ModelQuest — Flashcards

A Spark & Anvil printable flashcard deck. Quiz yourself on the ModelQuest cast's curriculum — one card at a time.

How to use these cards

Fold each sheet down the middle line so the answers are hidden. Read the **question** on the left, say your answer out loud, then **unfold** to check the right side. Testing yourself — then checking — is one of the most powerful ways to remember. Get one wrong? That's the best kind of practice: try it again tomorrow.

Want real flashcards? Cut along the fold line and the rows into cards — question on one side, answer on the other.

---

1. What an ML model is; learns patterns from data

✂ Question (fold →)	Answer
A model's output should be treated as:	**a prediction that can be wrong, not a certain truth** — Model outputs are fallible predictions, not guaranteed truths.
A 'feature' is:	**an input variable the model uses to make its prediction** — Features are the input attributes fed to the model.
To audit a model responsibly you first need to know:	**what it decides, on what data, and who is affected** — Responsible auditing starts with the model's purpose, data, and affected people.
A spam filter that improves as it sees more emails is:	**a machine-learning model** — Learning from examples to classify emails is machine learning.
An image classifier learns to recognize cats by:	**finding patterns in many labeled cat/not-cat images** — It learns cat-patterns from many labeled examples.
A model is NOT:	**a conscious being that understands like a human** — A model is a statistical pattern-learner, not a conscious understander.
A recommendation system (e.g. suggesting videos) is:	**a model predicting what you might like from past behavior data** — Recommenders learn from behavior data to predict preferences.
A key idea of AI4K12 Big Idea 3 (Learning) is that:	**computers can learn from data** — Big Idea 3: computers learn patterns from data (statistical inference).
Machine learning differs from traditional programming because:	**the model learns rules FROM data instead of being given explicit rules** — In ML the rules are learned from examples, not hand-coded.
A model's 'prediction' is:	**its output/guess for a new input** — A prediction is the model's output for an input it's given.
Supervised learning uses data that is:	**labeled with the correct answers** — Supervised learning trains on labeled examples.
The quality of a model depends heavily on:	**the quality and representativeness of its training data** — Good, representative data is essential; poor data yields poor models.
A model's accuracy is measured on:	**data it did NOT train on (test data)** — Accuracy on unseen test data shows whether the model generalizes.
The everyday phrase that captures data's role is:	**garbage in, garbage out** — Poor input data yields poor model outputs — garbage in, garbage out.
The reason ML has advanced recently is largely:	**more data and computing power for learning algorithms** — Large datasets + compute powered modern learning algorithms.
Machine learning is a kind of:	**statistical inference — finding patterns in data** — ML is statistical inference: it generalizes patterns from data.
A model that groups data WITHOUT labels is doing:	**unsupervised learning (e.g. clustering)** — Unsupervised learning finds structure (like clusters) in unlabeled data.
The data a model learns from is called the:	**training data** — Training data is the set of examples a model learns patterns from.

A machine-learning model is best described as a system that:	**learns patterns from data to make predictions or decisions** — An ML model finds patterns in data and uses them to predict or decide.
A model that 'learns patterns from data' can only be as good as:	**the data and the care taken in building it** — A model's quality is bounded by its data and the care in its design.
A model that only ever saw daytime photos may fail at night because:	**its training data didn't represent nighttime** — A model generalizes poorly to conditions absent from its training data.
The synthesis of this kit is:	**an ML model learns patterns from data — so the data and its use deserve scrutiny** — Because models learn from data, scrutinizing the data and its use is essential.
If a model's training data is biased, its predictions will likely:	**reflect that bias** — Models inherit biases present in their training data.
A 'label' in supervised learning is:	**the correct answer attached to a training example** — Labels are the known correct outputs the model learns to predict.
'Inference' (using a trained model) is different from 'training' in that inference:	**applies the learned model to new inputs** — Training learns the model; inference uses it on new inputs.

2. Bias in → bias out; representation

✂ Question (fold →)	Answer
Historical data can embed bias because:	**it may reflect past unfair human decisions** — Historical records can encode past discrimination, which a model then learns.
The FIRST thing an auditor should inspect for bias is:	**the training data's composition** — Data composition is the primary source of model bias.
If a hiring model is trained mostly on data from one group, it may:	**perform worse or unfairly for under-represented groups** — Under-representation in training data leads to worse/unfair results for those groups.
'Sampling bias' in training data means:	**some groups appear more/less than they should** — Sampling bias skews which examples the model learns from.
'Label bias' occurs when:	**the correct answers in the data were themselves assigned unfairly** — If labels reflect biased human judgments, the model learns that bias.
Data bias is a problem because model decisions can:	**affect real people's opportunities and treatment** — Biased models can unfairly affect real people's lives.
Even with good intentions, a model can be biased if:	**the underlying data is biased** — Good intentions don't fix biased data — the data drives the outcome.
Balancing a dataset means:	**adjusting it so groups are fairly represented** — Balancing improves fair representation across groups.
Documenting a dataset's sources and gaps (a 'datasheet') helps by:	**making its limitations transparent** — Documenting data sources/limits supports transparency and bias-checking.
A facial-recognition system that works worse on some skin tones likely had:	**training data that under-represented those tones** — Under-representation in training data causes worse accuracy for those groups.
Collecting MORE diverse data can help bias because it:	**improves representation of previously under-served groups** — More representative data improves fairness for under-served groups.
If a group is entirely MISSING from training data, the model:	**has no basis to serve that group well** — A model can't reliably serve a group absent from its training data.
The synthesis of this kit is:	**models inherit the biases of their data — representative data and per-group testing are essential** — Because models inherit data bias, representative data and disaggregated testing are essential.
The best first step to reduce data bias is to:	**examine who is represented (and under-represented) in the data** — Auditing representation reveals gaps that cause bias.
A responsible team checks a model's performance:	**broken down by relevant groups** — Per-group (disaggregated) evaluation reveals disparities.
A model amplifying an existing societal bias is an example of:	**harm from biased data at scale** — Biased models can amplify societal biases across many decisions.
Bias in training data is best caught by:	**auditing the data and testing outcomes across groups** — Auditing data + per-group outcome testing surfaces bias overall accuracy hides.
The claim 'the algorithm is	**misleading — it reflects its data and design choices** — Algorithms embody

neutral' is:	the biases of their data and design; they aren't automatically neutral.
'Bias in, bias out' means:	**a model trained on biased data will make biased decisions** — Models inherit and can amplify biases present in their training data.
Fixing biased data is important because a model:	**cannot be fairer than the data it learned from** — A model's fairness is bounded by its data; fix the data to improve fairness.
A resume-screening model trained mostly on past hires from one demographic will likely:	**reflect that demographic's bias in its scoring** — It learns the biased historical pattern and reproduces it.
Bias can enter at which stage?	**data collection, labeling, AND model use** — Bias can enter at collection, labeling, and deployment — multiple stages.
A 'representative' training dataset:	**reflects the real population the model will serve** — Representative data mirrors the population the model will act on.
Overall accuracy can HIDE bias when:	**the model is accurate for the majority but poor for a minority** — A high overall score can mask poor performance on a smaller group.
A model trained only on data from wealthy neighborhoods may:	**generalize poorly to other neighborhoods** — Narrow training data generalizes poorly beyond its source.

3. Features; proxies leak protected attributes

✂ Question (fold →)	Answer
A feature 'distance from city center' could act as a proxy for:	**income or race (via housing patterns)** — Residential distance correlates with demographics, so it can proxy income/race.
To detect a proxy, you can check whether a feature:	**strongly correlates with a protected attribute** — A feature highly correlated with a protected attribute is a likely proxy.
A 'protected attribute' is a characteristic like:	**race, gender, age, or disability that laws/ethics protect from discrimination** — Protected attributes are traits (race, gender, age, etc.) shielded from discrimination.
Removing a protected attribute (like race) from the data does NOT guarantee fairness because:	**proxies can leak it back in** — Other correlated features can act as proxies, reintroducing the protected attribute.
An auditor who checks only inputs (not outcomes) will:	**miss proxy-driven bias** — Input-only audits miss bias that proxies create in outcomes.
The core reason proxies matter is that fairness depends on:	**effects on people, which proxies can quietly undermine** — Proxies can quietly undermine fair effects on people, the thing that matters.
Proxy problems are hardest when the proxy is:	**also legitimately predictive of the outcome** — A genuinely predictive proxy makes the fairness trade-off genuinely hard.
Proxies show that fairness requires looking at:	**the model's OUTCOMES, not just its inputs** — Because proxies hide bias in inputs, you must examine outcomes across groups.
If removing a feature barely changes accuracy but improves fairness, you should:	**consider removing it** — A feature with little accuracy value but fairness cost is a candidate for removal.
Including an irrelevant feature can:	**add noise or unintended bias** — Irrelevant features can inject noise or hidden proxy bias.
A model that predicts creditworthiness from shopping habits risks:	**proxying for protected attributes via behavior** — Behavioral features can correlate with protected traits, acting as proxies.
Auditing for proxies is part of ensuring a model is:	**fair in effect, not just in stated inputs** — Proxy audits target real-world fairness of outcomes, not just nominal inputs.
The danger of proxy variables is that a model can discriminate:	**even without explicitly using the protected attribute** — Proxies let a model discriminate indirectly, without the explicit attribute.
The lesson of proxies is that 'fairness through unawareness' (just hiding the attribute) is:	**insufficient** — Merely hiding the attribute fails because proxies remain — it's insufficient.
A 'proxy variable' is a feature that:	**stands in for another (often protected) attribute** — A proxy correlates with and substitutes for another attribute, often a protected one.
Detecting proxies requires:	**data analysis of correlations, not just reading the feature names** — You analyze correlations in the data; names alone don't reveal proxies.

A feature that is a strong proxy but also genuinely predictive creates:	**a trade-off between accuracy and fairness** — Predictive proxies force a fairness-vs-accuracy trade-off decision.
A model's use of a proxy can be:	**unintentional yet still harmful** — Proxy harm often happens without intent — impact matters, not intent.
The safest way to describe a proxy to a peer is:	**a sneaky stand-in for something you didn't mean to use** — A proxy is an unintended stand-in for another attribute.
A hiring model using 'years since graduation' might proxy for:	**age** — Years since graduation correlates with age — an age proxy.
'Feature selection' is the process of:	**choosing which input variables the model uses** — Feature selection decides which inputs the model considers.
The synthesis of this kit is:	**hiding a protected attribute isn't enough — proxies can leak it, so audit outcomes** — Because proxies leak protected attributes, fairness requires outcome auditing, not just input removal.
The responsible response to a discovered proxy is to:	**examine and mitigate its discriminatory effect** — You should assess and mitigate a proxy's discriminatory impact.
A proxy variable relates to bias because it can:	**reintroduce a protected attribute the team tried to exclude** — Proxies smuggle excluded attributes back into the model's decisions.
A ZIP code can act as a proxy for:	**race or income (via residential patterns)** — ZIP codes correlate with demographics, so they can proxy race/income.

4. The trade-off; fairness metrics

✂ Question (fold →)	Answer
The stakes of false positives vs false negatives:	**differ by context and should guide the threshold** — Which error is worse depends on context and should inform the decision threshold.
Improving fairness for one group can sometimes:	**slightly lower overall accuracy** — Fairness adjustments can trade a little overall accuracy — a real cost to weigh.
The synthesis of this kit is:	**accuracy and fairness can trade off — measure both, per group, and decide values transparently** — Measure accuracy and fairness per group and resolve trade-offs transparently with human values.
High overall accuracy can still be UNFAIR if:	**errors fall disproportionately on one group** — A model can be accurate overall yet make most of its errors on one group.
Choosing which fairness metric to prioritize is:	**a value-laden decision that should involve stakeholders** — Selecting a fairness criterion involves values and should include affected stakeholders. (No single 'right' metric.)
Fairness is best treated as:	**an ongoing consideration throughout design and deployment** — Fairness needs continual attention across the lifecycle, not a single check.
A 'false negative' is:	**predicting negative when the truth is positive** — A false negative misses a true positive case.
There is often a TRADE-OFF between accuracy and fairness because:	**optimizing purely for accuracy can worsen group equity** — Maximizing accuracy alone can degrade fairness across groups — a genuine trade-off.
'Fairness metrics' are used to:	**measure whether outcomes are equitable across groups** — Fairness metrics quantify whether groups are treated equitably.
In a disease screen, a false NEGATIVE is dangerous because:	**a sick person is told they're fine** — A false negative can leave a real case untreated — high stakes.
A confusion matrix helps fairness analysis by:	**showing the types of errors, which can be compared across groups** — Breaking down error types per group exposes fairness disparities.
A 'false positive' is:	**predicting positive when the truth is negative** — A false positive flags something as positive that isn't.
'Equal opportunity' fairness asks that:	**qualified individuals have equal chances across groups** — Equal-opportunity fairness seeks equal true-positive rates across groups.
To evaluate fairness you should measure accuracy:	**separately for each relevant group** — Per-group accuracy reveals disparities that an overall number hides.
The most responsible summary of a model's performance includes:	**accuracy AND fairness metrics across groups** — A responsible report covers both accuracy and per-group fairness.
Different fairness definitions (e.g. equal accuracy vs equal false-positive rates):	**can conflict — you often can't satisfy all at once** — Multiple fairness criteria can be mathematically incompatible.

A responsible team facing an accuracy-fairness trade-off should:	**make the trade-off explicit and involve affected stakeholders** — Surface the trade-off and bring in affected voices — don't decide it silently.
The accuracy-fairness trade-off is ultimately:	**a decision about values, made by people not the model** — The trade-off is resolved by human value judgments, ideally with stakeholders.
Fairness metrics disagreeing means teams must:	**choose which notion fits the context, transparently** — When metrics conflict, choose the context-appropriate one transparently.
A model with 95% overall accuracy that is only 60% accurate for one group is:	**unfair despite high overall accuracy** — High overall accuracy hides a large per-group disparity — unfair.
The reason fairness can't be fully automated is that:	**it depends on human values and context** — Fairness embeds context-specific values that require human judgment.
A model with 99% accuracy that fails mostly for a small group shows:	**accuracy can hide group-level unfairness** — Overall accuracy can mask disparities concentrated in a small group.
Reporting ONLY overall accuracy is irresponsible when:	**group disparities are possible and consequential** — When disparities matter, overall accuracy alone hides the fairness picture.
A threshold change that boosts accuracy but harms a group should be:	**evaluated for its fairness impact, not adopted blindly** — Evaluate fairness impact before adopting an accuracy-boosting threshold change.
A model's 'accuracy' is:	**the fraction of predictions it gets right** — Accuracy measures the share of correct predictions.

5. Overfitting vs generalization

✂ Question (fold →)	Answer
'Cross-validation' helps by:	**testing the model on multiple held-out splits of the data** — Cross-validation estimates generalization across several data splits.
More data helps overfitting because it:	**makes memorizing noise harder and the true pattern clearer** — Abundant data drowns out noise and reveals the real pattern.
A responsible ML report should include:	**test/generalization performance, not just training accuracy** — Reporting generalization performance is essential for honesty.
A model that scores 99% on training but 70% on new data is:	**overfitting** — The large train-vs-test gap indicates overfitting.
An overfit model is dangerous in deployment because:	**it may perform far worse on real-world data than expected** — Overfitting inflates test-time expectations that real data won't meet.
A model that memorizes exact training examples has effectively:	**not learned the general pattern** — Memorizing examples isn't learning the underlying pattern.
A model with high training accuracy but low TEST accuracy is:	**overfitting** — A big train-vs-test gap signals overfitting.
The simplest guard against fooling yourself with training accuracy is:	**always evaluate on unseen data** — Unseen-data evaluation is the basic guard against overfitting illusions.
Overfitting is essentially a failure of:	**generalization** — At its core, overfitting is poor generalization.
If test accuracy is much lower than a demo suggested, suspect:	**overfitting to the demo/training data** — A demo-vs-reality gap often means overfitting to favorable data.
The synthesis of this kit is:	**a model that only memorizes hasn't learned — always test generalization on unseen data** — Real learning generalizes; verify it on unseen data, never trust training accuracy alone.
Choosing model complexity well means balancing:	**fitting the data and generalizing to new data** — The goal is enough complexity to fit but not so much it overfits.
A way to REDUCE overfitting is:	**use more training data or a simpler model** — More data or a simpler model improves generalization.
Generalization is a model's ability to:	**perform well on data it has never seen** — Generalization is good performance on new, unseen data.
Overfitting means a model:	**memorizes the training data but fails on new data** — An overfit model fits training noise and fails to generalize to unseen data.
Overfitting connects to responsible AI because an overfit model may:	**fail unpredictably for real people after deployment** — Overfit models can behave unpredictably in the real world, harming users.
A learning curve that shows training accuracy rising while test	**overfitting has begun** — Diverging train-up/test-down curves signal

accuracy falls indicates:	overfitting.
Adding MORE features without more data can:	**increase overfitting risk** — More features on limited data can let the model overfit.
A cause of overfitting is:	**a model too complex for the amount of data** — Excess complexity relative to data lets the model memorize noise.
A model that generalizes well:	**captures the true underlying pattern, not the noise** — Good generalization means learning signal, not noise.
The 'bias-variance trade-off' relates to overfitting because:	**high variance (complexity) tends to overfit; high bias (simplicity) tends to underfit** — Overfitting is high variance; underfitting is high bias — a modeling trade-off (distinct from social bias).
A held-out test set must NOT be used during:	**training or model selection** — Touching the test set during training/selection leaks information and inflates results.
Underfitting means a model is:	**too simple to capture the real pattern** — Underfit models miss the pattern — too simple, poor on both train and test.
To detect overfitting you compare performance on:	**training data vs held-out test data** — The train-vs-test gap reveals overfitting.
Regularization techniques reduce overfitting by:	**penalizing excessive model complexity** — Regularization discourages complexity, improving generalization.

6. FP/FN; the stakes differ

✂ Question (fold →)	Answer
The confusion matrix turns 'accuracy' into:	**a detailed picture of WHICH errors happen** — It reveals the specific error types behind an accuracy number.
'F1 score' combines:	**precision and recall into one number** — F1 is the harmonic mean of precision and recall.
In a cancer screen, MINIMIZING false negatives matters because:	**a missed cancer is very costly** — Missing a real case (false negative) can be life-threatening — prioritize recall.
There is often a trade-off between:	**precision and recall** — Raising precision often lowers recall and vice versa.
'Recall' answers:	**of the real positives, how many did the model find?** — Recall = true positives / all actual positives.
The stakes of FP vs FN:	**depend on the context and who is affected** — Which error is worse is context- and stakeholder-dependent.
To compare a loan model's fairness, check whether:	**false-positive/negative rates differ across groups** — Disparate error rates across groups signal unfairness.
If a model has high recall but low precision, it:	**catches most positives but with many false alarms** — High recall + low precision = finds real cases but flags many negatives too.
Choosing to optimize precision vs recall is:	**a value decision about which error is worse** — Prioritizing precision or recall reflects which error is worse — a value judgment.
Adjusting the decision THRESHOLD changes:	**the balance of false positives and false negatives** — Moving the threshold trades FP for FN and vice versa.
'Precision' answers:	**of those predicted positive, how many really are?** — Precision = true positives / all predicted positives.
The confusion matrix helps fairness because you can:	**compute error types separately for each group** — Per-group error breakdowns expose disparities.
A responsible team reports:	**the full confusion matrix, not just accuracy** — Reporting the full matrix shows the error profile behind a single number.
Understanding FP vs FN is essential because:	**the human cost of each error differs by situation** — The real-world cost of each error type varies, guiding the right trade-off.
For a rare disease (1 in 1000), always predicting 'no disease' gives:	**99.9% accuracy but useless recall** — It's accurate yet catches zero real cases — the accuracy paradox.
A 'false positive' is:	**predicting positive when the truth is negative** — A false positive flags a negative case as positive.
A 'true negative' is:	**correctly predicting the negative case** — A true negative is a correct negative prediction.
	look past overall accuracy to the KINDS of errors and who bears them —

The synthesis of this kit is:	Examining error types (FP/FN) per group reveals the real, human-weighted performance.
A confusion matrix shows:	**counts of correct and incorrect predictions by type** — It tabulates true/false positives and negatives.
In a spam filter, minimizing false POSITIVES matters because:	**deleting a real important email is costly** — A false positive here trashes a wanted email — prioritize precision.
A 'false negative' is:	**predicting negative when the truth is positive** — A false negative misses a real positive.
In a fraud detector, a false NEGATIVE means:	**real fraud went undetected** — A false negative is missed fraud — the truth was positive.
Overall accuracy can be MISLEADING when:	**the classes are imbalanced (one is rare)** — With a rare class, high accuracy can come from ignoring it — misleading.
A 'true positive' is:	**correctly predicting the positive case** — A true positive is a correct positive prediction.
A model tuned for high precision will tend to:	**only predict positive when very confident (fewer false alarms)** — High-precision tuning makes cautious positive predictions.

7. Opening the black box

✂ Question (fold →)	Answer
For a medical AI, explainability helps doctors:	**decide whether to trust and act on the recommendation** — Explanations let clinicians judge whether to trust and use the AI's suggestion.
Transparency builds:	**trust and accountability** — Openness fosters trust and enables accountability.
The 'right to an explanation' is the idea that:	**people affected by automated decisions can ask why** — Some frameworks grant affected people a right to an explanation of automated decisions.
A model too complex to explain is problematic when:	**it makes consequential decisions about people** — Opaque models are most problematic for high-stakes, person-affecting decisions.
A responsible deployment includes a way for users to:	**get an explanation and appeal a decision** — Users should be able to understand and appeal automated decisions.
A loan model that cannot explain WHY it denied an applicant is problematic because:	**the applicant can't understand or contest it** — Unexplained high-stakes denials leave people unable to understand or appeal.
A simple, interpretable model (like a short decision tree) trades:	**some accuracy for much more transparency** — Interpretable models can be a bit less accurate but far more transparent.
Transparency about a model includes disclosing:	**what data it used and its known limitations** — Transparency covers the data, purpose, and limitations of the model.
Choosing an interpretable model over a slightly-more-accurate opaque one is:	**a legitimate responsible-AI decision** — In high-stakes contexts, trading a little accuracy for interpretability is legitimate.
A model card documents:	**a model's intended use, performance, and limitations** — Model cards summarize purpose, performance across groups, and limits.
Post-hoc explanation methods try to:	**approximate why a black-box model made a decision** — Post-hoc methods estimate reasons for opaque model outputs.
The MORE consequential the decision, the more you should demand:	**transparency and explainability** — Higher stakes warrant more transparency and explainability.
A system that hides how it decides makes it hard to:	**detect bias, errors, or unfairness** — Opacity blocks detecting bias, error, and unfairness.
If a model can't explain WHY it denied someone, the fair response is to:	**provide a human review or an interpretable alternative** — Unexplainable high-stakes denials warrant human review or interpretable models.
Explainability matters for high-stakes decisions because:	**people affected deserve to understand and contest them** — When decisions affect people, they should be understandable and contestable.
A good explanation is:	**truthful, understandable, and actionable for the affected person** — Effective explanations are honest, clear, and useful to the person.

Explainability means a model's decisions can be:	**explained in terms a person can understand** — Explainability = decisions a human can understand and inspect.
A 'black box' model is one whose:	**internal decision process is hard to interpret** — Black-box models produce outputs whose reasoning is opaque.
Explainability is part of AI4K12 Big Idea 5 because it concerns:	**the societal impact of AI decisions** — Explainability is a societal-impact concern (Big Idea 5).
An explanation of a loan denial might state:	**the main factors that drove the decision** — A useful explanation names the key factors behind the decision.
Explainability supports fairness by:	**letting people spot and challenge unfair reasoning** — Visible reasoning lets stakeholders detect and contest unfairness.
The synthesis of this kit is:	**opening the black box — transparency and explainability — lets people understand, trust, and contest AI decisions** — Transparency and explainability empower people to understand, trust, and challenge AI decisions.
A 'feature importance' analysis shows:	**which inputs most influenced the model's decisions** — Feature importance reveals which inputs drive predictions — an explainability tool.
Transparency differs from explainability: transparency is more about:	**openness regarding the system's data, purpose, and limits** — Transparency = openness about the system; explainability = per-decision reasons.
Explainability is HARDER for:	**very large, complex models (like deep neural nets)** — Deep, complex models are the hardest to interpret.

8. Collect the least; on-device

✂ Question (fold →)	Answer
Informed consent for data means users:	**understand and agree to what is collected and why** — Informed consent requires clear understanding and agreement.
The trade-off with data minimization is sometimes:	**slightly less personalization for much more privacy** — Minimizing data can slightly reduce personalization but greatly improves privacy.
The principle 'privacy by design' means:	**building privacy protections in from the start** — Privacy by design bakes protections into the system from the outset.
A model that works with LESS personal data is generally:	**more privacy-protective** — Using less personal data reduces privacy risk.
Data minimization means:	**collecting only the data actually needed for the task** — Data minimization limits collection to what the task requires.
Collecting extra 'just in case' data is risky because:	**it increases the harm if there's a breach** — More stored data means more to lose in a breach.
The synthesis of this kit is:	**collect the least data, keep it on-device when possible, and protect what you must keep** — Minimize collection, prefer on-device, and safeguard necessary data — the privacy core.
Anonymization can FAIL when:	**other data can be combined to re-identify people** — Combining datasets can re-identify supposedly anonymous records.
Data minimization connects to responsible AI because:	**less data collected means less potential for misuse and harm** — Minimizing data shrinks the surface for misuse and harm.
Anonymization tries to:	**remove information that identifies individuals** — Anonymization strips personally-identifying details from data.
Third-party trackers embedded in an app are a privacy risk because:	**they send user data to outside companies** — Trackers exfiltrate user data to third parties — a privacy risk.
A privacy-respecting recommendation feature could work by:	**using on-device signals without uploading a profile** — On-device signals can personalize without transmitting a personal profile.
Collecting sensitive attributes (health, location) requires:	**extra care, justification, and protection** — Sensitive data demands heightened justification and safeguards.
For a kids' product especially, data practices should be:	**minimal, on-device, and privacy-protective** — Products for children warrant especially strict, minimal, protective data practices.
Retaining data forever is a privacy risk because:	**the longer it's kept, the more chances for misuse or breach** — Indefinite retention accumulates breach and misuse risk.
A weather app that uploads your full contact list violates:	**data minimization (it doesn't need contacts)** — Collecting contacts unrelated to weather breaks data minimization.
'On-device' processing improves privacy because:	**personal data doesn't have to leave the device** — On-device processing keeps personal data local, reducing exposure.

A responsible engineer, unsure whether to collect a field, should:	**default to NOT collecting it unless clearly needed** — When in doubt, don't collect — minimization as the default.
'Purpose limitation' means:	**using data only for the purpose it was collected for** — Data should be used only for its stated collection purpose.
A data breach exposes users to harms like:	**identity theft, profiling, or embarrassment** — Breaches can lead to identity theft, profiling, and other harms.
A model that only needs a yes/no answer should NOT:	**store every detail of the user's history** — If a yes/no suffices, storing full histories is unnecessary over-collection.
Aggregating data (counts, not individuals) protects privacy by:	**avoiding storage of identifiable records** — Aggregate statistics avoid keeping identifiable individual data.
The most privacy-protective default is:	**opt-in collection with clear disclosure** — Opt-in with clear disclosure respects users by default.
The safest data (from a privacy view) is:	**the data you never collected** — Uncollected data can't leak — minimization is a core privacy defense.
A privacy-respecting design asks:	**what is the LEAST data we can use to do this well?** — The guiding question is the minimum data needed for the task.

9. Who's affected; consent

✂ Question (fold →)	Answer
A responsible engineer facing a values disagreement should:	**surface it for stakeholder deliberation, not resolve it unilaterally by fiat** — Values disagreements belong in transparent stakeholder deliberation, not a unilateral fiat.
Ignoring a stakeholder group in the design tends to:	**produce harms that group could have flagged** — Excluded groups' concerns go unheard, risking preventable harm.
When stakeholders disagree, the responsible output is:	**a clear account of the trade-offs, not a declared winner** — Document the trade-offs and perspectives; don't declare a scored winner (anti-evangelism).
'Value-tagging' an option means:	**naming which value it prioritizes (e.g. privacy vs convenience)** — Value-tagging labels which value an option serves, with its trade-off — not a score.
A group with the LEAST power in a decision often:	**bears risks they had little say in** — Low-power groups often bear risks without a voice — a fairness concern to weigh.
The goal of stakeholder engagement is to:	**make better-informed, more just decisions** — Engaging stakeholders yields more informed, just decisions.
Consent obtained under pressure or without alternatives is:	**not truly free consent** — Consent isn't free if refusing has no real alternative or is pressured.
The affected COMMUNITY should be consulted because:	**those impacted often see harms the builders miss** — Affected communities surface impacts builders may overlook.
A responsible process for a contested deployment presents:	**the trade-offs and each stakeholder group's perspective** — Responsible deliberation lays out trade-offs and stakeholder views rather than declaring one side right.
The synthesis of this kit is:	**identify who's affected, seek meaningful consent, and deliberate trade-offs openly — without declaring a single 'right' side** — Responsible AI maps stakeholders, secures real consent, and deliberates trade-offs transparently (anti-evangelism).
Presenting 'both sides' of a contested deployment with evidence is:	**responsible deliberation** — Fairly presenting evidence and perspectives is responsible deliberation, not weakness.
Meaningful consent requires that people:	**understand what they're agreeing to and can decline** — Consent is meaningful only with understanding and a real option to decline.
Consent should be:	**specific, informed, and freely given** — Valid consent is specific, informed, and freely given (and revocable).
A 'stakeholder analysis' lists:	**the groups affected and how each is impacted** — Stakeholder analysis enumerates affected groups and their impacts.
A consent checkbox buried in dense legal text is:	**weak consent (people don't truly understand)** — Hidden, unreadable terms undermine genuine understanding — weak consent.

'Nothing about us without us' expresses the principle that:	**affected people should participate in decisions about them** — Affected people should have a voice in decisions that impact them.
A 'stakeholder' in an AI system is:	**anyone affected by or with an interest in the system's decisions** — Stakeholders include everyone affected by the system, not just its builders.
When an AI decision affects people who did NOT consent to the data use, that raises:	**an ethical concern to address** — Affecting non-consenting people is an ethical issue requiring attention. (Not a scored side — a concern to weigh.)
A deployment that benefits many but harms a vulnerable few requires:	**careful, transparent weighing of the trade-off** — Such trade-offs need transparent, careful deliberation — not an automatic verdict either way.
The right way to handle a values-laden deployment question in this app is to:	**map stakeholders and trade-offs, not pick the 'right' answer** — The app teaches mapping stakeholders and trade-offs; it does not score a 'correct' side (anti-evangelism).
In a deliberation about a deployment, the responsible facilitator:	**keeps trade-offs open and includes affected voices** — A responsible facilitator surfaces trade-offs and includes stakeholders — no forced verdict.
When affected people had no say in a data-use decision, a responsible team should:	**seek their input before proceeding** — Responsible practice seeks affected people's input (not a scored verdict).
Obtaining consent from children's data specifically requires:	**heightened protections and usually parental involvement** — Children's data needs heightened protection and typically guardian involvement.
Identifying stakeholders BEFORE deployment helps you:	**anticipate who could be harmed or benefited** — Mapping stakeholders surfaces potential harms and benefits early.
Considering stakeholders is part of AI4K12 Big Idea 5 because it concerns:	**AI's impact on society and people** — Stakeholder impact is a societal-impact (Big Idea 5) concern.

10. Deployment reshapes future inputs

✂ Question (fold →)	Answer
A responsible team plans for feedback loops by:	**designing monitoring and correction mechanisms before launch** — Anticipate loops with monitoring and correction built in.
Feedback loops mean model evaluation must be:	**ongoing, not just a one-time pre-launch check** — Because deployment changes data, evaluation must continue after launch.
A model retrained on its OWN outputs risks:	**compounding its errors (model collapse / drift)** — Retraining on self-generated data can compound errors and drift.
A hiring model that favors one profile will, over time:	**make the workforce data even less diverse, reinforcing itself** — Favoring a profile skews future hiring data, reinforcing the bias.
'Runaway feedback' describes:	**a loop that spirals in one direction unchecked** — Runaway feedback spirals without correction.
A predictive-policing model sent to areas it flags will:	**record more incidents there, reinforcing its own bias** — Sending patrols where it predicts crime generates more recorded incidents there — a self-reinforcing loop.
A/B testing a change can help detect a loop by:	**comparing outcomes with and without the intervention over time** — Comparing conditions over time surfaces loop effects.
A model that becomes MORE confident and MORE biased after deployment is showing:	**a feedback loop effect** — Growing confidence plus bias post-launch signals a feedback loop.
A search engine that ranks popular results higher can create a loop where:	**popular things get more clicks and become even more popular** — Ranking by popularity feeds a rich-get-richer loop.
The signal that a deployed model needs re-examination is:	**its real-world outcomes drifting from expectations** — Outcome drift from expectations flags a need to re-examine.
The reason a pre-launch test can MISS a feedback problem is:	**the loop only forms once the model influences real data** — Loops emerge from deployment, so static pre-launch tests can't see them.
Content-recommendation loops can affect society by:	**shaping what large groups see and believe** — Recommendation loops can shape collective attention and belief.
The responsible mindset is to treat a deployed model as:	**an intervention in a system that needs monitoring** — A deployed model is a system intervention requiring ongoing monitoring.
Feedback loops are dangerous because they can:	**amplify an initial bias over time** — Loops can compound small initial biases into large ones.
A recommendation system that shows you more of what you click can create:	**a filter bubble / narrowing of content** — Click-driven recommendations can narrow content into a filter bubble.
A hiring tool that keeps recommending one profile makes future training data:	**less diverse, reinforcing the bias** — Its own recommendations skew future data, a self-reinforcing loop.

Human oversight helps with feedback loops by:	**catching and correcting harmful drift the model won't self-correct** — Human oversight can catch drift a model won't correct on its own.
The key insight of feedback loops is that a model is:	**part of the system it's predicting, not a neutral observer** — A deployed model acts on the world and changes what it later sees.
The core lesson of feedback loops is that deployment is:	**not the end — the model keeps shaping its world** — Deployment begins an ongoing interaction between model and world.
A loop can be broken by:	**intervening (e.g. diversifying inputs, human oversight, monitoring)** — Deliberate interventions (fresh inputs, oversight, monitoring) can break harmful loops.
The synthesis of this kit is:	**a deployed model reshapes its own future data — monitor for loops that amplify bias** — Because deployment creates feedback loops that can amplify bias, ongoing monitoring is essential.
A feedback loop turns a small initial unfairness into:	**a larger, entrenched pattern** — Loops can entrench and enlarge an initial unfairness.
Feedback loops are part of Big Idea 5 because they concern:	**AI's evolving impact on society** — Loops are about AI's ongoing societal impact (Big Idea 5).
To detect a harmful feedback loop, you should:	**monitor outcomes over time after deployment** — Ongoing post-deployment monitoring reveals emerging loops.
A feedback loop in a deployed model means:	**the model's decisions influence the future data it learns from** — Deployed models can shape the very data they later train on — a feedback loop.

11. Oversight; when to keep a human

✂ Question (fold →)	Answer
A model recommending a medical treatment should:	**support a clinician's decision, not replace it** — In medicine the model informs; the clinician decides (human in the loop).
Over-reliance on automation ('automation bias') is when people:	**trust the model even when it's wrong** — Automation bias is over-trusting automated outputs even against evidence.
The right level of automation depends on:	**the stakes, reversibility, and error costs** — Stakes, reversibility, and error costs determine appropriate automation.
A downside of human review is:	**it is slower and doesn't scale as easily** — Human review adds time and scaling cost — a real trade-off.
The purpose of an APPEAL process is to:	**let affected people contest an automated decision** — Appeals give people recourse to challenge automated decisions.
Human oversight complements the model by adding:	**judgment, context, and accountability** — Humans bring judgment, context, and accountability machines lack.
'Human in the loop' means:	**a person reviews or can override the model's decisions** — Human-in-the-loop keeps a person able to review/override automated decisions.
For a content-moderation system at massive scale, a reasonable design is:	**automation with human review of hard/appealed cases** — Scale forces automation, with humans handling hard and appealed cases.
A well-designed system shows the human:	**the model's confidence and reasons, inviting scrutiny** — Showing confidence and reasons supports genuine human scrutiny.
For an AI that flags content for removal, human review of appeals is important because:	**automated errors can wrongly remove legitimate content** — Appeals with human review catch wrongful automated removals.
Automation bias is worse when the interface:	**presents the model's output as authoritative** — Authoritative-looking outputs increase over-trust.
Full automation is more acceptable when:	**errors are low-cost and easily reversible** — Low-cost, reversible-error tasks are safer to fully automate.
Keeping a human in the loop can help catch:	**errors, edge cases, and unfair outcomes the model misses** — Humans catch edge cases and unfairness models miss.
'Human on the loop' (vs in the loop) means the human:	**monitors and can intervene, but the system runs autonomously** — Human-on-the-loop monitors and can step in, without approving each decision.
For an irreversible decision (e.g. major medical or legal), you should lean toward:	**more human involvement** — Irreversibility argues for greater human involvement.
A good human-in-the-loop design gives the human:	**enough information and time to meaningfully review** — Meaningful oversight needs adequate context and time.
Human-in-the-loop is part of	**a check against automated harm** — Human oversight is a key check against

responsible AI because it provides:	automated harm.
Human oversight matters MOST for:	**high-stakes, consequential decisions** — The higher the stakes, the more human oversight matters.
A 'rubber stamp' human who always approves the model is:	**not providing real oversight** — Automatic approval isn't genuine oversight.
The synthesis of this kit is:	**match oversight to the stakes — keep a human able to catch and override consequential errors** — Calibrate human oversight to the stakes, preserving real override power for consequential decisions.
A human in the loop is a safeguard, but only if they:	**have the ability and authority to actually override** — Oversight requires genuine power to override, not mere presence.
The decision of how much to automate is:	**an ethical and practical judgment, not purely technical** — Automation level is an ethical/practical judgment call.
A system that lets NO ONE explain or override its decisions is:	**hard to hold accountable** — Without override/explanation, accountability breaks down.
Removing the human entirely from a high-stakes loop risks:	**unchecked errors harming people** — Full automation in high-stakes settings risks unchecked harm.
The core question for automation is:	**'what happens when the model is wrong, and who catches it?'** — Design around model failure: who catches and corrects errors?

12. Generative basics; hallucination + provenance

✂ Question (fold →)	Answer
Knowing content is AI-generated matters because:	**people should be able to judge its reliability and origin** — Disclosing AI origin lets people weigh reliability appropriately.
Watermarking or labeling AI content helps by:	**signaling provenance so people know it's AI-made** — Provenance labels/watermarks disclose AI origin.
If a generative model cites a source that doesn't exist, that is:	**a hallucinated citation — verify before trusting** — Models can fabricate citations; always verify references exist and say what's claimed.
A chatbot that invents a fake court case citation is:	**hallucinating (confident but false)** — A fabricated citation is a hallucination — verify before trusting.
A 'hallucination' in generative AI is:	**confident output that is false or made-up** — A hallucination is fluent, confident content that is factually wrong.
When using AI-generated images of people, you should consider:	**consent and the risk of deception** — AI images of people raise consent and deception concerns.
Generative models can also reflect:	**biases present in their training data** — Like other models, generative AI can reproduce training-data biases.
The single best practice with generative AI is to:	**verify, disclose provenance, and keep human judgment** — Verify facts, disclose AI origin, and apply human judgment.
You should treat a generative model's factual claims as:	**needing verification against reliable sources** — Generative outputs must be checked against trustworthy sources.
For a school assignment, passing off AI text as your own is:	**an academic-integrity problem** — Undisclosed AI authorship is an academic-integrity issue.
The reliability of generative output is HIGHEST for:	**common, well-represented topics** — Well-represented topics yield more reliable output than sparse or unknowable ones.
A responsible use of a generative model for a report is to:	**use it to draft, then verify and cite real sources** — Use it to draft, but verify facts and cite genuine sources.
A generative model asked about a niche fact may:	**confidently invent a plausible but wrong answer** — On sparse-data topics, models often fabricate confidently.
Provenance tools (like content credentials) aim to:	**track and disclose how media was created and edited** — Provenance systems record and disclose creation/editing history.
Generative AI is a tool for:	**augmenting human work, with human judgment on top** — It augments humans; human judgment and verification remain essential.
Generative AI relates to Big Idea 2 (Representation & Reasoning) because it:	**builds internal representations to produce new outputs** — Generative models use learned representations to reason toward new content.

'Provenance' of content means:	**where it came from and how it was produced** — Provenance is the origin and production history of content.
A 'deepfake' is:	**AI-generated fake media of a real person** — Deepfakes are AI-fabricated but realistic media of real people.
Asking a generative model to 'explain its reasoning' gives:	**a plausible-sounding story that may not be its true process** — Generated 'reasoning' is itself produced text, not a faithful trace of the computation.
Deepfakes raise concerns about:	**misinformation, consent, and trust in media** — Deepfakes threaten truth, consent, and media trust.
Hallucinations happen because the model:	**predicts plausible-sounding text, not verified truth** — Generative models optimize plausibility, not verified accuracy, so they can invent facts.
A safe habit with generative AI facts is to:	**cross-check with an authoritative source** — Cross-checking against authoritative sources guards against hallucination.
Generative AI models:	**create new content (text, images) based on patterns in training data** — Generative models produce new content from learned patterns.
The synthesis of this kit is:	**generative AI produces plausible content that can be false — verify facts and track provenance** — Because generative AI can confidently hallucinate, verification and provenance are essential.
The confidence of a generative model's tone is:	**not a reliable indicator of accuracy** — Fluent confidence doesn't mean the content is true.

13. Where content came from; credit

✂ Question (fold →)	Answer
Submitting AI-generated text as entirely your own work, undisclosed, is:	**a failure of honest attribution** — Undisclosed AI authorship misrepresents who created the work.
The synthesis of this kit is:	**track where content comes from and give credit — provenance and attribution keep information honest** — Provenance and attribution sustain trust, honesty, and respect for creators in the AI era.
Knowing an image's provenance helps a viewer:	**judge whether to trust it** — Provenance informs how much to trust a piece of media.
Attribution means:	**crediting the source or creator of content** — Attribution gives credit to the original source/creator.
Attribution matters for AI training data because:	**creators' work is used and may deserve credit or consent** — Training data comes from creators whose work raises credit/consent questions.
A deepfake with no provenance is dangerous because:	**viewers can't tell it's fabricated** — Without provenance, fabricated media can pass as real.
When unsure whether to credit a source, the safe choice is to:	**credit it** — When in doubt, attribute — err toward giving credit.
A key habit for the AI era is to:	**ask 'where did this come from?' about content** — Questioning content origin is a vital AI-era habit.
Attribution and provenance together support:	**accountability for content** — Knowing origin and credit enables accountability.
Plagiarism is:	**using others' work without proper credit** — Plagiarism is uncredited use of others' work.
Passing off AI-generated work as fully your own can mislead about:	**who or what actually created it** — Undisclosed AI authorship misrepresents the true creator.
Giving credit is part of responsible AI because it:	**respects creators and supports an honest information ecosystem** — Attribution respects creators and keeps information honest.
A responsible creator using AI tools should:	**disclose the tools/sources used where it matters** — Disclosing AI/tool use where relevant is responsible attribution.
Respecting intellectual property means:	**honoring creators' rights and giving credit** — Respecting IP means honoring rights and crediting creators.
For AI-assisted schoolwork, honest attribution means:	**noting where and how AI was used** — Honesty means disclosing how AI contributed.
Content credentials attach:	**metadata about how content was made and edited** — Content credentials embed provenance metadata (creation/edit history).

The 'source' of a model's output can be hard to trace because:	**it blends patterns from vast training data** — Generative output blends many sources, complicating attribution.
Provenance is important for combating:	**misinformation and fabricated media** — Provenance helps distinguish authentic from fabricated media.
Provenance systems help creators by:	**establishing a record of their authorship** — Provenance records support creators' authorship claims.
When AI is trained on artists' work without consent, it raises:	**questions of consent and credit** — Training on creators' work without consent raises credit and consent issues.
Provenance metadata can be stripped, so it is:	**helpful but not foolproof** — Provenance helps but can be removed, so it isn't a complete guarantee.
Provenance tracking records:	**the origin and history of a piece of content** — Provenance records where content came from and how it changed.
Claiming credit for a source you didn't create is:	**dishonest attribution** — Taking credit for others' work is dishonest.
A responsible platform labels AI-generated content to:	**preserve user trust and transparency** — Labeling AI content maintains transparency and trust.
A citation in a report serves to:	**let readers find and verify the original source** — Citations let readers locate and check original sources.

14. Real-world harm cases

✂ Question (fold →)	Answer
Analyzing a case, you should ask who was:	**affected, and whether they had a voice** — Ask who bore the harm and whether they were consulted.
Diverse design teams can help prevent harm because they:	**are more likely to anticipate varied impacts** — Diverse teams better anticipate impacts across groups.
When analyzing a harm case, the responsible focus is on:	**what went wrong in the system and how to fix it** — Focus on the system failure and remedies, not on scoring a side (anti-evangelism).
A case study is well-handled when students can:	**explain the harm mechanism and propose safeguards** — Success is explaining the mechanism and proposing safeguards.
The lesson from most algorithmic-harm cases is that:	**harm often traces to data, design choices, or lack of oversight** — Harm usually stems from data/design/oversight gaps — fixable causes.
Studying real algorithmic-harm cases helps us:	**learn to recognize and prevent similar harms** — Case study builds the skill to recognize and prevent recurring harms (not to score sides).
Studying a case where an AI denied benefits to eligible people, the responsible focus is on:	**the missing safeguards (appeals, oversight) to prevent recurrence** — Focus on systemic safeguards, not blame or a verdict (anti-evangelism).
A responsible retrospective on a harm assigns:	**systemic causes and improvements, avoiding scapegoating** — Good retrospectives find systemic causes and fixes, not scapegoats.
A risk-scoring tool used in decisions about people should be:	**audited for fairness and open to challenge** — High-stakes scoring tools need fairness audits and contestability.
A recommendation system amplifying harmful content shows:	**a feedback-loop / societal-impact harm** — Amplifying harmful content is a societal-impact/feedback-loop harm.
A pattern across cases is that harmed groups are often:	**those under-represented in the data or the design team** — Under-represented groups (in data or teams) are disproportionately harmed.
The responsible conclusion from a harm case is usually:	**specific, actionable safeguards — not a single villain** — Draw actionable safeguards, avoiding a simplistic villain or verdict.
The trauma-aware framing of a harm case avoids:	**sensationalizing or re-traumatizing affected people** — Handle harm cases with care, avoiding sensationalism or re-traumatization.
A hiring algorithm that downgraded resumes with certain words shows:	**how proxies can encode bias** — Penalizing certain words acted as a proxy for a protected trait — proxy bias.

A facial-recognition system misidentifying some groups more often illustrates:	**harm from unrepresentative training data** — Higher error rates for some groups trace to unrepresentative data — a real harm pattern.
Across all cases, the through-line is that harm is:	**preventable with responsible data, design, and oversight** — Responsible data/design/oversight can prevent these harms.
Learning from harm cases, a designer should build in:	**fairness audits, oversight, and recourse from the start** — Cases teach designing audits, oversight, and recourse up front.
A case where an AI denied benefits to eligible people shows the importance of:	**appeal processes and human oversight** — Wrongful denials show why appeals and oversight matter.
Transparency about past harms helps the field by:	**letting everyone learn and improve practices** — Openly studying harms improves field-wide practice.
The synthesis of this kit is:	**study real harms to extract safeguards — understand the mechanism, present perspectives, and never reduce it to a scored verdict** — Learn the harm mechanism, weigh perspectives, and derive safeguards — without a scored verdict (anti-evangelism).
The reason we present multiple perspectives on a contested case is:	**different stakeholders experience the impact differently** — Different stakeholders feel impacts differently, so multiple views are essential.
When a case involves contested VALUES (e.g. safety vs privacy), this app:	**presents the trade-offs and stakeholder views, without a scored verdict** — On value conflicts, the app maps trade-offs and views rather than scoring a side.
Harm cases often went undetected because teams measured:	**only overall accuracy, not per-group outcomes** — Relying on overall accuracy hid per-group harms.
When a case is politically charged, the educational goal is to:	**understand the mechanism of harm, not to advocate a position** — Focus on the harm mechanism and prevention, not political advocacy (anti-evangelism).
The most useful takeaway from a harm case is:	**a concrete design safeguard to prevent recurrence** — Extract a design safeguard, not blame — prevention is the goal.

15. Value-tagged, no verdict

✂ Question (fold →)	Answer
This app presents a contested deployment as:	**value-tagged options with trade-offs, not a scored 'right' choice** — Contested deployments are shown as value-tagged trade-offs, never scored (anti-evangelism).
'Value-tagging' a deployment option means labeling:	**which value it prioritizes and what it trades away** — Value-tagging names the prioritized value and the trade-off, without ranking it best.
Presenting only the benefits of a deployment (hiding harms) is:	**irresponsible/misleading** — Hiding harms while touting benefits misleads decision-makers.
Post-deployment monitoring is needed because:	**real-world behavior can differ from testing** — Real use can reveal issues tests missed — hence monitoring.
Deciding NOT to deploy is:	**sometimes the responsible choice** — Choosing not to deploy can be the responsible decision.
A responsible deployment plan includes:	**monitoring, rollback, appeals, and clear ownership** — A responsible plan has monitoring, rollback, appeals, and accountability.
The reason we don't 'score' the right deployment in this app is that:	**reasonable people weigh the values differently** — Value trade-offs are legitimately weighed differently, so the app maps rather than scores them.
The app's role in a contested deployment scenario is to help you:	**reason through the trade-offs, not to hand you the answer** — It builds trade-off reasoning skill; it doesn't dispense a verdict (anti-evangelism).
Before deploying, a responsible team asks:	**who could be harmed, and how would we know?** — Anticipating harms and detection is central to responsible deployment.
A deployment that helps many but seriously harms a vulnerable few should be:	**deliberated transparently, weighing the trade-off with stakeholders** — Such trade-offs need transparent stakeholder deliberation, not an automatic verdict.
When stakeholders disagree about deploying, a responsible team:	**documents the trade-offs and decides transparently with input** — Document trade-offs and decide transparently with stakeholder input — no forced verdict.
An impact assessment before deployment evaluates:	**potential effects on affected groups** — Impact assessments evaluate effects on affected groups pre-launch.
The best deployment	**transparently, with evidence, trade-offs, and affected voices** — Transparency,

decisions are made:	evidence, trade-offs, and stakeholder voice make the best decisions.
A 'staged rollout' deploys to a small group first to:	**catch problems before wide release** — Staged rollouts surface issues before full deployment.
A reversible, low-stakes deployment can proceed with:	**lighter safeguards than an irreversible high-stakes one** — Safeguards should scale with stakes and reversibility.
A responsible deployment decision weighs:	**benefits, harms, and who bears each** — It weighs benefits and harms and considers who bears them.
Including affected communities in the deployment decision:	**improves legitimacy and surfaces overlooked harms** — Community input improves legitimacy and reveals hidden harms.
A responsible team treats deployment as:	**a decision with ongoing responsibility, not a one-time launch** — Deployment carries ongoing responsibility, not a one-and-done launch.
The synthesis of this kit is:	**responsible deployment weighs benefits, harms, and stakeholders transparently — a value judgment the app helps you reason, not one it scores** — Deployment is a transparent value judgment weighing benefits/harms/stakeholders — reasoned, not scored (anti-evangelism).
A deployment decision is ultimately:	**a value judgment informed by evidence and stakeholders** — It's a value judgment informed by evidence and stakeholder input.
The capstone skill here is being able to:	**lay out a deployment's trade-offs and reason about them responsibly** — The skill is structured, responsible trade-off reasoning about deployment.
A deployment that maximizes accuracy but ignores a harmed group is:	**a trade-off to deliberate, not an automatic win** — Maximizing accuracy while harming a group is a genuine trade-off for deliberation, not an auto-win.
A deployment decision framework should make the VALUES:	**explicit, so people can debate them openly** — Making values explicit enables open, honest debate.
If evidence of harm emerges after launch, the responsible action is to:	**investigate and be ready to pause or fix** — Investigate and be prepared to pause or remediate.
A 'kill switch' or rollback plan exists to:	**stop or undo a deployment if harm appears** — Rollback capability lets you halt/undo a harmful deployment.

16. Audit a model end-to-end + decide

✂ Question (fold →)	Answer
The audit should recommend ongoing:	**monitoring and re-auditing after changes** — Recommend continued monitoring and re-audit after changes.
For a contested deployment question, the audit should:	**present value-tagged trade-offs, not a scored verdict** — On value questions the audit maps trade-offs, never a scored verdict (anti-evangelism).
A model audit's FIRST step is to:	**define what the model decides and who is affected** — Start by framing the model's purpose and affected stakeholders.
Auditing explainability asks:	**can the model's decisions be understood and contested?** — Explainability audit checks understandability and contestability.
Auditing for feedback loops requires:	**monitoring outcomes over time after deployment** — Loop audit needs ongoing post-deployment outcome monitoring.
The affected community's role in an audit is to:	**provide perspective on real-world impact** — Affected communities inform the audit with lived impact.
The output of an audit should be:	**documented findings and recommended actions** — Audits produce documented findings and actionable recommendations.
A good auditor is:	**independent and skeptical, following the evidence** — Independence and evidence-led skepticism define a good auditor.
A thorough model audit checks data, fairness, explainability, privacy, feedback loops, and:	**human oversight/recourse** — A full audit also verifies human oversight and user recourse.
Documentation like model cards and datasheets supports auditing by:	**recording purpose, performance, and limits transparently** — Model cards/datasheets give auditors transparent records.
The synthesis of ModelQuest is:	**responsible AI means auditing data, fairness, transparency, privacy, and impact — and reasoning trade-offs openly, not evangelizing a verdict** — The through-line: audit the whole lifecycle for bias/fairness/transparency/privacy/impact, and reason value trade-offs openly (anti-evangelism).
A model audit ties together every earlier concept because it checks:	**data bias, fairness, explainability, privacy, loops, and oversight** — The capstone audit integrates all the app's concepts.
Auditing the training data checks for:	**bias, gaps, and representativeness** — Data audit checks bias, coverage gaps, and representation.
Checking for proxy variables is part of auditing for:	**hidden/indirect discrimination** — Proxy checks catch indirect discrimination.

The responsible auditor's deliverable respects that some questions are:	**value judgments for stakeholders, presented as trade-offs** — Value questions are surfaced as stakeholder trade-offs, not decided by the auditor alone.
An audit finding that overall accuracy hid a group disparity shows the value of:	**disaggregated evaluation** — Per-group evaluation reveals disparities overall numbers hide.
A complete audit considers oversight by asking:	**is there appropriate human review and recourse?** — Audit oversight checks for human review and user recourse.
The single most important auditor question is:	**who could this harm, and are there safeguards?** — Center the audit on potential harm and safeguards.
Auditing across the LIFECYCLE means checking:	**data, model, deployment, AND ongoing monitoring** — A full audit spans data, model, deployment, and monitoring.
When an audit finds a fairness problem, the responsible next step is to:	**investigate the cause and remediate** — Investigate root cause and remediate the fairness problem.
Auditing performance means measuring accuracy:	**overall AND per group** — Audit performance overall and disaggregated by group.
Auditing is valuable BEFORE deployment because it:	**catches problems while they're still cheap to fix** — Pre-deployment audits catch issues early and cheaply.
A responsible audit conclusion is:	**calibrated to the evidence, stating both strengths and risks** — Audit conclusions honestly state strengths and risks, matched to evidence.
Auditing privacy checks whether the model:	**collects only necessary data and protects it** — Privacy audit verifies data minimization and protection.
If an audit can't explain a high-stakes model's decisions, it should recommend:	**an interpretable alternative or human review** — Recommend interpretability or human oversight when explanations are impossible.

Keep going

You practiced **400 cards** from ModelQuest. Come back to the ones you missed — spaced-out practice makes them stick.

Spark & Anvil is a 501(c)(3) public charity. Every app, story, and book is free, forever — no ads, no in-app purchases, no tracking.

Keep exploring online

Play it now	More free books
	
`https://spark-and-anvil.org/play/modelquest`	`https://spark-and-anvil.org/books`

Scan a code with any phone camera, or type the address. Every app, story, and book is free, forever — no account, no tracking.