

Machine Learning Interview Questions: Overview and Preparation Map



Prep for ML practitioners, early-career data scientists, and career switchers



Interviewers test fundamentals, coding, problem framing, experimentation, and applied judgment



Typical loop: recruiter screen, technical screen, coding, ML concepts, case study, behavioral



2026 signals: conceptual questions dominate, SQL ~26%, LLM questions now major

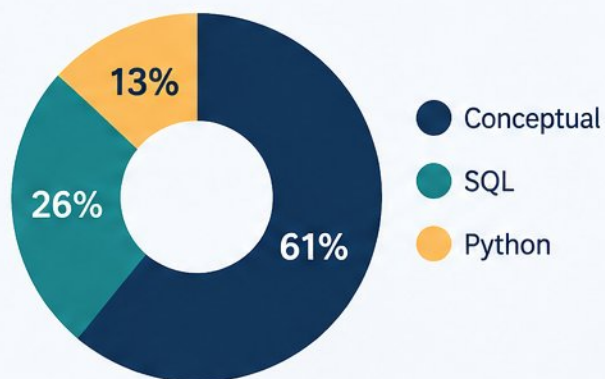


Practice by question category; build frameworks and connect answers to real examples

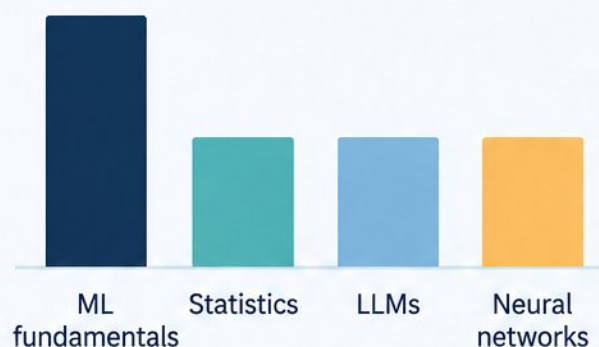


What the 2026 Interview Data **Actually** Shows

- 389 questions across 49 companies: 61% conceptual, 26% SQL, 13% Python



- ML fundamentals lead; statistics, LLMs, and neural networks tied for second

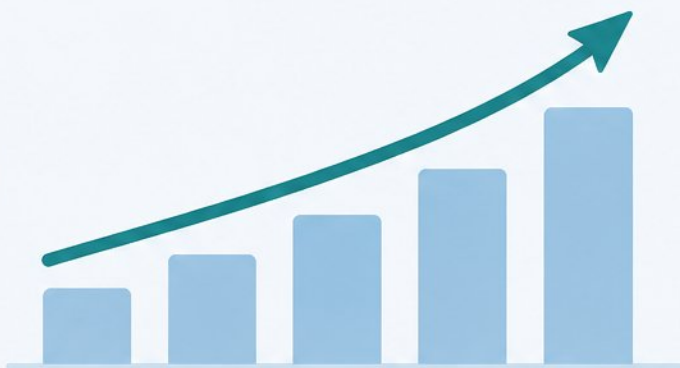


- SQL is still a gatekeeper: joins are 30% of SQL questions



Joins are **30%** of SQL questions

- LLM questions grew fast, concentrated at Google, OpenAI, Anthropic, DeepMind, Meta



- Company mix matters: Amazon leans SQL, AI labs show zero SQL



Role Differences: Data Scientist, ML Engineer, Applied Scientist, and Analytics



Data Scientist



ML Engineer



Applied Scientist



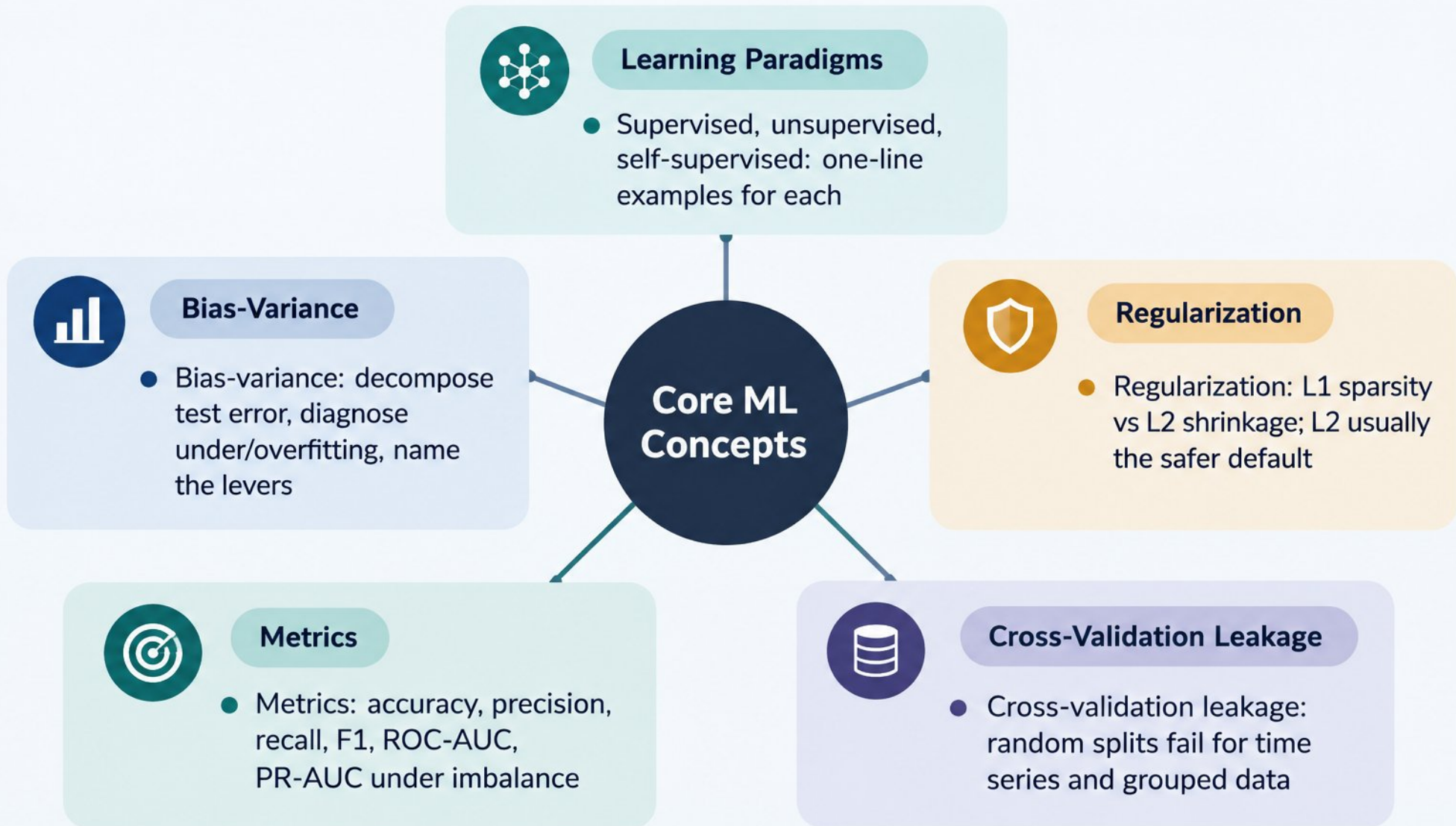
Analytics

- Product DS loops: SQL, statistics, experimentation, metrics, communication
- ML engineer loops: coding quality, ML fundamentals, serving, monitoring, reliability
- Applied scientist loops: research-to-production judgment plus modeling depth
- Same churn prompt, different evidence expected from each role
- Ask your recruiter for round types before allocating prep time



Core ML Concepts and How Interviewers Probe Them

Key ideas, what to know, and how they connect



Modeling Questions and Answer Frameworks



Start with a framework

- Start every "build a model for X" prompt with a framework: goal, metric, data, baseline, iterate.



Baseline first

- Baseline first: dummy classifier, logistic regression, tree ensembles — complexity only when justified.



Explain algorithms

- Explain algorithms from memory: linear and logistic regression, trees, random forests, gradient boosting, k-means, PCA.



Name tradeoffs deliberately

- Name tradeoffs deliberately: interpretability vs. accuracy, training vs. inference cost, latency, memory, calibration.



Churn case study

- Churn case study: prediction time, label horizon, features, validation, and the action the score drives.

Coding Rounds: Python, NumPy, pandas, and SQL



Four patterns: implement a metric, transform data, write numerical routines, fix buggy pipelines



Python: pandas groupby, merges, pivot, date handling, NumPy broadcasting, apply vs vectorize



From scratch: k-NN, one gradient descent step, precision and recall, softmax, k-fold CV



SQL: joins, GROUP BY with HAVING, window functions, CTEs, date functions, retention queries



Practice loop: timed problems, narrate tradeoffs aloud, test edge cases, review failure modes



Python / pandas

```
import pandas as pd
import numpy as np
# group, merge, pivot, date handling
df = pd.read_csv("data.csv")
result = df.groupby("key").agg({"value": "mean"})
```



NumPy

```
import numpy as np
# broadcasting, apply vs vectorize
a = np.array([1, 2, 3])
b = np.array([10, 20, 30])
c = a + b # broadcasting
```



SQL

```
WITH recent_users AS (
    SELECT user_id, signup_date
    FROM users
)
SELECT u.user_id, COUNT(*)
FROM recent_users u
GROUP BY u.user_id
HAVING COUNT(*) > 0
```

SQL Deep Dive:

Joins, Window Functions, and Metric Questions



Joins dominate:

30 of 100 SQL questions, counting 30% overall



Window functions next at 15:

ROW_NUMBER, RANK, LAG, LEAD, running totals



Subqueries and CTEs follow

at 12 and 9 respectively



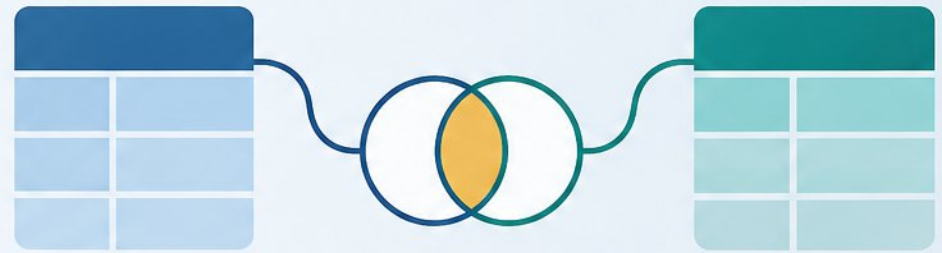
Most prompts are metric definition:

fix unit, window, and denominator first



Defensive querying:

check row counts, COALESCE nulls, dedupe duplicate events



```
SELECT
  ROW_NUMBER() OVER ( ... ),
  RANK() OVER ( .. ),
  LAG( ... ),
  LEAD( ... ),
  -- running totals
```

```
WITH cte AS (
  SELECT ...
)
SELECT ...
FROM cte;
```

Data Quality, Feature Engineering, and Experimental Design



Data quality prompts

- Data quality prompts: missing values, outliers, duplicates, leakage, label noise, train-serve skew



Feature choices

- Feature choices: scaling, encoding, interactions, point-in-time correctness, domain features



Validation splits

- Validation splits: stratification, group splits, time-based splits, nested cross-validation



A/B testing

- A/B testing: hypothesis, power, guardrails, CUPED, sequential peeking, multiple comparisons



Offline up but online down

- Offline up but online down: check drift, leakage, metric mismatch, feedback loops

Deep Learning and Transformer Questions

1

Neural network basics: activations, loss functions, backpropagation, and optimizers

2

Why Adam sometimes generalizes differently from SGD

3

Overfitting controls: dropout, batch norm, early stopping, weight decay

4

Architecture choices: CNNs for images, transformers for most sequences

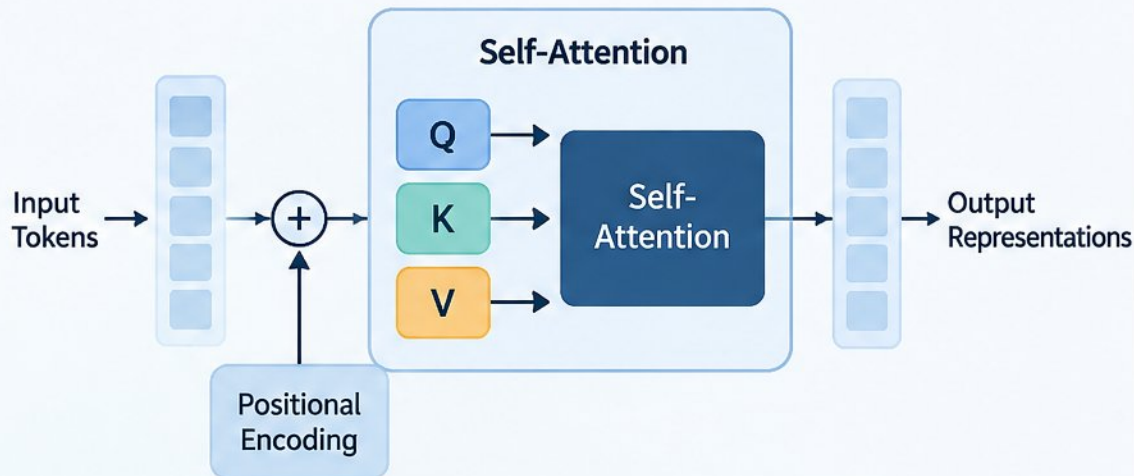
5

Transformer intuition: Q/K/V, self-attention, RoPE, $O(n^2)$ cost

6

Inference awareness: KV cache, FlashAttention, quantization, speculative decoding

Transformer Block (Cutaway View)



Inference Optimizations



KV cache



FlashAttention



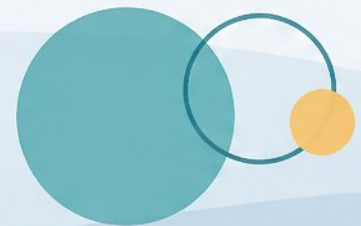
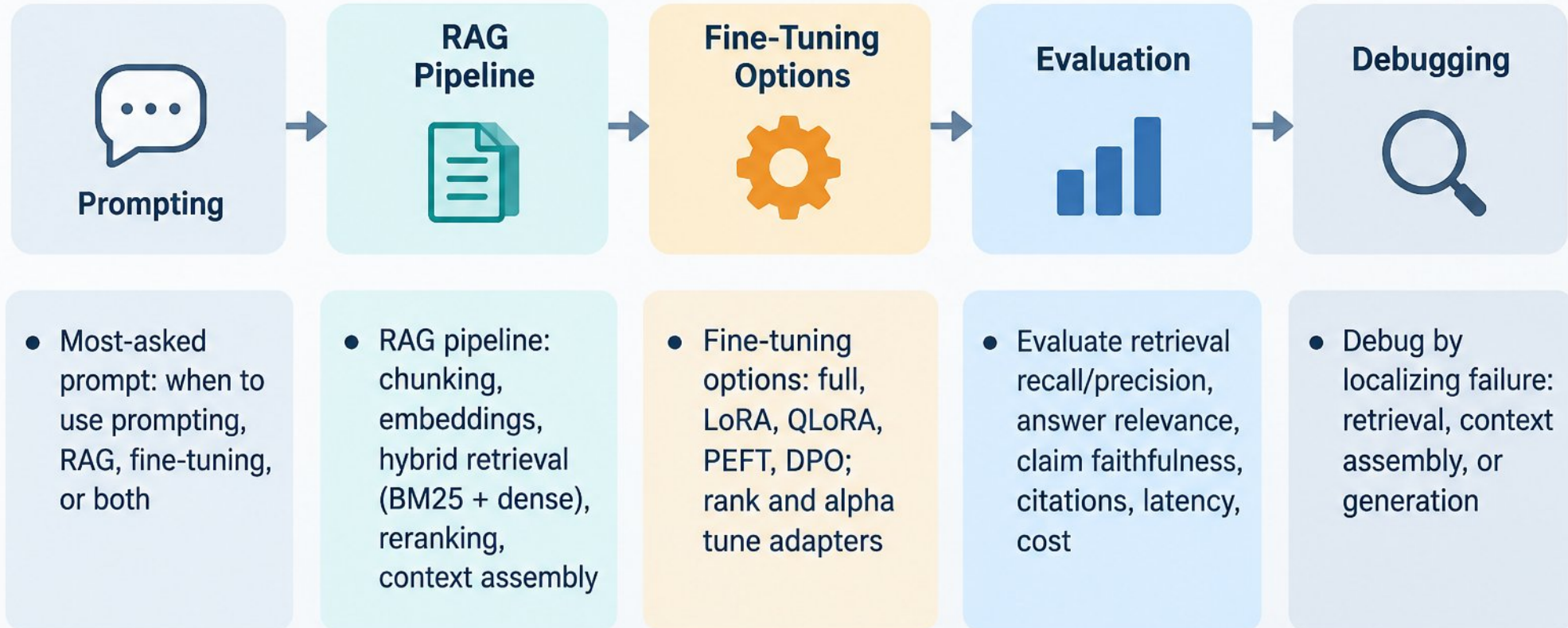
quantization



speculative decoding

LLM, RAG, and Fine-Tuning Interview Questions

Key concepts. Typical workflows. What interviewers ask.



Applied AI System Design and Evaluation



Framework: clarify requirements and latency budget, then sketch the pipeline



Deep dive on hardest components, then address failure modes and evaluation



Common prompts: recommendation, fraud detection, search ranking, ad CTR, RAG assistant



Production concerns: offline vs online eval, A/B testing, training-serving skew

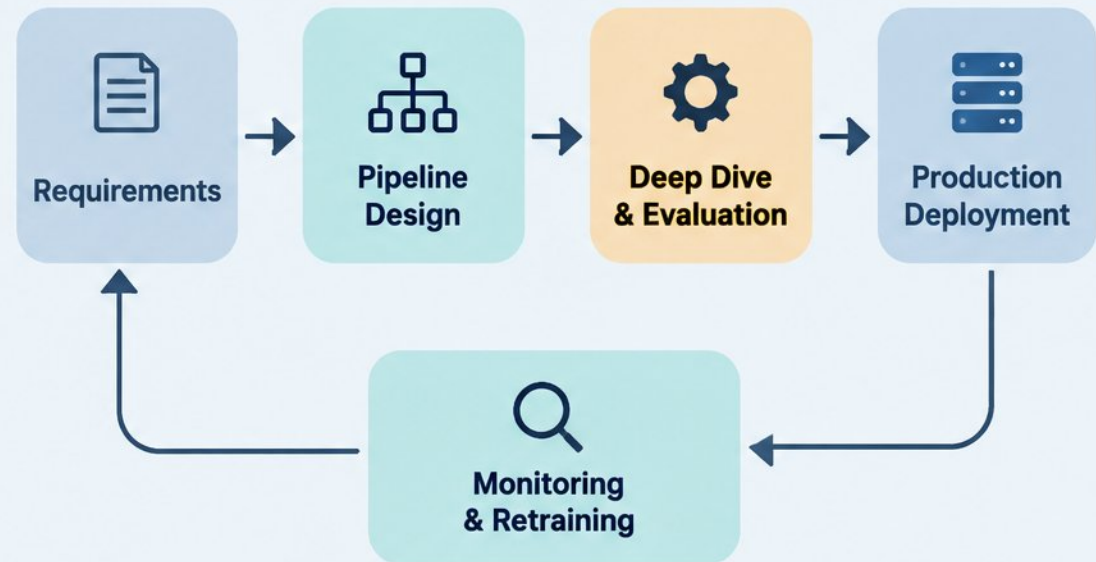


Monitoring: prediction drift, feature shift, calibration, KPI tracking, retraining triggers



Senior signal: name tradeoffs across quality, latency, cost, consistency, reliability

From Design to Evaluation to Reliable Impact



Explainability, Ethics, and Responsible AI

PEOPLE • PRINCIPLES • SAFER AI



Interpretability

- Interpretability tools: feature importance, SHAP, partial dependence, and local explanations



Fairness

- Fairness: protected groups, disparate impact, proxies, and the accuracy-fairness tradeoff



Privacy

- Privacy: PII handling, consent, data minimization, retention limits, regulatory awareness



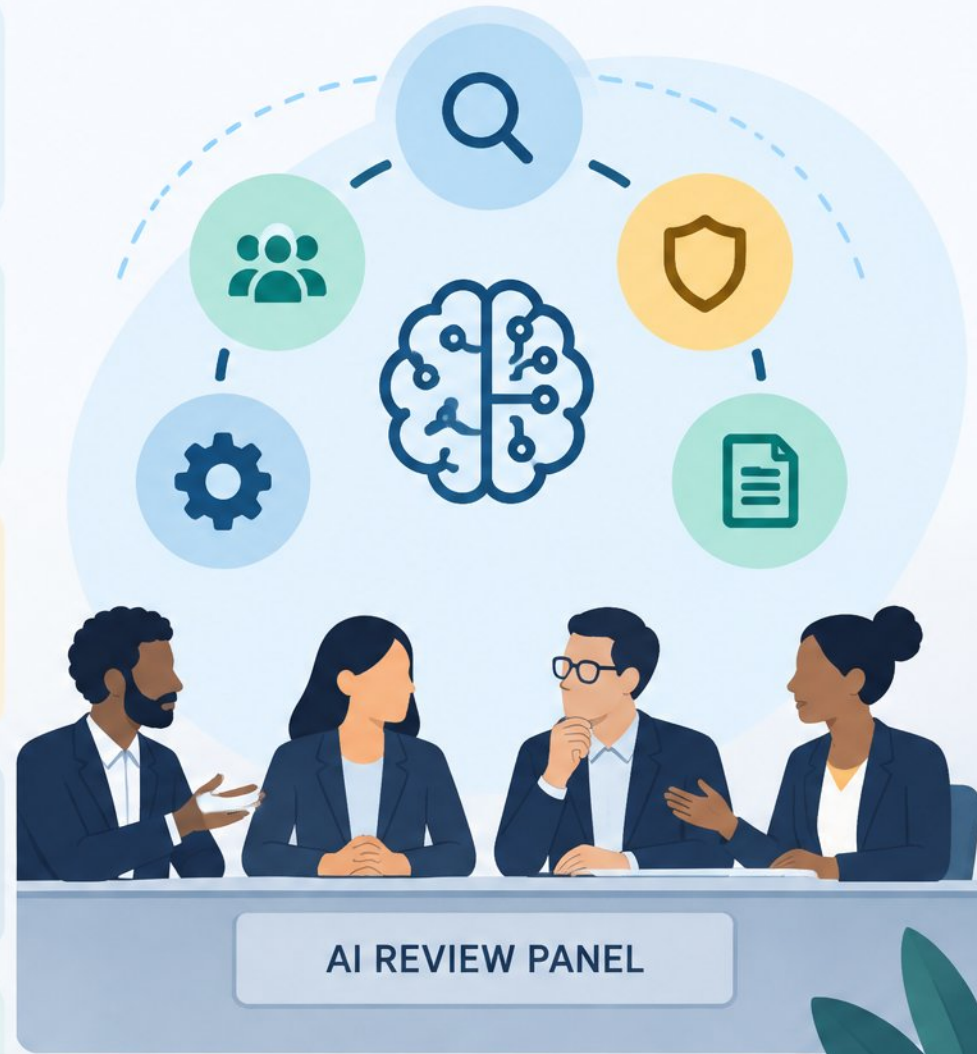
Model risk

- Model risk: drift monitoring, retraining triggers, human review, and fallback behavior



Ethics answer pattern

- Ethics answer pattern: name the risk, propose mitigation, define a measurable check



Behavioral, Project Deep-Dive, and Communication Questions

Structure your stories. Show your thinking. Communicate with impact.

STAR-L for Project Stories



Project sequence: problem, data, baseline, approach, evaluation, failure, impact



Distinguish personal ownership from team contribution; quantify only measured outcomes



Common prompts: difficult stakeholder, tight deadline, failed model, production incident



Non-technical audiences: skip jargon, lead with impact, state uncertainty honestly



Four-Week Practice Plan, Mock Interviews, and Final Preparation



Weeks 1-2

drill SQL joins, window functions, pandas, and NumPy basics



Weeks 3-4

practice ML cases, system design, LLM/RAG concepts, then mock interviews



Mocks

10-min concepts, 25-min coding/case, 10-min behavioral, then self-review



Final week

redo missed problems, rehearse stories aloud, prep interviewer questions



Build a question bank by concept, difficulty, and answer framework



Track recurring weak areas instead of re-reading everything



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/f7d01aec/public