Deep research on all AI/LLM questions asked by famous companies, most frequently asked.

Below is a curated bank of high-signal AI/ML/LLM interview questions that recur across big tech and frontier labs (Google, DeepMind, Meta, Microsoft, Amazon, OpenAI, Anthropic, etc.), based on public guides and candidate reports.[1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13] [14] [15]

They are grouped by theme, with brief prep notes. This is designed for ML / LLM engineer, applied scientist, and AI-focused SWE roles.

A. Core ML fundamentals (15–18 questions)

1. **Explain the difference between supervised, unsupervised, and semi-supervised learning.**
   Focus on objectives, label availability, and typical algorithms.[2] [13]
2. **Bias–variance trade-off: what is it and how do you manage it in practice?**
   Connect to model complexity, regularization, data size, and under/overfitting.[14] [2]
3. **Compare linear regression, logistic regression, decision trees, random forests, and SVMs. When would you use each?**
   Talk assumptions, interpretability, robustness to outliers, non-linearity.[13] [2]
4. **What is regularization? Explain L1 vs L2 and when you'd prefer one over the other.**
   Include effect on weights, sparsity, optimization behavior.[2]
5. **How would you handle a dataset where the positive class is <1% of the data?**
   Discuss rebalancing (SMOTE, class weights), anomaly framing, evaluation with PR curves.[2]
6. **Differentiate precision, recall, F1, ROC–AUC, PR–AUC. Which would you use for highly imbalanced data and why?**
   Be very crisp on trade-offs and business context.[16] [2]
7. **What metrics would you use to evaluate a recommendation system / ranking model?**
   Hit rate, NDCG, MAP, coverage, diversity, business KPIs.[3] [17] [5]
8. **Explain k-means. What are its limitations and how do you choose k?**
   Cover initialization, distance assumptions, non-convex objective, elbow/silhouette.[3]
9. **Naive Bayes vs other classifiers: when would you pick it?**
   High-dimensional sparse text, independence assumption, fast baselines.[16]
10. **How do you handle missing data and noisy features in an ML pipeline?**
    Imputation strategies, robust models, feature filtering, leakage risks.[18] [16]

11. **Explain cross-validation. When is k-fold preferable to a simple train/validation split?**
    High variance / limited data scenarios, time-series caveats.
12. **What is overfitting? List concrete techniques to prevent it.**
    Regularization, early stopping, data augmentation, ensembling, simpler models.[8] [2]
13. **How do you choose between a simpler model (e.g., logistic regression) and a complex one (e.g., deep net) in production?**
    Trade-off accuracy vs interpretability, latency, data volume, infra cost.[4] [14]
14. **What is a generative model? Contrast it with discriminative models and give examples.**
    E.g., VAEs, GANs, autoregressive LMs vs logistic regression, SVM.[13]
15. **Explain feature engineering. Give an example where it mattered more than model choice.**
    Connect to domain understanding and downstream metrics.[13]
16. **When would you use anomaly detection instead of supervised classification?**
    Rare events, label scarcity, evolving patterns (fraud, abuse).[5] [2]

B. Deep learning & optimization (10–12 questions)

17. **Explain what a neural network is at a high level.**
    Architecture, non-linearities, learning via gradient descent.[13]
18. **Describe backpropagation and how gradients are computed efficiently.**
    Chain rule, computational graph, shared sub-expressions.
19. **What is the difference between SGD, mini-batch SGD, and full-batch gradient descent?**
    Convergence behavior, noise, compute/memory trade-offs.
20. **Common causes of exploding/vanishing gradients and how to mitigate them.**
    Initialization, normalization, residual connections, gradient clipping.
21. **BatchNorm vs LayerNorm: what problem do they solve and when do you use each?**
    Training stability in CNNs vs sequence/transformer settings.
22. **Explain dropout and weight decay. When would you use each?**
    Different regularization mechanisms, effect on test performance.[8]
23. **How do you decide when to stop training a model?**
    Early stopping with validation curves, patience heuristics, overfitting signals.[19] [8]
24. **Describe gradient boosting (e.g., XGBoost, GBDT). Why might it outperform neural nets on some tabular tasks?**
    Bias/variance properties, handling heterogeneous features.[13]
25. **How do you optimize a model for inference on resource-constrained devices (mobile/edge)?**
    Quantization, pruning, distillation, smaller architectures.[2]
26. **You see training loss going down but validation loss diverging. How do you debug?**
    Regularization, data leakage, distribution shift, early stopping, augmentation.

C. NLP & LLM-specific questions (15–18 questions)

These have become common at Google, Amazon, Microsoft, Meta, OpenAI, Anthropic, and DeepMind for ML/AI roles.[6] [9] [10] [11] [12] [15] [5] [3] [8] [2] [13]

27. **Explain the architecture of a transformer.**
    Multi-head self-attention, positional encodings, feed-forward blocks, residuals, layer norm.
28. **Self-attention: what problem does it solve compared to RNNs/CNNs for sequence modeling?**
    Parallelization, long-range dependencies, path length.
29. **Contrast autoregressive (causal) language modeling with masked language modeling.**
    Training objective, typical uses (GPT-style vs BERT-style).
30. **What is "in-context learning" in LLMs and why is it surprising from a classical ML perspective?**
    No parameter update, emergent capabilities from scale.
31. **Explain tokenization and subword methods (BPE, WordPiece, SentencePiece). Why not character or word only?**
    Trade-offs in vocabulary size, OOV handling, efficiency.
32. **How would you adapt or fine-tune a pre-trained LLM for a specific downstream task with limited labeled data?**
    LoRA/PEFT, instruction tuning, RAG as an alternative, domain adaptation.
33. **Describe Retrieval-Augmented Generation (RAG). When is RAG preferable to pure fine-tuning?**
    Freshness, data governance, cost; designing retriever, index, reader.[6]
34. **How would you design and optimize a RAG system for internal documents (end-to-end)?**
    Data ingestion, chunking, embedding selection, vector store, retrieval strategy, caching, evaluation.[6]
35. **What are "hallucinations" in LLMs? How would you measure and reduce them?**
    Definition, grounded generation, RAG, constrained decoding, post-verification.[2]
36. **Explain RLHF (Reinforcement Learning from Human Feedback) at a high level.**
    Supervised SFT, reward model, PPO/other RL, pros/cons vs pure SFT.
37. **How would you reduce toxicity or biased outputs in an LLM-powered product?**
    Dataset curation, filtering, alignment techniques, safety classifiers, prompt-time and post-hoc filters.[7] [20] [12]
38. **Compare classical NLP models (e.g., TF-IDF + logistic regression) with transformer-based LLMs for a classification task.**
    Performance, data requirements, latency, interpretability.[5] [13]
39. **You must serve LLM responses under a strict latency SLO. How do you optimize the system?**
    Smaller models, quantization, caching, early-exit layers, request batching, speculative decoding.[2]

40. **How would you evaluate an LLM-based question-answering system?**
    Automatic metrics (EM/F1, BLEU, ROUGE, BERTScore), human eval, task-specific business metrics.[5] [2]

41. **Design guardrails / safety filters for an LLM API.**
    Input classification, output classification, rate limiting, red-teaming, escalation paths.[20] [12] [7]

42. **Explain embeddings. How would you build a semantic search engine using them?**
    Embedding model choice, vector index (ANN), similarity metrics, re-ranking.

D. ML & LLM system design (15–18 questions)

System design for ML is a core component at Google, Meta, Amazon, Microsoft, OpenAI, Anthropic, etc.[17] [21] [9] [11] [22] [1] [4] [18] [14] [3] [16] [8] [5] [6] [2]

43. **Design a personalized news feed / content ranking system (e.g., Facebook/Instagram/YouTube).**
    Cover signals, feature engineering, candidate generation + ranking, feedback loops, experimentation.[21] [17] [18] [8]

44. **Design a product recommendation system for a large e-commerce site (Amazon-style).**
    Candidate generation, personalization, cold-start, ranking, latency/scale.[3] [5]

45. **Design an autocomplete / type-ahead search system. How would you make it ML-driven?**
    Query logs, trie/baseline, embeddings, ranking, caching, abuse/quality controls.[16] [8] [3]

46. **Design an ML system to detect policy-violating or offensive content (ads, posts, multimedia).**
    Problem framing, multi-modal signals, thresholds, human review flows, appeals.[8]

47. **Design a fraud or abuse detection system for a marketplace or payments platform.**
    Features, real-time vs batch, feedback loops, adversarial behavior.[4] [21]

48. **Architect an end-to-end ML pipeline for a ranking or recommendation use case.**
    Data ingestion, feature store, offline training, online inference, monitoring, retraining strategy.[4] [3] [5]

49. **What architecture would you use to serve low-latency ML predictions at high QPS?**
    Microservices, autoscaling, caching, model versioning, canary/shadow deploys.[4] [5]

50. **How would you design a feature store that supports both offline training and online inference?**
    Consistency, latency, backfills, schema/versioning.[4]

51. **Design a large-scale recommendation system for short videos or news, end-to-end.**
    Often asked at OpenAI / Meta / big tech; emphasize retrieval–ranking–re-ranking loops.[6]

52. **Design an LLM-powered assistant for internal developer productivity (code assistant, doc Q&A).**
    Requirements, data security, RAG, privacy, logging, evaluation.

53. **How would you handle data drift and model degradation in a production ML system?**
    Drift detection, automatic retraining, guardrails, model rollback.[5]

54. **Describe how you would structure experimentation and A/B testing for a new ML feature.**
    Unit of randomization, metrics, guardrails, sample size, interpreting results.[5] [4] [2]
55. **How would you design APIs for secure and efficient interaction with AI models?**
    Authentication, quotas, request validation, streaming, safety layers, observability.[7]
56. **Design a system to ensure safe deployment of AI models in production.**
    Staging, shadow mode, red-teaming, safety tests, rollback paths, audit logs.[20] [7]
57. **Build an ML system that detects if a marketplace listing is about selling a weapon.**
    Labeling strategy, text representation/LLM, thresholds, legal/ethical constraints.[4]

E. Data, evaluation, and experimentation (8–10 questions)

58. **Given six million search queries, how would you pick a representative sample to label or analyze?**
    Stratified sampling, time-based, long-tail vs head, bias considerations.[16]
59. **How do you design an experiment to test the impact of a new ranking model on user engagement?**
    A/B test setup, primary/secondary metrics, guardrail metrics, duration.[5] [4]
60. **Explain the ROC curve, sensitivity, specificity, and the confusion matrix.**
    When ROC is misleading vs PR curve, threshold selection.[16]
61. **How would you evaluate a computer vision model beyond accuracy?**
    Per-class metrics, IoU, calibration, latency, robustness.[3] [2]
62. **How do you test and validate reliability of an AI system under stress conditions?**
    Load testing, adversarial inputs, failover behavior, graceful degradation.[7]
63. **You deploy a new model and business KPIs improve but offline metrics worsen (or vice versa). How do you debug this?**
    Metric mismatch, leakage, experiment bugs, user behavior shifts.
64. **How would you structure an experiment to test a value proposition across different customer segments?**
    Segmentation, interaction effects, multiple-testing and interpretation.[4]

F. Safety, alignment, and ethics (Anthropic / OpenAI-style) (6–8 questions)

These are especially common at Anthropic and OpenAI, but now bleed into other labs as well.[11] [12] [23] [15] [20] [7] [6]

65. **How would you identify and mitigate potential risks when deploying an LLM-based system?**
    Enumerate failure modes, red-teaming, monitoring, human-in-the-loop.
66. **How do you balance performance optimization with model interpretability and safety?**
    Trade-offs between black-box models vs simpler or post-hoc explanation.

67. **How would you implement guardrails to ensure ethical AI usage in a real-world application?**
    Policy definition, enforcement mechanisms, user education, escalation paths.[7]
68. **Describe how you'd assess and reduce bias in an ML or LLM system.**
    Dataset audits, fairness metrics, debiasing techniques, stakeholder involvement.
69. **Anthropic's view: "Capability without alignment is a liability." How does that influence design decisions in ML systems?**
    Show that you think about emergent behavior, long-term impacts, alignment constraints.[12] [15] [20]
70. **How do you collaborate with cross-functional teams (policy, legal, product) to align ML solutions with ethical principles?**
    Concrete examples of trade-off decisions, communication, and documentation.[7]

G. Behavioral & project deep-dive (AI-focused) (10–12 questions)

Nearly every company combines ML depth with strong behavioral probing.[24] [9] [10] [25] [1] [17] [18] [21] [19] [8] [3] [5] [4]

71. **Walk me through an ML/LLM project you led end-to-end. What was the impact?**
    Expect deep follow-ups on data, modeling, infra, metrics, trade-offs.
72. **Tell me about a time you worked with messy or incomplete data. How did you ensure model reliability?**
    Focus on robustness, validation, stakeholder communication.[18]
73. **Describe a time when a model you deployed failed or caused issues in production. What did you do?**
    Amazon/Microsoft love this; show ownership and post-mortem thinking.[26] [24] [5]
74. **Give an example of a production incident that happened after you left the team (or when you were off-call). How did you respond?**
    Classic Amazon "ownership beyond your box" question.[24]
75. **Tell me about a challenging trade-off between model performance and system constraints (latency/cost/interpretability).**
    Show pragmatic engineering judgment.
76. **How do you stay up-to-date with the latest in AI/ML/LLMs?**
    OpenAI and DeepMind explicitly ask variants of this.[9] [10] [19] [8]
77. **Describe a time you pushed back on a product or leadership request because of ML or safety concerns.**
    Emphasize principled reasoning and communication.
78. **Why do you want to work on AI/LLMs at <company>?**
    Tailor to each org's mission (Google scale and infra, OpenAI/Anthropic safety and frontier models, Meta recommender systems, etc.).[23] [17] [11] [8]
79. **Tell me about a time you had to debug a complex ML system (data + model + infra).**
    Great place to talk about distributed training, logging, and observability.[9] [6]

80. **What is the greatest accomplishment of your career related to AI/ML, and why?**
    Meta, Amazon, Google all ask variants of this.[17] [21] [18]

How to use this list effectively

- **Map to companies:**
  - Google / DeepMind: heavy on ML theory + system design (ranking, autocomplete, content moderation, LLM infra).[10] [14] [8] [13]
  - Meta: news feed / recommendation / ads evaluation, ML system design, product impact.[1] [21] [17] [18]
  - Amazon: ML system design with strong behavioral & ownership focus (latency, scale, incidents).[27] [26] [24] [3] [5]
  - Microsoft: ML + Azure-centric system design, experimentation, applied scientist flavor.[28] [16] [2] [4]
  - OpenAI / Anthropic: deep ML + safety/alignment, LLM/RAG design, debugging complex training/inference systems.[15] [11] [12] [23] [9] [20] [6] [7]
- **Practice style:** for each question, prepare:
  - A 30–60s high-level answer.
  - A 5–10 minute deep-dive with concrete examples and, where relevant, equations and diagrams.
  - 1–2 war stories from your own work that demonstrate the concept under real constraints.

If you share your target companies and role level (e.g., "L5 ML engineer at Google vs OpenAI applied scientist"), a tailored subset of ~30 priority questions plus a focused prep plan can be outlined next.

⁂

1. https://prepfully.com/interview-guides/meta-machine-learning-engineer
2. https://www.test-king.com/blog/80-essential-interview-questions-for-microsoft-machine-learning-ai-engineer-roles/
3. https://igotanoffer.com/blogs/tech/amazon-machine-learning-engineer-interview
4. https://www.interviewquery.com/interview-guides/microsoft-machine-learning-engineer
5. https://www.interviewquery.com/interview-guides/amazon-machine-learning-engineer
6. https://prachub.com/companies/openai/positions/machine-learning-engineer
7. https://www.amainterview.ai/interview-questions/machine-learning-engineer-at-anthropic
8. https://igotanoffer.com/blogs/tech/google-machine-learning-engineer-interview
9. https://www.interviewquery.com/interview-guides/openai-machine-learning-engineer
10. https://www.educative.io/blog/google-deepmind-interview-questions
11. https://prachub.com/companies/openai/categories/machine-learning

12. http://www.interviewnode.com/post/ace-your-anthropic-ml-interview-top-25-questions-and-expert-answers
13. https://www.vervecopilot.com/interview-questions/top-30-most-common-google-ai-ml-interview-questions-you-should-prepare-for
14. https://www.interviewquery.com/interview-guides/google-machine-learning-interview-questions
15. https://www.linkedin.com/posts/santoshrout_ace-your-anthropic-ml-interview-top-25-questions-activity-7302011802710654976-Yn6R
16. https://www.topbots.com/microsoft-data-science-interview-questions/
17. https://igotanoffer.com/blogs/tech/facebook-machine-learning-engineer-interview
18. https://www.interviewquery.com/interview-guides/facebook-machine-learning-interview-questions
19. https://www.coursera.org/articles/openai-interview-questions
20. https://www.interviewnode.com/post/ace-your-anthropic-ml-interview-top-25-questions-and-expert-answers-2026-version
21. https://www.datainterview.com/blog/meta-machine-learning-engineer-interview
22. https://aimcqs.com/google-ml-engineer-interview-guide
23. https://www.vervecopilot.com/interview-questions/top-30-most-common-anthropic-interview-questions-you-should-prepare-for
24. https://www.linkedin.com/pulse/amazon-machine-learning-engineer-interview-guide-questions-foster-m2fef
25. https://igotanoffer.com/en/advice/openai-interview-questions
26. https://prepfully.com/interview-guides/amazon-machine-learning-engineer
27. https://interviewkickstart.com/blogs/interview-questions/amazon-machine-learning-interview-questions
28. https://datalemur.com/blog/microsoft-data-science-interview
29. https://www.hellointerview.com/blog/meta-ai-enabled-coding
30. https://www.reddit.com/r/developersIndia/comments/1ot60wt/insane_interview_with_microsoft_applied_scientist/