100 Machine Learning Interview Questions and Answers (Basic to Advanced)

Below are 100 Machine Learning interview questions spanning from fundamental concepts to advanced techniques. Each answer now includes more detailed explanations and reasoning to provide deeper insights into each topic.

---

1. What is Machine Learning?

Answer:
Machine Learning (ML) is a subset of artificial intelligence where computers learn patterns from data without being explicitly programmed. Instead of hand-crafting rules, we feed algorithms large amounts of examples from which they infer generalizable patterns. This allows ML systems to improve their performance on a task, such as classification or prediction, as they gain more experience (data).

2. What are the three main types of Machine Learning?

Answer:

1. **Supervised Learning:** Involves labeled data where the target outcome (label) is known. Models learn a mapping from inputs (features) to outputs (labels). Examples: classification (spam detection) and regression (price prediction).

2. **Unsupervised Learning:** Works with unlabeled data, finding hidden structures or patterns, like grouping similar items (clustering) or reducing dimensionality. Examples: customer segmentation using K-means clustering.

3. **Reinforcement Learning:** An agent interacts with an environment, receiving rewards or penalties. Over time, it learns a policy that maximizes cumulative reward (e.g., a game-playing agent improving its strategy over many rounds).

3. What is the difference between supervised and unsupervised learning?

Answer:

- **Supervised Learning:** Uses labeled data, meaning we know the correct answers for training examples. The model's goal is to learn a function that accurately maps inputs to the known outputs, such as predicting a house's price based on features.

- **Unsupervised Learning:** Deals with unlabeled data, where no predefined targets exist. The model tries to discover underlying patterns or structures in the input features alone, for instance grouping similar images or detecting anomalies.

4. Define overfitting.

Answer:

Overfitting occurs when a model learns not only the general patterns but also the random noise or fluctuations in the training data. Such a model performs extremely well on training data but fails to generalize to unseen data, leading to poor performance in the real world. Overfitting is often indicated by a large gap between high training accuracy and low test accuracy.

5. How can you prevent overfitting?

Answer:

Prevention methods include:

- **Regularization (L1, L2):** Adding a penalty to large coefficients encourages simpler models.
- **Early Stopping:** Halting training when validation performance stops improving.
- **Data Augmentation:** Increasing the training set size or diversity.
- **Cross-Validation:** Ensuring robust estimates of model performance.
- **Model Simplification:** Using fewer parameters or simpler models.
- **Ensembling:** Combining multiple models to smooth out irregularities.

6. What is underfitting?

Answer:

Underfitting happens when a model is too simple or constrained and fails to capture the underlying trends in the data. It results in poor performance on both training and test sets. For example, using a linear model for a highly nonlinear problem can cause underfitting.

7. Explain the bias-variance trade-off.

Answer:

- **Bias:** Error from simplifying assumptions in the model; high-bias models are too rigid and underfit.
- **Variance:** Error from sensitivity to small data fluctuations; high-variance models overfit.

The trade-off involves balancing these two. A model with too high bias underfits, while too high variance overfits. Achieving a good compromise results in better generalization.

8. What is a training set, validation set, and test set?

Answer:

- **Training Set:** The model learns from this data, adjusting parameters to minimize errors.

- **Validation Set:** Used to tune hyperparameters and select the best model variant. It helps prevent overfitting to the training set.
- **Test Set:** A final, unseen dataset used after all tuning is complete to estimate the model's true generalization performance.

9. Why do we use cross-validation?

Answer:

Cross-validation splits data into multiple folds, using different folds as validation each time. This yields a more robust and reliable estimate of model performance. It reduces variance in evaluation and helps in optimal model/parameter selection, especially when data is limited.

10. What is regularization and why is it important?

Answer:

Regularization adds a complexity penalty (often on the size of coefficients) to the loss function. It discourages overly complex models, reducing overfitting and enhancing generalization. For example, L2 (Ridge) regularization shrinks coefficients, while L1 (Lasso) promotes sparsity by setting some coefficients to zero.

11. Explain L1 and L2 regularization.

Answer:

- **L1 (Lasso):** Uses the absolute values of weights as a penalty. It tends to produce sparse solutions, effectively performing feature selection by forcing some weights to zero.

- **L2 (Ridge):** Uses the squared weights as a penalty, spreading out the penalty more smoothly. Ridge does not usually zero out coefficients but shrinks them, stabilizing the solution and reducing variance.

12. What is logistic regression used for?

Answer:

Logistic regression is a supervised classification algorithm modeling the probability of a binary outcome. Instead of predicting continuous values, it predicts class probabilities (e.g., spam vs. not spam). The logistic (sigmoid) function ensures outputs are between 0 and 1, representing probability.

13. Define a confusion matrix.

Answer:

A confusion matrix is a table that visualizes the performance of a classification model by comparing predicted labels against actual labels. It has four main cells:

- **True Positive (TP):** Correct positive predictions.
- **True Negative (TN):** Correct negative predictions.
- **False Positive (FP):** Incorrectly predicted positives.

- **False Negative (FN):** Missed positives (predicted negative but actually positive).

14. What are precision and recall?

Answer:

- **Precision:** Of all predicted positives, how many are truly positive? High precision means few false alarms (FP).
  Formula: Precision = TP / (TP + FP)

- **Recall (Sensitivity):** Of all actual positives, how many did we correctly identify? High recall means catching most of the positives.
  Formula: Recall = TP / (TP + FN)

15. What is the F1-score?

Answer:

The F1-score is the harmonic mean of precision and recall. It provides a single metric that balances both, especially useful if there's class imbalance or if both precision and recall are important.
F1 = 2 * (Precision * Recall) / (Precision + Recall)

16. What is accuracy, and when is it not a good metric?

Answer:

Accuracy = (TP + TN) / (All Samples). It's the percentage of correct predictions.
However, in imbalanced datasets (e.g., detecting rare diseases), accuracy can be misleading. If 99% of cases are negative, a naive model always predicting negative achieves 99% accuracy but is worthless. In these cases, metrics like F1-score, precision, recall, or AUC are more informative.

17. What is ROC AUC?

Answer:

ROC (Receiver Operating Characteristic) curve plots the True Positive Rate vs. False Positive Rate at various thresholds. AUC (Area Under the Curve) measures how well the model ranks positive instances higher than negatives. An AUC of 1.0 means perfect discrimination, while 0.5 is random guessing.

18. Define Mean Squared Error (MSE).

Answer:

MSE is a regression metric defined as the average of the squared differences between predicted and actual values. By squaring errors, large deviations are penalized heavily. MSE = (1/n) * $\Sigma(\text{actual - predicted})^2$.

19. What is the purpose of gradient descent?

Answer:

Gradient descent is an optimization technique that iteratively adjusts parameters to minimize a loss function. By computing the gradient (slope) of the loss with respect to parameters, it updates parameters in the direction that reduces error, eventually converging to (or near) a minimum.

20. What is a learning rate in gradient descent?

Answer:

The learning rate controls the size of the steps taken when updating parameters. A too large learning rate might overshoot minima and diverge. A too small one leads to very slow convergence. Choosing a good learning rate is crucial for efficient training.

21. Explain the concept of feature engineering.

Answer:

Feature engineering transforms raw data into meaningful representations that improve model performance. It can involve:

- Creating interaction terms (e.g., multiplying features).
- Normalizing or scaling features.
- Using domain knowledge to craft more relevant features. Better features can simplify learning, increase accuracy, and sometimes reduce the complexity needed in the model.

22. How do you handle missing data?

Answer:

Approaches include:

- Dropping rows or columns with missing values (if few and not critical).
- Imputing using mean, median, or mode.
- Using predictive models to impute missing values.
- Employing algorithms tolerant of missing data. The choice depends on data quantity, distribution, and the importance of the features.

23. What is a decision tree?

Answer:

A decision tree is a flowchart-like structure of binary decisions. Each node splits the data based on a feature test (e.g., “Is feature X ≤ value?”). Splits aim to produce pure leaf nodes. Decision trees are easy to interpret but prone to overfitting if not pruned or regularized.

24. What are ensemble methods?

Answer:

Ensemble methods combine multiple base models (often called “weak learners”) to achieve better predictive performance. By aggregating diverse predictions, ensembles reduce variance, bias, or both. Examples: Bagging

(Random Forest), Boosting (XGBoost), and Stacking.

25. Explain Random Forest.

Answer:

A Random Forest is a bagging ensemble of decision trees. Each tree is trained on a bootstrap sample and a random subset of features, increasing diversity. The final prediction is typically the majority vote for classification or average for regression. This reduces overfitting and improves stability compared to a single decision tree.

26. What is boosting?

Answer:

Boosting trains models sequentially. Each new model focuses on correcting errors from the previous ones, giving more weight to misclassified instances. Over iterations, it creates a strong model from many weak learners, reducing both bias and variance. Examples include AdaBoost and Gradient Boosting.

27. Explain XGBoost.

Answer:

XGBoost (Extreme Gradient Boosting) is a highly optimized gradient boosting library. It improves speed and performance via efficient implementations of tree splitting, uses regularization to avoid overfitting, handles missing data gracefully, and supports parallel computation. It’s popular in many ML competitions for its accuracy and speed.

28. What is the curse of dimensionality?

Answer:

As the number of features (dimensions) grows, data becomes sparse. Models need exponentially more data to achieve the same level of accuracy, making learning difficult. High-dimensional spaces also complicate distance-based methods and can lead to overfitting.

29. What is PCA (Principal Component Analysis)?

Answer:

PCA is a linear dimensionality reduction method that identifies directions (principal components) of maximum variance in the data. By projecting onto a few principal components, PCA reduces noise, speeds up training, and mitigates the curse of dimensionality, while retaining most of the informative structure.

30. What is LDA (Linear Discriminant Analysis)?

Answer:

LDA is both a classification and dimensionality reduction technique. For supervised dimensionality reduction, LDA seeks projections that best separate classes by maximizing between-class variance and minimizing within-class variance. It’s commonly used for tasks where clear class boundaries exist.

31. How do you select the number of clusters in K-means?

Answer:

Techniques include:

- **Elbow Method:** Plot within-cluster sum of squares (WCSS) vs. k, and pick k at the "elbow" point.
- **Silhouette Score:** Measures how similar each point is to its own cluster vs. other clusters. Higher silhouette indicates better clustering.
- **Domain Knowledge:** Practical insights sometimes guide k selection.

32. What is hierarchical clustering?

Answer:

Hierarchical clustering creates a hierarchy of clusters. **Agglomerative** starts with each point as a single cluster and merges them step-by-step. **Divisive** starts with one cluster and splits it. The result is often visualized as a dendrogram, allowing you to choose a clustering level.

33. Explain model interpretability and why it matters.

Answer:

Model interpretability is understanding how and why a model makes specific predictions. It matters for:

- Trust and Transparency: Stakeholders need to know if decisions are fair and reliable.
- Compliance: Some regulations require explanations of automated decisions.
- Debugging: Interpretability helps identify errors in data or model logic.

34. What is a kernel in SVM?

Answer:

A kernel function transforms data into a higher-dimensional space, making complex separations appear linear. Common kernels: linear, polynomial, RBF (Gaussian). Kernels let SVMs solve nonlinear classification problems efficiently without explicitly computing coordinates in the higher space.

35. What is regularization in linear models?

Answer:

Regularization adds a penalty to large coefficients, controlling model complexity. In linear models, it prevents weights from becoming too large (overfitting). Examples: Ridge (L2) and Lasso (L1) regularization. It leads to simpler, more generalizable models.

36. Compare batch gradient descent and stochastic gradient descent.

Answer:

- **Batch Gradient Descent:** Uses the entire training set to compute the gradient for each update. More stable but can be slow for large datasets.
- **Stochastic Gradient Descent (SGD):** Uses one (or a small batch) example at a time, making quicker updates and often converging faster in practice, especially useful with very large datasets.

37. What is the difference between parametric and non-parametric models?

Answer:

- **Parametric Models:** Have a fixed number of parameters. Assume a functional form (e.g., linear regression). Faster, need less data, but more prone to bias if assumptions are wrong.
- **Non-parametric Models:** Number of parameters grows with data. Make fewer assumptions, can fit complex patterns, but risk overfitting and can be slower.

38. Explain what a validation curve is.

Answer:

A validation curve plots model performance (e.g., accuracy) on both training and validation sets against a single hyperparameter (like regularization strength). It helps identify where the model overfits or underfits and guides in choosing an optimal hyperparameter value.

39. What is early stopping?

Answer:

Early stopping monitors validation performance during training and halts as soon as the validation score stops improving, preventing overfitting by not letting the model over-train on the noise of the training set.

40. Define Transfer Learning.

Answer:

Transfer Learning leverages knowledge learned from one (usually large) task/domain and applies it to another related task/domain with limited data. Often seen in deep learning, where a model pre-trained on ImageNet is fine-tuned on a smaller dataset, accelerating training and improving performance.

41. What is a confusion matrix, and what are its components?

Answer:

Already covered in Q13, but to reiterate: A confusion matrix is a table comparing predicted vs. actual labels. It includes TP, TN, FP, FN. From these, you derive metrics like precision, recall, and accuracy, understanding errors and successes in classification.

42. Explain the concept of stratified sampling.

Answer:

Stratified sampling splits the dataset so that each fold (in cross-validation) or subset maintains the original class proportion. This ensures that models are evaluated on representative data, especially important for imbalanced classes.

43. What is data leakage?

Answer:

Data leakage occurs when information that would not be available at prediction time is inadvertently used to train the model. This often leads to overly optimistic performance estimates that vanish in real-world deployment. An example is scaling data using the whole dataset, including the test set, before model training.

44. What is a ROC curve?

Answer:

A ROC curve plots TPR (Recall) vs. FPR (1 - Specificity) at various classification thresholds. It shows how well the classifier can separate positive and negative classes. The AUC summarizes the curve’s overall performance.

45. How do you handle class imbalance?

Answer:

Techniques include:

- **Oversampling the minority class (e.g., SMOTE).**
- **Undersampling the majority class.**
- **Adjusting class weights in the model’s loss function.**
- **Focusing on metrics like F1-score or AUC rather than accuracy.**

46. What is a cost function (or loss function)?

Answer:

A cost (loss) function quantifies how far off the model’s predictions are from the true values. Minimizing this loss guides training. For regression, MSE is common; for classification, cross-entropy is often used.

47. Differentiate between Gini impurity and entropy in decision trees.

Answer:

Both measure impurity. **Entropy** (from information theory) measures unpredictability, while **Gini impurity** measures how often a randomly chosen sample would be misclassified. Both guide tree splits, but Gini is often slightly faster and commonly used in practice.

48. What is a hyperparameter?

Answer:

A hyperparameter is a model configuration set before training (e.g., learning rate, number of hidden layers, regularization strength). Unlike model parameters learned from data, hyperparameters are chosen via tuning methods like grid search or Bayesian optimization.

49. How do you tune hyperparameters?

Answer:

Techniques:

- **Grid Search:** Exhaustively tries all parameter combinations.
- **Random Search:** Randomly samples hyperparameter space, often more efficient.
- **Bayesian Optimization:** Models the objective function to intelligently choose hyperparameters.
- **Hyperband/Successive Halving:** Efficient resource allocation strategies.

50. Explain ensemble averaging and voting.

Answer:

Ensemble averaging (for regression) or majority voting (for classification) combines predictions from multiple models. For classification, hard voting chooses the class with the most votes; soft voting averages predicted probabilities. This leverages model diversity to boost overall accuracy and robustness.

51. What is gradient boosting?

Answer:

Gradient boosting builds an ensemble of weak learners (usually shallow trees), each correcting the residual errors of the previous ensemble. By moving down the gradient of the loss function, it incrementally improves performance, often resulting in highly accurate models.

52. How does AdaBoost work?

Answer:

AdaBoost starts with a weak classifier and reweights samples, giving more importance to previously misclassified examples. New weak learners focus on harder instances. The final model is a weighted sum of these weak learners, achieving strong performance despite each weak learner’s simplicity.

53. Explain the concept of model drift.

Answer:

Model drift (or data drift) occurs when the data distribution changes over time, rendering the model’s original patterns less relevant. To handle it, we might retrain regularly, monitor predictions, and adapt the model to evolving conditions.

54. What are one-hot encoding and label encoding?

Answer:

- **One-hot encoding:** Transforms a categorical feature into multiple binary features, each representing a category. Removes any implied ordinal relationship.
- **Label encoding:** Assigns an integer to each category. It's simpler but can misleadingly imply an order among categories.

55. Why might you prefer a smaller model?

Answer:

A smaller model:

- Reduces the risk of overfitting.
- Is faster to train and predict.
- Is easier to interpret and maintain.
- Requires fewer computational resources.

In many practical scenarios, a simpler model that performs almost as well as a complex one is preferred due to cost and maintainability.

56. What is a baseline model and why is it useful?

Answer:

A baseline model sets a minimal performance standard (e.g., predicting the mean for regression or the majority class for classification). It helps understand if complex models provide real improvements over trivial solutions.

57. Explain the concept of latent variables.

Answer:

Latent variables are hidden factors not directly observed but inferred from data. For example, "customer satisfaction" might influence survey responses. Techniques like factor analysis or topic modeling uncover these latent dimensions, explaining observed patterns more fundamentally.

58. How do AutoML tools assist in model building?

Answer:

AutoML tools automate tasks like feature engineering, model selection, and hyperparameter tuning. They free practitioners from manual trial-and-error, accelerating experimentation and can produce strong models with less human intervention.

59. What is a pipeline in ML?

Answer:

A pipeline chains together preprocessing steps (scaling, encoding) and the model into a single workflow. This ensures the exact same transformations are applied consistently, simplifies code, and makes hyperparameter tuning and deployment more straightforward and reproducible.

60. Define overparameterization.

Answer:

Overparameterization occurs when a model has far more parameters than necessary to represent the target function. Neural networks often are overparameterized but still generalize well with proper regularization. Without controls, overparameterization typically leads to overfitting.

61. What is the difference between feature selection and feature extraction?

Answer:

- **Feature selection:** Chooses a subset of existing features to reduce dimensionality.
- **Feature extraction:** Creates new features (e.g., through PCA or embeddings) that summarize or transform original features into a lower-dimensional space.

62. Why is scaling features important?

Answer:

Scaling ensures all features contribute equally to the model and that gradient-based methods converge more easily. Without scaling, features with large magnitudes may dominate the learning process, skewing the model's behavior and making it harder to find optimal solutions.

63. Explain SMOTE.

Answer:

SMOTE (Synthetic Minority Over-sampling TEchnique) creates synthetic minority class samples by interpolating between existing minority samples. This technique helps address class imbalance without simply replicating existing minority examples, improving model sensitivity to minority classes.

64. What is online learning?

Answer:

Online learning updates model parameters incrementally as each new data point arrives. It's useful for streaming or large-scale scenarios where batch retraining is costly. The model evolves continuously and can adapt quickly to changing data distributions.

65. Compare ridge and lasso regression.

Answer:

- **Ridge (L2):** Shrinks all coefficients but rarely zeroes them out. It reduces variance and stabilizes solutions.
- **Lasso (L1):** Encourages sparsity by pushing some coefficients exactly to zero, performing feature selection. Good when many features are irrelevant.

66. What does an ROC AUC of 0.5 mean?

Answer:

An AUC of 0.5 indicates the model is no better than random guessing. The model has no discriminative power to rank positive instances above negatives.

67. Explain data augmentation.

Answer:

Data augmentation artificially expands the training set by applying transformations to existing examples. For images, this could be rotations, flips, or brightness changes. It reduces overfitting and improves robustness by making the model invariant to these transformations.

68. What is the main idea behind Bayesian methods in ML?

Answer:

Bayesian methods treat parameters and predictions as distributions rather than fixed values. Prior beliefs are updated with observed data, resulting in a posterior distribution. This approach quantifies uncertainty and can be more robust when data is scarce, guiding more informed decisions.

69. How do you handle outliers?

Answer:

Options:

- Remove or cap outliers if they’re data errors.
- Apply transformations like log or Box-Cox to reduce their influence.
- Use robust models/metrics less sensitive to outliers (e.g., median-based loss).
- Detect them using statistical methods (z-scores, IQR) before deciding on an approach.

70. Define anomaly detection.

Answer:

Anomaly detection identifies unusual patterns or rare events that deviate significantly from the majority. Applications include fraud detection, fault diagnosis, and intrusion detection. Methods include isolation forests, one-class SVM, or statistical deviation analysis.

71. What is a learning curve?

Answer:

A learning curve plots the model’s performance on training and validation sets as a function of the training set size. It shows if more data would help (if validation score is still improving) or if the model is too simple or complex (underfitting or overfitting patterns).

72. How is the R^2 score interpreted?

Answer:

R^2 measures what fraction of the variance in the target variable is explained by the model. An R^2 of 1.0 means perfect fit. R^2 near 0 means the model explains almost none of the variance, and negative values indicate the model is even worse than a trivial mean predictor.

73. What is the purpose of model calibration?

Answer:

Model calibration ensures predicted probabilities match observed frequencies. A well-calibrated model saying “70% chance” should see the event happen about 70% of the time. Calibration techniques like Platt scaling or isotonic regression improve the reliability of probability estimates.

74. Explain monotonicity constraints.

Answer:

Monotonicity constraints force predictions to consistently increase (or decrease) with certain features. For example, “price should not decrease as quality score increases.” Such constraints improve interpretability and ensure predictions follow known domain rules.

75. What is a kernel trick?

Answer:

The kernel trick allows algorithms like SVMs to operate in a transformed feature space without explicitly computing coordinates in that space. By using kernel functions (e.g., RBF), it efficiently captures complex nonlinear relationships while keeping computations tractable.

76. Define meta-learning.

Answer:

Meta-learning, or “learning to learn,” trains models on multiple tasks so that they can quickly adapt to new tasks with few examples. It leverages prior knowledge to reduce data requirements and speed up learning in new domains.

77. What is the difference between ML and DL (Deep Learning)?

Answer:

Deep Learning is a subset of ML using neural networks with many layers to learn hierarchical feature representations automatically from raw data. Traditional ML often requires manual feature engineering, whereas DL can learn features end-to-end, given sufficient data and computation.

78. Explain batch normalization (in the DL context).

Answer:

Batch normalization normalizes layer inputs over a mini-batch, stabilizing and speeding up training. It reduces internal covariate shift, allows higher learning rates, and often improves both convergence speed and final performance.

79. Why use dropout in neural networks?

Answer:

Dropout randomly deactivates a proportion of neurons during training, preventing over-reliance on specific features and encouraging multiple neurons to learn robust patterns. This reduces overfitting and improves generalization.

80. What are attention mechanisms in neural networks?

Answer:

Attention mechanisms allow models to focus on different parts of the input when making predictions, assigning learned weights to elements. For example, in machine translation, attention highlights important words in the source sentence for each target word, improving performance on long sequences.

81. Define reinforcement learning.

Answer:

Reinforcement Learning (RL) trains an agent to act in an environment to maximize cumulative rewards. The agent learns by trial and error, receiving feedback from its actions. It’s used in robotics, game playing (like AlphaGo), and resource management.

82. What is the purpose of Q-learning in RL?

Answer:

Q-learning is an off-policy RL algorithm that learns a value function (Q-values) mapping state-action pairs to expected future rewards. By continually updating Q-values, the agent identifies an optimal policy to achieve the highest long-term reward.

83. Explain the concept of generalization in ML.

Answer:

Generalization is a model’s ability to perform well on unseen data. A good generalizing model doesn’t just memorize training examples—it learns underlying patterns that hold true beyond the training set, enabling it to make accurate predictions on new samples.

84. What is model drift, and how do you mitigate it?

Answer:

Model drift arises when data distribution shifts over time. Mitigation involves:

- Monitoring performance regularly.
- Periodically retraining the model with fresh data.

- Employing online learning or adaptive models that update continuously.
- Using alerts or triggers when input distributions or error rates significantly change.

85. Define adversarial examples.

Answer:

Adversarial examples are inputs deliberately crafted with small perturbations that fool ML models into making incorrect predictions, despite appearing normal to humans. This exposes vulnerabilities in models, prompting research into robust and secure ML techniques.

86. What is explainability (XAI)?

Answer:

Explainable AI focuses on techniques and tools to make model decisions understandable by humans. It includes methods like LIME or SHAP to show which features influenced a prediction, aiding trust, compliance, and debugging of complex black-box models.

87. How do you choose an appropriate evaluation metric?

Answer:

Match the metric to the task and business goals. For imbalanced classification, F1 or AUC may be better than accuracy. For ranking tasks, consider MAP or NDCG. For regression, MSE or MAE might suffice. Also consider costs of false positives vs. false negatives.

88. What is multi-class vs. multi-label classification?

Answer:

- **Multi-class:** Each instance belongs to exactly one of several mutually exclusive classes.
- **Multi-label:** Each instance can have multiple associated labels simultaneously. For example, a single image can contain both a cat and a dog.

89. Explain zero-shot learning.

Answer:

Zero-shot learning predicts classes that were not seen in the training phase. The model leverages semantic relationships, textual descriptions, or attribute-based representations to generalize to new classes without direct training examples.

90. What is few-shot learning?

Answer:

Few-shot learning aims to achieve good performance from only a handful of labeled examples. By leveraging prior knowledge or meta-learning, the model adapts quickly to new tasks with very limited data.

91. How do you debug a model that’s performing poorly?

Answer:

- Check data quality and distribution shifts.
- Inspect if labels are correct.
- Try simpler baseline models or different feature engineering.
- Evaluate different metrics to identify the problem.
- Use interpretability tools (LIME, SHAP) to understand predictions.
- Perform error analysis on specific subgroups.

92. Explain the concept of model serving and MLOps.

Answer:

Model serving is the deployment of ML models into production, making predictions available through APIs or batch processes. MLOps extends DevOps principles to ML, covering continuous integration/deployment, monitoring, reproducibility, model versioning, and automated retraining, ensuring reliable and efficient ML systems at scale.

93. What is an embedding?

Answer:

An embedding maps high-dimensional, sparse data (words, items, users) into dense, low-dimensional vectors capturing semantic similarities and patterns. For example, word embeddings like Word2Vec place similar words close together in vector space, improving downstream tasks like NLP.

94. How do you ensure fairness in ML models?

Answer:

- Audit data for biased distributions.
- Remove or mask sensitive attributes if possible.
- Employ fairness-aware algorithms or regularization.
- Evaluate metrics like disparate impact or equalized odds.
- Continually monitor and refine models to prevent discrimination.

95. What is incremental learning?

Answer:

Incremental (or continual) learning updates models with new data without retraining from scratch. It’s useful when data arrives continuously or when recalculating from the entire historical dataset is too expensive. Careful methods prevent catastrophic forgetting of previously learned information.

96. Why use probabilistic models?

Answer:

Probabilistic models provide not just predictions but also uncertainty estimates. This aids decision-making under uncertainty. For example, knowing a model’s confidence helps determine when to defer decisions or gather more data.

97. Explain active learning.

Answer:

Active learning selects the most informative samples to label next, reducing labeling costs. The model queries an oracle (e.g., a human annotator) only for samples that will most improve the model if labeled, accelerating learning with fewer labeled samples.

98. How do you handle concept drift?

Answer:

Concept drift means the relationship between input and output changes over time. Mitigation includes:

- Continual monitoring of performance.
- Periodic or triggered retraining.
- Weighted ensembles giving more weight to recent data.
- Adaptive online learning techniques.

99. Define monotone constraints in gradient boosting.

Answer:

Monotone constraints ensure that as a particular feature value increases, the model’s predictions either never decrease or never increase (depending on the constraint). It encodes domain knowledge (e.g., price should not decrease with quality) and improves trust and interpretability.

100. When would you use Bayesian Optimization?

Answer:

Bayesian Optimization is used for efficient hyperparameter tuning when model evaluations are expensive. It builds a surrogate model of the objective function and uses an acquisition function to select promising hyperparameter configurations, often converging to good solutions with fewer trials than grid or random search.