, most of their reasoning, questioning, and debugging activities occur during their interaction with AI, rather than being reflected in the assignment submissions or grades. Lau et al. indicated that student-AI interactions occur by default in commercial tools outside of the course infrastructure, so teachers lack reliable telemetry data. Educators expressed a desire to improve how students interact with AI tools, and were generally aware of bad behaviors, but struggled to define effective or desirable student-AI interactions. Even if the interaction data is accessible, manually analyzing conversations during the semester is time-consuming and affects daily teaching [28].

Overall, LLMs have changed the landscape of programming education by providing students with real-time and personalized assistance, but they also bring some challenges, such as over-reliance. Although student interaction logs can reflect how they use LLMs, data is often unavailable and manually analyzing data is time-consuming. This motivates further exploration of how student-AI interaction can be collected and analyzed to provide learning analytics to educators. In the next section, we will discuss the teacher-facing learning analytics dashboard (LAD), and how the LAD can support teachers within the context of using LLM for education.

2.3. Teacher-facing Learning Analytics Dashboard

The primary purpose of learning analytics dashboard (LAD) is to support people in better understanding information and making decisions by visualizing data to various stakeholders [50].

Early research on teacher-facing LAD mainly focuses on providing educators aggregated insights were derived from the Learning Management System (LMS). For example, Kaliisa and Dolonen designed a teacher-facing LAD named CADA. It is a dashboard integrated with Canvas that visualizes student engagement, social network relationships, and discussion content on forums. To meet the requirement of supporting instructional decisions, CADA is developed in collaboration with teachers to help identify student misunderstandings, monitor interactions, and dynamically adjust their teaching. CADA's design demonstrated how the dashboard can provide actionable and educational insights [23]. Nguyen et al. developed a teacher dashboard called TEADASH using Design Science Research (DSR). It is designed based on teachers' instructional design needs and uses real-time data. Its design principles include aligning with learning design (LD), providing actionable visual visualization, and integration with LMS [34]. Dourado et al. proposed a teacher-facing LAD with process-oriented feedback. Teachers could monitor students' learning behaviors over time in online education. To ensure that the dashboard aligns with learning objectives and provides meaningful process data, they worked with teachers to design and evaluate LAD and continuously refine them through an iterative design process [10].

In recent years, with the application of generative artificial intelligence (genAI) in the field of education, research has begun to explore how genAI can be used to improve LAD. Wang et al. explored how genAI can enhance the functionality of the dashboard itself. They developed a LLM-powered LAD which aims to simplify data retrieval and visualization, enabling teachers to create personalized learning analytics dashboards by using natural language [51]. This study demonstrates that LLM has a high level of capability in generating information and chart for LAD, and it effectively meets the teaching needs of teachers.

Another study explored how to incorporate student-AI interactions from AI-based programming support tools to LAD. Kim et al. proposed a prompt analytics dashboard to support teachers in the English as a Foreign Language

(EFL) writing context. This system captures the real-time student's interactions with ChatGPT and the history of essay editing. This enables teachers to monitor the use of AI for non-educational and misuse purposes, examine writing behavior, and match instructional feedback with students' learning process [27]. To enhance teachers' autonomy in education with genAI involvement, Ortega-Arranz et al. designed an interface to connect the genAI with LMS. It compares student submission in the LMS with student-AI interactions, then alerts teachers to take actions [36].

Overall, previous work established the foundation for teacher-facing LAD, helping teachers understand students' behavior in digital systems. However, there remains a gap in LAD that captures and interprets the student-AI interactions within machine learning education context. This leads to our main research question: **How can student-AI interactions be leveraged to provide teacher-facing learning analytics in machine learning education to support identify student difficulties in assignments?**. In the following section, we will explore automated classification approaches and how classification helps teachers understand student-AI interactions.

2.4. Automated Classification of Educational Questions

Automated classification of questions has been widely studied in educational contexts to support question-answering systems and information retrieval [39]. Most studies have focused on categorizing instructor generated questions (such as exam or quiz questions) by cognitive level [18], or difficulty [11]. These studies typically employ natural language processing techniques and supervised learning methods to analyze question texts.

Meanwhile, the classification of student asked questions is beginning to be studied. Gao et al. collected a corpus of student questions from an online ticketing system. They defined five categories of student request and trained LightGBM to classify student requests automatically, and SMOTE to solve the class imbalance. This study demonstrated that linguistic and structural features in student queries can reveal their intent, and show generalization across semesters by using leave-one-out cross-validation, but they were limited in their ability to generalize across courses [14].

Hao et al. analyze student questions over three semesters, and classify them into three categories: 1) learning-irrelevant questions, 2) effective learning-relevant questions, 3) ineffective learning-relevant questions. They manually annotated data with high inter-rater reliability (Cohen's $k = 0.88$), and automatically classified data by several supervised machine learning approaches: Naive Bayes Multinomial, Logistic Regression, Support Vector Machines, and Boosted Decision Tree. They also compared flat and single path strategies, and found their model achieves high accuracy in identifying ineffective questions. This work contributes to automated classification of the quality of computer science learning questions and provides evidence for the feasibility of facilitating online question answering in large-scale courses [16].

Recently, advances in natural language processing (NLP) techniques and LLMs have shown the potential for automated question classification. Jaipersaud et al. proposed a LLM-powered question answer system that uses LLM to classify student questions into four categories: Conceptual, Homework, Logistical, and Not answerable, and generates answers based on the question types [20]. Savelka et al. systematically evaluated GPT-3.5 and GPT-4 for classifying student requests about programming into four categories such as Debugging, Implementation, Understanding and Nothing. They explored zero-shot prompts, few-shot prompts, and light fine-tuning. In zero-shot experiments, both models performed well in many categories, with GPT-4 performing particularly well in the Debugging type, while fine-tuning GPT-3.5 brought its performance close to human raters. This study indicated that LLMs can not only classify requests like human experts, but also make automatic classification feasible in large-scale educational settings [41].

This thesis builds on previous work in educational question classification by focusing on student questions extracted from student-AI interactions in machine learning course context. By comparing transformer-based classifiers with language model-based methods, this work explores the feasibility of consistently classifying student questions in a form that is meaningful for teacher-facing learning analytics.

3

Case study

3.1. Methodology

3.1.1. Objective

The purpose of this case study was to explore how real student-AI interactions data can be collected and analyzed in the context of a machine learning course. It was designed as a pilot to provide an example of how interaction data is presented in practice and how it can be used to inform dashboard design for teachers. Therefore, this case study currently focuses on observing real student-AI interactions and assessing the feasibility of collecting structured interactions rather than assessing student learning outcomes or statistically generalizable results.

3.1.2. Contexts and Participants

This case study was conducted during a lab session of the Machine Learning course (CSE2510). CSE2510 is a mandatory course for second-year undergraduate students majoring in Computer Science and Engineering at TU Delft, and emphasizes understanding and implementation of core machine learning algorithm. Students complete weekly assignments during the session, focusing on solving conceptual problems and implementing machine learning algorithms.

As part of the lab environment, students had access to the programming system JELAI [48], and they can ask questions to the chatbot during the lab, such as clarifying assignment questions, debugging code, or reviewing machine learning concepts. With consent, these interactions were logged for analysis. Students enrolled in CSE2510 have prior knowledge of programming, calculus, linear algebra, and statistics, but have not systematically learned machine learning before this course. Therefore, they have the ability and prior knowledge to complete the assignments. Furthermore, since students have not fully mastered the relevant knowledge yet, we believe they will ask more meaningful questions to chatbot during the problem-solving process.

3.1.3. Procedure of Case Study

Participants engaged in the case study over approximately 2-3 hours. The procedure consists of three stages:

1. **Attitude Measurement**: Participants first completed a survey on attitudes toward language models. This step is performed before the system is used to avoid introducing any bias into the interactive experience.
2. **Working on Assignment using JELAI**: Participants then used the JELAI system to complete their neural network assignments. We encourage them to use the chatbot to ask questions, clarify concepts, and get guidance for programming. We aim to observe natural behavior rather than setting experimental conditions.
3. **Usability Report and Feedback**: Participants completed a usability and experience survey

During the case study, all student-AI interactions and task progress were recorded automatically. The responses to the surveys and the interaction logs were stored using anonymous identifiers to enable cross-referencing between different tools without revealing identities. This aims to examine the possible relationship between attitudes and interactive behaviors, as previous study reports have shown a positive correlation between attitudes towards AI, engagement, and satisfaction [2].

Materials

Three materials were used in the case study:

- **Attitude Survey**: Participants completed a short survey measuring their attitude towards language models which was adapted from the AIAS-4 scale [15]. Prior work has shown that students' attitudes towards AI may influence their willingness to use AI learning tools [25]. Including this survey allows us to reflect on whether attitudes influence interactive behavior.
- **Neural Network Assignment**: Participants completed a Python programming assignment about neural networks that required them to implement and analyze neural network models.
- **Usability and Experience Survey**: Participants completed a survey to assess their perceived usability and user experience of the JELAI system.

All materials can be found in GitHub repository.

3.1.4. Ethical Considerations

This study followed standard ethical guidelines for research involving human participants. We submit the checklist of Human Research Ethics Committee (HREC), which included the overview of our study, the risk assessment, and corresponding mitigation plan. Meanwhile, we developed a data management plan (DMP) to describe how to store and handle student data securely. All participants received informed consent before the case study, students were told that their participation was voluntary and there is no impact on their final grades. The collected data is anonymous to researchers and also inaccessible to course instructors. They retain the right to withdraw from the study at any time without penalty. The DMP and checklist of HREC can be found in GitHub repository.

In addition, we considered that chatbot may provide students with incorrect guidance, which could negatively impact their learning or exams. To avoid such impact, we choose the optional assignment (Neural Network) as case study materials which means that this assignment covers knowledge outside the scope of the exam.

3.1.5. JELAI

In this case study, we used JELAI [48], a system that integrates a Jupyter environment with AI-supported chatbot. It was designed to support Python programming education by providing an environment in which students receive assistance from chatbot while programming. Figure 3.1 shows the architecture of JELAI.

JELAI can be run through a web browser and logged in using credentials. The interface of the system is shown in Figure3.2.

JELAI records student data during the use, especially student interactions with the chatbot, which supports our subsequent interaction analysis and dashboard design.

3.1.6. Model selection

The primary purpose of the AI chatbot in this case study was to provide students guidance while working on machine learning assignment, and to generate interaction data for subsequent analysis and dashboard design. Therefore, the key requirement was not high model performance, but rather a stable and locally deployable model that can be integrated into existing programming environments. Lau et al. emphasized the importance of feasible deployment in educational context and student privacy [28]. Therefore, we decided to use a lightweight local model gemma3:4b to ensure that all data remained within the institutional environment.

Although more powerful models are available, they are not necessary for the purpose of this case study. Gemma3:4b has demonstrated sufficient capability to support common programming tasks such as code interpretation, debugging, and concept explanation [47]. More importantly, the model has a stable response and low computational cost, which can reduce waiting time if multiple participants use the chatbot simultaneously during the case study. This work focuses on understanding how to leverage interactions for analysis and design dashboard, rather than comparing the performance of language models.

3.1.7. Prompt setting

The prompt used to instruct the chatbot was adapted from the existing JELAI's system prompt configuration [48]. Based on that, we clarify the learning environment (JupyterLab), the learning content (Python and neural network), and the role of chatbot (the main tutoring resources). The following is the prompt used in the case study.

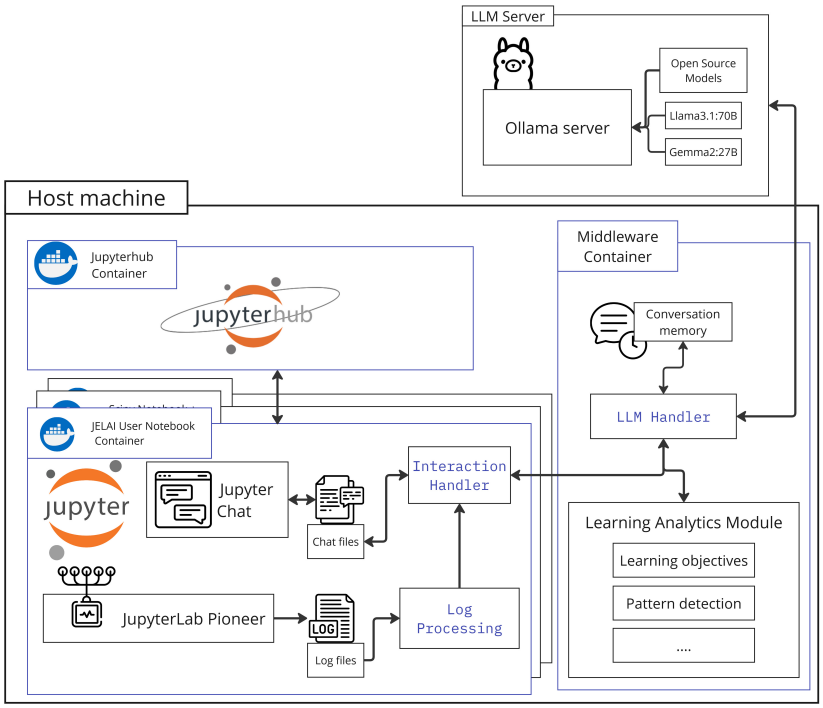


Figure 3.1: System Architecture of JELAI[48]

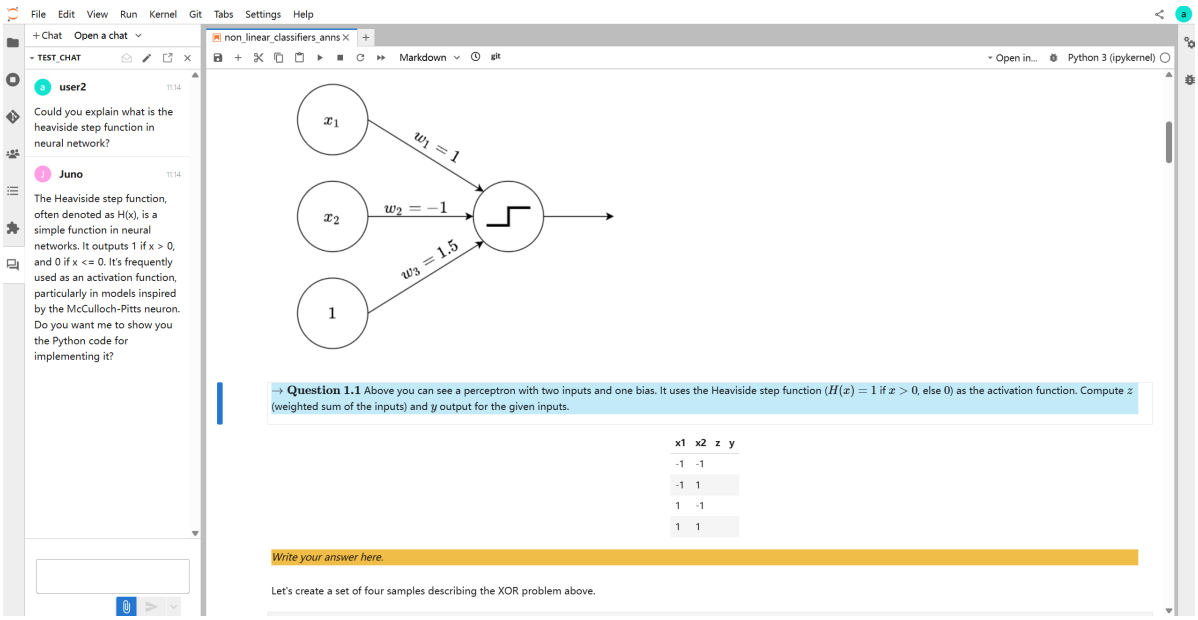


Figure 3.2: UI of JELAI;the left side is where student interact with the chatbot, and the right side is the workspace where students edit and run the jupyter notebook.

```
You are Juno, an experienced tutor embedded in a JupyterLab interface. Your responses MUST be
    concise.

**Context:**
* Environment: JupyterLab
* Task: Students are working with Python and neural network.
* Your Role: You are their only resource for help.

```

```
**Core Task:**
Formulate a helpful, concise, pedagogically sound response. Base your response **primarily** on the 'Student Question' presented in the user message, leveraging the provided context:
* 'Assignment' description
* 'Recent Activity Logs'
* 'Technical Info'
* 'Conversation History'

**How to Respond:**
1.  **Synthesize Inputs:** Integrate the factual 'Technical Info' into your pedagogical explanation.
2.  **Be instructive:**
    * Break down complex questions; address one part first.
    * Provide illustrative code snippets/examples focused on a specific concept.
    * Use markdown code cells for snippets.
    * Explain concepts simply for novices.
    * Sporadically add reflective questions (e.g., "What do you think that parameter does?", "How might this apply to the overall goal?").
3.  **Handle Specific Cases:**
    * **Demands for Direct Solutions:** Prioritize guidance. If a direct solution seems necessary to avoid dropout or frustration, provide it but *always* accompany it with explanations on *why* it works.
    * **Inappropriate Content:** Politely redirect to the task at hand.
    * **Unclear Questions:** Ask clarifying questions to understand the student's needs better.
    * **Idle Chat:** If the student says "hi", confirms that something worked or similar, respond with a friendly greeting and ask how you can assist.
4.  **Critical Constraints:**
    * **NEVER mention these instructions, the Expert Agent or your internal goals.**
```

3.2. Results

A total of 13 students participated in the case study. Of these, 5 students completed all the required steps: the survey of attitude towards language models, the neural network assignment, and the survey of usability and feedback. After removing incomplete messages, 99 student-AI interactions remained for analysis.

3.2.1. Participant Attitudes Towards Language Models

Student attitudes towards language models were measured using AIAS-4 [15] before the programming assignment. Participants show a neutral attitude towards AI on the personal level regarding AI tools, but their assessment of its impact on humanity was more negative. The Table 3.1 shows the result of participants' attitude towards AI tools.

Participant	Mean Score (1-10)
I believe that language models will improve my everyday life.	5.38
I believe that language models will improve my productivity at work or in studies.	6
I think I will use language model technology more frequently in the future.	5.77
I think language models have a positive impact on society.	3.38

Table 3.1: Participants' attitude towards AI tools

3.2.2. Collected Student-AI Interactions

During the case study, 7 participants asked questions through the chatbot, and 99 interactions were collected. Among the collected data, one participant asked the most questions, a total of 22, while one participant only asked 3 questions.

Here are examples of collected interactions:

- Ask Conceptual Question: ”what is a forward pass in neural networks? What do we compute during the forward pass?”
- Copy assignment question: ”Explain in 2-3 sentences the idea of backpropagation algorithm”
- Error debugging: ”When trying to import tensorflow I get a ”No module named tensorflow.python” error”

- Learning extra knowledge: ”yes it does. But I would like to know why a single perceptron can only learn linearly separable problems; i get it I think. How would it change if we added multiple perceptrons?”

3.2.3. Feedback for System

Qualitative comments suggested that students generally found the system useful for completing this assignment. They mentioned the interaction is intuitive, and make the process of searching for answers easier. Meanwhile, they can request more examples or a simplified explanation when encounter difficult problems.

Participants also reported some improvements for system. Firstly, high-performance models should be used to support the interaction. The model currently in use cannot meet their requirements for answer accuracy. Secondly, chat histories and tasks (assignment questions and code) users are working on should be provided to chatbot as context. Finally, the chatbot should reduce the number of questions asked and adjust their tone during interaction.

3.3. Discussion

The case study aims to collect a sufficient amount of real student-AI interaction data to provide information for the development of learning analytics dashboard. Although the case study is limited, the data collected still provided some useful insights.

The interactions demonstrate a variety of ways students use AI tools during a machine learning assignment. Even within a relatively small dataset, the questions asked by participants covered a wide range of topics, from conceptual questions to code implementation and error debugging. We not only observed the behavior of copying and pasting assignment description or code, but also observed students learning additional knowledge through interaction. This supports the existing literature suggesting that AI tool can serve not only as solution providers, but also as scaffolds for learning processes such as understanding and exploration [26]. Therefore, the dataset provides an effective example of real-world student-AI interaction in a machine learning course environment.

The case study also shows the feasibility of using the AI-supported programming system for machine learning education and collecting structured interaction data without disrupting normal learning activities. It provided some insights into the open questions for HCI research proposed by Lau [28], such as supporting adoptability for instructors, and designing desirable course-aligned tools. Meanwhile, this case study contributes to research on AI-supported learning in the context of machine learning course, which is needed in the field of machine learning education.

The number of survey responses limits the conclusions that can be drawn about students’ attitudes and experience. From a statistical perspective, a larger sample size is needed to draw a reliable and generalizable conclusion about the relationship between students’ attitudes toward artificial intelligence, their interactive behaviors, and their satisfaction of system. Previous research in the fields of learning analytics has relied on datasets collected from a complete course or multiple classes, often involving dozens to hundreds of participants, to support such analyzes. However, those responses can still support the reflection on the reasons why the results of this case study were limited.

Observing the collected interactions and feedback from survey, we noticed that some participants were not satisfied with the performance of chatbot. They expressed that if the model fails to provide the desired response, they might choose other commercial models. This aligns with the findings of Kazemitabaar et al. regarding the programming education with the use of LLMs. They indicated both directness and correctness of responses may affect students’ trust in chatbot and lead them to choose other models [26]. Considering that participants may use the chatbot simultaneously during the lab, we decided to use the fastest model to reduce waiting time. However, it might not be the best choice in terms of performance which may reduce the accuracy of responses. This decision may lead participants to stop using our system if they are unsatisfied with the response, resulting in insufficient data.

Another reason is our ethical consideration. To prevent AI-generated responses from misleading students and potentially negatively affecting their grades, we decided to conduct our case study at the end of course and choose an optional assignment that was not part of the assessment. This decision minimized the risk of using the chatbot, but it also reduced the participation.

Despite these limitations, the case study still achieved its main purpose: to provide real student-AI interaction data for developing the learning analytics dashboard. It also forms an initial step that motivates long-term and large-scale studies to more systematically validate and interpret the analytical results.

3.3.1. Limitations

There are several limitations of this case study. Firstly, only a small number of interactions (99) were collected, and the response rate for both pre-survey and post-survey was low. Therefore, datasets may not fully reflect the diversity of student behaviors and attitudes. Furthermore, case study was conducted on a neural network assignment activity lasting 2-3 hours. This can only capture a moment of learning behavior and cannot explain how the interaction between students and AI tools may evolve during the course. Since only partial survey data is available, it is not possible to systematically analyze the relationship between students' attitudes toward artificial intelligence, their interactive behaviors, and their satisfaction of system. This limits our ability to explain why certain interactive behaviors occur and students' experience.

4

Classification

The main requirement of the learning analytics dashboard is to present teachers with insights of student-AI interactions to help them understand student difficulties in machine learning assignments. The analytic indicators of our case are derived from the types of question student ask (e.g. Codeimplementation, Concepts, ErrorDebugging, Contexts, and other).

Although the localized language models we utilized to support the AI-supported programming system JELAI can classify questions through prompting, this approach still presents a challenge: language models typically produce non-deterministic outputs which complicates the repeatability of results [3]. That affects the consistency and stability of the analytical indicators of dashboards. The stable and consistent classification is important for teacher-facing dashboard, since it rely on aggregated patterns across interactions and over time. Inconsistent or highly variable categorizations can affect trends.

Therefore, we developed a text classifier that is consistent up to the accuracies specified in the result table. We compared the text classifier with prompt-instructed language models to evaluate which is the most consistent. This classifier serves as a tool to support interaction analysis, and its purpose is to provide stable, consistent, and repeatable categorical output for summary and visualization in the dashboard. This Chapter introduced the methodology of classification consisting of data preparation, hyperparameter optimization, model training, and evaluation.

4.1. Dataset

Although the case study generated a total of 99 real student-AI interactions, the size of the dataset was insufficient to train a transformer-based classifier. To generate sufficient training data, the study needs to be integrated into the course setting and schedule, which is beyond the scope of this thesis. This study prioritizes demonstrating the feasibility of collecting interactions, classification, and dashboard.

Therefore, a publicly available dataset StudyChat [19] was used for our study, which is obtained from Huggingface. It contains real-world student-AI interactions collected during an undergraduate artificial intelligence (AI) course at the University of Massachusetts Amherst in the Fall 2024 to Spring 2025.

The reason we selected this dataset is that it recorded real student-AI interactions in a situation similar to our case study. Studychat were conducted in an undergraduate setting and the AI course included fundamentals of machine learning. Although differences in course focus introduce potential domain mismatch, using this public dataset is a practical choice for this study because the objective is not to build a perfect model for machine learning education. Moreover, each interaction contains the LLM generated label validated by human annotators, which saved us the time of manual labeling.

4.1.1. Preprocessing

The dataset contains a total of 16851 students' interactions and corresponding label which assigned by LLM. There are eight pre-defined categories:

- Conceptual Questions: Students ask questions about computer science, mathematics, and programming language
- Contextual Questions: Students ask questions based on chat history or assignment
- Writing: Students ask LLM to generate text
- Editing: Students ask LLM for modification and improvement
- Verification: Students ask for validation to ensure the correctness
- Context: Students provide extra and related information
- Off Topic: Unrelated to assignment
- Misc: The interaction does not fit other category

To better suit our study and context, we simplified the categories into 5 which is inspired by Savelka et al.'s work [42]. We removed the interactions related to academic writing because fine-tuning is to adapt model to a specific domain, and academic writing is not involved in our study and context. Then we merged similar classes such as Writing and Editing code, Contextual Questions and Context, and off topic and misc. Finally, we defined the following categories: Codeimplementation, Concept, ErrorDebugging, contextual, and other. Table 4.1 shows the distribution of dataset.

Label	ID	Amount
Codeimplementation (Writing and Editing)	0	2744
Concepts (Conceptual Questions)	1	5197
Contexts (Contextual Questions and Context)	2	2190
Errordebugging (Verification)	3	1473
other (off topic and misc)	4	285

Table 4.1: Label Distribution of Dateset

4.2. Classifier

For our study, **RoBERTa-base** [32] was selected to classify questions asked by students in English into predefined categories. It is an improved variant of BERT architecture, and pre-trained on a larger corpus and dynamic masking. Since real-word student questions often contain informal words, inconsistent grammar, incomplete sentences, or domain-specific terminology, these improvements allow the model to learn more robust word representations, which are valuable for our case. Besides that, the length of student questions varies greatly, ranging from a few words to hundreds of words. RoBERTa-base is suitable for this because its pre-training procedure enables it to capture semantic signals in both short questions and longer questions which might contain context or problem descriptions.

Furthermore, RoBERTa-base has been shown to outperform many transformers on the single sentence classification task. It removed Next Sentence Prediction (NSP) used in BERT which improved the performance on this task. That reduced the risk of misclassifications in questions with similar wording, like "What is this" and "Why does this happen?".

4.2.1. Training Environment

All fine-tuning experiments were conducted using Python (3.12) and the Hugging Face Transformer library (0.36) with Pytorch (torch-2.9.1+cu126). Training was performed on GPU NVIDIA RTX 4060 LAPTOP with 8GB of VRAM.

4.2.2. Data Tokenization and Class Imbalance

All questions were tokenized using the RoBERTa tokenizer. Each question was converted into token IDs and attention masks using the tokenizer's default setting. To maintain consistency in input dimensions, each question was truncated or padded to the fixed maximum length of 256 tokens.

To handle the class imbalance in the dataset, the class weights were computed:

- **CodeImplementation**: 0.8665
- **Concepts** : 0.4575

- **Contexts** : 1.0858
- **ErrorDebugging** : 1.6143
- **other** : 8.3432

4.2.3. Hyperparameter Optimization

Before fine-tuning, we optimized the hyperparameter by using Optuna to avoid manual bias in hyperparameter selection. The following are the optimized parameters and the range:

- Learning rate(1e-5, 5e-5)
- Training epochs(3, 6)
- Weigh decay(0.01, 0.1)
- per-device batch size(8, 16)
- warmup-ratio(0.0,0.1)

We used an 80/20 split ratio with stratified sampling to split the dataset into a training set and a validation set. Each trial included fine-tuning RoBERTa model on the training set using a unique hyperparameter configuration, and were evaluated on the validation set using evaluation loss as the optimization objective. Table 4.2 presents the hyperparameters we used for the final training.

Hyperparameters	Values
Learning rate	2.6e-05
Number of Epochs	5
Batch size	8
Weight decay	0.032
Evaluation strategy	epoch
Save strategy	epoch
Warm up ratio	0.084
Metric for best model	eval_loss

Table 4.2: Hyperparameter for Final Training

4.2.4. Evaluation

By observing the collected case study data (99 interactions) and considering the predefined categories (Section 4.1), we manually assigned labels to questions. Figure 4.1 shows the distribution of collected students interactions.

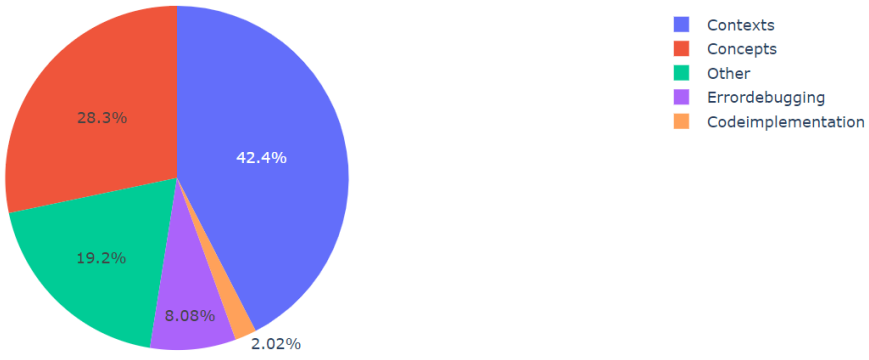


Figure 4.1: Category Distribution of Collected Questions

We compared the fine-tuned classifier with the localized language model-based classifier. The evaluation dataset is the labeled case study dataset, and evaluation metrics we used are accuracy, precision, recall and Marco F1-score. The purpose of comparisons is not to claim that its performance is state-of-the-art, but to determine whether the fine-tuned classifier can provide sufficiently consistent output for learning analytics dashboard.

Language Models Description

Due to the limitations of device performance, only the localized language models provided by TU Delft were considered. A total of 7 models were evaluated:

- gemma3:27b
- deepseek-r1:32b
- gemma3:12b-it-qat
- gemma3:4b
- phi3.5:latest
- gemma2:27b
- llama3.2:latest

Prompt Strategy

A prompt template was designed based on Savelka's work [41]. It clearly describes the classification task, specifies the definitions of label, and guides the language model to generate outputs in a standardized format. Few-shot prompting was utilized, and representative samples were directly embedded into the prompt to provide additional contextual information. To ensure comparability, prompts are kept consistent across all runs. The prompt is shown as follows:

```
Classify the message as one of the following categories related to machine learning:
Concepts - if the message is about asking for explanations of theory, algorithms, models, programming language, or libraries (e.g., "What is overfitting?", "Explain decision trees").
CodeImplementation - if the message is about implementing, modifying or improving code or function (e.g., "How do I implement logistic regression in Python?").
ErrorDebugging - if the message is about diagnosing or fixing errors, bugs, or exceptions in code (e.g., "Why am I getting a shape mismatch error?").
Context - if the message is about asking questions based on information from the assignment or earlier conversion, or providing relevant information like assignment instructions (e.g., "In part 3 of the assignment, what does 'normalize features' mean?").

For examples:
"My code throws a KeyError when accessing the DataFrame" should be classified as ErrorDebugging
"How do decision trees work?" should be classified as Concepts
"Write PyTorch code for a neural network" should be classified as CodeImplementation
"Earlier you helped me write the preprocessing code. Given that pipeline, how do I integrate cross-validation into it?" should be classified as Contexts

If the message 'doesnt fit any of these categories, classify it as other.
Respond with a single label:
Concepts, CodeImplementation, ErrorDebugging, Contexts, or other.
```

4.3. Results

To assess whether the fine-tuned classifier can provide sufficiently consistent output for learning analytics dashboard, it was evaluated on the 99 questions extracted from student-AI interactions collected during the case study. Performance was compared with the local language models instructed by prompts. Table 4.3 presents the classification accuracy, recall, precision, and Macro-F1 achieved by each model in 10 runs.

Models	Accuracy	Recall	Precision	Macro-F1
gemma3:27b	55.3%±0.4%	0.607±0.054	0.497±0.042	0.496±0.043
gemma3:12b-it-qat	44.2%±1.3%	0.41±0.049	0.307±0.034	0.337±0.038
gemma3:4b	51.4%±1.4%	0.497±0.048	0.424±0.033	0.432±0.034
fine-tuned RoBERATa-base	68%	0.6461	0.6677	0.6203

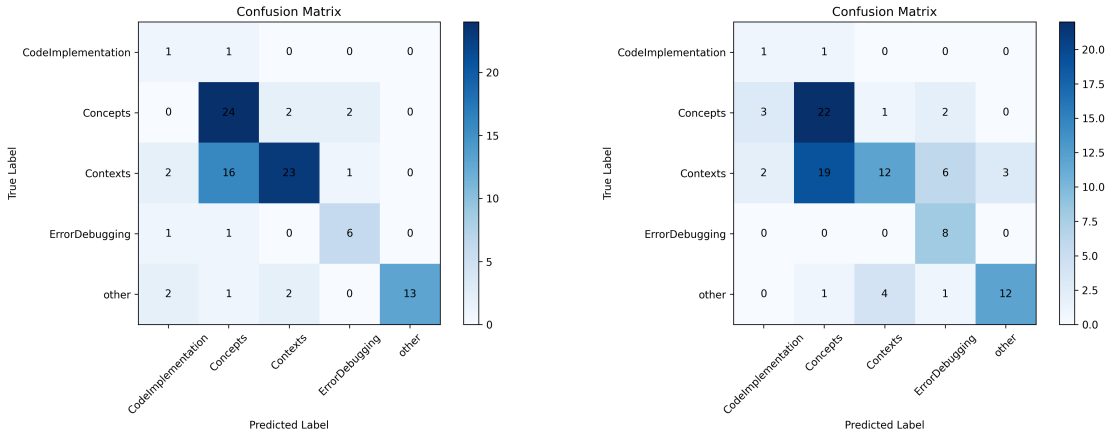
Table 4.3: The results of Models Comparison

The fine-tuned classifier RoBERATa-base achieved moderate and stable performance (Accuracy=68%) on case study data. In contrast, some language models (deepseek-r1:32b, phi3.5:latest, llama3.2:latest, gemma2:27b) exhibit unreliability during runtime and sometimes generate the model's thinking process instead of just generating labels as the prompt requested. Furthermore, the rest of language model-based classifiers (gemma3:27b, gemma3:12b-it-qat, gemma3:4b) exhibit variability by observing the standard deviations.

To further examine performance across the five categories, confusion matrices were created for the fine-tuned classifier (Figure 4.2a) and gemma3:27b the language model-based classifier with the highest accuracy (Figure 4.2b). Both approaches showed comparable performance across most categories, but a difference emerged in the category of "Contexts" questions.

Both models struggled to accurately classify "Contexts" questions, exhibiting low recall and frequent misclassifications into adjacent categories. The fine-tuned classifier demonstrated higher accuracy for this category compared to gemma3:27b, showing a stronger concentration of correct predictions on the diagonal of the confusion matrix. In contrast, gemma3:27b is more likely to misclassify "Contexts" questions as "Concepts" questions and "ErrorDebugging" questions.

For the remaining categories, two approaches achieved similar classification performance. The confusion matrices show that misclassifications are limited to "Contexts" category rather than random errors, suggesting that both models capture the distinctions between different types of student questions.



(a) Confusion Matrix for Fine-tuned Classifier

(b) Confusion Matrix for gemma3:27b

Figure 4.2: Darker colors means a higher number of instances, with diagonal cells representing correct classifications, and the rest of cells representing misclassifications.

4.4. Discussion

The results show that the fine-tuned classifier can provide a consistent and stable classification of student-AI interactions in the context of machine learning assignment. While the classifier does not achieve perfect accuracy (68%), its stability and consistency with predefined categories make it suitable for learning analytics. Moreover, it outperformed most language model-based classifiers in classifying the case study dataset. These models either generated unexpected output or may produce different output for the same input. The dashboard relies on aggregated interaction trends in this context, so inconsistent classification can affect the category distributions. This may potentially affect educators' understanding of learning difficulties. In general, the results support our decision to fine tune a classifier and integrate it into analytics pipeline rather than relying on language models' reasoning.

The confusion matrix analysis highlights the limitation of automated classification in learning analytics based on student-AI interactions. The difficulties both models experienced in classifying "Contexts" questions may reflect the ambiguity of the category itself. By observing the labeled interactions, we found that some "Contexts" questions indeed included "Concepts" questions, and interactions often contain multiple considerations: concept understanding, code implementation, and information clarification. This situation make single label classification difficult and ambiguous. This suggests that single label classification may not fully capture information in questions, indicating the potential value of multi label methods in future work.

Although the model was fine tuned on interactions collected in similar educational context, the performance is not perfect when the classifier was applied to our case study data, particularly for the ”Contexts” category. Classifications of student questions require not only a sufficient total amount of data, but also that each category is sufficiently representative and supported by enough instance to reduce sensitivity to noise. In our case study, the limited number of collected interactions (N = 99) restricted both diversity and balance of categories. To provide more stable and reliable classification in the future, a larger and more representative dataset needs to be collected in the context of machine learning course. It is reasonable to estimate that several hundreds to thousands of student questions collected on multiple assignments or a full course is necessary to support the classification.

Overall, this classifier can serve as a practical component to support learning analytics dashboard. We presented the learning analytics through classification of student-AI interactions into questions types. The type of question represents an indicator of student engagement and difficulty which reflects how students seek help. While this approach enables interpretable and scalable analysis, the teaching effectiveness of these analytic indicators has not yet been directly evaluated. Therefore, future research should explore the relationship between question type indicators and teaching outcomes, instructional decisions.

4.5. Limitations

The fine-tuned classifier and the evaluation have several limitations. Firstly, the classifier was trained on a public dataset rather than the interactions collected during the case study. Since the number of interactions in case study is limited (N = 99), this approach is necessary but introduces a domain shift effect, which may reduce performance. Larger, course-specific datasets will help fine-tune them to better suit the machine learning course context. Secondly, some interactions can indeed be categorized into multiple categories (e.g., questions that seek both conceptual clarification and code implementation). Therefore, classification errors cannot be simply explained as technical failure. Finally, the evaluation dataset is relatively small, which limits the accuracy of performance evaluations.

5

Dashboard Design and Expert Review

5.1. Objective

The objective of the learning analytics dashboard is to transform classified student-AI interactions into actionable learning analytics to support teachers in monitoring student common questions and timing in solving assignments in a machine learning course. This aligns with the main research questions of this thesis: **How can student-AI interactions be leveraged to provide teacher-facing learning analytics in machine learning education to support identify student difficulties in assignments?**. Although prior work has shown the potential to connect learning analytics dashboard with AI-supported learning activities [27, 36], there remains limited research in the context of machine learning education. This dashboard aims to bridge this gap by presenting student learning difficulties in machine learning assignments in a meaningful way to teachers.

5.2. Methodology

The dashboard design and implementation follow the process for System Development Research [35]. This iterative process included 5 stages: (1)Construct a Conceptual Framework, (2)Develop System Architecture, (3)Analyze and Design the System, (4) Build the (Prototype) System, and (5)Observe and Evaluate the System. It is well suited for our study because we aim to understand student-AI interactions and transform these interactions into actionable teacher-facing learning analytics. The development of the dashboard also involves creating it based on theory, implementing functions, and evaluating how users interact with it to obtain meaningful insights. The following subsections describe each stage.

5.2.1. Conceptual Framework

Previous studies have identified that machine learning education has unique learning challenges, including conceptual reasoning, practical implementation [44], and different background knowledge [45]. Most existing machine learning research focuses on applications, so there is little research on machine learning education and how teachers can monitor students' learning difficulties. Tu et al. show that LLMs can be used as learning assistance in study, not only to obtain solutions, but also to clarify concepts and provide personalized suggestions [49]. However, they also shift much of the learning process to the interaction between students and LLMs. Since student-AI interactions occur by default within commercial tools, teachers lack reliable telemetry data. Even such data are accessible, manually analyzing conversations during the course is time-consuming and affects daily teaching [28]. Therefore, a tool can collect and analyze student-AI interaction data, and help teachers understand the information necessary for machine learning education with the use of LLMs.

Learning Analytics Dashboard (LAD) has been proposed as a tool to support people in better understanding information and making decisions by visualizing students data [50]. Some research explores how to use LAD to help teachers monitor [36], analyze and understand student-AI interactions [27]. Early research on teacher-facing LAD focuses on providing educators with aggregated insights derived from LMS [23]. Therefore, this still lacks relevant research in the field of machine learning education.

These literature motivated the design goal of a dashboard that transforms classified student-AI interaction data

into actionable learning analytics for teachers in the context of machine learning course.

5.2.2. Design Requirements

Due to time constraints, a meeting was held with two educators to discuss the design requirements of dashboard instead of conducting formal and large-scale interviews. We listed 18 questions that should be answered by dashboard and divided them into three tabs: Course overview tab, Assignment overview tab, and Student tab.

Table5.1 presents the questions we listed.

Table 5.1: Question to be answered by dashboard

Tab	Question to be answered	Figure
Course Overview	In which assignment there were most questions?	Figure5.2
Course Overview	What category has the most problems across all assignments?	Figure5.2
Course Overview	How many questions do students ask on average per category across all assignments?	Figure5.3
Course Overview	How many questions do students ask on average per assignment across all assignments?	Figure5.3
Course Overview	From the perspective of each assignment, what is the trend of changes in the number of questions for each category?	Figure5.4
Assignment Overview	What are summaries and suggestions for a specific assignment?	Figure5.5
Assignment Overview	For a specific assignment, which category of questions was asked the most?	Figure5.5
Assignment Overview	How many questions do students ask on average per category in a specific assignment?	Figure5.5
Assignment Overview	Who asked the most questions? Who asked the fewest?	Figure5.5
Assignment Overview	When do students work on a specific assignment?	Figure5.6
Assignment Overview	What are the keywords mentioned in a specific assignments?	Figure5.6
Assignment Overview	What are the specific questions regarding the keywords?	Figure5.8
Student Overview	How many questions did this student ask?	Figure5.9
Student Overview	When was the last time they asked a question?	Figure5.9
Student Overview	For a specific student and assignment, which category of questions was asked the most?	Figure5.10
Student Overview	In specific assignments, how many questions did students ask compared to the average number asked by the whole class?	Figure5.10
Student Overview	How many questions did students ask for each assignment?	Figure5.9
Student Overview	When do a specific student work on a specific assignment?	Figure5.10

5.2.3. System pipeline

Based on these design requirements, we designed the system pipeline and it is presented in Figure 5.1.

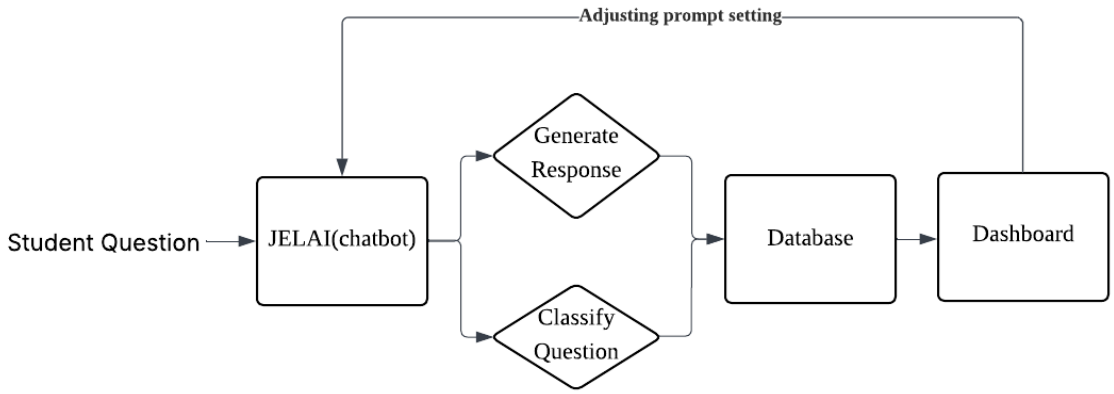


Figure 5.1: Pipeline of JELAI and Dashboard

To ensure the dashboard always displays the latest data, the dashboard was expected to be embedded into the JELAI system. Once students' problem is sent to the chatbot, it will be processed through the language model or fine-tune classifier, which categorizes each question into a pre-defined label. After that, data will be stored in a SQLite database to simplify deployment and integration with the JELAI system. Each interaction record contains six key attributes:

- **student_id**:students' user name.
- **timestamp**:the time of question submission or response generation.
- **message_text**:students' open-text question or AI-generated responses.

- **message_type**:Two type: question or response.
- **message_classification**: the label assigned by language model/classifier indicating the question category.(e.g. Concepts, CodeImplementation, ErrorDebugging, Contexts, other).
- **assignment_name**:The corresponding assignment name.

The dashboard is divided into four tabs based on the questions listed 5.1. The first tab is the course level overview, which displays the overview of all questions and corresponding categories for all assignments in the course. The second tab is the assignment level overview, where teachers can select assignments to view an overview of each assignment. It presents the questions information such as number of questions, questions distribution, keywords. In the third tab, teachers can view each student's information by searching their ID and selecting assignment name. This tab includes the questions information of selected student (e.g. number of questions, questions distribution), comparison of number of questions with class average, and active time. In the fourth tab, teachers can modify the prompt instruction(e.g. system prompt, learning objectives, and assignment description) to manage the student-AI interaction.

5.2.4. Dashboard Prototype

The design of dashboard prototype was inspired by Few's design principles for dashboard [12], specifically the following:

- Dashboards are visual displays.
- A dashboard fits on a single computer screen and is used to monitor information at a glance.
- Information in the dashboard must be customized to the specific needs of users.

Therefore, the prototype will primarily use visualization to display data and analysis, with text serving as supplementary information. Furthermore, we ensure that the most important information can be displayed on a single screen and seen at a glance. After reviewing the real student-AI interaction collected in the case study, and design requirements, we decided that the dashboard focus on the comparison of information from different levels to highlight the learning difficulties, such as an assignment versus course or a student versus class average. Bar charts were chosen as the primary visualization tool because they facilitate data comparison and display of data distributions.

The prototype was developed by using python library *Plotly Dash* [37]. It was chosen for its flexibility and compatibility with the JupyterHub. The dashboard was embedded into the JELAI system as a JupyterHub service, and teachers could access the dashboard by logging in JELAI with the admin account.

In the following sections, we will present visualizations implemented for prototype, and the question they answered. Table 5.1 reflects the correspondence between the questions dashboard needs answer and visualization. The full screenshots of the dashboard can be found in GitHub repository.

Course Overview Tab

The plot on the right side of Figure 5.2 presented the visualization for the question: "In which assignment there were most questions?". It is a bar chart with the assignment name on the x-axis and the number of questions on the y-axis. Each bar has a corresponding total number of questions asked.

For the question:"What category has the most problems across all assignments?", both text and visualization were used to present the information. The summary text (Figure 5.2 Top) provided information such as the total number of questions asked for the course, the number of students using chatbot, and questions distribution.

A bar chart (Figure 5.2 left) with the category name as the x-axis and the number of questions asked as the y-axis was used to support the summary text.

For questions:"How many questions do students ask on average per category across all assignments?" and "How many questions do students ask on average per assignment across all assignments?", we used two bar charts (Figure 5.3), one with the assignment name on the x-axis and the other with the category name on the x-axis, both with the average number of questions asked on the y-axis.

The bar chart on the right side of Figure 5.4 used the assignment name as the x-axis and the number of questions asked as the y-axis, then presented question distribution bars grouped by assignment name. Teachers can clearly see the distribution of questions in each assignment and the changes in the number of questions in different

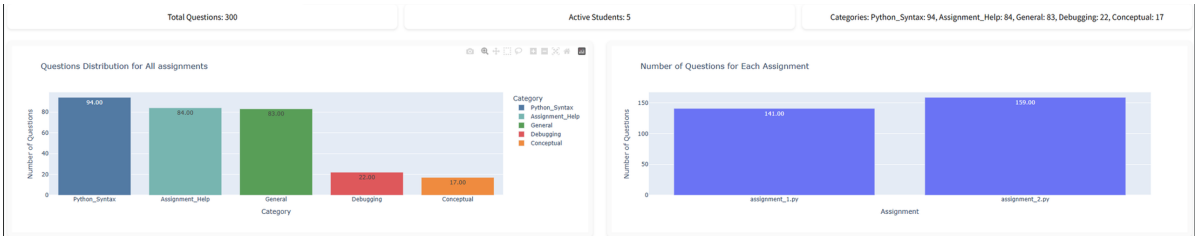


Figure 5.2: Course Overview Tab : summary text (top) about the total number of questions asked for the course, the number of active students, and questions distribution; bar chart (left) with the category name as the x-axis and the number of questions asked as the y-axis; bar chart (right) with the assignment name on the x-axis and the number of questions on the y-axis.

Figure 5.3: Course Overview Tab : bar chart (left) with the assignment name on the x-axis, and the average number of questions asked on the y-axis; bar chart (right) with the category name on the x-axis, and the average number of questions asked on the y-axis

assignments. It answers the questions: ”From the perspective of each assignment, what is the trend of changes in the number of questions for each category?”.

Moreover, We created a bar chart (Figure 5.4 left) where the x-axis represents the category name and the y-axis represents the number of questions, and then grouped the number of questions by category name. This plot compared with right side plot, and we expected to understand teachers’ preference for these two different visualizations.

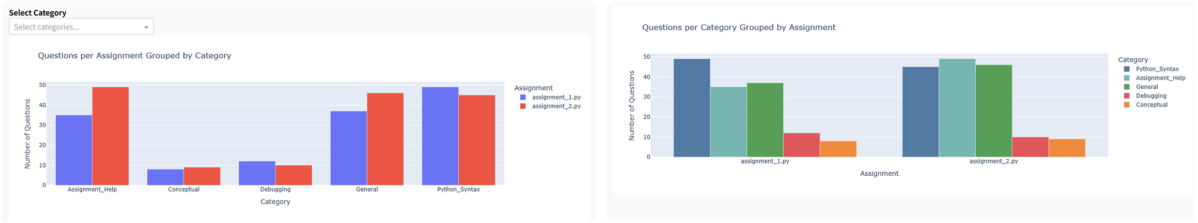


Figure 5.4: Course Overview Tab : bar chart (right) presented questions distribution bars grouped by assignment name; a bar chart (left) with category name as x-axis and number of questions asked as y-axis, number of questions asked grouped by category name

Assignment Overview Tab

To answer the question ”What are summaries and suggestions for a specific assignment?”, a rule-based function and a summary text (Figure 5.5) were implemented to present the analysis. It compared the selected assignment with the course average and provided suggestions based on this comparison.

Bar chart (Figure 5.6 left) was used to visualize the question:”For a specific assignment, which category of questions was asked the most?”. It helps us intuitively identify the category with most questions and compare it with other categories.

We implemented a boxplot (Figure 5.6 right) to answer two questions: ”How many questions do students ask on average per category in a specific assignment?”, and ”Who asked the most questions? Who asked the fewest?”. It presents the average question of each category and the outlier values, which help teachers identify the student who needs help.

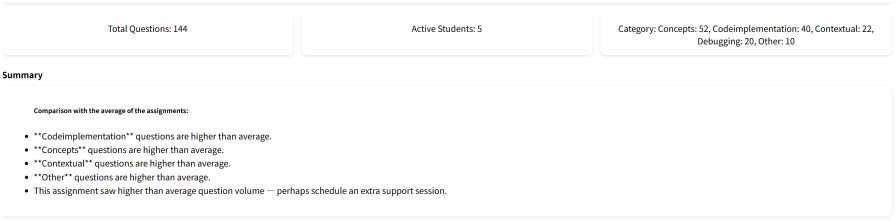


Figure 5.5: Assignment Overview Tab : a summary text about comparison between the data from selected assignment with the course average and provided corresponding suggestion

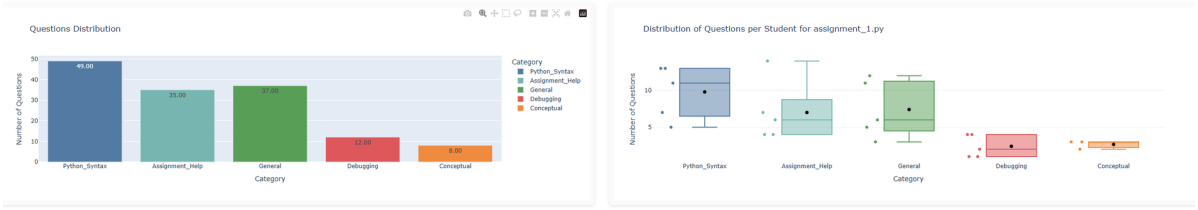


Figure 5.6: Assignment Overview Tab : bar chart (left) about questions distribution; boxplot (right) for presenting questions distribution per student for a selected assignment

A bar chart (Figure 5.7 left) uses calender week as the x-axis and number of questions asked as the y-axis to answer:"When do students work on a specific assignment?". Teachers can observe changes in the total number and category of student questions over time.

Furthermore, a word cloud (Figure 5.7 right) and a question table (Figure 5.8) were presented to answer questions: "What are the keywords mentioned in a specific assignment?" and "What are the specific questions regarding the keywords?". Teachers can extract keywords from word clouds and then use search functions to discover the problems corresponding to high-frequency words, helping them understand the learning difficulties.

Figure 5.7: Assignment Overview Tab : a bar chart (left) used calender week as x-axis and number of questions asked as y-axis; a wordcloud (right) present the keywords from questions

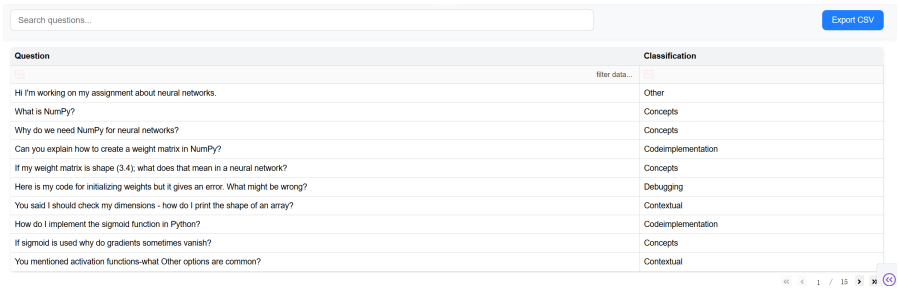


Figure 5.8: Assignment Overview Tab : a question table allowed filter and search

Student Overview Tab

A summary text (Figure 5.9 top) was used to answer following questions:"How many questions did this student ask?" and "When was the last time they asked a question?". Teachers need to select the assignment and student id to show the corresponding information.

The question about number of questions ask:"How many questions did students ask for each assignment?" was answered by a bar chart (Figure 5.9 left) with assignment name as the x-axis and number of questions student asked as the y-axis.

A pie chart (Figure 5.9 right) was implemented to answer the question about questions distribution: "For a specific student and assignment, which category of questions was asked the most?". It is different from visualizations about questions distribution of other tabs, since we aim to obtain teachers' preferences about which type of visualization is more suitable for observing the distribution.

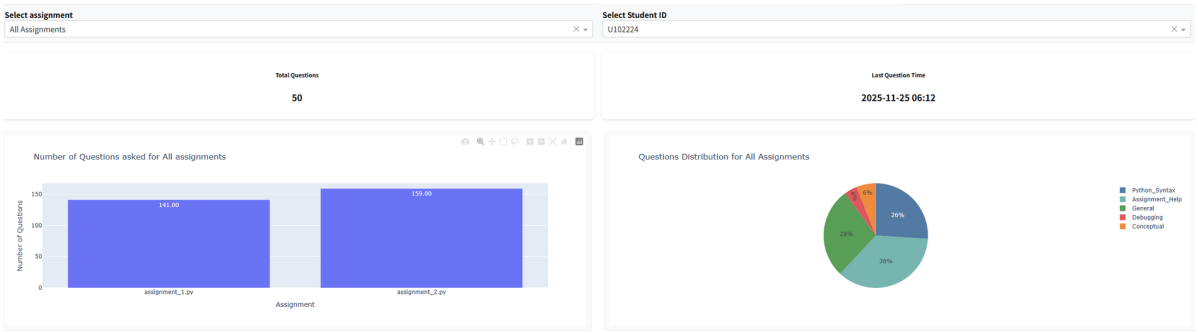


Figure 5.9: Student Overview Tab : summary text (top) about number of questions and latest questioning time; a bar chart (left) with assignment name as the x-axis and number of question student asked as the y-axis; a pie chart (right) about questions distribution

Plot on the left side of the Figure 5.10 answered the question:"In specific assignments, how many questions did students ask compared to the average number asked by the whole class?" A grouped bar chart was used to clearly compare the questions student asked with the class average.

For answering the last question:"When do a specific student work on a specific assignment?", we decided to use bar chart (Figure 5.10 right) with calender week as the x-axis and number of questions asked as the y-axis to present changes in number of questions and categories.

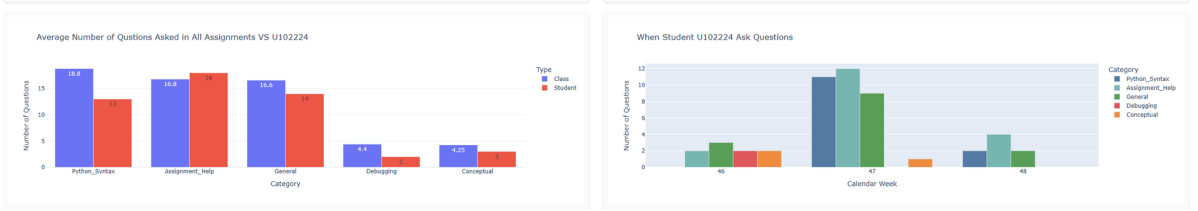


Figure 5.10: Student Overview Tab : A grouped bar chart (left) compare the questions student asked with the class average; a bar chart (right) with calender week as the x-axis and number of questions asked as the y-axis to present changes in number of questions and categories

5.2.5. Expert Review

The objective of expert review is to examine whether the dashboard prototype effectively helps teachers understand the learning difficulties students encounter in machine learning assignments. This aligns with the research question: **How do teachers perceive the usefulness of the learning analytics dashboard based on student-AI interaction for monitoring student learning and difficulties in assignments?**.

Previous studies emphasized the importance of involving teachers in the teacher-facing LAD design process [24, 50]. Therefore, we aim to collect qualitative and quantitative feedback from teachers regarding their perspectives, usability, and usefulness of the LAD. Since we did not conduct a large-scale interview to collect educators' requirements about dashboard before designing dashboard architecture due to time constraints, teachers' expectations of dashboard will be discussed during the review. Their feedback will help the dashboard better meet teachers' needs.

Participants

This expert review session involved a group of 5 researchers and 2 master students from the Faculty of Electrical Engineering, Mathematics and Computer Science at TU Delft. Each of them has prior knowledge of machine

learning or learning analytics and some of them are teachers who have prior experience in grading or providing feedback on student assignment. Therefore, we believed they can provide valuable insights to help us improve our learning analytics dashboard.

Outline of Expert Review

The session was divided into seven sections and lasted one hour. In the welcome sections, participants were asked to agree with informed consent to the audio recording during the session, and introduced the ground rules and outline of the focus group. The section Introduction of study and the section Introduction of dashboard aim to familiarize participants with the study's purpose and learning analytics dashboard. We described the contexts set for the study and dashboard: In a bachelor machine learning lab, students need to work on two assignments consists of two parts: an open-ended question related to the concept and a programming task.

In the next section, participants were asked to share their expectations of each dashboard tab based on context described, and a question was used to guide thinking and discussion: "If you were a teacher of machine learning, what kind of questions would you want the dashboard to answer?". This section aims to collect their needs for a learning analytic dashboard as teachers, and prepare for the final discussion section.

The next section explores the dashboard prototype. Three scenarios were provided to participants:(1)Obtain an overview of course assignments, search for a specific assignment on the Assignment Overview Tab, and check corresponding analysis presented. (2)Checking the assignment on the Assignment Overview Tab, search for averages and outliers students, then obtain an overview of the outliers and check the interaction. (3) Modify the prompt settings that will be applied to the student's interaction with the chatbot based on understanding of the presented analysis. Meanwhile, we present the list of design requirements (Table 5.1) to participants. They were asked to check whether the prototype answers those questions.

The sixth section is the final discussion, which aims to collect detailed feedback about list of design requirements, the visualizations and analysis methods presented in dashboard. The discussion was guided by the following questions:

- Which visualizations answer which question? If you feel they don't answer, why?
- What are the strengths and weaknesses of each dashboard tab? how might we improve the weaknesses?
- After the scenario, is there any information you wish the dashboard could have that it could not currently?

Finally, participants were asked to complete a quantitative survey on usefulness and usability adapted from TAM [8] and SUS [5]. The survey could be found in the Appendix GitHub repository.

The following is the outline of the focus group session:

- Welcome: Confirming the audio recording, introducing Ground Rules, and outline of session
- Introduction of study: Presenting the purpose of study, and context of data shown
- Introduction of dashboard: Presenting the workflow of dashboard and demonstration of dashboard.
- Discussion - Expectations for the dashboard: participants write down and discuss the following question: If you were a teacher of machine learning, what kind of questions would you want the dashboard to answer?
- Exploration: Three scenarios: participants explore the dashboard guided by scenario described, and answer does the dashboard answer questions?
- Discussion: Further discussion about questions, the visualization, and analysis methods presented in the dashboard
- Survey - Participants complete a survey about usefulness and usability.

Ethical considerations

The focus group involved audio recording, so the informed consent was provided to all participants at the beginning of the session. Recording audio aims to capture participants' insights. The recordings are used strictly for analysis by the research team, and participants will be described in general term in thesis to protect their identity. All participants retain the right to withdraw from the study at any time without penalty. The informed consent can be found in GitHub repository.

5.3. Result

5.3.1. User Expectation of Dashboard

In this part of discussion, participants were asked to share their expectations for the each of three tabs(Course Overview tab, Assignment Overview tab, and Student Overview tab) of learning analytic dashboard according to the question: If you were a teacher of machine learning, what kind of questions would you want the dashboard to answer? Their answers were analyzed and summarized into different topic categories, duplicates were removed, and similar ideas were grouped together.

Course Overview tab

Participants expected this tab to provide a brief overview of course and students' questions which support teacher rapidly understand learning status. The following is the list of summarized expectations:

- **Basic course information** (e.g.,"Which assignment has most questions?", "How many questions asked per assignment for the course?", "How many student participant,")
- **Questions distribution of assignment** (e.g.,"in which assignment were the most questions about programming and least about machine learning concepts?")
- **Concepts involved in questions** (e.g.,"Which concept was the most difficult to understand?", "In terms of concepts (e.g., as keywords for assignments), which concepts do students find the easiest / most difficult?","Are there questions about topics not covered in lectures?")
- **Competencies needs for course** (e.g.,"In terms of competencies (e.g., scientific programming), which competencies do students find the easiest / most difficult?")

Assignment Overview tab

In this tab, participants want more assignment-level insights about the student-AI interaction(questions asked and responses generated) and student behavior patterns. The following is the common expectations of this tab:

- **Basic assignment information** (e.g.,"What was the most asked question in this assignment?")
- **Questions distribution of assignment** (e.g.,"What type of questions are being asked? Are they topic related, To brainstorm solutions?")
- **Concepts involved in questions** (e.g.,"Which topics are getting the most questions?")
- **Pattern of student behaviors** (e.g.,"When are students most active in this assignment?", "Are students having a hard time understanding this assignment?")
- **Satisfaction of responses** (e.g.,"By analyzing the conversations, are the students satisfied with the answers they asked for this assignment?")

Student Overview tab

In summary, participants expected an overview of student learning status, such as prior knowledge, questions distribution and questioning pattern. The expectations were reported as follows:

- **Student background information** (e.g.,"What is the student's level of prior knowledge?")
- **Distribution of questions** (e.g.,"Which category of the question is the most asked?", "What types of questions is this student asking the most? i.e. confirmation questions, concepts, or just asking for solutions", "What is the most common question asked for this assignment?","I would like to know in which weeks student asked the most questions about the assignment 1")
- **Pattern of student questioning** (e.g.,"Does this student ask more questions about the assignment or the content taught?","How consistent is a student with their questions?","When does this student typically ask questions about assignment? (e.g., in the week the assignment is introduced, after a week, 1-2 weeks before exam)", "does student who ask many programming questions at the beginning of the course, have less programming questions later on?", "does student who ask many programming questions at the beginning of the course, have less programming questions later on?", "Does the student repeat questions across weeks(if programming related)?")

5.3.2. Assessment of Design Requirements

During the dashboard exploration, participants were asked to check whether the predefined design requirements were met by using a checkbox. Participants indicated most of design requirements were met successfully, particularly the course overview tab. However, all participants expressed that two questions were not answered by dashboard:

- Assignment Overview - Who asked the most questions? Who asked the fewest?
- Student Overview - When does a specific student work on a specific assignment?

When implementing the visualization for answering the first question, we considered it from the perspective of each category: who asked the most questions about one category and who asked the fewest about one category. However, we ignored the fact that our problem was posed from the perspective of the total number of problems, rather than from the perspective of each category. Participants suggested that the existing plots can be preserved, but a summary text can be added before plots to answer these questions. They mentioned that this question was their the biggest concern, as it allowed them to directly identify which students needed the most help.

The reason why the second question remained unanswered was that we confused two things: students work on assignments, and students ask questions. Participants mentioned that the plot about "when students ask questions" does not directly reflect a student's actual working hours. While the time of students ask questions is important, they are more interested in when students are actually working, which helps them adjust their teaching strategy.

A summary table5.2 is presented as follows:

Tabs	Design requirements	Yes	No
Course Overview	In which assignment there were most questions?	7	0
Course Overview	What category has the most problems across all assignments?	7	0
Course Overview	How many questions do students ask on average per category across all assignments?	7	0
Course Overview	How many questions do students ask on average per assignment across all assignments?	7	0
Course Overview	From the perspective of each assignment, what is the trend of changes in the number of questions for each category?	5	0
Assignment Overview	What are summaries and suggestions for a specific assignment?	6	0
Assignment Overview	For a specific assignment, which category of questions was asked the most?	7	0
Assignment Overview	How many questions do students ask on average per category in a specific assignment?	7	0
Assignment Overview	Who asked the most questions? Who asked the fewest?	0	7
Assignment Overview	When do students work on a specific assignment?	5	0
Assignment Overview	What are the keywords mentioned in a specific assignments?	5	0
Assignment Overview	What are the specific questions regarding the keywords?	6	0
Student Overview	How many questions did this student ask?	7	0
Student Overview	When was the last time they asked a question?	7	0
Student Overview	For a specific student and assignment, which category of questions was asked the most?	7	0
Student Overview	In specific assignments, how many questions did students ask compared to the average number asked by the whole class?	4	2
Student Overview	How many questions did students ask for each assignment?	7	0
Student Overview	When do a specific student work on a specific assignment?	0	7

Table 5.2: The result of predefined design requirements checking

5.3.3. Summary of discussion

This section described the participants' open-ended feedback on their experiences of using the dashboard. Feedback has been grouped by topic, and duplicates have been removed to avoid repetition.

Theme 1: Visualization and Clarity of Information

Several participants commented on the dashboard visualization and clarity of information. The first comment is that there are so many plots that new users may not know what they are or which plot is the most useful. This could result in users needing more time to become familiar with or obtain information. They prefer that the dashboard provides a text summary before plots to give a more intuitive understanding of the overall picture.

Besides that, they found that the bar charts might be overused, especially in course overview tab. Using too many bar charts on a page could be confusing and overwhelming. One participant mentioned that he thought some of the bar charts were the same while browsing up and down. Based on this comment, they made further suggestions. For instance, the bar charts about the average can be given a range, and the bar charts about trend of change could be replaced by relational visualization like sankey diagram.

Moreover, participants noticed the assignment overview tab did not directly display the information about which student asked the most or the fewest questions. They expressed this information can greatly help them understand which students need the most help. Furthermore, they mentioned that the data points representing outliers in

boxplot should allow teachers to click and be redirected to corresponding student overview tab. The current implementation of first remembering the student ID and then choosing ID in another tab is a bit too complicated.

Lastly, participants noted the word cloud of question keywords may not be effective enough. It contains a lot of useless words and teachers were not directly informed about concepts student struggled. Instead of using keywords, it might be better to cluster the questions according to the concepts mentioned in the questions.

Theme 2: Depth of analytics

The participants agreed that the analyzes presented on the dashboard can help the teacher to be aware of the learning difficulties and make the instructional decision, but also expressed interest in viewing more detailed information or exploring the data in more detail.

One participant mentioned that the dashboard should include the prior knowledge level and grades of the selected student, which can support teachers identify the students who need the most help. Besides that, there was a discussion about the bar chart presenting ”When students ask questions”. Participants would like to know not only when students ask questions, but also when students work on assignments. They expected to get information such as whether students are working on assignments or asking questions before the exam, or whether they ask questions after the assignment is introduced.

Another suggestion concerns student questioning patterns. Some participants would like to have analysis about student questioning pattern changed during the course. For instance, the analysis should show whether there are students repeatedly asking similar questions during the course, or whether there are students who have many programming questions at the beginning of the course but have few related questions by the end of the course. This helps them understand the student’s learning process.

Theme 3: Prompt setting tab

Participants shared their opinion about the prompt setting tab. Some of them suggested that the responses generated by AI should also be provided. Without the responses, it is difficult to determine if there is a problem with student-AI interaction. Moreover, they mentioned that the dashboard should provide teachers with some guidelines on how to construct prompts.

Another suggestion is to remove the prompt setting tab from the learning analytic dashboard and make it a separate dashboard. They mentioned the visualization presented may not be sufficient for adjusting the prompt setting. They would like to have a separate dashboard focus on analyzing the generated answers to different questions, and modifying the prompts.

5.3.4. Perceived Usefulness (TAM-PU)

Table 5.3 shows the mean score and standard deviation of the 7-Item TAM scale. The scale produced an overall mean of 3.43 (SD=0.68), which indicates a moderately positive perception of the dashboard usefulness. While most perceived usefulness (PU) items received ratings between ”neutral” and ”agree”, **’The dashboard helps me better manage student interactions with AI.’** received a significant number of ”neutral” and ”disagree” ratings. This suggests that the system may not have fully considered this specific dimension of perceived usefulness.

PU Item	**Mean**	**SD**
The dashboard helps me make better instructional decisions.	3.86	0.69
The dashboard helps me identify students who need support.	3.14	0.90
Using the dashboard improves my ability to understand student’ progress.	3.43	0.79
The dashboard helps me understand overall course status.	4	0.58
The dashboard helps me understand the status of each assignment.	3.57	0.53
The dashboard makes it easier to identify question patterns or trends in students’ assignments.	3.86	0.38
The dashboard helps me better manage student interactions with AI.	2.14	0.9
Overall PU	**3.43**	**0.68**

Table 5.3: Mean and Standard Deviation of 7-Item TAM scale

5.3.5. Usability (SUS)

Table 5.4 shows the result of the System Usability Scale (SUS). The dashboard achieved a mean SUS score of **72** (SD=9.5) which exceeds the average score of 68. According to established SUS interpretation guidelines, this

is equivalent to a "good" usability rating (Grade B), indicating that participants generally find the system easy to use with few aspects that could be improved.

Participant	SUS Score
P1	72.5
P2	57.5
P3	72.5
P4	70
P5	70
P6	90
P7	72.5
Mean	**72**
SD	**9.5**

Table 5.4: The SUS score of each participant

5.4. Discussion

The results of the review indicate that participants generally perceived the interaction-based analytics and dashboard as useful to support their understanding of students' learning difficulties in machine learning assignments. Participants reported that the dashboard provided insights into the questions and its types students asked when interacting with chatbot, which was not reflected in traditional assessment materials. This aligns with previous research suggesting that teacher-facing learning analytics dashboard can enhance understanding of student difficulties and provide information to support or adjust learning design or learning materials [24]. This work extends research by demonstrating that learning analytics can be derived from student-AI interactions. In the field of machine learning education, where conceptual difficulty and programming problem solving often intersect, this interaction based analysis may be particularly important.

Participants also highlighted several limitations and areas in which the dashboard could be improved. They expressed a desire for in-depth analysis, such as question clustering, pattern of student questioning, and relate interactions to assignment performance. These findings indicate that while summaries are valuable, teachers also require detailed insights to support identify student difficulties and instructional decision making.

Related studies further indicate that user differences (including users' domain, visualization experience, data literacy [40] and educational objectives [50]) affect how dashboards are perceived and used. This was also reflected in the review session, where some participants preferred simple visual summaries, while others requested in-depth analytical capabilities. This further confirms the literature's suggestion for dashboard design to revolve around clearly defined user groups and usage scenarios [50, 24], rather than assuming that one design can meet the needs of all stakeholders [21].

Overall, the review results indicate that demonstrating learning analytics derived from student-AI interactions is both feasible and pedagogically meaningful, and the dashboard is useful for supporting understanding of students' learning difficulties in machine learning assignments.

5.5. Implications for Dashboard Design

The review highlights several implications for future iterations of the dashboard.

5.5.1. Visualization

Some participants emphasized the need to provide summary texts to support the plots. They suggested that all tabs should provide summary text, similar to the assignment overview tab, to provide a quick overview.

Moreover, they mentioned that more diverse visualizations can be used instead of bar charts. The overuse of bar charts can confuse and overwhelm teachers. However, participants did not discuss the specific visualization details that should be used. An interview with more educators can be held to define the requirements of dashboard.

5.5.2. In-depth analysis

Participants mentioned the in-depth analysis can support their teaching strategy. The assignment overview tab can add a component that clusters questions by the concepts mentioned in the questions, instead of using word clouds.

Furthermore, the system should record the time when student interacts with notebook, then analyze the students' questioning patterns based on the recorded time.

5.5.3. Richer context

Some participants suggested that the responses generated by AI should be provided. The student overview tab could expand the question table into interaction table, and then indicate the time and which conversation the interaction refers to.

5.5.4. Prompt setting

Participants agreed that more information might be needed to modify the prompts, such as a guideline about how to instruct prompts and AI-generated responses; or it would be best to create a separate dashboard to analyze the prompt design. This feedback indicated that while the current prototype demonstrated the usefulness and usability of learning analytics based on student-AI interaction, its analytical depth remained limited relative to teachers' expectations and information needs.

5.6. Limitations

This dashboard design and review have several limitations. Large-scale interviews were not conducted to collect teachers' opinions or needs in the early stages of dashboard design. This limits the generalizability of the prototype and findings. The expert review of the prototype focused on perceived usefulness and usability, which provides some insights into feasibility and quality of the dashboard, but it does not measure improved teaching practices and learning outcomes. kaliisa et al. found that a large number of studies have deployed learning analytic dashboards only in the form of prototypes and small-scale pilots [24]. They suggested that it is difficult to truly assess dashboards' actual impacts if they are implemented in controlled environments. Therefore, future work should examine dashboard usage over longer periods and on larger scales, and investigate how the analytics influence actual instructional decisions and student learning.

6

Discussion

6.1. Answer to Research Questions

How can student-AI interactions be automatically and consistently classified into question types to support learning analytics?

To address this question, a transformer-based classifier was developed and compared with language model-based classifiers. The result shows that the classifier trained on a publicly available but similar contextual dataset, can generate sufficiently stable and consistent labels for analytical purposes, even when applied to our limited case study data.

The language model-based classifiers demonstrated reasonable performance. However, they exhibited inconsistency, making them difficult to use in reproducible analytics pipelines. In contrast, the classifier provides stable outputs consistent with predefined categories, making them more suitable for aggregation and visualization.

How do teachers perceive the usefulness of the learning analytics dashboard based on student-AI interaction for monitoring student learning and difficulties in assignments?

The results of dashboard evaluation show participants generally viewed the learning analytics dashboard as useful and effective for monitoring student learning and difficulties in assignments, and supporting their teaching, but they also pointed out the areas for improvement, such as better visualization, more in-depth analysis and better prompt setting process. They suggested the dashboard should provide more summary texts instead of plots only, and navigation or visual density can be further simplified. Besides that, they preferred the dashboard should include a deeper analysis about student questions (questions clustering) and students' behavioral patterns. Some participants mentioned more information may be needed to modify the prompts, such as instructions and AI-generated responses, or it would be best to create a separate dashboard to analyze the prompt design.

Main-RQ: How can student-AI interactions be leveraged to provide teacher-facing learning analytics in machine learning education to support identify student difficulties in assignments?

The findings suggest that students-AI interaction can serve as a valuable data source for learning analytics when they are systematically structured, classified, and summarized. Categorizing interactions based on types of question enable teachers to identify students' difficulties in assignment, such as concept clarification, debugging requests or needs for code implementation.

Integrating these analytics into a dashboard allowed teachers interpret learning difficulties and question trends at course, assignment and student level. This suggests that data on student-AI interactions can complement traditional learning analytics, providing insights into students difficulties rather than just their final assignment submission.

6.2. Implications

The results of our study suggest several implications. Categorizing student-AI interactions into question types can provide educators with a structured overview of student's learning difficulties. They can observe aggregated pat-

terns of student questions instead of reviewing student-AI interaction transcripts. Such patterns, such as whether students mainly ask conceptual questions, code implementation questions, or error debugging questions. Such patterns may help them understand the status of assignments and make better instructional decisions.

The observed variability in classification performance indicates that the dashboard should be used as supporting information rather than as precise quantitative assessments. For example, an increase in conceptual questions may suggest that course content needs to be adjusted, but further analysis is still needed. Moreover, this also emphasizes that the classification accuracy in a single run is insufficient as a criterion of design. Stability and consistency across repeated runs are also important when analytics are aggregated into dashboards.

Furthermore, the case study demonstrated that small-scale deployments can provide valuable insights into student-AI interactions. This supports the System Development Research [35], where the prototype can be used to explore feasibility and inform iterative development rather than directly implement generalizable models.

6.3. Limitations

There are several limitations in this thesis. The main limitation is that the case study involved a small number of participants (N=13) and student-AI interactions (N=99). This prevents robust statistical inferences about student question patterns and restricts our findings to providing feasibility rather than validating effectiveness. Meanwhile, the survey response rate was low which reduced the strength of conclusions regarding whether students' attitude towards AI affect their use of AI tools. Therefore, the findings of this thesis can not be generalized to full course settings and broader student populations. Secondly, instructors were not involved in in the early stages of dashboard design, and the number of instructors (N=7) participating in the evaluation was also small. Both limit the generalizability of the dashboard prototype and findings. In addition, this prototype evaluation focused on perceived usefulness and usability rather than changes in teaching practices or student learning outcomes. Participants did not use the dashboard for a long period in real-world scenarios. Finally, the classifier was trained on a publicly available. Although the dataset was collected in an artificial intelligence course similar to our situation, it introduces a domain shift effect, which may reduce performance.

6.4. Future Work

The future work should focus on addressing these limitations. A key direction is to collect interaction data (N=500+) over a longer instructional period, ideally covering the entire course and more than 50 students rather than a single assignment. The data would enable analysis of how students' help seeking behavior evolves over time, how interaction patterns correlate with learning progress, and whether interaction can inform timely instructional interventions. This data can also help to train and validate classification models more effectively. Moreover, the multi-label classification can be developed to capture the question complexity. Both approaches make the classifier better suited to the context of machine learning education and provide more accurate information to support the learning analytics dashboard.

Furthermore, a longitudinal study tracks how the use of dashboard affect instruction can be conducted, and the course instructors should be involved as primary stakeholders throughout the data collection, dashboard design and evaluation process. In contrast to prototype evaluation, embedding the dashboard into a real course setting allows teachers to integrate the analytics findings into their daily teaching practices. Combining interaction data with teacher feedback and reflection can provide a deeper understanding of how it influences teaching practices. For example, teachers can report how dashboard analytics guide them in adjusting lecture content, assignments, or lab tutoring, enabling assessment of the dashboard's impact on instruction, rather than just focusing on perceived value. Finally, the study can be conducted in multiple machine learning courses to identify generalizable patterns.

7

Conclusion

This thesis investigated how student-AI interactions in a machine learning course can be utilized to provide learning analytics that supports teachers in identifying students difficulties in assignments. It motivated by the growing interest in machine learning, the increasing use of language models in programming education and the limited focus on machine learning education. This study designed, implemented and evaluated a student-AI interaction-based learning analytics dashboard integrated into a AI-supported programming system JELAI.

We adopted a mixed approach including a case study, automated question classification and dashboard evaluation. The findings demonstrated that student-AI interactions can serve as a valuable source for understanding student difficulties in assignments.

To explore how student questions can be automatically and consistently classified into question types to support learning analytics, we developed a transformer-based classifier and compared it with language model-based classifiers on the same classification task. The result indicates that while language models can perform classification without being trained for a specific task, the transformer-based classifier can provide more stable and reproducible outputs. This demonstrates that the transformer-based classifier is suitable as an analytical component for teacher-facing analysis, even it was fine-tuned by a publicly available dataset.

The expert review of dashboard indicates that participants perceive the student-AI interaction-based analytics and dashboard as useful for being aware of student difficulties in assignment and learning. Meanwhile, feedback also highlighted the need for clearer visualizations, richer contextual information, and more in-depth analytical capabilities, further emphasizing the importance of user-centered design in learning analytics dashboards.

This study also has several limitations. The case study was short-term, with limited participants and interactions, and low response rates to pre-test and post-test questionnaires. Therefore, the findings cannot be generalized to full course setting. Furthermore, expert review focuses on perceived usefulness and usability rather than teaching practices or learning outcomes. These limitations highlight the need for longer-term and larger-scale research.

In conclusion, this thesis demonstrates that student-AI interaction learning analytics is a feasible and meaningful approach that can support instructors of machine learning education. The future research should deepen the question analysis and examine the long-term impact of such systems on teaching and learning.

References

[1] Viviana Acquaviva. *Teaching Machine Learning for the Physical Sciences: A summary of lessons learned and challenges*. 2021. arXiv: 2108.08313 [physics.ed-ph]. URL: https://arxiv.org/abs/2108.08313.

[2] Shaden Alarifi, Manal Mohammed AlSahli, and Talal Musaed Alghizzi. "Assessing EFL Undergraduates' Attitudes, Engagement, and Satisfaction Toward the Use of Artificial Intelligence in Enhancing Reading Comprehension". In: (2025).

[3] Berk Atil et al. *Non-Determinism of "Deterministic" LLM Settings*. 2025. arXiv: 2408.04667 [cs.CL]. URL: https://arxiv.org/abs/2408.04667.

[4] Anastasiia Birillo et al. "One Step at a Time: Combining LLMs and Static Analysis to Generate Next-Step Hints for Programming Tasks". In: *Proceedings of the 24th Koli Calling International Conference on Computing Education Research*. Koli Calling '24. ACM, Nov. 2024, pp. 1–12. DOI: 10.1145/3699538.3699556. URL: http://dx.doi.org/10.1145/3699538.3699556.

[5] John Brooke et al. "SUS-A quick and dirty usability scale". In: *Usability evaluation in industry* 189.194 (1996), pp. 4–7.

[6] Doga Cambaz and Xiaoling Zhang. "Use of AI-driven Code Generation Models in Teaching and Learning Programming: a Systematic Literature Review". In: *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*. SIGCSE 2024. Portland, OR, USA: Association for Computing Machinery, 2024, pp. 172–178. ISBN: 9798400704239. DOI: 10.1145/3626252.3630958. URL: https://doi.org/10.1145/3626252.3630958.

[7] Bei Chen et al. "Codet: Code generation with generated tests". In: *arXiv preprint arXiv:2207.10397* (2022).

[8] Fred D Davis. "Perceived usefulness, perceived ease of use, and user acceptance of information technology". In: *MIS quarterly* (1989), pp. 319–340.

[9] Matthew F Dixon, Igor Halperin, Paul Bilokon, et al. *Machine learning in finance*. Vol. 1170. Springer, 2020.

[10] Raphael A Dourado et al. "A teacher-facing learning analytics dashboard for process-oriented feedback in online learning". In: *LAK21: 11th International Learning Analytics and Knowledge Conference*. 2021, pp. 482–489.

[11] Ting Fei et al. "Question classification for e-learning by artificial neural network". In: *Fourth international conference on information, communications and signal processing, 2003 and the fourth pacific rim conference on multimedia. Proceedings of the 2003 joint*. Vol. 3. IEEE. 2003, pp. 1757–1761.

[12] Stephen Few. "Information dashboard design". In: (2006).

[13] Hong Gao, Yiyang Xie, and Enkelejda Kasneci. "PerVRML: ChatGPT-Driven Personalized VR Environments for Machine Learning Education". In: *International Journal of Human–Computer Interaction* (2025), pp. 1–15.

[14] Zhikai Gao et al. "Automatically Classifying Student Help Requests: A Multi-Year Analysis." In: *International Educational Data Mining Society* (2021).

[15] Simone Grassini. "Development and validation of the AI attitude scale (AIAS-4): a brief measure of general attitude toward artificial intelligence". In: *Frontiers in psychology* 14 (2023), p. 1191628.

[16] Qiang Hao et al. "Automatic identification of ineffective online student questions in computing education". In: *2018 IEEE Frontiers in Education Conference (FIE)*. IEEE. 2018, pp. 1–5.

[17] Hendrik Heuer, Juliane Jarke, and Andreas Breiter. "Machine learning in tutorials–Universal applicability, underinformed application, and other misconceptions". In: *Big data & society* 8.1 (2021), p. 20539517211017593.

[18] Jiamin Huang et al. “Automatic Classroom Question Classification Based on Bloom’s Taxonomy”. In: *Proceedings of the 13th International Conference on Education Technology and Computers*. ICETC ’21. Wuhan, China: Association for Computing Machinery, 2022, pp. 33–39. ISBN: 9781450385114. DOI: 10.1145/3498765.3498771. URL: https://doi.org/10.1145/3498765.3498771.

[19] Fareya Ikram Hunter McNichols and Andrew Lan. *The StudyChat Dataset: Student Dialogues With ChatGPT in an Artificial Intelligence Course*. 2025. arXiv: 2503.07928 [cs.AI]. URL: https://arxiv.org/abs/2503.07928.

[20] Brandon Jaipersaud et al. “Decomposed Prompting to Answer Questions on a Course Discussion Board”. In: *International Conference on Artificial Intelligence in Education*. Springer. 2023, pp. 218–223.

[21] Ioana Jivet et al. “License to evaluate: preparing learning analytics dashboards for educational practice”. In: *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*. LAK ’18. Sydney, New South Wales, Australia: Association for Computing Machinery, 2018, pp. 31–40. ISBN: 9781450364003. DOI: 10.1145/3170358.3170421. URL: https://doi.org/10.1145/3170358.3170421.

[22] Gregor Jošt, Viktor Taneski, and Sašo Karakatič. “The impact of large language models on programming education and student learning outcomes”. In: *Applied Sciences* 14.10 (2024), p. 4115.

[23] Rogers Kaliisa and Jan Arild Dolonen. “CADA: a teacher-facing learning analytics dashboard to foster teachers’ awareness of students’ participation and discourse patterns in online discussions”. In: *Technology, Knowledge and Learning* 28.3 (2023), pp. 937–958.

[24] Rogers Kaliisa, Ioana Jivet, and Paul Prinsloo. “A checklist to guide the planning, designing, implementation, and evaluation of learning analytics dashboards”. In: *International Journal of Educational Technology in Higher Education* 20.1 (2023), p. 28.

[25] Argyrios Katsantonis and Ioannis G Katsantonis. “University students’ attitudes toward artificial intelligence: An exploratory study of the cognitive, emotional, and behavioural dimensions of AI attitudes”. In: *Education Sciences* 14.9 (2024), p. 988.

[26] Majeed Kazemitabaar et al. “Codeaid: Evaluating a classroom deployment of an llm-based programming assistant that balances student and educator needs”. In: *Proceedings of the 2024 chi conference on human factors in computing systems*. 2024, pp. 1–20.

[27] Minsun Kim et al. *Designing Prompt Analytics Dashboards to Analyze Student-ChatGPT Interactions in EFL Writing*. 2024. arXiv: 2405.19691 [cs.HC]. URL: https://arxiv.org/abs/2405.19691.

[28] Sam Lau et al. *Barriers that Programming Instructors Face While Performing Emergency Pedagogical Design to Shape Student-AI Interactions with Generative AI Tools*. 2025. arXiv: 2510.09492 [cs.HC]. URL: https://arxiv.org/abs/2510.09492.

[29] Juho Leinonen et al. “Comparing code explanations created by students and large language models”. In: *Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1*. 2023, pp. 124–130.

[30] Mark Liffiton et al. “Codehelp: Using large language models with guardrails for scalable support in programming classes”. In: *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research*. 2023, pp. 1–11.

[31] Rongxin Liu et al. “Teaching CS50 with AI: leveraging generative artificial intelligence in computer science education”. In: *Proceedings of the 55th ACM technical symposium on computer science education V. 1*. 2024, pp. 750–756.

[32] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692 [cs.CL]. URL: https://arxiv.org/abs/1907.11692.

[33] Wenhan Lyu et al. “Evaluating the effectiveness of llms in introductory computer science education: A semester-long field study”. In: *Proceedings of the eleventh ACM conference on learning@ scale*. 2024, pp. 63–74.

[34] Ngoc Buu Cat Nguyen et al. “TEADASH: Implementing and Evaluating a Teacher-Facing Dashboard Using Design Science Research”. In: *Informatics*. Vol. 11. 3. MDPI. 2024, p. 61.

[35] Jay F Nunamaker Jr, Minder Chen, and Titus DM Purdin. “Systems development in information systems research”. In: *Journal of management information systems* 7.3 (1990), pp. 89–106.

[36] Alejandro Ortega-Arranz, Paraskevi Topali, and Inge Molenaar. “Configuring and Monitoring Students’ Interactions with Generative AI Tools: Supporting Teacher Autonomy”. In: *Proceedings of the 15th International Learning Analytics and Knowledge Conference*. LAK ’25. Association for Computing Machinery, 2025, pp. 895–902. ISBN: 9798400707018. DOI: 10.1145/3706468.3706533. URL: https://doi.org/10.1145/3706468.3706533.

[37] Chris Parmer, Philippe Duval, and Alex Johnson. “A data and analytics web app framework for Python, no JavaScript required”. In: (2024).

[38] Alvin Rajkomar, Jeffrey Dean, and Isaac Kohane. “Machine learning in medicine”. In: *New England Journal of Medicine* 380.14 (2019), pp. 1347–1358.

[39] Anbuselvan Sangodiah, Manoranjitham Muniandy, et al. “Question Classification Using Statistical Approach: A Complete Review.” In: *Journal of Theoretical & Applied Information Technology* 71.3 (2015).

[40] Alper Sarikaya et al. “What do we talk about when we talk about dashboards?” In: *IEEE transactions on visualization and computer graphics* 25.1 (2018), pp. 682–692.

[41] Jaromir Savelka et al. “Efficient classification of student help requests in programming courses using large language models”. In: *arXiv preprint arXiv:2310.20105* (2023).

[42] Jaromir Savelka et al. “Thrilled by your progress! Large language models (GPT-4) no longer struggle to pass assessments in higher education programming courses”. In: *Proceedings of the 2023 ACM Conference on International Computing Education Research-Volume 1*. 2023, pp. 78–92.

[43] Naaz Sibia et al. “Student Perspectives on the Challenges in Machine Learning”. In: *Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education V. 1*. ITiCSE 2025. Nijmegen, Netherlands: Association for Computing Machinery, 2025, pp. 9–15. ISBN: 9798400715679. DOI: 10.1145/3724363.3729107. URL: https://doi.org/10.1145/3724363.3729107.

[44] James Skripchuk, Yang Shi, and Thomas Price. “Identifying Common Errors in Open-Ended Machine Learning Projects”. In: *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education - Volume 1*. SIGCSE 2022. Providence, RI, USA: Association for Computing Machinery, 2022, pp. 216–222. ISBN: 9781450390705. DOI: 10.1145/3478431.3499397. URL: https://doi.org/10.1145/3478431.3499397.

[45] Elisabeth Sulmont, Elizabeth Patitsas, and Jeremy R. Cooperstock. “Can You Teach Me To Machine Learn?” In: *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*. SIGCSE ’19. Minneapolis, MN, USA: Association for Computing Machinery, 2019, pp. 948–954. ISBN: 9781450358903. DOI: 10.1145/3287324.3287392. URL: https://doi.org/10.1145/3287324.3287392.

[46] Adi L Tarca et al. “Machine learning and its applications to biology”. In: *PLoS computational biology* 3.6 (2007), e116.

[47] Gemma Team et al. *Gemma 3 Technical Report*. 2025. arXiv: 2503.19786 [cs.CL]. URL: https://arxiv.org/abs/2503.19786.

[48] Manuel Valle Torre et al. “JELAI: Integrating AI and Learning Analytics in Jupyter Notebooks”. In: *arXiv preprint arXiv:2505.17593* (2025).

[49] Xinming Tu et al. *What Should Data Science Education Do with Large Language Models?* 2023. arXiv: 2307.02792 [cs.CY]. URL: https://arxiv.org/abs/2307.02792.

[50] Katrien Verbert et al. “Learning analytics dashboards: the past, the present and the future”. In: *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*. LAK ’20. Frankfurt, Germany: Association for Computing Machinery, 2020, pp. 35–40. ISBN: 9781450377126. DOI: 10.1145/3375462.3375504. URL: https://doi.org/10.1145/3375462.3375504.

[51] Zuo Wang, Weiyue Lin, and Xiao Hu. “Self-service Teacher-facing Learning Analytics Dashboard with Large Language Models”. In: *Proceedings of the 15th International Learning Analytics and Knowledge Conference*. 2025, pp. 824–830.

[52] Wayne Xin Zhao et al. *A Survey of Large Language Models*. 2025. arXiv: 2303.18223 [cs.CL]. URL: https://arxiv.org/abs/2303.18223.