## **📈 Model Performance & Evaluation**

### **1. Why would a model’s performance degrade over time?**

* **Answer**: Reasons include **concept drift**, **data drift**, **pipeline failures**, **labeling issues**, or a **retraining gap**.

### **2. How do you handle model overfitting?**

* **Answer**: Use regularization (L1, L2, dropout), increase training data, perform cross-validation, or simplify the model.

### **3. What is precision vs recall, and when to prefer one?**

* **Answer**: **Precision** is the proportion of true positives among predicted positives; prefer it when false positives are costly. **Recall** is the proportion of true positives among all actual positives; prefer it when false negatives are costly.

### **4. How to evaluate a model on imbalanced data?**

* **Answer**: Use metrics like **F1 Score**, **PR AUC**, or **Matthews Correlation Coefficient**. Techniques like **resampling** or **cost-sensitive learning** can also be helpful.

### **5. What is calibration in classification models?**

* **Answer**: Calibration ensures predicted probabilities align with actual outcomes. Methods include **Platt scaling** and **isotonic regression**.

### **6. What metrics would you use for regression?**

* **Answer**: Metrics include **RMSE**, **MAE**, and **R²** score.

### **7. What are learning curves?**

* **Answer**: Learning curves plot model performance during training to diagnose underfitting or overfitting.

### **8. How do you compare models fairly?**

* **Answer**: Use **nested cross-validation** and **paired t-tests** to ensure identical data splits and fair evaluation.

### **9. What is a confusion matrix?**

* **Answer**: A confusion matrix shows counts of **true positives**, **false positives**, **true negatives**, and **false negatives** for classification models.

### **10. What is threshold tuning?**

* **Answer**: Threshold tuning involves adjusting the classification cutoff to optimize for metrics like precision or recall.

### **11. If your AI model was performing well earlier but now its accuracy has decreased, what should be your approach?**

* **Answer**: Investigate **data drift**, **concept drift**, **pipeline issues**, **labeling consistency**, and perform **A/B testing**.

### **12. What is model robustness?**

* **Answer**: Model robustness refers to stability when input data is slightly perturbed. It can be evaluated with **adversarial examples**.

### **13. What is model interpretability and how is it evaluated?**

* **Answer**: Interpretability means understanding why a model made a decision. Evaluation methods include **SHAP**, **LIME**, and **Partial Dependence Plots**.

### **14. How to validate models in time series?**

* **Answer**: Use **time-aware splits** such as forward chaining or expanding windows to avoid data leakage.

### **15. What are micro, macro, and weighted averages?**

* **Answer**: **Micro** averages over all classes, **macro** averages per class, and **weighted** averages account for class support.

## **🔍 Retrieval-Augmented Generation (RAG)**

### **16. What are common reasons a RAG model fails even when the answer exists in the documents?**

* **Answer**: Issues include **retrieval miss**, **context truncation**, **encoding mismatch**, **ranking failure**, or **model hallucination**.

### **17. How do you debug a RAG system when the model "hallucinates" answers?**

* **Answer**: Check if **retrieved passages** contain relevant information, evaluate **retriever recall**, and use **saliency analysis** to ensure proper attention.

### **18. What metrics are used to evaluate RAG systems?**

* **Answer**: **Retriever metrics** include **Recall@k**, **MRR**, **Precision@k**; **Generator metrics** include **BLEU**, **ROUGE**, **BERTScore**; and **End-to-End metrics** include **EM**, **F1**, and **faithfulness/factual consistency** metrics.

### **19. How to handle latency issues in RAG at inference time?**

* **Answer**: Use **Approximate Nearest Neighbor (ANN)** methods, reduce **chunk size**, or precompute results for high-frequency queries.

### **20. What are advanced retrieval strategies beyond dense retrieval?**

* **Answer**: Techniques include **hybrid retrieval**, **multivector retrieval**, **retrieval-augmented reranking**, and **retrieval with feedback**.

### **21. How to fine-tune retrieval for domain-specific tasks?**

* **Answer**: Use **hard negatives**, **contrastive loss**, and **in-batch negatives** to scale training.

### **22. What are failure modes specific to multi-hop RAG?**

* **Answer**: Issues include **sparse linking**, **bridge entity failure**, and **context fusion problems**.

## **🔧 Hyperparameter Tuning**

### **23. What is hyperparameter tuning?**

* **Answer**: Hyperparameter tuning is the process of selecting the optimal set of parameters for a model to improve performance.

### **24. What methods are used for hyperparameter tuning?**

* **Answer**: Common methods include **grid search**, **random search**, **Bayesian optimization**, and **genetic algorithms**.

### **25. What is the difference between grid search and random search?**

* **Answer**: **Grid search** exhaustively searches over a predefined set of hyperparameters, while **random search** selects hyperparameters randomly within a specified range.

### **26. What are the challenges in hyperparameter tuning?**

* **Answer**: Challenges include the **curse of dimensionality**, **overfitting** to validation data, and the **high cost** of exhaustive search.

### **27. How do you perform hyperparameter tuning for deep learning models?**

* **Answer**: Use techniques like **learning rate schedules**, **batch size optimization**, **dropout rates**, and **optimizer choice** tuning. Methods like **Bayesian optimization** or **random search** can be effective.

### **28. What is learning rate scheduling, and why is it important?**

* **Answer**: Learning rate scheduling involves adjusting the learning rate during training to improve convergence. Popular schedules include **exponential decay**, **step decay**, and **cosine annealing**.

### **29. What is early stopping in the context of hyperparameter tuning?**

* **Answer**: Early stopping involves halting training when the model's performance on the validation set stops improving to prevent overfitting.

## **🤖 Deep Learning & NLP**

### **30. What are transformers, and how do they work?**

* **Answer**: Transformers are neural networks designed for sequential data. They use **self-attention** mechanisms to process input in parallel, significantly improving performance in tasks like language modeling and translation.

### **31. What is BERT, and how does it work?**

* **Answer**: BERT (Bidirectional Encoder Representations from Transformers) is a transformer model that pretrains on vast text data using unsupervised learning and fine-tunes on specific tasks using supervised learning.

### **32. What is GPT, and how does it differ from BERT?**

* **Answer**: GPT (Generative Pretrained Transformer) is a causal language model that generates text autoregressively, while BERT is bidirectional and focuses on understanding context in text for downstream tasks.

### **33. What is the role of attention mechanisms in deep learning?**

* **Answer**: Attention mechanisms allow the model to focus on relevant parts of the input sequence, helping it to capture long-range dependencies and improving performance, especially in NLP tasks.

### **34. What is RLHF (Reinforcement Learning with Human Feedback)?**

* **Answer**: RLHF combines reinforcement learning with human feedback to train models. Humans provide ratings or corrections to model outputs, which are then used to update the model’s parameters.

### **35. What are diffusion models in generative AI?**

* **Answer**: Diffusion models generate data by reversing a diffusion process that gradually adds noise to data. This process is then learned and reversed to generate new, high-quality samples.

### **36. What is the difference between RNNs and transformers?**

* **Answer**: RNNs (Recurrent Neural Networks) process sequences step-by-step, while transformers use self-attention mechanisms, enabling them to handle long-range dependencies more effectively and in parallel.

## **🛠️ Deployment & MLOps**

### **37. What is MLOps?**

* **Answer**: MLOps (Machine Learning Operations) is the practice of combining machine learning with software engineering and DevOps principles to automate and streamline the deployment, monitoring, and maintenance of machine learning models.

### **38. What are common tools for model deployment?**

* **Answer**: Tools like **Docker**, **Kubernetes**, **TensorFlow Serving**, **FastAPI**, **Flask**, and **ONNX** are widely used in model deployment.

### **39. What is model versioning in MLOps?**

* **Answer**: Model versioning involves keeping track of different versions of models deployed in production, which allows for rollback, experimentation, and better model management.

### **40. How do you ensure model monitoring in production?**

* **Answer**: Use **real-time monitoring tools** like **Prometheus**, **Grafana**, and logging frameworks to track performance metrics like accuracy, latency, and data drift.

## **🧠 Advanced Topics**

### **41. What is the difference between batch and online learning?**

* **Answer**: **Batch learning** trains on the entire dataset at once, while **online learning** processes data in small batches or one sample at a time, allowing models to adapt continuously.

### **42. How do you perform transfer learning?**

* **Answer**: Transfer learning involves fine-tuning a pretrained model on a new task or dataset, leveraging the knowledge learned from a previous task.