Model Performance & Evaluation

1. Why would a model's performance degrade over time?

- **Concept Drift**: The relationship between features and labels changes over time (e.g., customer behavior).
- **Data Drift**: Input feature distribution changes (e.g., sensor calibration).
- **Pipeline Issues**: Feature extraction changes, schema updates, or label noise.
- **Retraining Gap**: Model becomes stale as new data patterns emerge.

2. How do you handle model overfitting?

- **Regularization**: L1, L2, dropout.
- **Cross-Validation**: Ensures generalization to unseen data.
- **Early Stopping**: Stops training when validation performance stops improving.
- **Simplify Model**: Reduce parameters or switch to less complex models.
- **Data Augmentation**: Create synthetic diversity in input data.

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

- **Precision**: TP / (TP + FP). High when false positives are costly (e.g., spam).
- **Recall**: TP / (TP + FN). High when false negatives are costly (e.g., cancer detection).
- **F1-Score**: Harmonic mean of precision and recall.

4. How to evaluate a model on imbalanced data?

- **Use Proper Metrics**: F1, PR AUC, ROC AUC.
- **Resampling**: Oversampling minority (SMOTE), undersampling majority.
- **Class Weighting**: Penalize misclassification of rare class.
- **Ensemble Methods**: Boosting or bagging can help balance bias and variance.

⚙️ Hyperparameter Tuning

5. What is hyperparameter tuning and why is it important?

- Optimizing settings like learning rate, max depth, etc., to improve model performance.
- Not learned during training; must be set before.

6. Compare grid search vs random search.

- **Grid Search**: Tests all combinations. Exhaustive but inefficient.
- **Random Search**: Tests random samples. More efficient for high-dimensional problems.

7. How to tune models with large datasets?

- **Subsampling**: Use representative sample.
- **Low-Fidelity Approximations**: Shorter training, fewer epochs.
- **Early Stopping**: Abort poor runs early.
- **Distributed Tuning**: Use tools like Ray Tune or Optuna.

8. Typical hyperparameters by model:

- **XGBoost**: `max_depth`, `eta`, `subsample`, `colsample_bytree`.
- **Neural Nets**: Learning rate, batch size, number of layers, dropout rate.
- **SVM**: Kernel type, `C`, `gamma`.

🧠 Retrieval-Augmented Generation (RAG)

9. What is RAG and how does it work?

- Combines retrieval (e.g., FAISS, BM25) with generation (LLM) for grounded, factual QA.
- Pipeline: Embed query -> Retrieve top docs -> Append to prompt -> Generate answer.

10. Why does RAG fail even if the data has the answer?

- **Retrieval Failures**: Poor embeddings, irrelevant chunking.
- **Prompting Errors**: Model ignores retrieved context.
- **Hallucination**: LLM generates without grounding.

11. How to improve RAG performance?

- **Better Embeddings**: Use domain-tuned models.
- **Chunking Strategy**: Optimal granularity with overlap.
- **Hybrid Retrieval**: Combine dense and sparse search.
- **Prompt Design**: Force model to ground answers in retrieved content.

🤖 Machine Learning Fundamentals

12. Difference between bias and variance?

- **Bias**: Error from wrong assumptions. High bias = underfitting.
- **Variance**: Error from sensitivity to small changes. High variance = overfitting.

13. What is cross-validation?

- **K-Fold**: Train/test split repeated K times with different folds.
- **Stratified CV**: Maintains class ratios. Useful for classification.

14. What is regularization?

- **L1 (Lasso)**: Adds sparsity. Good for feature selection.
- **L2 (Ridge)**: Penalizes large weights. Stabilizes learning.

15. ROC vs PR Curve

- **ROC**: Good when classes are balanced.
- **PR Curve**: Better when data is imbalanced.
- **AUC**: Area under curve used for evaluation.

📊 Data Handling and Feature Engineering

16. How do you handle missing values?

- **Imputation**: Mean, median, mode.
- **Model-based Imputation**: KNN, regression.
- **Drop Rows/Columns**: If missingness is too high.

17. What is feature selection?

- **Filter Methods**: Correlation, chi-square.
- **Wrapper Methods**: Recursive Feature Elimination (RFE).
- **Embedded**: Lasso, tree-based feature importance.

18. Handling Categorical Variables

- **One-hot Encoding**: For nominal categories.
- **Label Encoding**: For ordinal.
- **Target Encoding**: Use label mean; be careful of leakage.

🚀 Deployment & MLOps

19. How do you monitor a model in production?

- **Data Drift Monitoring**: Track input distribution.
- **Performance Monitoring**: Real-world accuracy or latency.
- **Logging & Alerting**: Use Prometheus, Grafana.

20. What is CI/CD in ML?

- **CI**: Automatically test and validate model pipelines.
- **CD**: Automatically deploy updated models.
- Tools: MLflow, DVC, Kubeflow, Airflow.

🧠 Deep Learning & NLP

21. What are vanishing and exploding gradients?

- Gradients too small or large during backprop.
- Solutions: ReLU, batch norm, residual connections.

22. What is transfer learning?

- Using pre-trained model on new task.
- Fine-tune final layers with task-specific data.

23. How do attention mechanisms work?

- Score importance of tokens using queries, keys, values.
- Allows models to focus on relevant parts of input.

24. Fine-tuning vs Prompt Engineering

- **Fine-tuning**: Change model weights.
- **Prompting**: Steer output via smart input only (zero/few-shot).

1. **What is the Central Limit Theorem and why is it important?**
2. **Explain the difference between Type I and Type II errors.**
3. **What is a p-value and how is it interpreted?**
4. **Define confidence interval.**
5. **What is the law of large numbers?**
6. **Explain Bayes' Theorem with an example.**
7. **What is the difference between descriptive and inferential statistics?**
8. **Define and differentiate between variance and standard deviation.**
9. **What is the difference between population and sample?**
10. **Explain the concept of hypothesis testing.**
11. **What are the assumptions of linear regression?**
12. **What is multicollinearity and how can it be detected?**

13. Explain the difference between correlation and causation.
14. What is heteroscedasticity?
15. Define and explain the significance of skewness and kurtosis.
16. What is the difference between parametric and non-parametric tests?
17. Explain the concept of overfitting and underfitting.
18. What is the purpose of ANOVA?
19. Describe the Chi-square test and its applications.
20. What is the difference between a t-test and z-test?

Machine Learning

21. What is the difference between supervised and unsupervised learning?
22. Explain the bias-variance tradeoff.
23. What is cross-validation and why is it important?
24. Describe the k-nearest neighbors algorithm.
25. What is the difference between classification and regression?
26. Explain how decision trees work.
27. What is ensemble learning?
28. Describe the random forest algorithm.
29. What is boosting and how does it differ from bagging?
30. Explain the concept of gradient boosting.
31. What is the purpose of regularization in machine learning?
32. Differentiate between L1 and L2 regularization.
33. What is logistic regression and how does it work?
34. Explain the support vector machine algorithm.
35. What is the kernel trick in SVM?
36. Describe the Naive Bayes classifier.
37. What is the difference between hard and soft clustering?
38. Explain the k-means clustering algorithm.
39. What is hierarchical clustering?
40. Describe the concept of dimensionality reduction.
41. What is Principal Component Analysis (PCA)?
42. Explain the concept of feature selection.
43. What is the curse of dimensionality?
44. How do you handle missing data in a dataset?
45. What is imbalanced data and how can it be addressed?

46. Explain the ROC curve and AUC.
47. What is precision, recall, and F1-score?
48. Describe the confusion matrix.
49. What is a learning curve?
50. Explain the concept of early stopping in training models.

🧮 Deep Learning

51. What is the difference between machine learning and deep learning?
52. Explain the structure of a neural network.
53. What is backpropagation and how does it work?
54. Describe the vanishing gradient problem.
55. What are activation functions and why are they important?
56. Compare ReLU, sigmoid, and tanh activation functions.
57. What is the purpose of dropout in neural networks?
58. Explain convolutional neural networks (CNNs).
59. What are pooling layers in CNNs?
60. Describe recurrent neural networks (RNNs).
61. What is the difference between RNN and LSTM?
62. Explain the concept of attention mechanism.
63. What are generative adversarial networks (GANs)?
64. Describe the architecture of a GAN.
65. What is transfer learning?
66. How do you prevent overfitting in deep learning models?
67. What is batch normalization?
68. Explain the concept of gradient clipping.
69. What are autoencoders and their applications?
70. Describe the concept of sequence-to-sequence models.

🗣️ Natural Language Processing (NLP)

71. What is tokenization in NLP?
72. Explain stemming and lemmatization.

73. What is the Bag of Words model?
74. Describe the TF-IDF approach.
75. What are word embeddings?
76. Compare Word2Vec, GloVe, and FastText.
77. What is the difference between count vectorization and TF-IDF?
78. Explain the concept of n-grams.
79. What is named entity recognition (NER)?
80. Describe sentiment analysis.
81. What is topic modeling?
82. Explain Latent Dirichlet Allocation (LDA).
83. What is the difference between rule-based and machine learning-based NLP?
84. Describe the architecture of the Transformer model.
85. What is BERT and how does it work?
86. Explain the concept of attention in NLP models.
87. What are the challenges in NLP?
88. How do you handle out-of-vocabulary words?
89. What is language modeling?
90. Describe the process of text classification.

Data Engineering & Big Data

91. What is the difference between OLAP and OLTP?
92. Explain the concept of data warehousing.
93. What is ETL and how does it work?
94. Describe the architecture of Hadoop.
95. What is MapReduce?
96. Explain the components of the Hadoop ecosystem.
97. What is Apache Spark and its advantages over Hadoop?
98. Describe the concept of data partitioning.
99. What is the role of a data pipeline?
100. Explain the CAP theorem.
101. What is data sharding?
102. Describe the concept of eventual consistency.
103. What are NoSQL databases and when to use them?
104. Compare SQL and NoSQL databases.
105. What is data normalization?

106. Explain the concept of data lake.
107. What is the difference between batch processing and stream processing?
108. Describe the architecture of Apache Kafka.
109. What is the role of a message broker?
110. Explain the concept of schema evolution.

🧪 Model Evaluation & Tuning

111. What is hyperparameter tuning?
112. Explain grid search and random search.
113. What is cross-validation and its types?
114. Describe the concept of model selection.
115. What is overfitting and how can it be prevented?
116. Explain the bias-variance tradeoff.
117. What is the purpose of a validation set?
118. Describe the concept of bootstrapping.
119. What is the difference between bagging and boosting?
120. Explain the concept of stacking in ensemble methods.
121. What are the common metrics for regression models?
122. Describe the use of ROC-AUC in classification problems.
123. What is the purpose of a confusion matrix?
124. Explain the concept of precision-recall tradeoff.
125. What is the Matthews correlation coefficient?
126. Describe the concept of Cohen's Kappa.
127. What is the significance of learning curves?
128. Explain the concept of model drift.
129. What is the role of feature importance in model evaluation?
130. Describe the process of model calibration.

🛠️ Tools & Technologies

131. What is the difference between Python and R for data science?

132. Explain the use of Pandas in data analysis.
133. What are NumPy arrays and their advantages?
134. Describe the functionality of Matplotlib and Seaborn.
135. What is scikit-learn and its applications?
136. Explain the use of TensorFlow and PyTorch.
137. What is the purpose of Jupyter Notebooks?
138. Describe the role of SQL in data science.
139. What are the common data visualization tools?
140. Explain the use of Docker in data science projects.
141. What is the purpose of Git and GitHub?
142. Describe the concept of virtual environments in Python.
143. What is Apache Airflow and its use cases?
144. Explain the role of Kubernetes in deploying ML models.
145. What is MLflow and how does it help in model management?
146. Describe the use of Tableau and Power BI.
147. What is the purpose of cloud platforms like AWS, GCP, and Azure?
148. Explain the concept of serverless computing.
149. What are REST APIs and their significance?
150. Describe the use of Flask and Django in deploying machine learning models.

Problem-Solving & Case Studies

151. How would you approach a dataset with missing values?
152. Describe a time when you improved a model's performance.
153. How do you handle imbalanced datasets?
154. Explain how you would detect and handle outliers.
155. Describe your process for feature engineering.
156. How do you select the appropriate model for a given problem?
157. Explain how you would deploy a machine learning model to production.
158. Describe a challenging data science problem you faced and how you solved it.
159. How do you ensure the reproducibility of your data science projects?
160. Explain how you would handle a situation where your model's performance degrades over time.

🧠 Behavioral & Situational Questions

161. **Tell me about yourself and your experience in data science.**
162. **Why do you want to work in this company?**
163. **Describe a time when you had to learn a new tool or technology quickly.**
164. **How do you prioritize tasks when working on multiple projects?**
165. **Describe a

To enrich the **RAG (Retrieval-Augmented Generation)** section and balance its depth with other topics, here's how you could expand it with more **advanced troubleshooting scenarios, failure modes, and evaluation metrics**:

🔍 Retrieval-Augmented Generation (RAG)

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

- **Retrieval Miss**: Relevant document isn't among top-k retrieved due to embedding quality, vector store issues, or incorrect query formulation.
- **Context Truncation**: Retrieved documents are truncated or cut off due to input length limits.
- **Encoding Mismatch**: Embedding model used for indexing differs from one used for retrieval.
- **Ranking Failure**: Reranker may misprioritize lower-relevance documents.
- **Model Hallucination**: Generator ignores retrieved content or blends it with parametric knowledge incorrectly.

2. How do you debug a RAG system when the model "hallucinates" answers?

- Check if **retrieved passages** contain relevant information.
- Evaluate **retriever recall**: What percentage of ground-truth answers are in top-k retrieved chunks?

- Perform **saliency analysis** or use **attention heatmaps** to verify if model is attending to the right input.
- Use **contrastive examples** to test factual consistency.

3. What metrics are used to evaluate RAG systems?

- **Retriever Metrics**:
  - *Recall@k, MRR, Precision@k*.
- **Generator Metrics**:
  - *BLEU, ROUGE, METEOR, BERTScore*.
- **End-to-End Metrics**:
  - *EM (Exact Match), F1*, or domain-specific metrics.
  - *Faithfulness* and *Factual Consistency* via tools like **FEVER score**, **QAGS**, or human evaluation.

4. How to handle latency issues in RAG at inference time?

- Use **Approximate Nearest Neighbor (ANN)** methods like FAISS or ScaNN.
- Reduce **chunk size** or optimize **embedding dimensionality**.
- Precompute results for high-frequency queries (cache hits).

5. What are advanced retrieval strategies beyond dense retrieval?

- **Hybrid Retrieval**: Combine BM25 with dense retrieval.
- **Multivector Retrieval**: Use multiple query vectors per document (e.g., ColBERT).
- **Retrieval-augmented reranking**: Use rerankers like T5 or BGE-m3.
- **Retrieval with Feedback**: Incorporate user or model feedback to improve future retrievals.

6. How to fine-tune retrieval for domain-specific tasks?

- Curate **hard negatives** during training (documents similar but incorrect).
- Use **contrastive loss** or **triplet loss**.
- Employ **in-batch negatives** to scale training without manual annotation.

7. What are failure modes specific to multi-hop RAG?

- **Sparse linking**: Documents needed for different hops may not co-occur.

- **Bridge entity failure**: The bridge concept connecting hops may be missing or missed in retrieval.
- **Context fusion issues**: Generator fails to combine multiple retrieved contexts correctly.

Would you like me to add this expanded RAG section directly to the document or continue enhancing other sections (e.g., Transformers, RLHF, Diffusion Models) with similar depth?

You said:

add this expanded RAG section directly to the document

ChatGPT said:

It looks like there was an issue inserting the RAG section directly into the document. Here's the full section — you can copy and paste it into your document:

🔍 Retrieval-Augmented Generation (RAG)

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

- **Retrieval Miss**: Relevant document isn't among top-k retrieved due to embedding quality, vector store issues, or incorrect query formulation.
- **Context Truncation**: Retrieved documents are truncated or cut off due to input length limits.
- **Encoding Mismatch**: Embedding model used for indexing differs from one used for retrieval.
- **Ranking Failure**: Reranker may misprioritize lower-relevance documents.
- **Model Hallucination**: Generator ignores retrieved content or blends it with parametric knowledge incorrectly.

2. How do you debug a RAG system when the model "hallucinates" answers?

- Check if **retrieved passages** contain relevant information.
- Evaluate **retriever recall**: What percentage of ground-truth answers are in top-k retrieved chunks?
- Perform **saliency analysis** or use **attention heatmaps** to verify if model is attending to the right input.
- Use **contrastive examples** to test factual consistency.

3. What metrics are used to evaluate RAG systems?

- **Retriever Metrics**:
    - *Recall@k, MRR, Precision@k*.
- **Generator Metrics**:
    - *BLEU, ROUGE, METEOR, BERTScore*.
- **End-to-End Metrics**:
    - *EM (Exact Match), F1*, or domain-specific metrics.
    - *Faithfulness* and *Factual Consistency* via tools like **FEVER score**, **QAGS**, or human evaluation.

4. How to handle latency issues in RAG at inference time?

- Use **Approximate Nearest Neighbor (ANN)** methods like FAISS or ScaNN.
- Reduce **chunk size** or optimize **embedding dimensionality**.
- Precompute results for high-frequency queries (cache hits).

5. What are advanced retrieval strategies beyond dense retrieval?

- **Hybrid Retrieval**: Combine BM25 with dense retrieval.
- **Multivector Retrieval**: Use multiple query vectors per document (e.g., ColBERT).
- **Retrieval-augmented reranking**: Use rerankers like T5 or BGE-m3.
- **Retrieval with Feedback**: Incorporate user or model feedback to improve future retrievals.

6. How to fine-tune retrieval for domain-specific tasks?

- Curate **hard negatives** during training (documents similar but incorrect).

- Use **contrastive loss** or **triplet loss**.
- Employ **in-batch negatives** to scale training without manual annotation.

7. What are failure modes specific to multi-hop RAG?

- **Sparse linking**: Documents needed for different hops may not co-occur.
- **Bridge entity failure**: The bridge concept connecting hops may be missing or missed in retrieval.
- **Context fusion issues**: Generator fails to combine multiple retrieved contexts correctly.

Balancing **speed** and **accuracy** in AI models, especially when dealing with large datasets or real-time applications, involves optimizing the model architecture, processing techniques, and deployment strategies. Here are some key strategies to achieve that balance:

1. Model Complexity and Size

- **Smaller Models**: Use simpler models (e.g., shallow neural networks, decision trees) for faster inference times, though this may compromise accuracy.
- **Larger Models**: More complex models (e.g., deep neural networks, transformers) typically deliver higher accuracy but at the cost of slower inference times. You can reduce the model size using techniques like **knowledge distillation**.

2. Early Stopping in Training

- **Stopping Early**: During training, monitor performance on a validation set and stop the training once the model stops improving. This can prevent overfitting, speeding up both training and deployment without sacrificing accuracy.

3. Approximate Methods

- **Approximate Nearest Neighbors (ANN)**: When dealing with large-scale datasets, ANN algorithms like **FAISS**, **ScaNN**, and **HNSW** can speed up retrieval and search tasks with minimal loss of accuracy.

- **Model Quantization**: Reduce the precision of your model’s weights (e.g., using 8-bit integers instead of floating-point numbers). This can significantly reduce computation time without a large drop in accuracy.

4. Model Optimization for Inference

- **Pruning**: This technique removes unimportant weights from the model (those that don’t contribute significantly to predictions), making it smaller and faster, but potentially impacting accuracy slightly.
- **Distillation**: Transfer knowledge from a large, high-accuracy model to a smaller, more efficient one. The distilled model is faster but can retain much of the original model's accuracy.

5. Hardware Acceleration

- **GPU/TPU**: Leveraging hardware like GPUs or TPUs can provide significant speedups in deep learning tasks. However, this requires tuning your model to make full use of the hardware.
- **Edge Devices**: For real-time applications (e.g., mobile or IoT), optimize models for low-power devices by using techniques like **TensorFlow Lite** or **ONNX** for efficient execution.

6. Efficient Data Processing Pipelines

- **Data Preprocessing**: Improve speed by optimizing data pipelines. Use batching and parallelization to process multiple inputs simultaneously. This will speed up training and inference without sacrificing accuracy.
- **Feature Selection/Engineering**: Selecting only the most relevant features for your model can improve speed without a significant drop in accuracy.

7. Adaptive Techniques

- **Adaptive Thresholding**: Use dynamic thresholds based on confidence levels. For example, if the model’s confidence in an answer is high, you can use a faster, less accurate model for inference.
- **Early Prediction**: Implement a system where the model performs light computations for simpler, quicker cases and switches to a more complex model only when necessary.

8. Transfer Learning

- Use pretrained models (e.g., BERT, GPT, etc.) for faster convergence. Fine-tune these models on your specific task. Transfer learning can significantly improve accuracy with much less computational cost during training and testing.

9. Model Ensemble

- **Ensemble Methods**: Combining several models, such as in **bagging** (e.g., Random Forest) or **boosting** (e.g., XGBoost), can improve accuracy. However, these methods are slower due to multiple models, so balancing how many models you use is key to maintaining speed.

10. Evaluation and A/B Testing

- Test and evaluate models with both speed and accuracy as key criteria. If needed, choose the model that performs well enough for production but doesn't overcomplicate the system.
- **A/B Testing**: Try different versions of models in parallel (e.g., faster but slightly less accurate models vs. more accurate but slower models) and assess which best fits user needs.

11. Batching Inference Requests

- When dealing with multiple inference requests (e.g., in a server setup), use **batch processing**. This can help reduce the time spent on redundant calculations and improve throughput.

Conclusion:

The tradeoff between speed and accuracy is often a matter of context. While aiming for the fastest model, ensure that the drop in accuracy is acceptable for the business or application requirements. Start with the most efficient models and iteratively apply techniques like pruning, quantization, and hardware optimization for balancing speed and accuracy.