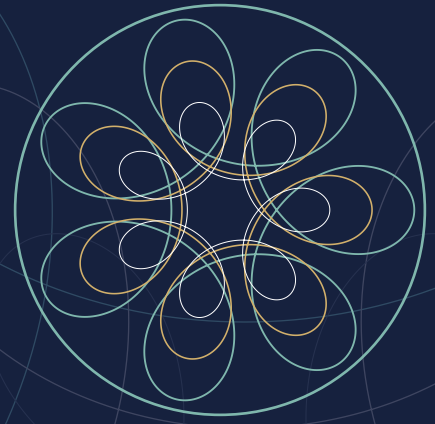




International Institute of
Applied AI Technology



Course guide for AI-BAP

Certified AI Business Analysis Professional

Elicit, model and validate requirements for solutions that include machine learning, generative AI and intelligent automation, and judge when AI is the right answer to a business need.

About this guide

Elicit, model and validate requirements for solutions that include machine learning, generative AI and intelligent automation, and judge when AI is the right answer to a business need.

This guide sets out what the AI-BAP examination covers and how to prepare for it. It assumes you already practice business analysis, and concentrates on what changes when AI is part of the work.

How to use it

Work through the modules in order. Each one sets out the ideas, lists the topics the exam can test, and ends with an exercise. Do the exercises on your own projects where you can: the exam questions are written around the same situations.

Who it is for	Business analysts, product owners, systems analysts and requirements engineers who now work on projects where some of the solution is AI.
Recommended experience	Two years of business analysis or requirements work, or an equivalent qualification in the field.
Study time	About 40 hours
Exam	50 multiple-choice questions in 90 minutes, online. Pass mark 75%, which is 38 correct. Up to 3 attempts per enrollment.
Certification	Valid for 3 years, with a downloadable certificate and a verifiable credential number.

Learning outcomes

When you have completed this program you will be able to:

- Explain how rule-based, machine learning and generative systems differ, and frame a business problem as the right kind of AI task
- Decide whether a need is best met by AI, conventional automation or a process change, and build a business case that counts the real costs
- Write requirements for probabilistic behavior: measures, thresholds, confidence routing and fallbacks
- Assess the data a proposed AI solution depends on before committing to it
- Specify grounding, safety and evaluation requirements for generative AI solutions
- Model human-in-the-loop work, exception paths and feedback loops
- Define acceptance criteria, trial approaches and monitoring for models in production

- Turn responsible-AI principles into testable, traceable requirements

How the exam is built

Every attempt draws 50 questions from a larger bank, spread evenly across the 8 modules, so each module contributes about 6 questions. Most questions describe a situation and ask what a practitioner should do, which measure applies or what the result means. Everything the exam tests is explained in this guide; the questions themselves are not.

Module 1	AI essentials for the business analyst	about 12%
Module 2	Strategy analysis and the AI business case	about 12%
Module 3	Requirements for probabilistic systems	about 12%
Module 4	Data requirements and readiness	about 12%
Module 5	Requirements for generative AI solutions	about 12%
Module 6	Modeling human and machine work together	about 12%
Module 7	Acceptance, deployment and monitoring	about 12%
Module 8	Responsible AI requirements and governance	about 12%

Contents

Module 1. AI essentials for the business analyst

Module 2. Strategy analysis and the AI business case

Module 3. Requirements for probabilistic systems

Module 4. Data requirements and readiness

Module 5. Requirements for generative AI solutions

Module 6. Modeling human and machine work together

Module 7. Acceptance, deployment and monitoring

Module 8. Responsible AI requirements and governance

Preparing for the exam

Glossary

Module 1

AI essentials for the business analyst

What AI systems are, how they differ from conventional software, and the vocabulary an analyst needs to work with data scientists and engineers without becoming one.

In this module

- Rule-based, machine learning and generative systems compared
- Training, inference, features, labels and models in plain language
- Framing a business problem as classification, regression, ranking, clustering, extraction or generation
- The CRISP-DM life cycle and where business analysis fits in it

Three kinds of system

Conventional (rule-based) software follows logic that people write: if a claim is over \$5,000, send it to a supervisor. Given the same input, it always gives the same output, and when it is wrong the fault can be traced to a specific rule.

A machine learning (ML) system learns a pattern from examples instead of being told the rule. Shown thousands of past claims marked as fraudulent or genuine, it learns which combinations of features tend to go with fraud. It then produces a prediction for each new claim, usually with a score that expresses how confident it is. It will be wrong some of the time, and the analyst's job is to make sure everyone agrees in advance how often that is acceptable and what happens when it is.

A generative AI system, such as a large language model (LLM), produces new content: text, summaries, answers, images or code. It is trained on very large amounts of general data and then directed with instructions called prompts. Its output can be fluent and still be wrong, so it needs its own kind of requirements, covered in Module 5.

The working vocabulary

A model is the learned pattern, stored as a file that software can call. Training is the process of building the model from historical examples. Inference is using the trained model on new cases in day-to-day operation.

Features are the input facts the model uses, such as claim amount, policy age or the words in an email. The label (or target) is the answer the model is trained to predict, such as fraudulent or genuine. Labeled data is historical data where the right answer is known. Supervised learning learns from labeled data; unsupervised learning finds structure, such as groups of similar customers, without labels.

A prediction is usually returned with a score between 0 and 1. A threshold turns that score into a decision: for example, scores of 0.8 and above are treated as fraud. Choosing the threshold is a business decision, not a technical one, because it trades one kind of error against another.

Framing the problem

Most of the value an analyst adds at the start of an AI initiative is in stating the problem precisely. The same business need can often be framed in more than one way, and each framing implies different data, measures and risks.

- Classification: which category does this belong to? (route an email to a team; approve or refer an application)
- Regression: what number should we expect? (days to settle a claim; next month's call volume)
- Ranking or recommendation: which items matter most, in what order? (which leads to call first)
- Clustering: which items are similar to each other? (customer segments, with no labels in advance)
- Anomaly detection: which items are unusual? (transactions unlike the customer's normal pattern)
- Extraction: which facts are in this document? (invoice number, amount and date from a scanned invoice)
- Generation: produce new content (draft a reply; summarize a case file)

When AI is the right answer

AI is not automatically the best solution. A useful test asks four questions. Volume: are there enough cases for automation to pay off? Variability: do the cases differ in ways that fixed rules cannot capture? Data: is there enough relevant, lawful, historical data, with known outcomes, to learn from? Tolerance for error: can the business live with some wrong answers, and is there a way to catch them?

If the rule can be written down completely and never varies, such as a tax rate set in legislation, conventional logic is cheaper, more accurate and easier to audit. If the problem is caused by a broken process, such as a form that collects the wrong information, fixing the process comes first. AI is strongest where the volume is high, the patterns are real but hard to state as rules, and good historical data exists.

Where business analysis fits in the AI life cycle

CRISP-DM (Cross-Industry Standard Process for Data Mining) is the most widely used life cycle for analytics and ML work. Its six phases are business understanding, data understanding, data preparation, modeling, evaluation and deployment. The phases loop back on each other: evaluation often sends the team back to business understanding or data preparation.

Business analysis leads business understanding (what decision are we improving, how will success be measured) and evaluation (does the model meet the business need, not only the technical metrics), and contributes heavily to data understanding and deployment. Classic business analysis techniques, such as stakeholder analysis, process modeling, interviews and requirements traceability, apply throughout. What changes is that requirements must describe acceptable error, data and ongoing monitoring.

Stakeholders unique to AI work

Beyond the usual sponsor, users and IT, AI initiatives bring in data owners (who can approve use of each data source), data scientists and ML engineers, model risk or validation teams (common in banking and insurance), privacy and legal advisers, ethics or responsible AI review boards, and the people affected by the model's decisions, who may never use the system themselves. Missing one of these groups is a common cause of late surprises.

Worked example: one request, three framings

A contact center asks for "AI to handle complaints." Framed as classification, the model would sort incoming complaints into categories so they reach the right team. Framed as ranking, it would order the queue so the most urgent complaints are handled first. Framed as generation, it would draft a reply for an agent to edit.

Each framing needs different data (past categories, past escalations, past replies), different measures (share routed correctly, time to handle urgent complaints, agent edit rate) and carries different risks. The analyst's first deliverable is a one-page problem statement that names the framing, the decision it supports and how success will be measured, agreed by the sponsor before any data work starts.

Key terms

Model	The learned pattern produced by training, which software calls to make predictions.
Training / inference	Building the model from historical examples / using it on new cases in operation.
Feature	An input fact the model uses to make its prediction.
Label (target)	The known correct answer in historical data that the model learns to predict.
Supervised / unsupervised learning	Learning from labeled examples / finding structure without labels.
Threshold	The score at which a prediction is turned into a decision or action.
CRISP-DM	Six-phase life cycle: business understanding, data understanding, data preparation, modeling, evaluation, deployment.

Exercise

Take three requests from your own backlog. For each, state the business question, the type of AI task it would be (if any), and what the output would look like to the person who receives it.

Self-check

- Can you explain to a sponsor, without jargon, why an ML model will sometimes be wrong?
- For a request you know, which of the seven framings fits best, and what would its output look like?
- Which of the four suitability questions would most often rule AI out in your organization?

Module 2

Strategy analysis and the AI business case

Deciding whether AI is worth doing, comparing it with simpler options, and setting success measures that the business, not only the data team, understands.

In this module

- Current state, future state and the change strategy for an AI initiative
- Comparing options: process change, rules, robotic process automation, buy or build
- Costs and benefits that include the cost of errors and of human review
- Business outcome measures versus model measures; proof of concept versus pilot

Start from the business need

Strategy analysis defines the business need, describes the current state, defines the future state and works out a change strategy to move from one to the other. On AI initiatives the temptation is to start from the technology ("we need an AI strategy") rather than the need. The analyst brings the conversation back to the problem: what decision or task is slow, costly, inconsistent or error-prone today, and how will we know it has improved?

Measure the current state before proposing anything. How many cases are there, how long does each take, what does each cost, and how often is the current process wrong? Without today's error rate there is nothing to compare a model against, and many AI business cases fail because the baseline was never measured.

Comparing the options

Most needs that arrive labeled "AI" can be met in several ways, and the analyst should present them side by side:

- Process change: remove the step, change a form, change a policy. Often the cheapest and fastest option.
- Conventional rules or workflow automation: suits stable, fully specified logic.
- Robotic process automation (RPA): software that repeats keystrokes in existing systems. Suits structured, repetitive tasks; it does not learn or handle variation well.
- Buy an AI product or service: faster to start, but the organization depends on the vendor's data handling, accuracy claims and roadmap.
- Build a custom model: most control and fit, but needs data, skills and long-term ownership for monitoring and retraining.

Costs and benefits specific to AI

An AI business case has costs that conventional projects do not: preparing and labeling data, the human review of low-confidence cases, monitoring the model after go-live, periodic retraining, and the cost of the errors the model will make. A model that automates 80% of cases still leaves 20% for people, and the people now handle the hardest cases, which may take longer than average.

Each type of error has a cost. Missing a fraudulent claim costs the claim amount; wrongly flagging a genuine one costs investigator time and customer goodwill. The business case should estimate both, because they drive the threshold, the staffing for review and ultimately whether the model is worth building.

Measures of success

Distinguish business outcome measures from model measures. Business outcome measures are what the sponsor cares about: cost per case, time to decision, customer satisfaction, losses prevented. Model measures, such as precision and recall (Module 3), describe how well the model predicts. A model can score well on its measures and still fail to improve the business outcome, for example if staff ignore its recommendations or it is used on the wrong cases. Both kinds of measure belong in the requirements, and each model measure should be traced to the business outcome it supports.

Feasibility, proofs of concept and pilots

Assess feasibility on three fronts before committing: data (does enough relevant, lawful data exist?), technical (can a model reach the needed performance, and can it be integrated?) and organizational (will people use it, and can the organization own it after launch?).

A proof of concept (PoC) answers a narrow technical question cheaply, usually on historical data: can a model reach the needed accuracy at all? A pilot tests the full solution, including people and process, in a limited live setting: does it work in practice and deliver the outcome? A successful proof of concept is not evidence that the business outcome will be achieved; that needs a pilot.

Questions for vendors

When buying an AI product, ask: on what data was the model trained, and is it relevant to our cases? How was its accuracy measured, and can we test it on our own sample before signing? Will our data be used to train the vendor's models for other customers? Where is our data stored and processed? How will we be told about model changes? Can we get the explanations, logs and records our regulators or auditors need? What happens to our data if we leave?

Worked example: the hidden review cost

An insurer handles 100,000 claims a year at 30 minutes each. A vendor's model can approve 70% of claims automatically. The remaining 30% go to adjusters, but these are now the complex claims and take 50 minutes each on average.

Before: $100,000 \times 30 = 3,000,000$ minutes. After: $30,000 \times 50 = 1,500,000$ minutes. The real saving is 50% of adjuster time, not the 70% the headline figure suggests. The business case must also add the cost of the errors among the automatically approved claims, monitoring and retraining. The analyst's role is to make these numbers visible before the investment decision, not after.

Key terms

Current state / future state	How the work is done and performs today / how it should work after the change.
Baseline	Measured performance of the current process, including its error rate, used for comparison.
RPA	Robotic process automation: software that repeats user actions in existing systems; it does not learn.
Business outcome measure	A measure the sponsor cares about, such as cost per case or time to decision.
Model measure	A measure of predictive performance, such as precision or recall.
Proof of concept	A small, usually offline test of whether something is technically possible.
Pilot	A limited live trial of the full solution, including people and process.

Exercise

For a request in your organization, compare three options (process change, conventional automation, AI) in a one-page table showing cost, time to value, error handling and data needed.

Self-check

- What is the measured error rate of a process you support today? If you do not know, how would you find out?
- Name one AI request you have seen that would have been better solved by a process change.
- What would you ask a vendor who claims 98% accuracy?

Module 3

Requirements for probabilistic systems

Conventional software is right or wrong. A model is right most of the time. Its requirements must say how often, measured how, for whom, and what happens to the cases it gets wrong or is unsure about.

In this module

- The confusion matrix: true and false positives and negatives
- Accuracy, precision and recall in business language, and why accuracy misleads
- Thresholds, confidence routing and the cost of each kind of error
- Non-functional requirements for models: latency, throughput, monitoring and retraining

Four outcomes of every prediction

Take a model that flags insurance claims as possible fraud. Every prediction falls into one of four boxes, together called the confusion matrix. A true positive is a fraudulent claim correctly flagged. A false positive is a genuine claim wrongly flagged, sometimes called a false alarm. A false negative is a fraudulent claim the model missed. A true negative is a genuine claim correctly left alone.

Business stakeholders rarely care about the matrix itself, but they care a great deal about its two error boxes, because false positives and false negatives have different costs and affect different people. Writing requirements for an AI system starts with asking which error matters more.

Accuracy, precision and recall

Accuracy is the share of all predictions that are correct. It is easy to explain and often misleading. If 5% of claims are fraudulent, a model that never flags anything is 95% accurate and completely useless. When one outcome is rare, which is common in fraud, defects, safety incidents and urgent requests, accuracy hides failure.

Precision answers: of the items the model flagged, how many were really positive? It is $TP / (TP + FP)$. Low precision means many false alarms, which wastes reviewers' time and erodes trust in the model.

Recall answers: of all the items that really were positive, how many did the model catch? It is $TP / (TP + FN)$. Low recall means many misses. Recall is sometimes called sensitivity or the detection rate.

Precision and recall pull against each other. Flagging more items usually catches more real cases (recall rises) but includes more false alarms (precision falls). The F1 score combines the two into a single number, which is convenient for comparing models but hides the trade-off the business needs to see, so requirements should usually state precision and recall separately.

Worked numbers

Of 1,000 claims, 50 are fraudulent. The model flags 60 claims, of which 40 are fraudulent. Precision is $40 / 60 = 67\%$: one flag in three is a false alarm. Recall is $40 / 50 = 80\%$: one fraud in five is missed. Accuracy is $(40 + 930) / 1,000 = 97\%$, because 930 of the 950 genuine claims were correctly left alone. The 97% sounds excellent; precision and recall tell the business what it is actually getting.

Thresholds and confidence routing

Most models output a score. The threshold decides what score triggers an action. Raising the fraud-flag threshold means fewer, more certain flags: precision goes up and recall goes down. Lowering it catches more fraud but creates more false alarms.

A common design uses two thresholds. Items the model is very confident about are handled automatically; items in the uncertain middle go to a person; and, in some designs, items it is confident are negative are passed straight through. The requirement must state each threshold or how it will be set, who reviews the uncertain items, and how many of them to expect, because that number drives staffing. If 10,000 documents arrive a day and 15% fall below the confidence threshold, 1,500 need human review every day.

Where the cost of each error is known, the threshold can be chosen to minimize total cost. A missed fraud worth \$8,000 and a false alarm costing \$40 of investigator time justify a low threshold: many false alarms are worth it to catch more fraud. A marketing model where a wasted offer costs little and a missed customer costs little more needs a different balance.

Writing the requirement

A useful requirement for a model states the measure, the level, the test basis and the handling of exceptions. Weak: "The system shall classify emails accurately." Strong: "The system shall assign the correct category to at least 92% of emails in the agreed evaluation set, with recall of at least 95% for the Complaint category; emails scored below 0.80 confidence shall be routed to the triage team."

Performance may differ between groups of cases, such as product lines, languages or customer segments. Where it matters, the requirement should set a minimum for each important segment, not only an overall figure, so that good average performance cannot hide poor performance for one group.

Non-functional requirements for models

- Latency: how quickly a prediction must be returned (for example, within 300 milliseconds during an online application).
- Throughput: how many predictions per hour or per day the system must handle at peak.
- Availability and fallback: what the business process does if the model is unavailable.
- Monitoring: which measures are tracked after go-live, how often, and who is alerted when they fall below agreed levels.

- Retraining: how often, or on what trigger, the model is retrained, and what approval is needed before a retrained model goes live.
- Explainability and records: what must be logged for each decision so it can be explained and audited later (see Module 8).

Worked example: setting the threshold with the business

A hospital uses a model to flag discharge summaries that may be missing a medication review. A miss can harm a patient; a false alarm costs a pharmacist five minutes. The analyst runs a workshop with the pharmacy lead, using the model's results at three candidate thresholds on last quarter's records: at 0.3, recall is 97% with 220 flags a day; at 0.5, recall is 91% with 120 flags; at 0.7, recall is 78% with 60 flags.

The pharmacy lead chooses 0.3 and adds a pharmacist to the rota, because missing reviews is the costlier error. The decision, the evidence and the owner are recorded, and the requirement states both the recall target and the expected daily review volume.

Key terms

Confusion matrix	Table of true positives, false positives, false negatives and true negatives.
False positive	An item wrongly flagged as positive (a false alarm).
False negative	A real positive the model missed.
Accuracy	Share of all predictions that are correct; misleading when one outcome is rare.
Precision	Of the items flagged, the share that were really positive: $TP / (TP + FP)$.
Recall	Of all real positives, the share the model caught: $TP / (TP + FN)$.
F1 score	A single figure combining precision and recall.
Confidence routing	Sending items the model is unsure about to a person, based on its score.

Exercise

For a document classification scenario, write the requirement that states the acceptable error rate, the test set it will be measured on, and what happens to low-confidence items.

Self-check

- In a process you know, which costs more: a false positive or a false negative? Who bears each cost?
- Can you calculate precision and recall from a confusion matrix without looking up the formulas?
- How would you explain to a sponsor why 97% accuracy might not be good enough?

Module 4

Data requirements and readiness

Most AI projects that fail do so on data. The analyst documents what data exists, where it comes from, how good it is, whether it represents the cases the model will see, and whether it may lawfully be used.

In this module

- Data inventories, lineage and source-to-feature mapping
- Data quality dimensions and representativeness
- Labeling: guidelines, agreement between labelers and gold-standard sets
- Data leakage, train/validation/test splits, and permitted use of personal data

Inventory, ownership and lineage

A data inventory lists each source the model may use: what it contains, which system holds it, who owns it, how far back it goes, how often it is updated and what restrictions apply. Lineage records where each item of data came from and what was done to it on the way, so that a figure used by the model can be traced back to its origin.

A source-to-feature map links each model input (feature) to the source field or fields it is built from and the transformation applied, for example "customer tenure in months = today minus account open date, from the core banking system." This map is the AI equivalent of a data dictionary and is essential for testing, audits and later changes.

Every source needs a data owner who approves its use for this purpose. Discovering at the end of a project that the owner, or the law, does not permit the use is one of the most expensive surprises in AI work.

Data quality dimensions

Data quality is usually assessed on six dimensions, widely used in data management practice:

- **Completeness:** are the needed values present? (40% of records missing the postal code is a completeness problem.)
- **Accuracy:** do the values correctly describe the real world? (a recorded birth date that is not the customer's real birth date)
- **Consistency:** do the same facts agree across systems and records? (a customer marked active in one system and closed in another)
- **Timeliness:** is the data up to date and available when the decision is made?
- **Validity:** do values follow the required format and rules? (a date field containing text)
- **Uniqueness:** is each real-world item recorded once, without duplicates?

Representativeness and history

A model learns from the past and assumes the future will look similar. Training data must represent the cases the model will actually see: the same products, regions, channels, languages and customer groups, in realistic proportions. A model trained only on online applications may perform poorly on branch applications.

Historical data also carries the decisions and habits of the past. If past loan approvals were unfair to a group, a model trained to imitate those approvals will learn the same unfairness. The analyst should ask whether the historical outcome is the outcome the business actually wants to reproduce.

Labels and labeling

Supervised models need labeled examples. Sometimes labels already exist, such as the outcome of past investigations. Often they must be created by people reading cases and recording the right answer. Labeling needs clear written guidelines with examples, especially for borderline cases, so that different labelers make the same call.

Inter-annotator agreement measures how often two labelers give the same label to the same item. Low agreement means the categories or the guidelines are unclear, and no model can be expected to do better than people can agree. A gold-standard set is a sample labeled very carefully, often by experts, and used to check labelers and to evaluate the model.

Leakage and data splits

Data leakage happens when the training data includes information that would not be available at the moment the model is used. For example, a field recording "investigation opened" is filled in only after a claim is suspected; a fraud model that uses it will look excellent in testing and fail in production. The analyst can spot leakage by asking, for each feature, when the value becomes known in the real process.

Data is normally split into a training set (used to build the model), a validation set (used to tune it) and a test set held back until the end to give an honest estimate of performance. If the test set is used repeatedly to make design choices, or overlaps with the training data, the reported performance will be too optimistic. For data that changes over time, testing on the most recent period gives a more realistic estimate than a random split.

Permitted use of personal data

Privacy laws in most jurisdictions share common principles. Purpose limitation: data collected for one purpose may not automatically be used for a different one. Data minimization: use only the personal data needed for the purpose. Retention: keep data no longer than needed. Consent and notice: people may need to be told, and sometimes asked, before their data is used in a new way. Sensitive data, such as health information, usually carries stricter rules.

The analyst does not decide what the law allows, but makes sure the question is asked for every source and the answer is recorded, with privacy or legal advisers. A datasheet for a dataset, a short document recording its origin, contents, collection method, known limitations and permitted uses, is a good way to capture this.

Worked example: the field that knew the answer

A team builds a model to predict which new customers will cancel within 90 days. In testing it is 99% accurate. The analyst walks through each feature with the operations team, asking when the value is known. One field, "retention offer made," is only filled in after an agent has already decided the customer is about to cancel.

The field is removed. Performance drops to a realistic level, and the model is now useful at the point of decision, when the customer signs up. The analyst adds a check to the requirements: every feature must be available at the moment of prediction, confirmed against the process model.

Key terms

Data lineage	Record of where data came from and how it was transformed.
Source-to-feature map	Links each model input to its source fields and the transformation applied.
Completeness / accuracy / consistency	Values present / values true to the real world / facts agree across systems.
Timeliness / validity / uniqueness	Up to date and available when needed / follows format rules / no duplicates.
Inter-annotator agreement	How often different labelers assign the same label to the same item.
Gold-standard set	A carefully labeled sample used to check labelers and evaluate the model.
Data leakage	Training data that contains information not available when the model is used.
Data minimization	Using only the personal data needed for the purpose.

Exercise

Build a one-page data readiness assessment for a proposed model, rating each source on completeness, accuracy, consistency and timeliness, and noting who owns it and whether its use is permitted.

Self-check

- For a model you might build, who owns each source of data, and have they approved this use?
- Which quality dimension is weakest in your organization's customer data?
- How would you check whether a feature could leak the answer?

Module 5

Requirements for generative AI solutions

Assistants, summarizers and drafting tools built on large language models need requirements for grounding, accuracy checking, safety, confidentiality and cost that classic requirements templates do not cover.

In this module

- How large language models work, and why they produce confident errors
- Prompts, retrieval-augmented generation (RAG) and grounding in approved sources
- Guardrails, prompt injection and confidentiality requirements
- Evaluating generated output: rubrics, reference sets and human review

What a large language model does

A large language model (LLM) is trained on very large amounts of text to predict the next word in a sequence. That simple mechanism, at scale, lets it write, summarize, translate, answer questions and follow instructions. It does not look facts up unless it is connected to a source; it generates what is likely given its training and the prompt.

Because of this, an LLM can produce fluent, confident statements that are false, such as invented policy clauses, citations or figures. This is called a hallucination (or fabrication). It is a property of how the model works, not a bug that will simply be fixed, so requirements must assume it can happen and say how it will be prevented, detected and handled.

LLMs are also non-deterministic by default: the same question can get differently worded answers on different occasions. Requirements and tests must allow for this, judging answers on whether they are correct and complete rather than on exact wording.

Prompts, grounding and RAG

A prompt is the instruction and context given to the model. A system prompt sets standing rules the user does not see, such as "Answer only from the documents provided; if the answer is not in them, say so."

Retrieval-augmented generation (RAG) is the most common pattern for business use. When a user asks a question, the system first searches an approved collection of documents, then gives the most relevant passages to the model with the instruction to answer from them. This grounds the answer in the organization's own content and allows the answer to cite its sources, so that users can check them.

RAG moves much of the quality problem into the document collection. If the policy library contains outdated or conflicting versions, the assistant will confidently quote them. Requirements for a RAG solution therefore include which sources are in scope, who owns them, how often they are refreshed, how superseded versions are removed and how access permissions are respected, so that a user is never

shown content they are not allowed to see.

Choosing generative AI or something else

Generative AI suits tasks where the output is language and some variation is acceptable: drafting, summarizing, explaining, answering questions from documents. Where the output must be one of a fixed set of values, such as a category, a yes/no decision or a number read from a form, a classification or extraction model, or plain rules, is usually more reliable, cheaper and easier to test. Asking an LLM to make a high-stakes decision on its own, with no grounding or review, is rarely appropriate.

Guardrails and security

Guardrails are the controls around the model: input checks, output checks, topic restrictions, filters for harmful or confidential content, and rules for when to refuse or hand over to a person. Requirements should state what the assistant must decline to do, such as giving legal advice or discussing individual employees, and what it says when it declines.

Prompt injection is an attack in which text supplied by a user, or hidden in a document or web page the system reads, tries to override the system's instructions ("ignore your previous instructions and..."). Requirements should call for testing against it and for limiting what the assistant can do, especially if it can take actions such as sending email or updating records.

Confidentiality matters in two directions. Staff may paste confidential information into a public tool; requirements should specify an approved service where prompts and data are not retained or used to train the provider's models. And the assistant itself must not reveal information to people who are not entitled to it.

Evaluating generated output

Because answers vary in wording, generative systems are evaluated with a reference set of realistic questions or tasks, each with a known good answer or the key facts a good answer must contain, and a rubric: the criteria a reviewer uses to score each answer, such as factually correct, complete, grounded in the cited source, appropriate in tone, and declined when it should be. Subject-matter experts review a sample, and the share of answers meeting the rubric becomes the acceptance measure.

Useful operational measures include the share of answers with a valid citation, the share of out-of-scope questions correctly declined, user ratings, how often users edit drafted text before sending it, and cost and response time per request. LLM services are usually charged by the volume of text processed (tokens), so cost requirements should be estimated from expected usage.

Worked example: an HR policy assistant

Requirements agreed with HR included: answers drawn only from the current published policy library, with a link to the section used; if no relevant section is found, the assistant says so and gives the HR contact address; no answers about named individuals; superseded policy versions removed from the library within one working day of a new version being published; and staff prompts not retained by the service provider.

Acceptance used 150 real staff questions collected from the HR inbox, including 20 deliberately out of scope. HR advisers scored each answer against a four-point rubric. The pass mark was 90% of in-scope answers correct and grounded, and every out-of-scope question declined. The first run failed on outdated documents in the library, a content problem the model could not fix.

Key terms

Large language model (LLM)	A model trained on large amounts of text to generate language by predicting likely next words.
Hallucination	Fluent, confident output that is false or not supported by any source.
Prompt / system prompt	The instruction given to the model / standing instructions set by the system, not the user.
RAG	Retrieval-augmented generation: retrieving approved content and having the model answer from it.
Grounding	Tying generated answers to specific, checkable sources.
Guardrails	Controls on what goes into and comes out of the model, including refusal rules.
Prompt injection	Text that tries to override the system's instructions, from a user or hidden in content.
Rubric	The criteria used to score generated answers consistently.

Exercise

Write five requirements for an internal policy assistant that answers staff questions from the HR policy library, covering sources, citations, out-of-scope questions, confidentiality and how answer quality will be evaluated.

Self-check

- Why will connecting an assistant to an out-of-date document library produce confident wrong answers?
- For a generative tool you know, what should it refuse to do, and what should it say when it refuses?
- How would you build a reference set to test it?

Module 6

Modeling human and machine work together

Process models that show where the model decides, where a person reviews, how exceptions flow, and how corrections feed back into the model, so that the combined system works and stays accountable.

In this module

- Human-in-the-loop, human-on-the-loop and fully automated patterns
- Swimlane and BPMN models with the model as a participant
- Decision tables alongside model outputs; exception paths
- Feedback loops, automation bias and review workload

Three patterns of oversight

In a human-in-the-loop design, a person reviews and confirms the model's output before it takes effect, at least for some cases. In a human-on-the-loop design, the system acts automatically while a person monitors it, can intervene and reviews samples or alerts. In a fully automated (human-out-of-the-loop) design, the system acts with no routine human involvement.

The right pattern depends on the impact of an error, the volume, the model's measured performance and any legal requirements. Decisions with significant effects on people, such as credit, employment or benefits, usually need a person in the loop, or at least a route for people to ask for human review. Low-impact, high-volume decisions can often be automated with monitoring.

Modeling the process

Swimlane diagrams and BPMN (Business Process Model and Notation) models work well for AI processes when the model is shown as its own participant or lane, alongside the reviewer and the systems. This makes visible exactly which decisions the model makes, which it only suggests, where a person must act, and what happens on each path.

Every model step has at least three exits that the process must handle: a confident result that proceeds, an uncertain result that goes to a person, and a failure (the model is unavailable or the input cannot be processed) that follows a fallback. Missing the failure path is one of the most common gaps in AI process designs.

Decision tables with model outputs

Model outputs rarely act alone. Business rules usually sit around them: a claim with a low fraud score is fast-tracked only if it is also under the policy limit and the customer has no open complaint. Decision tables, such as those defined in DMN (Decision Model and Notation), show these combinations clearly, with the model's score as one input among others. This keeps policy decisions in rules that the business owns and can change, rather than buried inside the model.

Feedback loops and corrections

When a reviewer corrects the model, for example changing the category it assigned, that correction is valuable new labeled data. The process should capture it in a structured form (the original prediction, the correction and the reason) so it can be used to monitor performance and to retrain the model. A correction made by simply retyping a value in another system is lost.

Feedback loops can also cause harm. If a model decides which cases are investigated, and only investigated cases produce new fraud labels, the model only ever learns about the kinds of cases it already suspects. Taking a small random sample of cases for review, regardless of score, provides unbiased data to check and retrain the model.

Automation bias and review design

Automation bias is the tendency of people to accept a system's suggestion without enough scrutiny, especially when they are busy and the system is usually right. A reviewer who confirms 400 suggestions an hour is not really reviewing. Review steps need realistic workloads, clear guidance on what to check, and sometimes design choices such as asking the reviewer to judge before seeing the model's suggestion, or inserting known test cases to check that reviews are genuine.

Review workload is calculated from volume and routing: if 12,000 items arrive a week, 25% go to review and each takes four minutes, review needs $3,000 \times 4 = 12,000$ minutes, or 200 hours a week. This figure belongs in the business case and the transition plan.

Roles and accountability

A responsibility assignment (RACI) for the AI solution names who is accountable for the business decisions it supports, who owns the model's performance, who monitors it, who can switch it off, and who approves changes such as new thresholds or retrained versions. A model cannot be accountable; a named person or role must be.

Worked example: invoice processing

The future-state swimlane has four lanes: supplier portal, extraction model, accounts payable clerk and ERP system. Invoices extracted with confidence of 90% or more on every key field are matched to the purchase order by a decision table; matches within tolerance are posted automatically. Invoices below 90% go to the clerk, who sees the image and the extracted fields side by side and corrects them; each correction is stored with the original value. If the model is unavailable, invoices queue for manual entry and the clerk is alerted.

Each month, 2% of automatically posted invoices are sampled at random for checking. Corrections and sample results are reported to the AP manager, who is accountable for the process, and are used for the quarterly retraining decision.

Key terms

Human-in-the-loop	A person reviews or confirms the model's output before it takes effect.
Human-on-the-loop	The system acts automatically while a person monitors and can intervene.
BPMN	Business Process Model and Notation, a standard for process diagrams.
DMN / decision table	Decision Model and Notation; a table of conditions and outcomes owned by the business.
Feedback loop	Using reviewers' corrections and outcomes to monitor and retrain the model.
Automation bias	The tendency to accept a system's suggestion without enough scrutiny.
Fallback path	What the process does when the model is unavailable or cannot process an input.

Exercise

Draw the future-state process for invoice handling where extraction is automated and a clerk reviews items below 90% confidence, including the exception path and how corrections are captured.

Self-check

- For a decision in your organization, which oversight pattern would you choose, and why?
- Does the process model you would draw have a path for when the model is unavailable?
- How would you know whether your reviewers are really reviewing?

Module 7

Acceptance, deployment and monitoring

Acceptance criteria for AI are statistical and agreed before the model is built. After go-live the model's performance changes as the world changes, so monitoring and retraining are requirements, not afterthoughts.

In this module

- Representative evaluation sets, baselines and segment-level results
- Overfitting, shadow mode, A/B tests and champion/challenger
- Go/no-go criteria and business sign-off
- Data drift, concept drift, retraining triggers, versioning and rollback

The evaluation set

The evaluation (test) set is the collection of real, correctly labeled cases on which the model is judged. It must be held out from training, represent the cases the model will meet in production (products, channels, seasons, languages), and be large enough for the rare categories that matter. If urgent complaints are 1% of email and the set has 500 emails, it holds about five urgent ones, far too few to measure recall for that category. The set can be deliberately enriched with rare cases, as long as results are reported separately for each category.

Agree the evaluation set, the measures and the pass marks before the model is built and before results are seen. Choosing the test after seeing the results invites moving the goalposts.

Baselines and segments

Results only mean something compared with a baseline: usually the current process's measured performance, and sometimes a simple rule ("flag every claim over \$10,000"). A model that is 88% correct sounds weak until you learn the manual process is 81% correct; a model that is 95% correct is not worth deploying if a simple rule achieves 94%.

Report results for each important segment as well as overall: by product, region, language, channel and, where relevant, by group of people affected. A model can pass overall and fail badly for one segment.

Overfitting

An overfitted model has learned the training data too closely, including its noise and quirks, so it performs very well on data it has seen and noticeably worse on new data. A large gap between training performance and test performance is the warning sign. Honest, held-out evaluation is the defense; the analyst's role is to make sure acceptance is based on test results, never training results.

Ways to trial a model safely

- Shadow mode: the model runs on live cases alongside the current process, but its outputs are recorded and not acted on. Its decisions are compared with what actually happened. No customer is affected, and it tests the model on real, current data.
- Pilot: the model is used for real in a limited scope, such as one region or one team, with close monitoring.
- A/B test: live cases are split at random between the current process (A) and the new one (B), so outcomes can be compared fairly.
- Champion/challenger: the current production model (the champion) keeps running while a new candidate (the challenger) is tested on a share of cases; the challenger replaces it only if it performs better.

Go/no-go and sign-off

Go/no-go criteria list, in advance, what must be true to go live: performance at or above the agreed levels overall and for each segment, better than the baseline, review staffing in place, monitoring working, fallback tested, users trained, and approvals from risk, privacy and the business owner. Sign-off should be presented in business terms: "Out of 1,000 typical emails, about 920 will reach the right team automatically, 60 will go to the triage team, and about 20 will be misrouted, compared with 70 misrouted today."

After go-live: drift and decay

Models degrade over time because the world changes. Data drift is a change in the inputs: new products, a new customer segment, a changed form, a new document layout. Concept drift is a change in the relationship between inputs and outcomes: fraudsters change their tactics, or customer behavior shifts after a price change, so patterns that once predicted the outcome no longer do.

Monitoring requirements state which measures are tracked (input distributions, score distributions, precision and recall on a reviewed sample, business outcomes), how often, the thresholds that trigger an alert, and who acts. Retraining can be scheduled (for example quarterly) or triggered by monitoring. A retrained model is a new version and goes through the same acceptance before it replaces the old one.

Every model in production should have a version number recorded with each decision, and a rollback plan: the ability to return quickly to the previous version or to the manual process if a new version misbehaves.

Worked example: shadow mode before go-live

A lender runs a new credit-limit model in shadow mode for eight weeks. Every application is still decided by the existing process; the model's recommendation is recorded beside it. At the end, the team compares recommendations with the actual decisions and early repayment behavior, overall and by region and product.

Results meet the go/no-go criteria in four regions but not in one, where a new broker channel had launched after the training data was collected: data drift before launch. The model goes live in four regions, and the fifth waits for retraining with the new channel's data.

Key terms

Evaluation set	Held-out, representative, labeled cases on which the model is judged.
Overfitting	Performing well on training data but worse on new data.
Shadow mode	Running the model on live cases without acting on its outputs, to compare.
A/B test	Randomly splitting live cases between the current and new process to compare outcomes.
Champion/challenger	Testing a new model against the current production model before replacing it.
Data drift	A change in the inputs the model receives.
Concept drift	A change in the relationship between inputs and the outcome.
Rollback	Returning to a previous model version or manual process.

Exercise

Write acceptance criteria for a customer-email triage model, including the size and make-up of the evaluation set, the baseline, the segment-level minimums and the go/no-go rule.

Self-check

- Could you explain the difference between data drift and concept drift with an example from your industry?
- What would your go/no-go checklist include beyond model performance?
- How would you present acceptance results to a sponsor who does not know what recall is?

Module 8

Responsible AI requirements and governance

Fairness, transparency, privacy and accountability written as requirements that can be traced and tested, and the frameworks and regulations that shape them.

In this module

- Affected groups, proxy variables and fairness measures
- Explainability by audience; transparency and notice
- Privacy impact assessment, decision records, model cards and audit trails
- Governance frameworks: NIST AI RMF, ISO/IEC 42001 and risk-based regulation such as the EU AI Act

From principles to requirements

Most organizations have published AI principles: fair, transparent, accountable, private, safe. Principles are not requirements; they cannot be tested. The analyst's job is to turn each principle into specific, measurable, traceable requirements for this solution, with an owner and a test, in the same way as any other requirement.

Fairness

Start by identifying the groups who could be affected differently, including those protected by human rights or anti-discrimination law, such as groups defined by sex, age, race, disability or religion, depending on the jurisdiction. Then decide how fairness will be measured for this decision.

Two common measures compare groups. Selection rate parity (often called demographic parity) compares the share of each group that receives the favorable outcome, such as approval. Error rate parity compares the model's mistakes across groups; for example, equal opportunity compares the share of genuinely qualified people in each group whom the model wrongly rejects. These measures can conflict, and in general they cannot all be satisfied at once, so choosing which matters most is a policy decision for accountable owners, documented with its reasons, not a technical default.

Removing a protected attribute, such as sex, from the model's inputs does not guarantee fairness. Other features can act as proxies, because they are closely linked to the protected attribute: postal code can track ethnicity; career gaps can track sex. Fairness must be tested on outcomes, which may require holding protected attributes securely for testing even when the model does not use them.

Explainability and transparency

Different audiences need different explanations. A customer who is declined needs the main reasons in plain language and what they can do about it. A front-line employee needs enough to handle the case and answer questions. An auditor or regulator needs to understand how the model works overall and to reconstruct individual decisions. A global explanation describes how the model behaves in general (which factors matter most overall); a local explanation describes why it produced a particular result for one case.

Transparency also means telling people when AI is involved: that they are talking with a chatbot rather than a person, that content was generated by AI, or that an automated system contributed to a decision about them, and how to ask for human review.

Privacy, records and documentation

A privacy impact assessment (PIA), called a data protection impact assessment (DPIA) in some jurisdictions, identifies the personal data involved, the risks to individuals and the controls, before the system is built. The analyst supplies much of its content: data flows, purposes, retention and who has access.

Decision records make individual decisions reconstructable: the input data used, the model version, the score, the threshold applied, the outcome, any human reviewer and their decision, and the time. Without them, a complaint or audit cannot be answered.

A model card is a short document that describes a model: its intended use and uses it is not suitable for, the data it was trained and evaluated on, its performance overall and by group, known limitations and ethical considerations. It travels with the model so that people who use or approve it understand what it can and cannot do.

Frameworks and regulation

The NIST AI Risk Management Framework (AI RMF), published by the US National Institute of Standards and Technology, organizes AI risk management into four functions: Govern (policies, roles and culture), Map (understand the context and identify risks), Measure (analyze and track the risks) and Manage (prioritize and act on them). ISO/IEC 42001 is an international standard for an AI management system, against which organizations can be certified, much as ISO/IEC 27001 does for information security.

Regulation is increasingly risk-based. The European Union's AI Act sorts AI uses into tiers: unacceptable-risk practices that are prohibited (such as social scoring that leads to unjustified treatment of people); high-risk uses with strict requirements for risk management, data quality, documentation, human oversight and accuracy (examples include AI used in recruitment, credit scoring, education and access to essential services); limited-risk uses with transparency duties (such as telling people they are interacting with a chatbot); and minimal-risk uses with no specific obligations. Even organizations outside the EU are affected when their systems are used there, and the same risk-based thinking appears

in guidance in many other countries. The analyst should find out early which tier and which obligations apply, with legal advice.

Traceability

Responsible AI requirements must be traced like any others: from the principle or regulation, to the requirement, to the design element, to the test and its result. "The share of qualified applicants wrongly rejected shall not differ by more than five percentage points between women and men on the evaluation set" can be tested; "the model shall be fair" cannot. Traceability is also the evidence an auditor or regulator will ask for.

Worked example: a recruitment screening tool

A company plans to screen job applications with a model. The analyst flags it early as a likely high-risk use under risk-based regulation such as the EU AI Act, and the legal team confirms it. Requirements added include: selection rates compared across sex and age bands on the evaluation set, with a defined minimum ratio; a local explanation for every rejected application, available to recruiters; every applicant told that automated screening is used and how to request a human review; decision records kept for the retention period; and a model card approved by HR and legal before go-live.

Testing shows the model does not use sex, but selection rates for women are much lower. Investigation finds a proxy: a feature counting years of continuous employment, which penalizes career breaks. The feature is removed and the model retrained, and the requirement and test are kept for future versions.

Key terms

Selection rate parity	Comparing the share of each group that receives the favorable outcome (also called demographic parity).
Equal opportunity	Comparing, across groups, how often genuinely qualified people are wrongly rejected.
Proxy variable	A feature closely linked to a protected attribute, such as postal code and ethnicity.
Global / local explanation	How the model behaves overall / why it produced one particular result.
PIA / DPIA	Privacy (data protection) impact assessment carried out before building the system.
Decision record	Log of inputs, model version, score, threshold, outcome and reviewer for each decision.
Model card	Short document of a model's intended use, data, performance by group and limitations.
NIST AI RMF	Framework with four functions: Govern, Map, Measure, Manage.
ISO/IEC 42001	International standard for AI management systems; certifiable.

Exercise

Add three testable responsible-AI requirements to an existing requirements document, covering fairness, explanation and record-keeping, and trace each to the test that will verify it.

Self-check

- Can you turn one of your organization's AI principles into a requirement that a tester could verify?
- Which features in a dataset you know could act as proxies for a protected attribute?
- What would you need to reconstruct a decision made by a model six months ago?

Preparing for the exam

The AI-BAP exam has 50 multiple-choice questions drawn at random from every module. You have 90 minutes and need 38 correct answers to pass.

Before you start

- Finish every module's exercise. Most questions describe a situation and ask what a practitioner should do.
- Re-read the module summaries: they state the main idea each question is built on.
- Set aside uninterrupted time. The timer keeps running if you leave the page.
- Have this guide to hand. The exam is open book, but you will not have time to look up every answer.
- Pace yourself: about 1 minute 48 seconds per question. Answer the ones you are sure of, then return to the rest; unanswered questions count as wrong.
- Learn the key terms at the end of each module, and practice the calculations in Module 3 and Module 6 until you can do them without notes.
- Read every option before choosing. Several may sound reasonable; pick the one that best fits the situation described.

Taking the exam

Log in to your account on the Institute's website and choose Take the exam. Check that your name is shown exactly as it should appear on your certificate. Your result is shown as soon as you submit, and if you pass, your certificate and credential number are issued straight away.

Each enrollment includes 3 attempts, and an attempt counts as soon as you start it. If you don't pass, you can try again after 24 hours. Each attempt draws a different set of questions.

Glossary

The key terms from every module, in alphabetical order.

A/B test	Randomly splitting live cases between the current and new process to compare outcomes.
Accuracy	Share of all predictions that are correct; misleading when one outcome is rare.
Automation bias	The tendency to accept a system's suggestion without enough scrutiny.
Baseline	Measured performance of the current process, including its error rate, used for comparison.
BPMN	Business Process Model and Notation, a standard for process diagrams.
Business outcome measure	A measure the sponsor cares about, such as cost per case or time to decision.
Champion/challenger	Testing a new model against the current production model before replacing it.
Completeness / accuracy / consistency	Values present / values true to the real world / facts agree across systems.
Concept drift	A change in the relationship between inputs and the outcome.
Confidence routing	Sending items the model is unsure about to a person, based on its score.
Confusion matrix	Table of true positives, false positives, false negatives and true negatives.
CRISP-DM	Six-phase life cycle: business understanding, data understanding, data preparation, modeling, evaluation, deployment.
Current state / future state	How the work is done and performs today / how it should work after the change.
Data drift	A change in the inputs the model receives.
Data leakage	Training data that contains information not available when the model is used.
Data lineage	Record of where data came from and how it was transformed.
Data minimization	Using only the personal data needed for the purpose.
Decision record	Log of inputs, model version, score, threshold, outcome and reviewer for each decision.
DMN / decision table	Decision Model and Notation; a table of conditions and outcomes owned by the business.
Equal opportunity	Comparing, across groups, how often genuinely qualified people are wrongly rejected.
Evaluation set	Held-out, representative, labeled cases on which the model is judged.
F1 score	A single figure combining precision and recall.

Fallback path	What the process does when the model is unavailable or cannot process an input.
False negative	A real positive the model missed.
False positive	An item wrongly flagged as positive (a false alarm).
Feature	An input fact the model uses to make its prediction.
Feedback loop	Using reviewers' corrections and outcomes to monitor and retrain the model.
Global / local explanation	How the model behaves overall / why it produced one particular result.
Gold-standard set	A carefully labeled sample used to check labelers and evaluate the model.
Grounding	Tying generated answers to specific, checkable sources.
Guardrails	Controls on what goes into and comes out of the model, including refusal rules.
Hallucination	Fluent, confident output that is false or not supported by any source.
Human-in-the-loop	A person reviews or confirms the model's output before it takes effect.
Human-on-the-loop	The system acts automatically while a person monitors and can intervene.
Inter-annotator agreement	How often different labelers assign the same label to the same item.
ISO/IEC 42001	International standard for AI management systems; certifiable.
Label (target)	The known correct answer in historical data that the model learns to predict.
Large language model (LLM)	A model trained on large amounts of text to generate language by predicting likely next words.
Model	The learned pattern produced by training, which software calls to make predictions.
Model card	Short document of a model's intended use, data, performance by group and limitations.
Model measure	A measure of predictive performance, such as precision or recall.
NIST AI RMF	Framework with four functions: Govern, Map, Measure, Manage.
Overfitting	Performing well on training data but worse on new data.
PIA / DPIA	Privacy (data protection) impact assessment carried out before building the system.
Pilot	A limited live trial of the full solution, including people and process.
Precision	Of the items flagged, the share that were really positive: $TP / (TP + FP)$.
Prompt / system prompt	The instruction given to the model / standing instructions set by the system, not the user.
Prompt injection	Text that tries to override the system's instructions, from a user or hidden in content.

Proof of concept	A small, usually offline test of whether something is technically possible.
Proxy variable	A feature closely linked to a protected attribute, such as postal code and ethnicity.
RAG	Retrieval-augmented generation: retrieving approved content and having the model answer from it.
Recall	Of all real positives, the share the model caught: $TP / (TP + FN)$.
Rollback	Returning to a previous model version or manual process.
RPA	Robotic process automation: software that repeats user actions in existing systems; it does not learn.
Rubric	The criteria used to score generated answers consistently.
Selection rate parity	Comparing the share of each group that receives the favorable outcome (also called demographic parity).
Shadow mode	Running the model on live cases without acting on its outputs, to compare.
Source-to-feature map	Links each model input to its source fields and the transformation applied.
Supervised / unsupervised learning	Learning from labeled examples / finding structure without labels.
Threshold	The score at which a prediction is turned into a decision or action.
Timeliness / validity / uniqueness	Up to date and available when needed / follows format rules / no duplicates.
Training / inference	Building the model from historical examples / using it on new cases in operation.



International Institute of Applied AI Technology

A global leader in applied artificial intelligence education

iiaait.org