BUILD WITH IT

Knowing If It Works

The least glamorous skill in applied AI, and the one that decides whether your thing actually works.

First edition, September 2026

Addaly

Series editor: Dr Mustafa Mun

Draft, open for academic review

BUILD WITH IT

Knowing If It Works

The least glamorous skill in applied AI, and the one that decides whether your thing actually works.

Series editor: Dr Mustafa Mun

Addaly

First edition, September 2026 · Draft, open for academic review

Knowing If It Works

First edition, September 2026, build 62a26075.

Written by AI models to a written standard, with Dr Mustafa Mun as series editor. Exam keys checked blind by a second model: 3,665 of 3,666 agreed. Academic review invited.

The manuscript was drafted with the assistance of a large language model and was edited and published under his name. That is stated plainly here rather than left to be discovered, because a reader is entitled to know how a book they are trusting was made.

How to cite: *Mun, M. (2026). Knowing If It Works. Addaly. addaly.com/learn/evaluating-ai.*

This book is the printed form of the course of the same name at addaly.com/learn/evaluating-ai. The website is the living version and is corrected first. Where the two differ, the website is right.

Free to read, free to download, free to print, and free to teach from. There is no paid edition and there never will be.

Every diagram in it is drawn from data by code, not generated as a picture, so the numbers on the page are the numbers in the argument. The answers to every check, test and examination question are at the back.

8 modules · 76 lessons · 28 figures · 8 case studies · 57,695 words · about 11 hours

Brief contents

1

MODULE 1

What counts as evidence

Turn a vague claim that a feature works into a fixed, labelled, held-out set with a stated agreement figure and a named baseline, and say what size of difference a set that size can and cannot detect.

Lesson 1 · 7 min

A demo is not a measurement

THE ONE THING TO KEEP

A result is a number, on a fixed set, next to a comparison — anything else is an anecdote.

The bot that worked perfectly on twelve messages

A small team in Lagos built a WhatsApp assistant that pulls delivery addresses out of customer messages. Before launch they pasted in twelve messages and read the outputs. All twelve were right. It looked good. They shipped.

In the first week the assistant handled about 3,000 messages and got roughly one in five wrong — not slightly wrong, but wrong in the way that sends a rider to the other side of the city. The twelve test messages had been written by the team, and the team writes addresses like "14 Adeola Odeku Street, Victoria Island". Customers write "opposite the blue mosque, after the second speed bump, call when you reach".

The model did not fail. The measurement failed. There was no measurement.

Three things "it looks good" cannot tell you

How often. Twelve passes feels like proof. It is not. If you see zero failures in twelve tries, a reasonable upper bound on your true failure rate is about three in twelve — one in four. A system failing a quarter of the time will still hand you twelve clean passes fairly often. Small informal samples can rule out "catastrophic". They cannot separate "very good" from "roughly okay", and that is exactly the distinction you need.

Compared to what. Better than the previous prompt? Better than a regular expression? Better than the cheaper model that costs an eighth as much? "Looks good" contains no comparison, so it cannot support a decision. Every real result is a difference between two things.

Read by whom. You are the worst possible reader of your own system's output. You know what the answer should be, so you fill in the gaps as you read. Hand the same output to someone who does not know the intended answer and watch how much of the "obviously fine" becomes "wait, which invoice is this about".

What a result actually looks like

A result has four parts: a number, the fixed set it was computed on, something to compare against, and enough version detail to reproduce it.

```
address-extract v3
  set:     address-eval-100  (frozen 2026-08-14)
  metric:  exact match on area + landmark
  score:   78%   (v2 on same set: 61%)   n=100
  model:   pinned id, temperature 0
```

That fits on four lines and it settles arguments. "I tried it and it seems better" starts them.

Why teams skip this

Not laziness. Three real reasons.

It feels slow. Building the set is an afternoon of unglamorous labelling while your colleague is shipping features. It feels premature. "We're still exploring, the prompt changes daily." That is precisely when you need it — you are making a change a day with no way to tell forward from backward. And it feels like admitting you might be wrong, because a number can disappoint you and a demo never will.

The afternoon pays for itself the first time two people disagree about whether a prompt change helped. Without a set, that argument is decided by seniority. With a set, it is decided in ten minutes.

The honest caveat

Evaluation does not make you right. It makes you *checkable*. A well-built eval set with a badly chosen metric will give you a confident number about something you do not care about, and the rest of this course is largely about that failure mode: measuring the wrong thing precisely.

Start anyway. A rough number on real examples beats a strong opinion on imagined ones, and you can improve a metric once you have one. You cannot improve a vibe.

BEFORE YOU MOVE ON

A team edits their support-reply prompt. They run the old version and the new version on the same 30 examples they have been using since January. Both versions produce a correct reply on all 30. What have they learned?

A. Very little: a set where both versions pass everything contains no example capable of telling them apart.

B. The new prompt is at least as good as the old one, since it did not lose on any example.

C. The change is not worth shipping, since it produced no improvement on the test set.

D. The problem is the 30 examples were hand-picked; 30 randomly sampled examples would have given a valid answer.

The answer and why: p. 405

Lesson 2 · 8 min

The decision the number has to serve

THE ONE THING TO KEEP

Write the decision, the threshold and the unacceptable failure before you see any result, or every number you get will look like agreement.

"Does it work" is not a question

A question is answerable when a specific answer would change what you do next. *Does the assistant work?* fails that test. Two people can read the same fifty outputs and disagree completely, because they are quietly answering different questions — one is asking whether this would embarrass us in public, the other whether it saves the support team an hour a day. Both are reasonable. Neither is the same measurement.

So before you pick a metric, write three lines. They take ten minutes and they save weeks.

The three lines

1. The decision. What will you actually do differently depending on the result? Ship or hold. Route all tickets through it or one in ten. Replace the human first draft, or keep the human and let the model suggest. If no decision hangs on the number, you are collecting it for decoration and you can stop now.

2. The threshold. What result would make you ship, and what result would make you not? State it as a number *before you see one*. "We ship if it pulls the correct amount and date from at least 90 of 100 invoices." A threshold written after the run is a rationalisation; every number looks like support for what you already wanted. A threshold written before is a commitment you can be held to, including by yourself at midnight.

3. The unacceptable failure. Most systems have one error that cannot be traded against anything else. A pharmacy assistant may not invent a dose. A refund bot may not promise money the policy does not allow. This does not belong in the average. It is a gate: score it separately, and let a single occurrence in a hundred fail the run, if that is genuinely the truth.

Three lines, dated, committed next to the eval set. That file is the closest thing applied AI has to a pre-registration, and it is the cheapest honesty device in this course.

A worked example

A clinic in Pune wants a WhatsApp bot that handles appointment requests. "Does it work" is unanswerable. Here is the same intention, written so it can be measured:

> ***Decision:*** *whether the bot answers messages directly outside office hours, or only drafts a reply for the receptionist to send in the morning.*

> ***Threshold:*** *on 100 real messages from last month, it extracts the requested doctor, date and time correctly in at least 92, and asks a clarifying question rather than guessing whenever any of the three is missing.*

> ***Unacceptable:*** *naming a doctor who does not work at the clinic, or confirming a slot that is not free. Zero tolerated in 100.*

Notice what changed. The vague version invited a demo. This version tells you what to collect, how many, what to label, and what result ends the argument.

Metrics that look like evidence and are not

Three keep appearing in decks:

- **Average user rating.** Response rates to thumbs-up widgets run at roughly one to two per cent of sessions, and the people who click are not a random sample. A 4.6 average tells you about the people who rate, not the people who use.

- **Tokens, sessions, messages.** Usage is not quality. A support bot whose message count doubled may be failing twice as often and being asked again.
- **A model's own confidence.** Ask a language model how sure it is and it will say 95%, often while wrong. Its stated confidence is text, produced the same way as the rest of the text, and it is not calibrated to anything.

The test for any proposed metric is one question: *if this number moved five points, what would we do?* If the honest answer is "nothing", it is not a metric, it is a mood.

Leading and lagging

The number you care about is usually slow. Tickets resolved without a human, refunds issued in error, hours saved per week — these take a month to move and are contaminated by everything else happening in the business.

So you keep two. A **lagging** metric that is the actual point, measured monthly and treated as the truth. And a **leading** metric on your fixed set, measured in ten minutes, which you believe *predicts* the lagging one. The relationship between them is a claim, and you should check it at least once: when the offline number moved and the real one did not, you have learned something more valuable than either number.

Say what would change your mind

The last line of the file is the useful one. Finish the sentence "we would abandon this approach if…". If nothing could produce that sentence, you are not running an evaluation. You are gathering support.

Most teams find this uncomfortable the first time, and most find that writing it makes the next three arguments shorter. The number is not there to prove you were right. It is there so that the room can stop guessing.

BEFORE YOU MOVE ON

A team is about to evaluate a summarisation feature. Which of these, written down before the run, does the most to keep the result honest?

A. A list of the eight metrics they will compute, so nothing important is missed.

B. A note that the model used is GPT-class and the temperature is 0.2, for reproducibility.

C. The number that would make them ship, the number that would make them stop, and the one failure they will not trade against anything.

D. A commitment to have two people read every output before drawing conclusions.

The answer and why: p. 406

Lesson 3 · 8 min

A hundred examples, made by hand

THE ONE THING TO KEEP

A hundred real labelled examples resolves fifteen-point differences and settles arguments; it cannot resolve three-point ones.

Where the examples come from

Real traffic, if you have it. Pull a week of production inputs, sample 100, and label them. That is the whole method and it beats every clever alternative.

If you have no traffic yet, you still have sources that are not your imagination: support tickets, the search box on your existing site, WhatsApp messages your sales person forwards you, the spreadsheet the ops team maintains by hand today. A used-car marketplace in Bogotá built its first eval set from 100 rows of the WhatsApp export its two agents had been answering manually for a

CASE STUDY · MODULE 1

Fifteen questions and a batch starting Monday

An illustrative case, built from documented patterns.

A physics coaching centre in Kota, two weeks before the new JEE batch, deciding whether an AI doubt-solver is ready

Vidyarthi Classes has 1,400 students across three batches and a doubt-solving problem that every coaching centre has: between nine at night and seven in the morning, nobody answers. The founder, Rakesh, had been paying two junior teachers to sit on the WhatsApp doubt groups until midnight, and they were leaving.

Over the summer a former student, now in his second year at an engineering college, built a doubt-solver on top of a hosted language model. You send a photo or a typed question, it replies with a worked solution. Rakesh tried it himself. He typed fifteen questions from the printed module on kinematics and rotational motion, read the answers, and all fifteen were right. Two were better explained than the printed solutions. He wanted it in every group by Monday, when the new batch started, because a competitor two streets away had started advertising "24x7 AI doubt support".

The head of physics, Mrs Iyer, had been teaching for twenty-two years and did not trust it. Not on principle: she had read three of the fifteen answers and thought they were fine. Her objection was that she had no idea what the other few thousand answers would look like. "The questions you typed are from our own module. Our students do not send those. They send a blurry photo of a question from a random book, with half the diagram cut off, and they write 'sir why not option C'."

Rakesh's position was that fifteen for fifteen was evidence, and that the cost of waiting was real. Every night the tool was not live was a night the competitor's advertisement was true and his was not. He estimated that three days of building a proper test would cost him the launch date, and that a fortnight of "it works, we tried it" was worth more than a number.

Mrs Iyer's position was that fifteen for fifteen told them almost nothing. If the true failure rate were one in four, fifteen clean passes would still happen often enough that nobody should be surprised. She also pointed out that Rakesh was the worst possible reader of the outputs, because he knew what the answer should be and filled in the gaps as he read. And she had a specific fear: a solver that confidently gives a wrong numerical answer to a student at 1 a.m., who then goes into the test believing it.

The two of them agreed on one thing, which was that the argument was going in circles. So they wrote down what a decision would actually require.

The decision was not "launch or not". It was whether the tool answered students directly in the groups, or whether it drafted an answer that a junior teacher approved in the morning. Mrs Iyer wanted a threshold: on a hundred real doubts from last year's groups, the tool had to give a correct final answer on at least 85, and for numericals the answer had to be correct to the printed solution. The unacceptable failure was a confident wrong numerical answer: she wanted that scored separately, and she wanted a single occurrence in a hundred to fail the run.

Building the set was the cost. The doubt groups from last year were still on two teachers' phones. Pulling a hundred real doubts, stratified so that photographed questions, typed questions, conceptual doubts and numericals were all represented, would take a day. Labelling them took two teachers, and the labels had to agree with each other, which meant a labelling guide and a check that the two teachers actually applied it the same way. Three days, two senior teachers, at the busiest time of the year.

Rakesh's counter-offer was to launch in one batch on Monday and "watch it closely". Mrs Iyer's counter-offer was that watching it closely was what he had just done with fifteen questions.

What would you have decided, and what would you have measured first?

YOUR ANSWERS

1. Rakesh had fifteen correct answers out of fifteen. Why did that not settle whether the tool was ready?

2. The two teachers disagreed on seven of the first twenty labels, and none of the disagreements were about physics. What does that tell you, and what should happen next?

3. The overall pass rate was 71%, below the 85% threshold, yet the centre shipped something. Was that a violation of the pre-registered decision?

STOP HERE

Write your answers before you turn the page. How it played out starts on p. 54.

This page is blank so that how it played out stays out of view until you turn the page.

CASE STUDY · MODULE 1

How it played out

They built the set. It took two and a half days, not three. A hundred and twenty real doubts came off last year's phones: 55 typed conceptual questions, 40 photographed numericals, 15 that were really a student asking why their own working was wrong, and 10 that were off-topic or abusive and should have been declined.

Before labelling the rest, the two teachers independently labelled the same twenty. They disagreed on seven. Not one of the seven was about physics. They were about what counted as "correct": one teacher accepted an answer with the right number and a wrong intermediate step, the other did not. They wrote the ruling down (the final answer and the method both have to be right for a numerical; for a conceptual doubt, the answer has to address the misconception the student actually has). On the next twenty they disagreed on two.

The result: 71% overall. On typed conceptual questions, 89%. On photographed numericals, 38%, mostly because the photographs were poor and the model solved a slightly different question from the one in the picture. Two answers out of 120 were confident wrong numericals with no hedge at all. That failed Mrs Iyer's gate on its own.

They did not launch it in the groups. They launched the drafting version: the tool answered typed conceptual doubts directly, tagged as AI-generated, and everything containing a photograph or a number went to the morning queue with the draft attached. The junior teachers' evening shift went from four hours to about ninety minutes. Rakesh kept the 120 items as a frozen set and the failed photographs went into a separate folder that became the first thing they tested when the model was upgraded three months later.

The questions, discussed

1. Rakesh had fifteen correct answers out of fifteen. Why did that not settle whether the tool was ready?

Fifteen passes puts only a loose upper bound on the failure rate: a system failing a quarter of the time would still produce fifteen clean passes fairly often. The sample was also not representative, since the questions came from the centre's own printed module rather than from what students actually send, and the reader was the person least able to spot a plausible wrong answer. A result needs a number on a fixed, representative set, next to a comparison.

2. The two teachers disagreed on seven of the first twenty labels, and none of the disagreements were about physics. What does that tell you, and what should happen next?

It tells you the criterion was not yet defined, not that the teachers were careless. Disagreement clustered on what counted as correct, so the fix was a written ruling in the labelling guide and a fresh twenty to check agreement had improved. Until two people can label the same items the same way, no model can be scored against the label and no judge could later automate it.

3. The overall pass rate was 71%, below the 85% threshold, yet the centre shipped something. Was that a violation of the pre-registered decision?

No. The decision written in advance was between answering students directly and drafting for teacher approval; the threshold governed the first option. The per-slice results showed conceptual doubts passed comfortably while photographed numericals did not, and the two confident wrong numericals failed the separately scored gate. Shipping the drafting mode, with direct answers only on the slice that cleared, is exactly what the three-line decision file was written to permit.

MODULE 1 TEST

Check what stuck

6 questions, one right answer each. This one is for you, not for a record. The answers, and the reason behind each, are in the key on p. 404. If two go wrong, re-read the module before the exam.

1 A team writes twelve test messages, runs them through a new address extractor, and all twelve come back right. What does that result allow them to conclude?

A. That the true failure rate is close to zero, because twelve consecutive passes would be very unlikely if the system were failing at any real rate.

B. Only that catastrophic failure is unlikely; a system failing one time in four would often still pass twelve straight.

C. That the common case is handled and only rare cases remain to be tested.

D. Nothing whatsoever, since twelve items is too few to support any inference.

2 The course insists that the ship-or-hold threshold be written down before any result is seen. What is the mechanism it is guarding against?

A. A threshold written after the run tends to land wherever the result did.

B. The p-value is only valid when the threshold is a fixed parameter of the test.

C. Writing it first lets the team stop labelling as soon as the threshold is reached, saving effort.

D. The metric cannot be chosen until the threshold is known, so the order is forced.

3 Refund requests are 1% of traffic but, after stratified sampling, 12% of the evaluation set. How should a headline pass rate be quoted?

A. Quote the set-wide pass rate exactly as it stands; stratifying changes which items are present in the set, not how each of them is scored.

B. Remove refund items until they are 1% of the set, then quote the plain average.

C. Report only the refund pass rate, since rare strata are where systems fail.

D. Weight each stratum's rate by its real traffic share, and say the result is less certain than its item count suggests.

4 Two labellers agree on 95 of 100 posts, but only 2% of posts are harmful, and Cohen's kappa comes out near zero. What is the right reading?

A. The labellers are unreliable and should be retrained before any more labelling is done, since 95% is not good enough for a safety class.

B. Raw agreement is the trustworthy figure here and kappa should be ignored for rare classes.

C. Chance agreement is already about 94%, so kappa has little room; report the marginals and the overlap on positives too.

D. The definition of harmful is broken and the labelling guide must be rewritten from scratch.

5 A team generates 500 Devanagari test inputs from templates and the system passes 96% of them. Which claim does that result support?

A. About 4% of production failures come from Devanagari input.

B. The system can handle Devanagari input of the shapes that were generated.

C. The headline pass rate will rise if the 500 items are added to the traffic set.

D. Devanagari-speaking users will see a 96% success rate.

6 After twenty rounds of prompt tuning, the development set scores 91% and the rarely-run test set scores 76%. What is the most likely explanation?

A. The test set is harder than the development set and should be re-sampled to match it.

B. Run-to-run variance at temperature zero comfortably explains a fifteen-point gap on sets this size.

C. Near-duplicates have leaked from the test set into the development set.

D. The prompt has been fitted to the development set's particular items rather than improved.

What you should not do is invent a number for them and put it on a dashboard, because a number on a dashboard gets optimised, and a fake number gets faithfully optimised into a worse product.

BEFORE YOU MOVE ON

A team measures its summarisation feature by cosine similarity between the generated summary and a human-written reference. After a prompt change the mean similarity rises from 0.71 to 0.79. What has been established?

A. That the new summaries use wording closer to the references; whether they are more accurate has not been tested.

B. That the summaries improved meaningfully — eight points on a zero-to-one scale is a large move.

C. Nothing, because embedding similarity between texts is essentially random noise.

D. A valid improvement that now needs a significance test across the 100 items before shipping.

The answer and why: p. 412

Lesson 12 · 9 min

Precision and recall, worked on real numbers

THE ONE THING TO KEEP

Precision, recall and accuracy come from the same four cells, and under class imbalance accuracy rewards a system that does nothing.

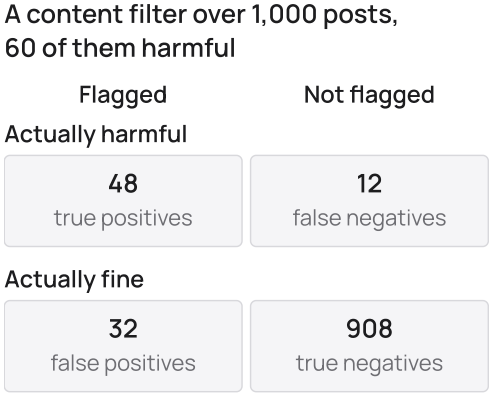
Four cells, and everything comes from them

Any yes/no classifier produces four outcomes on a labelled set. A content filter run over 1,000 posts, of which 60 are genuinely harmful:

- It flagged 80 posts.
- Of those 80, 48 really were harmful. These are **true positives**.
- The other 32 flagged posts were fine. **False positives**.
- 12 harmful posts went through unflagged. **False negatives**.
- 908 fine posts were correctly left alone. **True negatives**.

That is the confusion matrix. Every metric in this lesson is arithmetic on those four numbers, and you should be able to do all of it on paper.

Figure 2.1



Precision 48/80 = 0.60, recall 48/60 = 0.80, accuracy 95.6%. A filter that flags nothing at all scores 94.0% accuracy from the bottom-right cell alone.

Precision — when it flags, how often is it right?

```
precision = TP / (TP + FP) = 48 / 80 = 0.60
```

Recall — of the things that should have been caught, how many were?

```
recall = TP / (TP + FN) = 48 / 60 = 0.80
```

F1 — the harmonic mean of the two, used when you want one number:

EXAMINATION

Knowing If It Works: end-of-course examination

This paper covers the whole course:

- building a labelled set and reading its uncertainty
- choosing a metric that matches the decision
- deciding whether a difference is real
- evaluating the systems you are actually handed
- automating judgement without being misled by it
- evaluation inside the working loop
- evidence other people can act on
- keeping a shipped system honest over time

58 questions, the whole pool. The site draws 25 for a sitting, pass mark 70%. Work any 25 under your own conditions. Answers from page 456.

1 Module 1 · What counts as evidence

An intent router scores 78%. Always predicting the most common class scores 61%, twenty keyword rules score 74%, and two humans agree 88% of the time. What is the language model's contribution over the cheap baseline?

A. Seventeen points, the gap above the trivial baseline of always predicting billing.

B. Ten points, the distance from the model up to the ceiling where the two humans agree.

C. Four points, over the twenty keyword rules.

D. Seventy-eight points, since the baselines are context and not part of the result.

2 Module 1 · What counts as evidence

A team's offline pass rate rose eight points after a prompt change, and the monthly tickets-resolved figure it was supposed to predict did not move. What has been learned?

A. The leading metric's claim to predict the real one has failed a check.

B. One month is too short for a lagging metric to move, and the team should wait another quarter before reading it.

C. The lagging metric is too noisy to be useful and the offline number should be trusted instead.

D. The evaluation set needs more items before eight points can be believed.

3 Module 1 · What counts as evidence

After normalising and hashing, one string accounts for 6% of an evaluation set. What does the course say that means?

A. The string is a common case and should be kept in proportion so the headline reflects real traffic.

B. The set is fine, since 6% is below the level at which duplicates begin to distort a pass rate.

C. The string should be moved to the test set so it cannot inflate the development score.

D. It is not a set but a template; keep one copy and record its frequency.

4 Module 1 · What counts as evidence

A labelling guide has a one-sentence definition, three passing examples and three failing examples. Which element does the course say is most often skipped and does the most work?

A. A numeric scale from one to five so that partial quality can be recorded rather than forced into pass or fail.

B. Two boundary cases, each with a written ruling.

C. A list of the labellers' names and qualifications for the record.

D. A description of the model being evaluated so labellers know what to expect.

5 Module 1 · What counts as evidence

A system's score has plateaued at about 90% on a hand-labelled set. What does the course say the next honest step is?

A. Try another prompt variant, since a plateau shows the current wording has been exhausted.

B. Switch to a larger model, because a plateau at 90% is usually a capability ceiling rather than a data problem.

C. Re-label the failed items blind and count how many labels were wrong.

D. Add three hundred more items so the last ten points can be measured precisely.

Module 1 · What counts as evidence

The module starts on p. 11.

MODULE 1 TEST

Module 1 test, Q1, p. 56
A team writes twelve test messages, runs them through a new address extractor, and all twelve come back right. What does that result allow them…

Answer B: Only that catastrophic failure is unlikely; a system failing one time in four would often still pass twelve straight.

Why: With zero failures in twelve tries, a reasonable upper bound on the true failure rate is about one in four. Small informal samples can rule out catastrophe; they cannot separate very good from roughly okay, and the twelve items were written by the team rather than drawn from what customers send.

Module 1 test, Q2, p. 56
The course insists that the ship-or-hold threshold be written down before any result is seen. What is the mechanism it is guarding against?

Answer A: A threshold written after the run tends to land wherever the result did.

Why: A threshold chosen after seeing the number is a rationalisation; every number looks like support for what you already wanted. Written beforehand, dated and committed beside the set, it is a commitment the team can be held to, including at midnight. It has nothing to do with p-value validity, and stopping early once a threshold is reached is the optional-stopping error the statistics module warns about.

Module 1 test, Q3, p. 57
Refund requests are 1% of traffic but, after stratified sampling, 12% of the evaluation set. How should a headline pass rate be quoted?

Answer D: Weight each stratum′s rate by its real traffic share, and say the result is less certain than its item count suggests.

Why: A set-wide rate on a stratified set describes a world that does not exist. Either report per stratum and never average, or weight each stratum back to its production share. The weighted number rests on few items in the rare strata, so its uncertainty is wider than a count of items implies, and that should be said when it is quoted.

Module 1 test, Q4, p. 57
Two labellers agree on 95 of 100 posts, but only 2% of posts are harmful, and Cohen′s kappa comes out near zero. What is the…

Answer C: Chance agreement is already about 94%, so kappa has little room; report the marginals and the overlap on positives too.

Why: This is the kappa paradox. When one class dominates, expected agreement approaches observed agreement and the ratio becomes unstable. Kappa is asking how much better than guessing at these rates, and guessing is already 94% right. Report raw agreement, kappa and both marginals together, and on very rare classes prefer agreement measured only on the items either person flagged.

Module 1 test, Q5, p. 57

A team generates 500 Devanagari test inputs from templates and the system passes 96% of them. Which claim does that result support?

Answer B: The system can handle Devanagari input of the shapes that were generated.

Why: Synthetic items answer whether a system can handle a shape of input. They cannot say how often that shape occurs or how often it fails in traffic, because frequency is a property of your users and no generator has access to them. Generated items belong in a capability suite, tagged by source, and the harness should refuse to average them into the headline.

Module 1 test, Q6, p. 58

After twenty rounds of prompt tuning, the development set scores 91% and the rarely-run test set scores 76%. What is the most likely explanation?

Answer D: The prompt has been fitted to the development set's particular items rather than improved.

Why: Reading failures and changing the prompt to fix them, twenty times over, is overfitting with a human as the optimiser. The dev score becomes a record of how well a hundred specific examples were fitted. A test set that is looked at rarely is what makes the gap visible; a leak would make the test score look better, not worse.

THE LESSON CHECKS

Lesson 1, p. 14

A team edits their support-reply prompt. They run the old version and the new version on the same 30 examples they have been using since…

Answer A: Very little: a set where both versions pass everything contains no example capable of telling them apart.

Why: A result is a number, on a fixed set, next to a comparison — anything else is an anecdote.

B The new prompt is at least…: Reads a tie on examples neither version fails as evidence that the two versions are equivalent.

C The change is not worth shipping…: Treats absence of a measured gain as evidence the change is useless, when nothing was capable of measuring a gain.

D The problem is the 30 examples…: Assumes random sampling is the missing ingredient, when the deeper issue is that the set has no failures left to move.

Lesson 2, p. 18

A team is about to evaluate a summarisation feature. Which of these, written down before the run, does the most to keep the result honest?

Answer C: The number that would make them ship, the number that would make them stop, and the one failure they will not trade against anything.

Why: Write the decision, the threshold and the unacceptable failure before you see any result, or every number you get will look like agreement.

A A list of the eight metrics…: Confuses breadth of measurement with a stated commitment; eight metrics after the fact give eight ways to find a favourable one.

B A note that the model used…: Records the configuration, which matters, but nothing about it constrains how the result will be interpreted.

D A commitment to have two people…: Adds reviewer effort without fixing the underlying problem: two people with no stated criterion still disagree about what they are looking for.

Lesson 3, p. 21

Reviewing a first eval run, you find eight items where the model's answer was right and your own "expected" answer was wrong. What is the…

Answer A: Correct the eight labels, note the change, and re-run the earlier prompt versions so the old and new numbers still compare.

Why: A hundred real labelled examples resolves fifteen-point differences and settles arguments; it cannot resolve three-point ones.

B Leave them as they are —…: Applies the rule against tuning a set to your results so literally that known-wrong labels are preserved as truth.

C Remove those eight items, since ambiguous…: Mistakes hard-but-real cases for noise, and makes the set quietly easier every time it happens.

D Keep the labels and mark those…: Patches the score instead of the data, so the next person to run the set inherits the same eight wrong labels.

BUILD WITH IT

Knowing If It Works

The least glamorous skill in applied AI, and the one that decides whether your thing actually works.

WHAT IT TEACHES	Most AI features are shipped on a feeling. Someone tries eight examples, the outputs read well, and the thing goes live — and nobody finds out how often it fails until customers do.
WHAT IS INSIDE	Eight modules and 76 lessons, 28 figures drawn from data, eight case studies, eight module tests, a 58-question examination, and every answer with the reason behind it.
LICENCE	CC BY-NC-SA 4.0. Copy, print, share and adapt it for any non-commercial purpose, with credit, under the same licence. There is no paid edition.
THE COURSE	Free to read, with the lessons, the films and the examination, at **addaly.com/learn/evaluating-ai**

Addaly

First edition, September 2026 · build 62a26075

Scan to open the course