Effective Strategies For Teaching Machine Learning

Firas Moosvi[1], Fraida Fund[2], Varada Kolhatkar[1], Meiying Qin[3], Thomas Price[4], Lisa Zhang[5]
[1]University of British Columbia
[2]New York University
[3]York University
[4]North Carolina State University
[5]University of Toronto
firas.moosvi@ubc.ca, ffund@nyu.edu, kvarada@cs.ubc.ca, mqin@yorku.ca, twprice@ncsu.edu, lczhang@cs.toronto.edu

Abstract

As machine learning (ML) becomes integral in more disciplines, introductory courses in the field are attracting increasingly diverse audiences. Design of these introductory ML courses needs to be theoretically sound, but also intuitive, engaging, and accessible. Effective teaching of ML must go beyond teaching the theoretical or practical mechanics of algorithms. In this paper, we synthesize effective teaching strategies from 6 experienced ML instructors across 5 institutions to help students define appropriate ML problems, build intuition, develop reasoning skills, and apply models responsibly. We organize these strategies into eight thematic areas: preparing students for success, motivating learners through real-world relevance, integrating ethics and societal impact, avoiding common methodological pitfalls in model evaluation, guiding students on design decisions, adapting effective classroom practices, assessing student learning, and preparing for the future. On the companion website, we provide practical examples of classroom-tested activities, and in many cases, reflections on our experiences with the strategies. Our aim is for this paper to be a starting point for instructors aiming to improve learning in introductory ML courses. We hope this is a resource-rich guide for teaching ML to diverse learners, grounded in both pedagogy and practice.

Companion website with examples and artifacts — https://open-resources.github.io/teachingML
GitHub — https://github.com/open-resources/teachingML

Introduction

The increasing impact of Machine Learning (ML) and Artificial Intelligence (AI) on the world is undeniable. With the increase of ML/AI content in the ACM/IEEE-CS/AAAI's 2023Computer Science Curricular Guidelines (CS2023) (Kumar and Raj 2023), and an increase in university AI programs (Stanford Institute for Human-Centered AI 2024), more students will be learning ML/AI, and more educators will be teaching it. However, ML/AI has been reported as difficult, with both students and practitioners struggling to apply ML models effectively (Sulmont, Patitsas, and Cooperstock 2019a; Skripchuk, Shi, and Price 2022; Fund et al. 2025). Math anxiety/preparation is another obstacle (Allen, McGough, and Devlin 2022; Sibia et al. 2025). ML/AI educators need to address these challenges, while keeping pace with industry demands (Akgun and Hosseini 2025) engaging students with emergent ethical and professional challenges (Elbasi et al. 2025).

This paper is intended to be a "field guide" for new and developing ML educators. We synthesize strategies from the experiences of six ML instructors across five higher education institutions in North America. We provide guiding questions related to important challenges that new educators will face, and discuss the ways that we conceptualize these challenges. We provide summaries of practices that can be adopted in a variety of contexts and link to relevant literature. Finally, we offer a companion website that will serve as a showcase of some digital artifacts used in our classrooms to give educators concrete examples of the implementation of the effective strategies for teaching machine learning (Moosvi et al. 2025). Given the limited synthesis of ML teaching practices in the literature, our effort aims to support educators in developing effective and inclusive courses that meet the needs of a diverse set of learners.

ML education is a nascent but growing area of research and many experience reports present a single ML teaching approach (Chai and Gormley 2025), assignment (e.g., EAAI's model AI assignments) or curriculum (Zeng et al. 2021). Wollowski (2025) presents a detailed description of an introductory neural networks course, while Chai and Gormley (2025) discuss embodied classroom activities. Some prior work has focused on K-12 instruction, often with a broader focus on AI as a whole (Ali et al. 2025; Absalon and Deneux 2025). A number of research papers have also explored challenges facing ML instructors and students. Skripchuk, Shi, and Price (2022) and Zimmermann, Allin, and Zhang (2024) both identified common errors in ML programming assignments. Sulmont, Patitsas, and Cooperstock (2019a) explored additional challenges facing ML instructors, such as some students' lack of math background knowledge, or teaching students to navigate all the design decisions to create a whole ML model from scratch. In this paper, we present curated strategies for addressing questions facing ML instructors and present our own experiences.

Synthesizing Effective ML Teaching Strategies

The goal of this paper is to support teaching introductory ML courses to undergraduate and graduate students. We focus on courses that cover the CS-Core ML topics in the CS2023 guidelines (Kumar and Raj 2023), as well as similar courses for non-CS-majors.

The authors of this report each independently generated up to 10 teaching strategies they believed were important for teaching ML. Each strategy included a title, short description, and optionally, details of their experience with the strategy or other evidence for its effectiveness. Forty-four (44) total strategies were generated, consolidated, then shared anonymously with all authors. Each author considered all the strategies, noted the ones they used, and added additional comments and experiences. Through discussion and collaboration, the strategies were consolidated and catalogued in one of eight (8) themes that emerged from our analysis. Each theme relates to a question or challenge related to ML teaching and learning that we address in our classrooms:

- T1. **Preparing Students for Success** What are effective ways to address students' disparate prior knowledge of math and programming prerequisites?
- T2. **Motivating Students through Real-World Relevance** How can we make course content real-world relevant and motivating, so that they see the value of learning ML?
- T3. **Integrating Ethics and Societal Impact** How can we effectively integrate fairness, accountability, and broader social implications throughout an ML course?
- T4. **Avoiding Common Methodological Pitfalls in Model Evaluation** How can we address misconceptions and incorrect practices (e.g., flawed exploratory analysis, data leakage, misuse of evaluation metrics), so learners become adept and responsible practitioners?
- T5. **Guiding Students on Design Decisions** How can we help students navigate complex design decisions when building ML pipelines?
- T6. **Adapting Effective Classroom Practices** Which existing best practices for teaching are particularly effective for ML, and how can we adapt them to teaching ML?
- T7. **Assessing Student Learning** How can we assess student learning in ML, both formatively and summatively? What is important to assess?
- T8. **Preparing for the future** How can we prepare students to succeed in a field as fast-changing as ML? How should our courses adapt?

Each author led in the description of at least one theme, and the strategies shared were used by at least 2 authors, with the median strategy adopted by 4 authors.

Preparing Students for Success

Prior preparation in ML, particularly in math, is discussed in prior work (Shapiro and Fiebrink 2019; Li and Liu 2022; Kinnaird 2021; Tawfik et al. 2025), with competing perspectives on what mathematics is needed (Lau et al. 2022), and whether math preparation is even a challenge at all (Allen, McGough, and Devlin 2022). Authors had complementary techniques to handle math preparation, such as introducing math content only as needed, and using interactive visuals to convey the intuition behind key ideas.

Introduce math content at the time of need. Although there is some benefit to a math review session at the beginning of a course (Gagné and Briggs 1974), almost all authors structure their ML course to introduce math content right before it is needed. MQ aims to motivate students to understand why the math is needed. LZ uses a blended, "just-in-time" approach to math review, supporting students with video resources that use the same notation as in the ML course. The intention is to facilitate transfer of what student learned in a prerequisite course to a new setting, while being mindful of efficiency —in finding review resources, and different levels of support that students need—in structuring the video review resources (Zhang and Allin 2023).

Emphasize math topics central to ML. There are mathematical concepts that are not emphasized in prerequisite courses but are central to ML. For example, MQ emphasizes the geometric meaning of matrices, FM differentiates Euclidean and cosine distances, and LZ finds that students are not taught how to start from scalar quantities and find vectorized formulas. Indeed, Sibia et al. (2025) show that when provided the terminology, students express finding vectorization to be challenging, which may be why students have found neural networks and backpropagation to be challenging in other prior work (Allen, Devlin, and McGough 2021).

Motivating Students Through Real-World Relevance

Motivation plays a crucial role in learning. Students are more likely to engage deeply and persist through challenges when they understand why a topic matters and how it connects to their goals, interests, or prior experiences (Jang 2008; Hulleman and Harackiewicz 2009). This need for relevance is amplified in ML classrooms, which increasingly include learners from non-technical disciplines, students with varied professional backgrounds, and mature learners returning to study. For these learners, motivation is not just about sparking interest, it is also about building trust in the value of what they are learning.

Research in the learning sciences has long emphasized the connection between motivation and effective learning. Self-determination theory, for instance, suggests that intrinsic motivation increases when students feel a sense of autonomy, competence, and relevance (Ryan and Deci 2000; Deci and Ryan 1985). In the context of ML education, recent work by Tan et al. (2024) highlights that students' perceived usefulness of AI, career relevance, and personal interest in the subject are key drivers of engagement and sustained learning. These findings reinforce the pedagogical need to embed motivational elements throughout the ML curriculum, not just as an initial spark, but as an ongoing thread that connects each topic to meaningful applications. To accomplish this, we employ the following strategies:

Spark curiosity with hook questions. A common strategy is posing *hook questions*—short, surprising prompts or demonstrations with counterintuitive outcomes—to pique curiosity and provoke discussion (Lang 2021). These questions are especially useful at the beginning of a class or topic, helping students anticipate what they might learn and why it matters. For instance, VK and FM begin the lesson on recommendation systems by posing the question *What percentage of watch time on YouTube do you think comes from recommendations?* Students are often surprised that about 70% of YouTube watch time is driven by recommendations. This naturally leads to a discussion about how recommendation systems shape behaviour and content consumption and similar patterns appear on other platforms.

Make abstract ideas relatable. Another common strategy is selecting examples from familiar or personally meaningful domains. When abstract ML concepts are taught using relatable datasets, students are more likely to connect with the material and internalize the concepts (Ambrose et al. 2010). For instance, when motivating principal component analysis (PCA), VK uses a large-scale personality data set obtained from the Open-Source Psychometrics Project to show how the big five personality traits were identified using PCA. LZ uses students' own data from per-question test scores, with noise added to preserve privacy.

Use real-world case studies. Some of us also emphasized the motivational power of grounding abstract ML concepts in realistic case studies based on messy, real-world datasets. These case studies help students connect theory to practice, providing a narrative that situates each stage of the ML pipeline within a meaningful problem context. This sense of relevance is a key driver of engagement, especially for learners who want to apply ML in their own domains. Case studies also expose students to the complexity and messiness of real data, reinforcing foundational skills such as transforming ill-defined, real-world problems into meaningful ML tasks, a crucial step in problem framing. Several authors reported that such examples effectively prompted discussions around the appropriateness of ML for a given problem, the suitability of the available data, the importance of exploratory data analysis (EDA), modeling assumptions, limitations of evaluation metrics, and the role of domain knowledge before moving into modeling.

Align content with students' goals and backgrounds. As ML courses attract students from a wide range of disciplines and career paths, aligning course content with learners' goals becomes increasingly important. While some students, particularly those pursuing ML research, benefit from theoretical depth and algorithmic rigor, others may be better served by a focus on practical applications (Sibia et al. 2025) and responsible use (Barretto et al. 2021). For example, a public health student may gain more from learning about fairness metrics, interpretability, and model deployment in sensitive contexts than from mathematical derivations. In contrast, students who are looking for academic research may need stronger foundations in mathematical formalism and optimization. One challenge is that student interests could change over time Sibia et al. (2025). In any case, framing content around student interests can increase the perceived value of learning ML.

Integrating Ethics and Societal Impact

As ML use has increased, so too has its misuse. Substantial negative societal consequences of ML misuse (Anderljung, Hazell, and von Knebel 2024) highlights the need for ethics in ML education. Without explicit guidance however, students are unlikely to critically engage with the ethical implications of ML on privacy and well-being (McDonald and Pan 2020). Including ethics may also promote diversity and attract students from underrepresented groups (Barretto et al. 2021). However, ethics remains largely absent from curricula, with only 12% of 172 U.S. ML courses included it in 2019 (Saltz et al. 2019) and 8% of AI-related courses on global Massive Open Online Course (MOOC) platforms incorporating ethics and regulation content (Zhou et al. 2021).

Reflecting the broader interest in ethics-focused assignments, almost all authors emphasized its importance. Our teaching strategies varied: from delivering dedicated lectures on ethical issues to weaving ethical considerations throughout the course and incorporating ethics into project work beyond the classroom. Importantly, even in courses with explicit ethics sessions, the topic was not treated in isolation or as an afterthought. Ethical discussions were contextualized within technical content, reflecting the curricular patterns identified in Javed et al. (2022), where ethics is commonly embedded across disciplinary and topical boundaries rather than addressed as a standalone subject.

Integrate ethics in class. Three authors included ethics as a dedicated module with interactive activities and hands-on exercises. However, presenting ethics as a discrete module has the danger of conveying to students that it is a disparate topic, rather than integral to all of ML. For example, TP designed an ML system (meal voucher allocation), followed by an ethical analysis using stakeholder mapping and classical frameworks. This approach was effective because students initially invested in their own designs. LZ adapted the Value Cards toolkit (Shen et al. 2021) and guided students in exploring trade-offs, stakeholder perspectives, and ethical reasoning through a structured discussion. MQ introduced ethics through the technical challenge of tuning neural networks, leading to a discussion on fairness and accountability in AI-based grading. Each approach grounded ethical reflection in relatable, hands-on experiences. However, making the negative impact more personally identifiable, aligning technical and ethical considerations, avoiding false dichotomies ("should we build it?"), and oversimplification are all challenges that were identified.

Embed ethics alongside ML topics. In addition to treating ethics as a distinct module in the course, FF and FM integrated ethical reflection directly into core ML topics (similar to Grosz et al. (2019)). Ethical issues were introduced in various contexts, such as fairness in admissions, bias in data collection, and environmental impact in model training. Students explored consent and privacy during data analysis, and fairness metrics were taught alongside other topics. This approach encouraged students to think critically about ethical concerns throughout the course.

Provide opportunity for students to apply ethics out of class. TP, MQ, and LZ designed projects to deepen students' understanding of ethical issues in real-world ML applications. In TP's course, students worked on client-based and open-ended projects, exploring fairness, bias, and societal impact. LZ's assignments included building classifiers or gesture recognition systems, with discussions on data imbalance and subgroup performance. MQ also embedded ethics into student projects, supporting it with curated readings and milestone-based guidance. Students reported that these activities challenged their assumptions and helped them connect technical work to broader social consequences.

Avoiding Common Methodological Pitfalls in Model Evaluation

Evaluation of ML models is a foundational skill that holds particular importance in ML education. While the field continues to expand its influence across scientific and applied disciplines, surveys of ML-based research have identified hundreds of studies across at least 17 distinct fields that are compromised by evaluation errors (Kapoor and Narayanan 2023), which can lead to overoptimistic and ultimately misleading results. Meanwhile, prior work (Skripchuk, Shi, and Price 2022; Zimmermann, Allin, and Zhang 2024) suggests that students in machine learning courses make similar errors, with Zimmermann, Allin, and Zhang (2024) in particular reporting 93% of student projects in a course having some form of evaluation data error (improper train/test split, misuse of test data, inconsistent train/test splits, not considering feature differences in test set, and feature precognition). Other types of model evaluation errors, such as using an inappropriate evaluation metric or using a model that shows no meaningful learning, are also observed in student projects by Skripchuk, Shi, and Price (2022) and Zimmermann, Allin, and Zhang (2024).

Describe the impact of incorrect evaluation practices. Several of us present real-world instances of errors in evaluation, describing how they have played out in practice and the impact they had on policy decisions, publication of irreproducible results, and deployment of models with unacceptable performance. For example, FF uses a case study on the COVID-19 pandemic, in which multiple high-profile news outlets and government agencies used models that were not evaluated on an independent test set to predict future case counts and inform government policy. LZ uses several examples of high-profile real-world examples of failed machine learning projects.

Explicitly teach correct *and* incorrect evaluation. It is standard to teach the importance of setting aside a portion of the data as a test set and reserving it exclusively for final evaluation (Golden Rule of ML). However, students struggle to grasp what constitutes "using" the test set, leading to subtle errors such as normalization using statistics from the entire dataset (Skripchuk, Shi, and Price 2022). Thus, it can be helpful to explicitly enumerate both correct and incorrect ways of using data throughout the ML pipeline. For example, many of us demonstrate correct *and* incorrect preprocessing in lecture slides, and include practice questions presenting several code snippets for achieving the same task—such as standardizing a dataset using statistics from the entire dataset for both training and test sets, using training statistics for the training set and test statistics for the test set, or using training statistics for both sets—and ask students to identify the correct approach.

Practice using data with non-independent samples. Many of us integrate examples and assignments in which the data has non-independent samples, such as temporal data or data with multiple samples from a single respondent. Under these circumstances, a naive split of the data leads to an incorrect evaluation, introducing an opportunity to practice the correct approach to splitting this type of data. Similarly, we use *data with class imbalance*, where accuracy is an inappropriate metric, *data with a substantial proportion of missing data* where incorrect preprocessing practices will corrupt the evaluation, or *data with subgroups* where performance in different subgroups and group-level fairness must be considered, beyond overall accuracy.

Introduce anti-examples. An incorrect evaluation is often an overly optimistic one, which students may struggle to recognize as problematic ("If I'm doing something wrong, why is the accuracy so high?"). FF uses case studies of findings in the academic literature that are irreproducible due to data leakage errors (Fund et al. 2025). Students are tasked with reproducing the original (flawed) result, and then developing a correct evaluation showing that the model actually has less or no predictive value. Anecdotally, students develop a healthy skepticism of "too-good-to-be-true" results through this experience. Similarly, to highlight the importance of using an appropriate baseline for comparison, FF uses a case study of a published finding where the authors claim that a regression model is useful because its error is small relative to the scale on which the target variable is measured. However, on reproducing the work, students discover that the model error is similar to a "prediction by mean" strategy. Both Skripchuk, Shi, and Price (2022) and Zimmermann, Allin, and Zhang (2024) discuss using unreasonable results as a common error among students.

Evaluation of the model training pipeline goes beyond "Model Metrics": In addition to the standard "model metrics," students should also be guided on the implications of code efficiency, compute cycles, and the environmental impacts of their ML analysis choices. Most code in introductory ML courses executes in a short enough time span that it is tempting for students to "brute-force" their way through model training, or hyperparameter optimization. For example, searching the hyperparameter space systematically (grid search) with a very small step-size is rarely a good idea; instead, we teach students to consider a multi-step process with a larger step-size initially, followed by a smaller step-size in a more localized area, or using a random search. FF uses a lab assignment where students train the same model to a specified accuracy with different learning rates and batch sizes, and measure the energy consumption of the training process in each case ("Energy to Accuracy" metric). Students learn that modeling and training choices have an impact not only on the performance of the final model, but also on cost, time to train, and energy use.

Guiding Students on Design Decisions

A distinct challenge of teaching ML is that students must learn to make *design decisions* such as which preprocessing steps to use, which hyperparameters to tune, which models to compare, which evaluation metrics to use, how to tell if results are "good enough," and what to do if they are unsatisfactory. These decisions are equal part *art* and *science*, and our responses revealed an underlying tension: novices have not yet developed an intuition for ML design decisions and need concrete guidance on what to do and when (as also discussed in Sulmont, Patitsas, and Cooperstock (2019b)). On the other hand, simple if-then rules and heuristics can hide the complexity of ML decision making.

Explicit strategies (LaToza et al. 2020) are used by many authors to help students navigate tricky ML design decisions. In an ML context, these strategies guide students to: (1) identify salient contextual variables that inform the design decision, such as properties of the dataset, the model being trained, the problem statement, and the deployment context. (2) gather that information (through exploratory data analysis or experiments), (3) navigating a set of rules reflecting common practice to select the right design decision based on that context, and (4) justify the decision. For example, a strategy might help students decide which hyperparameters to tune, with what starting ranges, and with what tuning approach. This encourages students to intentionally make these design decisions, rather than relying on default library behaviors or blindly copying from examples.

Embracing nuance is an essential component of teaching ML guidelines, especially as an explicit strategy. Three authors explicitly teach this nuance, emphasizing that guidelines are *heuristics* that work well in many cases, and demonstrating edge cases where those heuristics fail. For example, it is often beneficial to scale features before model training, but in some situations this does not help (e.g., with many tree-based models) and has implications for model interpretation. Similarly, "best practices" sometimes do not actually improve model performance (for e.g., using hyperparameter tuning performs no better than using sensible defaults), and it is helpful to show this possibility to students, so they are not surprised when they encounter this scenario.

Contrasting cases (Schwartz and Bransford 1998) shows students two contrasting design choices and their effect on an outcome (e.g., model performance). For instance, to illustrate the impact of oversampling a minority class, students are shown the recall metric with and without oversampling. This helps students build intuition around ML design choices to supplement explicit strategies.

Adapting Effective Classroom Practices

Many of the teaching practices that we identified draw on existing effective strategies from the computing education and education literature. There is already substantial evidence for these practices, so here we focus on addressing how they are particularly applicable to ML education and how we adapted them to this context.

Peer Instruction (PI, also think/pair/share, or clicker questions) (Zingaro and Porter 2014) requires students to attempt short questions during class, first individually, and then again after discussing with their peers. The instructor can review the answer and address misconceptions that arise. PI is well-suited to ML because it is especially effective at building solid conceptual understanding (Balta et al. 2017). Distractor choices help address misconceptions (Yıldırım and Canpolat 2019) students frequently develop in ML, and can also help students build rapport through discussion, which is very helpful when in group projects.

Literate programming principles emphasize interleaving code with natural language explanation and instruction (Knuth 1984) . The resulting artifact serves as complex worked examples, an effective way to support learning (Barbieri et al. 2023). This paradigm works particularly well for ML, where instructors bundle instruction and examples into a code notebook (e.g. Jupyter). Tying together theory and implementation allows students to see the impact of design choices by running the code, making changes and iterating. Many open educational resources (e.g. (James et al. 2023; Zhang et al. 2023)) provide high-quality examples.

Algorithm visualizations are a staple of computing education (Fouh et al. 2014) and are particularly effective for ML education. All authors use interactive visualizations to support understanding the mathematics behind ML concepts. Visualizations can range from static visualizations to interactive widgets in Jupyter notebooks, purpose-built applications to explore algorithms, and even embodied activities and demonstrations. These visualizations are already used widely in ML education (e.g., 3blue1brown) and their efficacy is investigated in Renţea, Migut, and Krijthe (2025).

More broadly, VK and FM use embodied activities (Rodríguez-Jiménez, García-Merino et al. 2017) to make abstract concepts intuitive. In particular, VK incorporates learning activities that model algorithmic behaviour: e.g., to illustrate information flows in a Recurrent Neural Network (RNN), students pass a "hidden state" along a human chain, updating it with new input at each step. Similarly, to convey how Markov Models of Language rely only on limited prior context, students simulate text generation by choosing the next word based only on the last one or two. These physical representations make complex concepts tangible, supporting deeper understanding and retention.

Spiral learning is a pedagogical approach in which students revisit foundational concepts multiple times, with each iteration providing greater depth and complexity (Basu et al. 2020) . The first iteration begins with intuitive, simplified presentations that foster conceptual understanding and enable early comparison of ML algorithms. As the course progresses, these concepts are systematically expanded, incorporating more advanced mathematical models. This iterative structure promotes cumulative learning, reinforces retention through repetition, and strengthens connections between new and prior knowledge (Qin 2025), addressing some of the challenges of teaching deeply complex and connected topics in ML.

Assessing Student Learning

Frequent testing is the pedagogy of administering assessments on a repeating basis (Fitch, Drucker, and Norton 1951). This distributed retrieval practice improves long term retention and learning (Benjamin and Tullis 2010). There is a tendency in introductory ML courses to wait until there are "enough topics covered to assess students efficiently", e.g., teaching all parts of an analysis pipeline and then assessing students on the entire pipeline. However, there is value in assessing concepts piecemeal *and* cumulatively. The testing effect is well-documented as a phenomenon where testing students also helps them learn the material better, and retain it for longer (Roediger and Karpicke 2005).

On the other hand, it is worthwhile to consider the effects that testing — particularly high-stakes testing — has on student stress, anxiety, motivation, and well-being. There is mixed evidence on whether increased stress results in better learning on performance by improving focus and attention (Strack and Esteves 2015). But there is now overwhelming evidence that single-shot, time-constrained, high-stakes assessments (such as final exams) are poor choices to measure learning in all students, and also specifically disadvantage equity-deserving student groups (French, Dickerson, and Mulder 2024). The impact of testing on student (and instructor) well-being must be considered in course design.

Low-stakes assessments take many forms including exit-tickets (Angelo and Cross 2012), in-class clicker questions (Poirier and Feldman 2007; Sullivan 2009), worksheets, or even quizzes. The goal of low-stakes assessments is to help students solidify key concepts in assessments with low downside risk (to their grades), and complements active learning and frequent testing. Giving students opportunities to practice concepts with scaffolds in place to guide their learning may also encourage them to take bigger risks and engage with the material at a deeper level. If designed well, these techniques have been shown to promote transfer - students' ability to apply what they've learned in contexts different than the one they originally learned in (Son and Rivas 2016). Low-stakes assessments are also a fantastic way to improve classroom climate, improve learning for all students, and move closer to equitable outcomes for equity-deserving students (Cotner and Ballen 2017; Ballen, Salehi, and Cotner 2017; Ballen et al. 2017).

Prioritizing assessment of reasoning skills allows instructors to go beyond students memorizing facts, concepts, and principles to assess theoretical ML principles. For example, asking students to make authentic ML choices, evaluate choices that others have made, interpret outputs of models, and make decisions on model validity. Some authors give students the opportunity to practice their reasoning skills on lower-stakes programming labs and assignments. To challenge students, authors use problems that require insight into data and models, that can't be solved with off-the-shelf code or by leveraging additional computational resources.

Digital assessments and modern learning platforms can really help instructors create high quality authentic assessments that probe students' ability to think and reason in ML. Ideally, students should be assessed authentically with tools and workflows they are already familiar with so they can appropriately demonstrate proficiency with the course material authentically. Many of the authors use the PrairieLearn platform and in particular the "Workspace" paradigm where students are given a a computing environment (Jupyter, Visual Studio Code, Quarto, R, etc...) with the skeleton of an ML analysis along with some real data. Packages are pre-installed (including all documentation), and networking is turned off so students must complete the activities using only the provided resources. Within the workspace, students can run and execute code, process data, produce plots, and even run models and score them. This authentic assessment helps instructors evaluate the students on their ability to apply theoretical concepts, reason about machine learning choices and decisions, and write code.

Projects are an effective way to assess students' ability to "put it all together," navigate interconnected ML design decisions, wrestle with translating a real-world problem into an ML problem, experience authentic challenges of working with messy data, and build a sense of ownership over their work. Project topics can be *student-driven*, allowing them to connect it to their own interests (Patall, Cooper, and Robinson 2008) and pushing them to apply concepts in new contexts. Alternatively, the project could solve a problem for a real-world client, creating real authenticity. For example, TP worked with a local climate office to automate the detection of erroneous readings from their weather stations, and student projects demonstrated usable results. One caveat is that this does require a lot of instructor time. It is important that the project have sufficient *scaffolding*, especially if students choose their own topics. Examples of complete projects from prior years, and lists of appropriate datasets, can also help students to scope their projects appropriately. Breaking the project up into *milestones* with instructor feedback can help students avoid pitfalls and stay aligned with course learning objectives. Scaffolding can also occur *before* the project, such as with *project-aligned labs*. Here, earlier practice activities each parallel key steps of project (e.g., preprocessing, model selection, hyperparameters, etc.) and connect them together to ensure that students have already created a complete project-like artifact in a scaffolded, closed-ended way before they get to an independent project.

Preparing for the Future

There are curricular challenges in preparing students for future success in a field where new methods and applications emerge daily. The growing impact of Large Language Models fuel conversations about the extent to which ML courses should introduce advanced topics. At the same time, our group shared a common experience where ML courses tend to have a large amount of material, an experience shared by students (e.g., Sibia et al. (2025)). We discussed two strategies for preparing students for the future:

Focus on the Fundamentals. We share a common desire to help students thrive in tomorrow's ML landscape. While ignoring recent advances is not a good idea, emphasizing foundational skills that foster adaptability and lifelong learning is a more sustainable and empowering approach. Educators need to resist the temptation to continue adding new topics to ML courses, and clearly communicate the important learning objectives.

Carefully consider Generative AI policies. Neither prohibiting students from using Generative AI tools, nor ignoring them is likely to be successful– particularly for homework, assignments, labs, or projects. One alternative is to share the benefits, impacts, and pitfalls of using Generative AI tools during the learning process and co-create policies around its use (Chehak et al. 2025). Talking to students about the uses of Generative AI may also "level the playing field" for students that are fearful of using it, distrust it, or are unable to use it due to privacy or accessibility reasons.

Despite much discussion, none of the authors felt confident in the efficacy of Gen AI policies in their classes. We do not have compelling evidence that any strategy has been effective in encouraging students to improve their learning using AI tools. This has been studied in detail (García-Martínez et al. 2023), and remains an active area of research.

Limitations

In this work we have synthesized recommendations for teaching introductory ML based on our collective experiences and prior work. However, this approach has limitations. The strategies we highlighted are by no means exhaustive and only represent the experience of six instructors from five different institutions. We acknowledge that it will be impossible (and inadvisable) to implement all the strategies we highlighted in a single course. We intend for this experience report to be a starting point for instructors teaching introductory ML courses. Many of the strategies we have discussed are supported by prior empirical work, sometimes outside of ML; however, future work is needed to validate the efficacy of these strategies, and if any combinations are particularly effective for teaching ML. As a result, they should be interpreted as teaching advice to consider and adapt. An important next step is to repeat this study across a wider audience and in particular, have representation out of North America.

Ethical Statement

No large language models or other AI tools were used in the writing or the generation of ideas for this paper. Some authors used ChatGPT-4o to help proofread individual sentences and receive feedback on clarity and conciseness.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 2238108.

References

Absalon, M.; and Deneux, T. 2025. "AlphAI": Teaching AI Algorithms to K12 by Training Learning Robots and Visualizing How It Works. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 29096–29103.

Akgun, M.; and Hosseini, H. 2025. AI Education in a Mirror: Challenges Faced by Academic and Industry Experts. In *International Conference on Artificial Intelligence in Education*, 336–348. Springer.

Ali, S.; Jakubowicz, S.; Abodayeh, A.; Zubairi, A.; Aldossary, D.; and Breazeal, C. 2025. Designing Characters with AI: An Art & AI Learning Activity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 29104–29111.

Allen, B.; Devlin, M.; and McGough, A. 2021. Using the One Minute Paper to Gain Insight into Potential Threshold Concepts in Artificial Intelligence Courses. In *Computing Education Practice 2021*, 21–24. Durham United Kingdom: ACM. ISBN 978-1-4503-8959-4.

Allen, B.; McGough, A. S.; and Devlin, M. 2022. Toward a Framework for Teaching Artificial Intelligence to a Higher Education Audience. *ACM Transactions on Computing Education*, 22(2): 1–29.

Ambrose, S. A.; Bridges, M. W.; DiPietro, M.; Lovett, M. C.; and Norman, M. K. 2010. *How learning works: Seven research-based principles for smart teaching*. John Wiley & Sons.

Anderljung, M.; Hazell, J.; and von Knebel, M. 2024. Protecting society from AI misuse: when are restrictions on capabilities warranted? *AI & SOCIETY*, 1–17.

Angelo, T. A.; and Cross, K. P. 2012. *Classroom assessment techniques*. Jossey Bass Wiley.

Ballen, C. J.; Salehi, S.; and Cotner, S. 2017. Exams disadvantage women in introductory biology. *PLoS ONE*, 12(10): e0186419.

Ballen, C. J.; Wieman, C.; Salehi, S.; Searle, J. B.; and Zamudio, K. R. 2017. Enhancing Diversity in Undergraduate Science: Self-Efficacy Drives Performance Gains with Active Learning. *CBE Life Sciences Education*, 16(4): ar56.

Balta, N.; Michinov, N.; Balyimez, S.; and Ayaz, M. F. 2017. A meta-analysis of the effect of Peer Instruction on learning gain: Identification of informational and cultural moderators. *International Journal of Educational Research*, 86: 66–77.

Barbieri, C. A.; Miller-Cotto, D.; Clerjuste, S. N.; and Chawla, K. 2023. A meta-analysis of the worked examples effect on mathematics performance. *Educational Psychology Review*, 35(1): 11.

Barretto, D.; LaChance, J.; Burton, E.; and Liao, S. N. 2021. Exploring why underrepresented students are less likely to study machine learning and artificial intelligence. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*, 457–463.

Basu, D.; Kumar, H. K.; Lohani, V. K.; Barnette, N. D.; Back, G.; McPherson, D.; Ribbens, C. J.; and Plassmann, P. E. 2020. Integration and Evaluation of Spiral Theory based Cybersecurity Modules into core Computer Science and Engineering Courses. In *Proceedings of the 51st*

ACM Technical Symposium on Computer Science Education, SIGCSE '20, 9–15. New York, NY, USA: Association for Computing Machinery. ISBN 9781450367936.

Benjamin, A. S.; and Tullis, J. 2010. What makes distributed practice effective? *Cognitive Psychology*, 61(3): 228–247.

Chai, H.; and Gormley, M. R. 2025. Larger than Life In-Class Demonstrations for Introductory Machine Learning. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*, 220–226.

Chehak, M.; Debelius, M.; Holtschlag, J.; Kim, G.; Le, K.; Lyons, S.; Nutting, H.; and Oh, S. 2025. Designing an AI Policy: An Experiment in Co-Creation. *International Journal for Students as Partners*, 9(1): 151–160.

Cotner, S.; and Ballen, C. J. 2017. Can mixed assessment methods make biology classes more equitable? *PloS one*, 12(12): e0189610.

Deci, E. L.; and Ryan, R. M. 1985. *Intrinsic Motivation and Self-Determination in Human Behavior*. Springer.

Elbasi, E.; Nadeem, M.; Alzoubi, Y. I.; Topcu, A. E.; and Varghese, G. 2025. Machine Learning in Education: Innovations, Impacts, and Ethical Considerations. *IEEE Access*, 13: 128741–128770.

Fitch, M. L.; Drucker, A. J.; and Norton, J. A. 1951. Frequent testing as a motivating factor in large lecture classes. *Journal of Educational Psychology*, 42(1): 1–20.

Fouh, E.; Karavirta, V.; Breakiron, D. A.; Hamouda, S.; Hall, S.; Naps, T. L.; and Shaffer, C. A. 2014. Design and architecture of an interactive eTextbook–The OpenDSA system. *Science of computer programming*, 88: 22–40.

French, S.; Dickerson, A.; and Mulder, R. A. 2024. A review of the benefits and drawbacks of high-stakes final examinations in higher education. *Higher Education*, 88(3): 893–918.

Fund, F.; Saeed, M.; Malik, S.; and Ishak, K. 2025. Learning from Irreproducibility: Introducing Data Leakage Case Studies for Machine Learning Education. In *Proceedings of the 2025 ACM Conference on Reproducibility and Replicability*, ACM REP '25. New York, NY, USA: Association for Computing Machinery.

Gagné, R. M.; and Briggs, L. J. 1974. *Principles of instructional design*. New York, NY, USA: Holt, Rinehart & Winston.

García-Martínez, I.; Fernández-Batanero, J. M.; Fernández-Cerero, J.; and León, S. P. 2023. Analysing the impact of artificial intelligence and computational sciences on student performance: Systematic review and meta-analysis. *Journal of New Approaches in Educational Research*, 12(1): 171–197.

Grosz, B. J.; Grant, D. G.; Vredenburgh, K.; Behrends, J.; Hu, L.; Simmons, A.; and Waldo, J. 2019. Embedded EthiCS: integrating ethics across CS education. *Communications of the ACM*, 62(8): 54–61.

Hulleman, C. S.; and Harackiewicz, J. M. 2009. Promoting interest and performance in high school science classes. *science*, 326(5958): 1410–1412.

James, G.; Witten, D.; Hastie, T.; Tibshirani, R.; and Taylor, J. 2023. An introduction to statistical learning: Python edition.

Jang, H. 2008. Supporting students' motivation, engagement, and learning during an uninteresting activity. *Journal of Educational Psychology*, 100(4): 798.

Javed, R. T.; Nasir, O.; Borit, M.; Vanhée, L.; Zea, E.; Gupta, S.; Vinuesa, R.; and Qadir, J. 2022. Get out of the BAG! Silos in AI ethics education: Unsupervised topic modeling analysis of global AI curricula. *Journal of Artificial Intelligence Research*, 73: 933–965.

Kapoor, S.; and Narayanan, A. 2023. Leakage and the Reproducibility Crisis in Machine-Learning-Based Science. *Patterns*, 4(9): 100804.

Kinnaird, K. M. 2021. Teaching Computational Machine Learning (without Statistics). In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 17–22. PMLR.

Knuth, D. E. 1984. Literate programming. *The computer journal*, 27(2): 97–111.

Kumar, A. N.; and Raj, R. K. 2023. Computer Science Curricula 2023 (CS2023) Community Engagement by the ACM/IEEE-CS/AAAI Joint Task Force. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 2*, 1212–1213.

Lang, J. M. 2021. *Small Teaching: Everyday Lessons from the Science of Learning*. San Francisco, CA: Jossey-Bass, 2nd edition.

LaToza, T. D.; Arab, M.; Loksa, D.; and Ko, A. J. 2020. Explicit programming strategies. *Empirical Software Engineering*, 25(4): 2416–2449.

Lau, S.; Nolan, D.; Gonzalez, J.; and Guo, P. J. 2022. How Computer Science and Statistics Instructors Approach Data Science Pedagogy Differently: Three Case Studies. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education*, 29–35. Providence RI USA: ACM. ISBN 978-1-4503-9070-5.

Li, M.; and Liu, B. 2022. A Brief Discussion on the Reform of Mathematics Teaching in Artificial Intelligence Majors - Taking Matrix Computation and Optimization as Examples. In Cai, Z.; Chen, Y.; and Zhang, J., eds., *Theoretical Computer Science*, volume 1693, 132–141. Singapore: Springer Nature Singapore. ISBN 978-981-19815-1-7 978-981-19815-2-4. Series Title: Communications in Computer and Information Science.

McDonald, N.; and Pan, S. 2020. Intersectional AI: A study of how information science students think about ethics and their impact. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW2): 1–19.

Moosvi, F.; Fund, F.; Kolhatkar, V.; Qin, M.; Price, T.; and Zhang, L. 2025. Companion Site: Effective Strategies for Teaching Machine Learning [https://doi.org/10.5281/zenodo.17608647].

Patall, E. A.; Cooper, H.; and Robinson, J. C. 2008. The effects of choice on intrinsic motivation and related outcomes: a meta-analysis of research findings. *Psychological bulletin*, 134(2): 270.

Poirier, C. R.; and Feldman, R. S. 2007. Promoting Active Learning Using Individual Response Technology in Large Introductory Psychology Classes. *Teaching of Psychology*, 34(3): 194–196.

Qin, M. 2025. Approachable Machine Learning Education: A Spiral Pedagogy Approach with Experiential Learning. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1*, 924–930.

Renţea, I.; Migut, G.; and Krijthe, J. 2025. Are Interactive Visualizations in Machine Learning Education Helping Students? In *Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education V. 1*, 2–8.

Rodríguez-Jiménez, R.-M.; García-Merino, S.; et al. 2017. Enactive and embodied learning in higher education. *Functional Neurology, Rehabilitation, and Ergonomics*, 4: 5–9.

Roediger, H. L.; and Karpicke, J. D. 2005. Test-Enhanced Learning. *Psychological Science*, 17(3): 249–255.

Ryan, R. M.; and Deci, E. L. 2000. Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *American Psychologist*, 55(1): 68–78.

Saltz, J.; Skirpan, M.; Fiesler, C.; Gorelick, M.; Yeh, T.; Heckman, R.; Dewar, N.; and Beard, N. 2019. Integrating Ethics within Machine Learning Courses. *ACM Transactions on Computing Education*, 19(4): 1–26.

Schwartz, D. L.; and Bransford, J. D. 1998. A time for telling. *Cognition and instruction*, 16(4): 475–5223.

Shapiro, R. B.; and Fiebrink, R. 2019. Introduction to the Special Section: Launching an Agenda for Research on Learning Machine Learning. *ACM Transactions on Computing Education*, 19(4): 1–6.

Shen, H.; Deng, W. H.; Chattopadhyay, A.; Wu, Z. S.; Wang, X.; and Zhu, H. 2021. Value cards: An educational toolkit for teaching social impacts of machine learning through deliberation. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, 850–861.

Sibia, N.; Richardson, A.; Gao, A.; Petersen, A.; and Zhang, L. 2025. 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*, 9–15.

Skripchuk, J.; Shi, Y.; and Price, T. 2022. 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, 216–222. New York, NY, USA: Association for Computing Machinery. ISBN 9781450390705.

Son, J. Y.; and Rivas, M. J. 2016. Designing Clicker Questions to Stimulate Transfer. *Scholarship of Teaching and Learning in Psychology*, 2(3): 193–207.

Stanford Institute for Human-Centered AI. 2024. AI Index 2024 Annual Report, Chapter 6: AI Education and Talent. Accessed: 2024-10-22.

Strack, J.; and Esteves, F. 2015. Exams? Why worry? Interpreting anxiety as facilitative and stress appraisals. *Anxiety, Stress, & Coping*, 28(2): 205–214.

Sullivan, R. R. 2009. Principles for Constructing Good Clicker Questions: Going beyond Rote Learning and Stimulating Active Engagement with Course Content. *Journal of Educational Technology Systems*, 37(3): 335–347.

Sulmont, E.; Patitsas, E.; and Cooperstock, J. R. 2019a. Can you teach me to machine learn? In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 948–954.

Sulmont, E.; Patitsas, E.; and Cooperstock, J. R. 2019b. What is hard about teaching machine learning to non-majors? Insights from classifying instructors' learning goals. *ACM Transactions on Computing Education (TOCE)*, 19(4): 1–16.

Tan, C. Y.; Li, H.; Guo, W.; and Liem, G. A. D. 2024. Student motivation to learn artificial intelligence: A network perspective. *npj Science of Learning*, 9(1): 9.

Tawfik, M.; Petersen, A.; Porter, L.; and Zhang, L. 2025. Prerequisites and Performance in a Machine Learning Course: A Quantitative Analysis. In *Proceedings of the 25th Koli Calling International Conference on Computing Education Research*, 1–11.

Wollowski, M. 2025. Towards an AI Course Based on Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 29078–29086.

Yıldırım, T.; and Canpolat, N. 2019. An investigation of the effectiveness of the peer instruction method on teaching about solutions at the high-school level. *Education and Science*, 44(199): 127–147.

Zeng, Y.; Shah, A.; Thai, J.; and Zyda, M. 2021. Applied machine learning for games: A graduate school course. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 15695–15703.

Zhang, A.; Lipton, Z. C.; Li, M.; and Smola, A. J. 2023. *Dive into deep learning*. Cambridge University Press.

Zhang, L.; and Allin, S. 2023. Just-In-Time Prerequisite Review for a Machine Learning Course. In *Proceedings of the 25th Western Canadian Conference on Computing Education*, 1–2. Vancouver BC Canada: ACM. ISBN 9798400707896.

Zhou, X.; Tong, Y.; Lan, X.; Zheng, K.; and Zhan, Z. 2021. AI Education in Massive Open Online Courses: A Content Analysis . In *2021 3rd International Conference on Computer Science and Technologies in Education (CSTE)*, 80–85. Los Alamitos, CA, USA: IEEE Computer Society.

Zimmermann, R. M.; Allin, S.; and Zhang, L. 2024. Common Errors in Machine Learning Projects: A Second Look. In *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research*, Koli Calling '23. New York, NY, USA: Association for Computing Machinery. ISBN 9798400716539.

Zingaro, D.; and Porter, L. 2014. Peer instruction in computing: The value of instructor intervention. *Computers & Education*, 71: 87–96.