How to Choose the Right AI Technique

Align with the job to be done, the data you have, and the trust the answer requires.

TECHNIQUE + EXAMPLE	BEST WHEN	REQUIREMENTS	BAD FIT WHEN	TRADEOFF FOR PM
Rules **If this, then that** Example: HR vacation days based on employee tenure and hours worked.	**You know the exact criteria and what should happen.** If an expert can write it down in a spreadsheet, start with rules.	**Clear definition of what is in vs. out.** Few edge cases. No historical or labeled data required.	**The input is messy text, notes, or judgment calls.** Typos, vague text, and rule interactions make the system brittle.	**Simple and auditable, but hard to maintain when rules start stacking.**
Classical ML **Known inputs, unknown formula** Example: predict customer cancellation from usage, plan, tickets, and payment history.	**You know the inputs, but not how to weigh them.** Good for predictions you need to rank, threshold, or explain. If it says 80% risk, it should be right about 80% of the time.	**A table of past examples where the final answer is known.** Example: customer plan, usage, support tickets, and whether they canceled.	**You lack a large, structured dataset with consistent labels.** Weak for messy notes, images, or brand-new situations with no useful history.	**Great for prediction, but it can quietly get worse when behavior changes.** You need monitoring, retraining, and clear ownership.
Deep Learning **When you don't know what to measure** Example: detect car damage from photos. The useful clues are hard to define: dent shape, shadows, scratches, angle, lighting, and context.	**Best when a human cannot list every feature the system should look for.** The model learns features from examples. Best for vision, speech, handwriting, and other messy signals.	**Lots of examples and a clear task.** Example: thousands of photos labeled with damage type or severity.	**Your data is small or already fits neatly in a table.** Expensive to build/train and needs specialized ML skill to operate.	**Can find patterns people cannot describe, but is harder to trace and explain.** Harder for workflows that need a clear "why" behind every answer.
GenAI / LLMs **Read, reason, and draft** Example: summarize customer emails, identify the real issue, and draft a support reply.	**The input is messy human language and the output needs judgment.** Good for summarizing, extracting, explaining, rewriting, or drafting.	**Unstructured text and a strong eval set.** You may not need a training set, but you still need test cases.	**You need calibrated probabilities or exact consistent decision.** Hallucination, latency, cost, and governance become first-order risks.	**Fastest path through messy language, but risky as the final decision layer.**

How they work together: Customer Support

A customer wants to return a damaged product and get a replacement.

Customer submits RMA

LLM reads request
Summarize the issue, product, order, and urgency.
Why: language varies too much for rules.

Customer uploads proof

Deep learning checks photos
Inspect damage photos, labels, packaging, or serial numbers.
Why: raw media is hard to turn into spreadsheet columns.

Support checks history

ML scores risk
Predict fraud, escalation, repeat return, or priority risk.
Why: rules can't weigh all factors or know which past signals predict behavior best.

Support decides outcome

Rules apply policy
Approve RMA, request more proof, refund, replace, or escalate.
Why: decisions must be consistent.

Customer gets next steps

LLM drafts reply
Explain the decision, return label, replacement timing, and next steps.
Why: the reply needs to sound clear and human.

Rishabh Singh 2026