EdSutra

Teaching guide

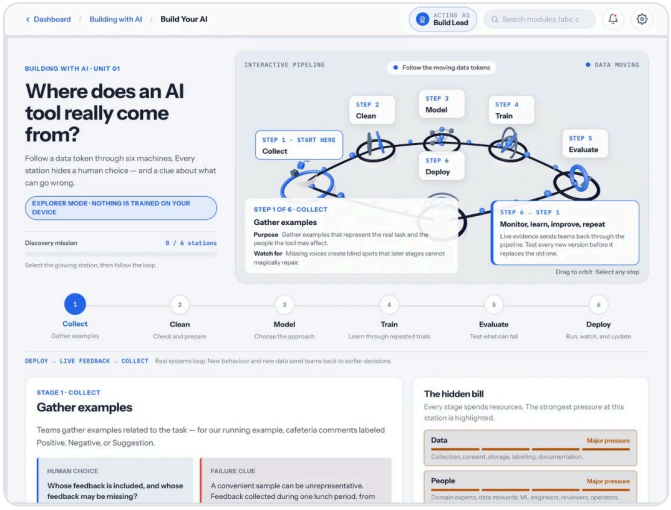
Module 06 · Building with AI · Unit 01 · Stage 3, Responsibility

Build Your AI

The Pipeline Lab

Grades 9–12 | 55 minutes | Role: **Build Lead**

Grade 9: About age 14–15 · Year 10 in the UK · Class 9 in India



The big idea

An AI tool is a chain of ordinary human decisions: collect, clean, model, train, evaluate, deploy. Each one changes quality, cost and who might be harmed. In the Pipeline Lab, training longer can make results on new comments worse, and a good overall score can hide missed Suggestions. Deciding to deploy depends on what a mistake would cost.

The learner's job

Learners are the **Build Lead** at Kestrel Works · Shop Floor. A customer is waiting. They do not care which part is difficult. Their task: take it from idea to working. Say out loud when it is not.

By the end, learners can say

- I can sequence the six pipeline stages and name a failure mode for each
- I can change a deterministic pipeline simulation and trace cause to effect
- I can diagnose why a simulated run underperforms and which stage caused it
- I can judge whether evidence is enough to deploy, revise, or run a limited trial

Before class

- One computer or tablet for each learner or pair, with EdSutra open (this Module is part of the full EdSutra studio).
- Open the studio once on each device before class, so the Unit still runs if the internet drops.
- Play the first screen and the start of the Pipeline Lab yourself, so you know where learners will click.
- Write the six station names on the board in a loop, with an arrow from Deploy back to Collect.
- Pairs work well: one learner makes the choices, the Build Lead says which station to change next, and why.
- Best after How Machines Learn and Neural Networks.
- Units open in order, and Module 1 comes first. To start a class here, switch on Open every Unit in Settings → Teachers & parents.

Words to introduce

Pipeline: The chain of steps that turns data into a working AI tool.

Epoch: One full pass of the training examples through the model.

Overfitting: Getting better on seen examples while getting worse on new ones.

Best checkpoint: The saved version that did best on new data; a candidate, not a pass.

Recall: Of all the real examples of one kind, how many the model found.

Confusion matrix: A table counting each right answer and each mix-up between kinds.

Drift: Real-world data changing after deployment, so the model fits it less well.

Running the lesson 55 minutes

Open 5 min

"When an AI tool goes wrong, people say "the computer messed up". Where else could the mistake have come from?"

Before anyone clicks, ask: if you keep training a model for longer and longer, will it keep getting better at sorting new comments? Take a hands-up vote and record it. Come back to it in Apply.

Step	What learners do	Ask the class	Listen for
1. Remember 5 min	Enter the pipeline and open the six stations in order, from Step 1 to Step 6: Collect, Clean, Model, Train, Evaluate, Deploy. Watch the data token travel around the loop.	• Which station comes right after Clean? • Where does the loop go after Deploy, and why?	Six stations in order, and Deploy sends live evidence back to Collect: the system is never really finished.
2. Understand 10 min	At each station, read the human choice and the failure clue, check the hidden bill of costs, and open the Human review queue to see which station each warning sends the team back to.	• Which human choice surprised you most? • Why might cleaning out every short comment hurt the sorter?	Every station is a human decision. Removing short comments like "More vegan?" throws away real Suggestions.
3. Apply 13 min	In the Pipeline Lab, answer the prediction, then change one choice at a time (sample, cleaning, model, epochs, pace) and press Recalculate. Drag along the training curve to find the best checkpoint.	• What happened to new-data results when you trained for many more epochs? • Which single change helped most? • Look at our vote. Were we right?	Training results kept improving while new-data results got worse. That widening gap is overfitting.
4. Analyze 12 min	Press Show the counts to open the confusion matrix, read each run's diagnosis in the Iteration history, then press Return to go back to the station that caused the problem.	• Overall accuracy looks fine. How well does it find Suggestions? • Which station did the history send you back to, and why?	Overall accuracy can hide a rare kind of comment. Weak Suggestion recall traces back to Collect or Clean.
5. Evaluate 10 min	At Deploy, decide whether staff should use the sorter with a person checking. Then judge a tool that flags cheating from quiz timing at 95% accuracy: pick a verdict and the reason that backs it.	• Which call did you make for the cafeteria sorter, and what evidence backed it? • If cheating is rare, why could many flags still be wrong?	Good enough depends on the cost of a mistake, not one accuracy number. Any choice needs a reason that fits.

Steps follow Bloom's revised taxonomy, in the order the Unit presents them.

Deepen the thinking

Critical thinking	A team trains longer and the training score rises. What evidence would you ask for before believing the model got better?
AI thinking	Deployment is not the finish line. What new kinds of comments might reach the sorter next term, and which station would they send the team back to?
Ethics and fairness	Aggressive cleaning removed short comments like "More vegan?". Whose voices get lost when data is cleaned or collected in one lunch period, and who should decide?
Connect to life	Think of an app you use that changed after an update. What might its team have seen that sent them back to an earlier stage?

Misconceptions to listen for

"AI fails because the computer went rogue."

Every station hides a human choice. The sorter's problems trace to ordinary decisions, like a narrow sample or aggressive cleaning.

"Training longer always makes a model better."

In the Pipeline Lab, training results keep improving while new-data results get worse. That widening gap is overfitting.

"High accuracy means the model is good."

Show the counts: overall accuracy can look fine while the sorter misses many Suggestions. Check every kind, and the cost of each mistake.

In real life the card learners see at the end of the Unit

- When a school or shop AI tool fails, the cause is usually an ordinary choice, like its data or testing.
- Some app updates come from monitoring: real-world feedback sends a model back to be retrained and retested.
- A tool sorting feedback can look accurate overall yet miss most of a rare kind, like suggestions.

Try it this week. Memorize the answers to five practice questions, then try five new ones on the same topic. The gap is like overfitting.

Watch out for. Training longer doesn't always help: a model can improve on practice data while getting worse on new data. That's overfitting.

Talk about it. What should a school test before using AI to spot cheating? Who could a mistake hurt?

Exit ticket

1. Name the six pipeline stations in order.

2. What is overfitting, and how does the training curve show it?

3. Why might a sorter with high overall accuracy still be failing someone?

Going further

The ML Pipeline Builder Lab trains real models on penguin or message data. A model that always guesses Normal scores high yet catches no spam, like the missed Suggestions here. For homework, use the In real life card's "Try it this week".