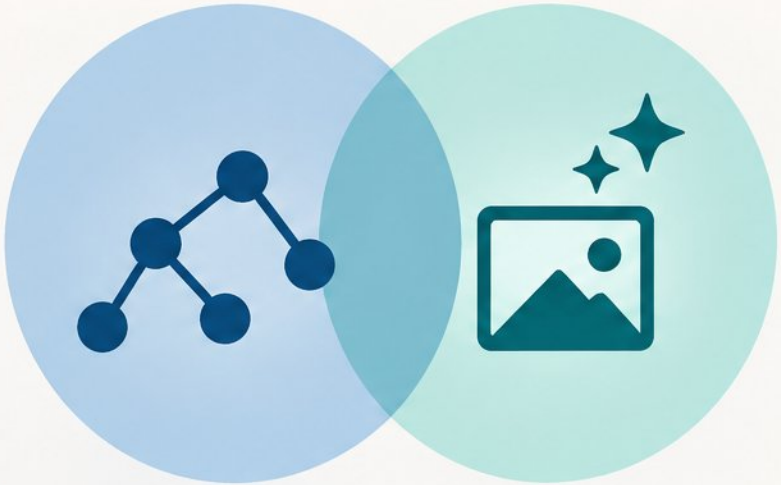
Generative AI Vs Machine Learning: Differences, Tradeoffs, and Use Cases



Goal: a shared, non-mathematical mental model for beginners and business teams

One-line distinction: ML learns patterns to predict or classify; GenAI learns patterns to create content

Why it matters: scoping, data needs, risk, cost, and evaluation all shift

Roadmap: foundations → differences → tradeoffs → use cases → decision framework → exercise

Running theme: "more AI" is not a strategy—matching the approach to the problem is

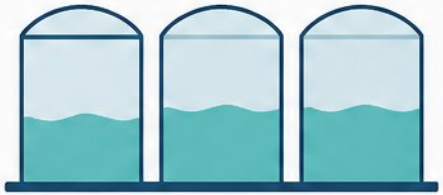
Foundations | Differences | Tradeoffs | Use Cases | Decision Framework | Exercise

Why This Distinction Matters Now

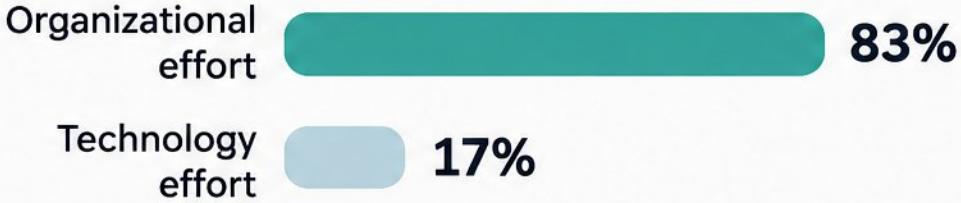
- Most enterprise AI failures are scoping failures, not model failures
- Only 1 in 4 agentic AI initiatives reaches production (HFS Research, 2026)
- 43% of AI spend is trapped in stalled pilots
- 77% of hard challenges are organizational: change management, data quality, process



1 in 4
initiatives in production



43%
of spend is locked in idle pilots



Learning checkpoint: Which approach fits here—and can you defend it?

Plain-Language Foundations: ML, Deep Learning, and GenAI

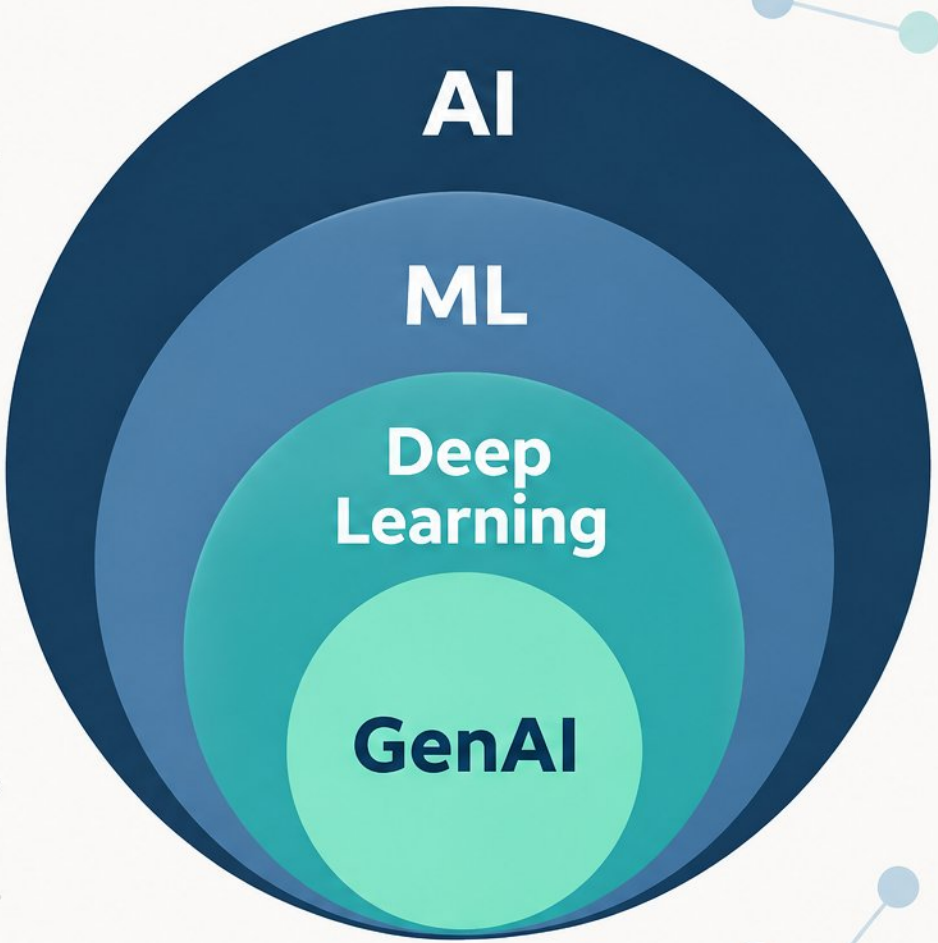
- **Machine learning** learns a function from examples: predict, classify, rank, forecast

- **Deep learning:** multi-layer neural networks powering both prediction and generation

- **Generative AI:** an ML subset producing text, images, or code—not a single label

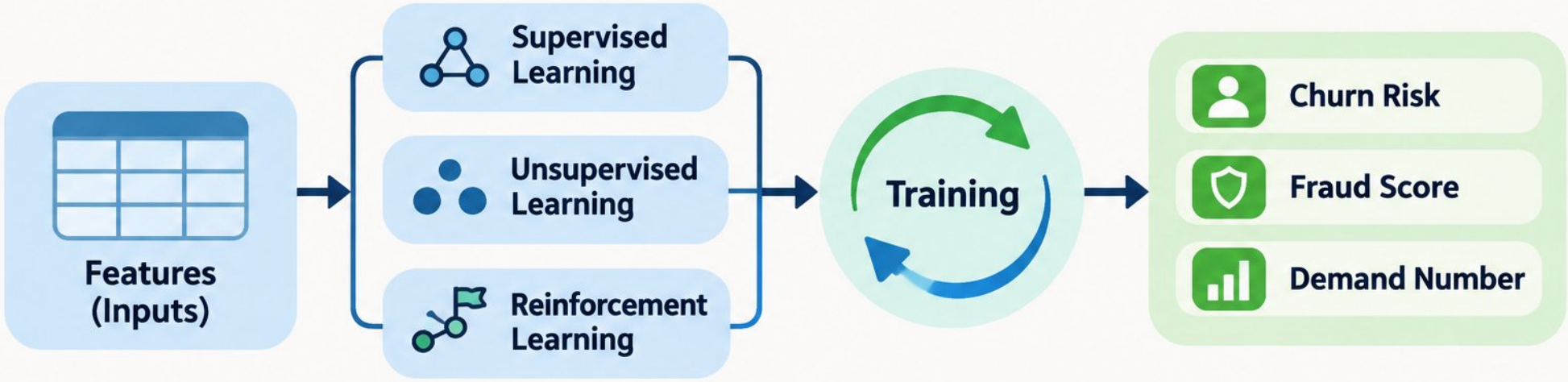
- **Retire three myths:** “all AI is generative”, “GenAI replaced ML”, “the model is a database”

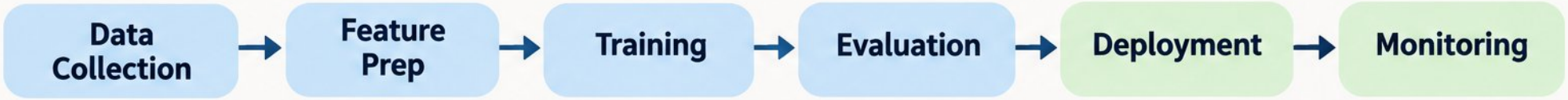
- **Mental model:** nested circles—AI ⊃ ML ⊃ deep learning ⊃ GenAI



How Machine Learning Works: Prediction and Classification

- **Core idea:** learn a mapping from inputs (features) to outputs (labels, scores, numbers)
- **Three families:** supervised, unsupervised, and reinforcement learning
- **Workflow:** data collection → feature prep → training → evaluation → deployment → monitoring
- **Business examples:** churn prediction, demand forecasting, fraud scoring, recommendation ranking
- Outputs are compact and scorable—measured by accuracy, precision, recall, or forecast error

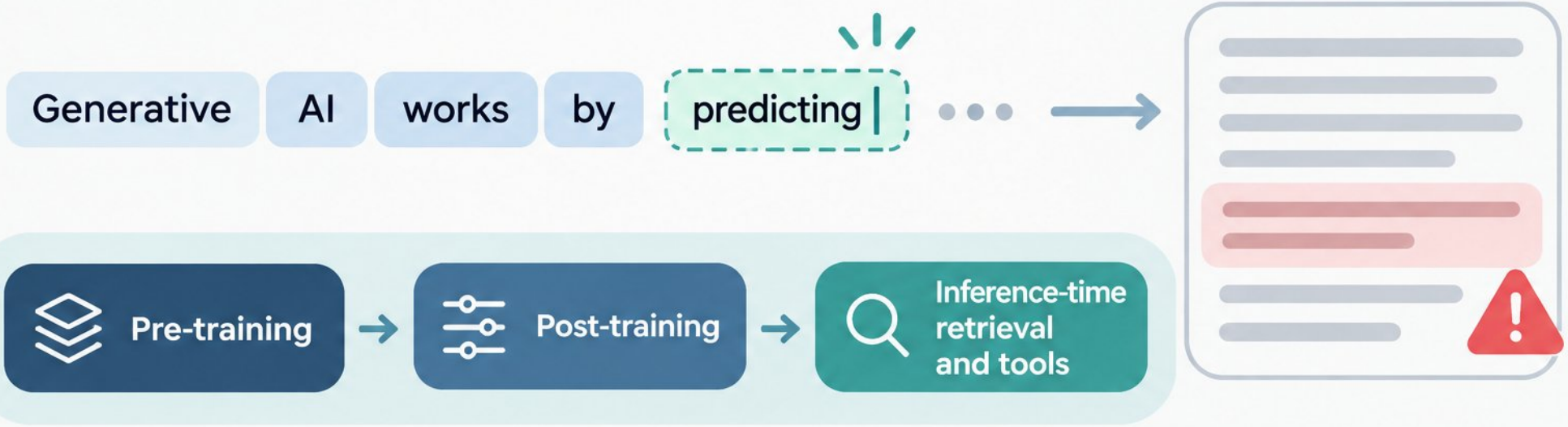




Create / Communicate

Predict / Decide

How Generative AI Works: From Next-Token Prediction to Content



- LLMs predict the next token; at scale, that produces fluent language
- Transformers use attention to weigh which earlier words matter most
- Life cycle: pre-training → post-training → inference-time retrieval and tools
- Probabilistic outputs: same prompt can differ, and plausible content may be wrong
- Multimodal generation extends the same pattern to images, audio, and code

Prompting, RAG, and Fine-Tuning: The Three Levers



- **Prompting:** instructions, context, input data, output format—lowest cost, always start here
- **RAG:** retrieve documents at runtime, paste into prompt; no weight changes; current and citable
- **Fine-tuning:** adjust model weights on labeled examples; teaches style and behavior, not fresh facts
- **Cost/complexity ladder:** prompting < RAG < fine-tuning; most projects never reach the last rung
- **Rule of thumb:** RAG for changing knowledge, fine-tuning for form—many systems combine both

Side-by-Side: Key Differences Between ML and GenAI

	ML (Machine Learning)	GenAI (Generative AI)
Objective	Prediction and decisions	Creation and communication
Input–output	Structured features → label or score	Prompt → generated content
Data	Curated labeled datasets	Massive unlabeled corpora plus prompts and retrieval
Evaluation	Accuracy, precision, recall	Relevance, groundedness, safety, human review
Determinism	Stable outputs	Variable probabilistic outputs—this drives cost and governance

Cost and Operational Tradeoffs

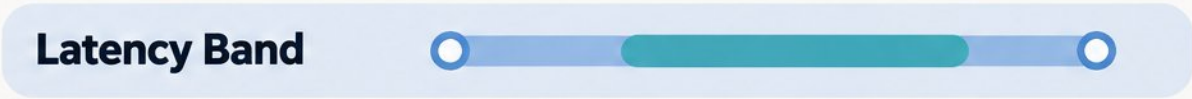
- ML cost sits in data, features, and pipeline upkeep
- GenAI cost sits in tokens, latency, and vendor spend
- Token spend is volatile: 82% of leaders worry, only 64% monitor budgets (EY, 2026)
- Hidden drivers: agent loops, extended thinking, unbounded consumption
- ML is repeatable; GenAI adds variable latency and shifting model quality
- Model choice is often interchangeable—advantage lives in orchestration







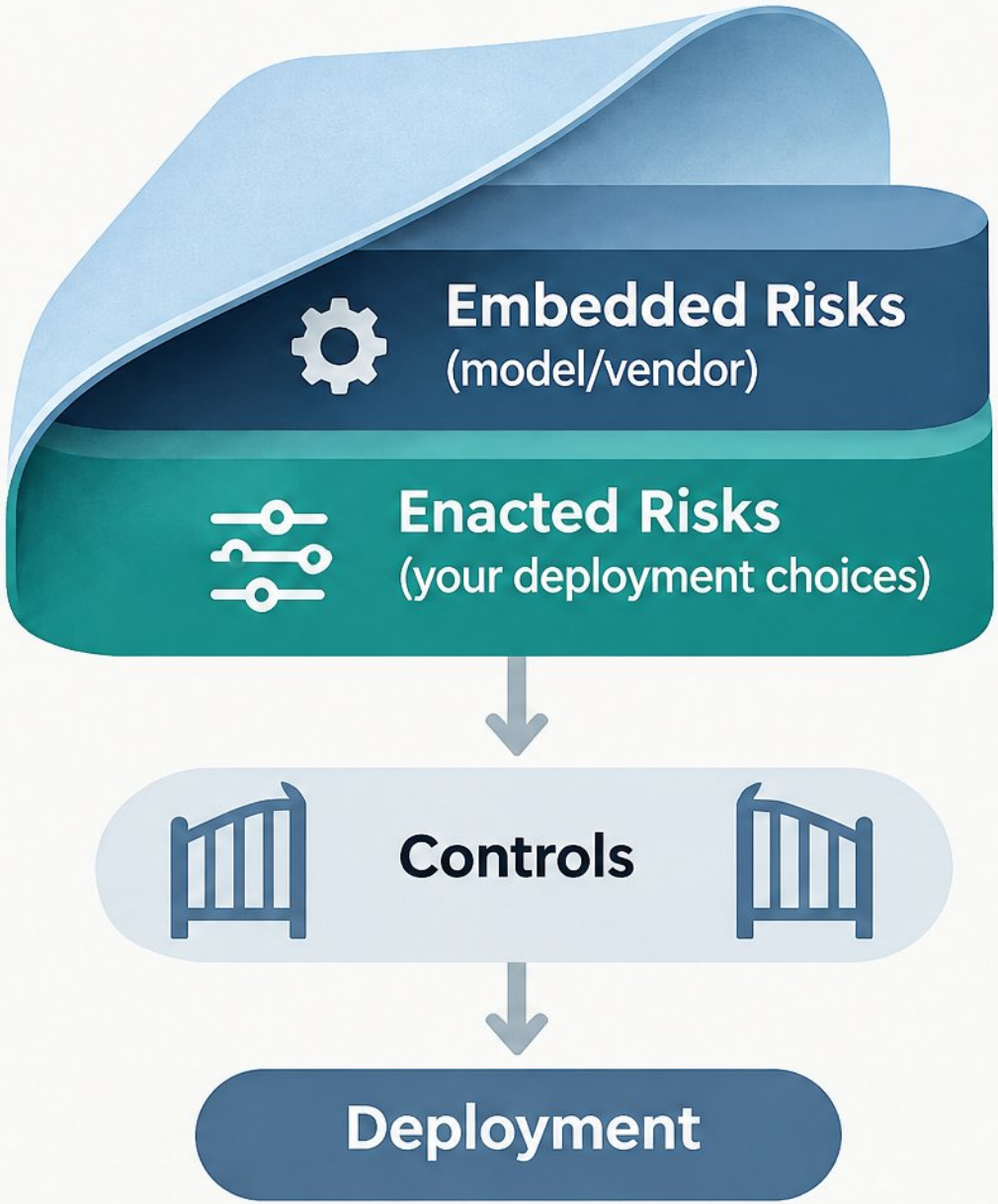






Risk, Safety, and Governance Tradeoffs

- **ML risks:** data drift, biased labels, stale features—mature monitoring helps
- **GenAI risks:** hallucination, prompt injection, data leaks, IP exposure, runaway spend
- Prompt injection is architectural: LLMs can't separate instructions from data
- Separate embedded risks (model/vendor) from enacted risks (your deployment choices)
- **Controls:** RAG grounding, tool limits, human review, spending caps, AI tool inventory



Use Cases: Where Machine Learning Still Wins

- **High-volume, structured, decision-oriented problems:** scoring, forecasting, segmentation, anomaly detection

- **Proven examples:** credit risk, demand planning, predictive maintenance, recommendation ranking, spam filtering

- **Why ML wins:** measurable targets, stable evaluation, auditability, low cost per decision

- **Signal checklist:** historical outcomes? numeric metric? repeatable and explainable answer?

- **Stakeholder framing:** justify with accuracy gains and cost-per-decision, not 'wow' demos

Predictive ML: Specimen Card

- **Credit risk**
- **Demand planning**
- **Predictive maintenance**
- **Ranking**

Repeatable, auditable decision outputs with a cost-per-decision footnote

Predict / Decide → **Create / Communicate**

Use Cases: Where Generative AI Shines—and Where It Fails

- **Content-centric wins:** drafting, summarizing, translating, ideation, chat interfaces, code assistance
- **Enterprise examples:** support copilots, marketing copy, meeting summaries, RAG document Q&A
- **Struggles:** high-stakes factual accuracy, deterministic calculations, strict compliance without guardrails
- **Field failures:** misheard orders in noisy drive-thrus, repeated upselling, trusting polished but wrong output
- **Mitigations:** RAG grounding, tool calling for numbers, human review for regulated content

Create / Communicate

Where It Shines

- Drafting
- Summarizing
- Translating
- Copilot Use Cases

Mitigation: RAG

Predict / Decide

Where It Fails

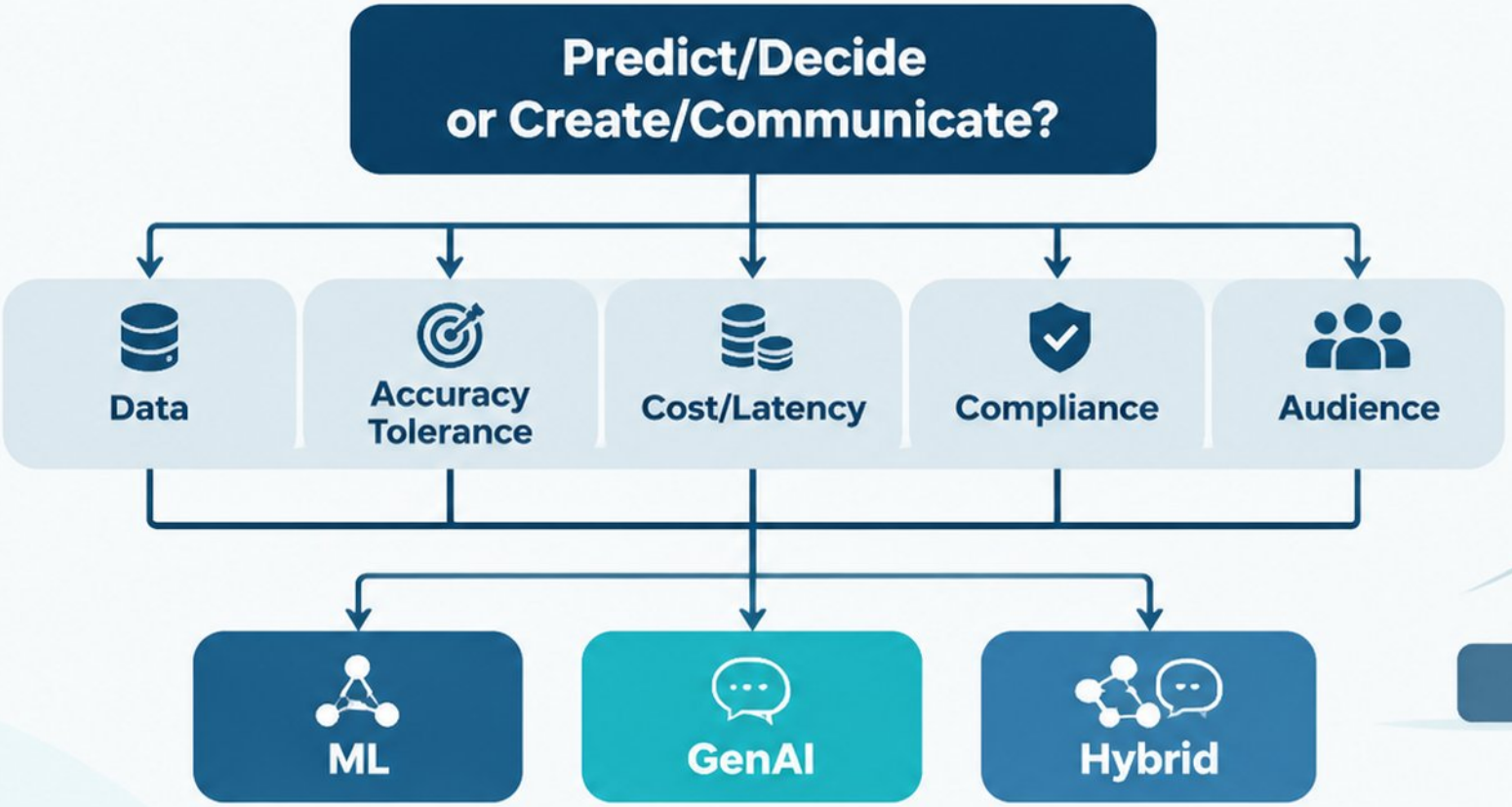
- High-stakes numeric question

Mitigation: Tool Calling

- Noisy-input misunderstanding

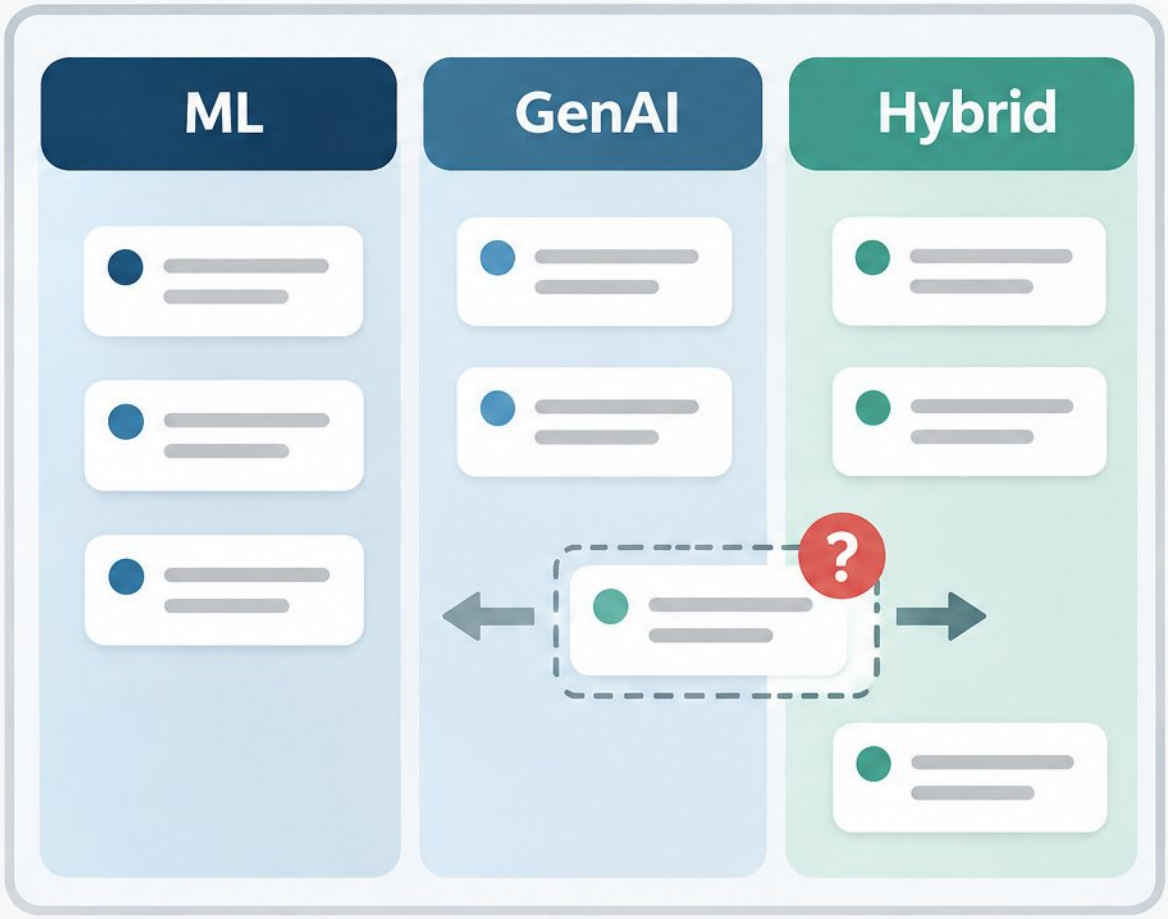
Mitigation: Human Review

A Decision Framework for Choosing an Approach



- Start with the problem type: predict/decide or create/communicate
- Screen five constraints in order: data, accuracy, cost/latency, compliance, audience
- Consider hybrids: ML scoring wrapped in a GenAI natural-language interface
- Apply a lightweight scoring checklist during intake, before any engineering commitment
- Define success metrics and evaluation plans before budget approval

Scenario Clinic: ML, GenAI, or Hybrid?

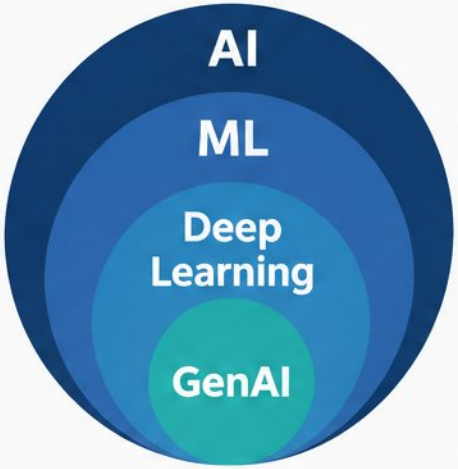


- Group exercise: classify 6–8 business scenarios as ML, GenAI, or hybrid
- Each set includes one deliberately ambiguous case—both answers defensible
- Debrief against the decision framework; surface where the room disagrees
- Reward reasoning about data, risk, and evaluation over labels alone
- Watch common fault lines: customer agents, document processing, numeric output

Key Takeaways and Next Steps

- ML optimizes decisions; GenAI creates content—best solutions combine both
- Budget monitoring, guardrails, and human review from day one
- Start small, measure honestly; expect ~12 months from pilot to production
- Build a use-case inventory and define evaluation criteria per use case
- Take-home: a one-page decision checklist to reuse with your team

Take-Home Decision Checklist



ML optimizes decisions; GenAI creates content—best solutions combine both

Budget for Guardrails

- [ ] Budget monitoring
- [ ] Guardrails
- [ ] Human review

Pilot to Production Timeline

Pilot → Production

Next Steps and Owners

PersonWise.

PersonWise • AI digital-human course platform



Scan the code or click anywhere to watch this course live, with voice Q&A

personwise.ai/t/1f836cfe/public