Machine Learning Basics Notes 📝

+ interview questions at the end

-- Amar Sharma

Notes on Machine Learning Basics

Definition of Machine Learning

Machine Learning (ML) is a subset of artificial intelligence (AI) that enables systems to learn patterns and make decisions without being explicitly programmed. It involves algorithms that improve automatically through experience.

Key Types of Machine Learning

1. Supervised Learning:

The model is trained on labeled data.

The aim is to map inputs to outputs based on example input-output pairs.

Examples:

Classification (e.g., spam detection).

Regression (e.g., predicting house prices).

2. Unsupervised Learning:

The model works on unlabeled data to find patterns or structures.

Examples:

Clustering (e.g., customer segmentation).

Dimensionality reduction (e.g., PCA).

3. Semi-Supervised Learning:

Combines labeled and unlabeled data to improve learning accuracy.

Example: Image recognition with a few labeled images.

4. Reinforcement Learning:

The model learns by interacting with an environment and receiving rewards or penalties.

Example: Game-playing AI like AlphaGo.

Key Concepts in Machine Learning

1. Features and Labels:

Features: Independent variables (input data).

Labels: Dependent variables (output data).

2. Training and Testing Data:

Training set: Data used to train the model.

Test set: Data used to evaluate model performance.

3. Model:

A mathematical representation of the relationship between input and output.

4. Algorithm:

A method to create a model by finding patterns in data (e.g., decision trees, neural networks).

Common ML Algorithms

1. Linear Models:

Linear Regression

Logistic Regression

2. Decision Trees and Ensembles:

Random Forest

Gradient Boosting (e.g., XGBoost)

3. Support Vector Machines (SVM):

Effective for high-dimensional data.

4. Neural Networks:

Composed of layers of interconnected nodes (deep learning).

5. Clustering Algorithms:

K-Means

DBSCAN

Model Evaluation Metrics

1. Classification Metrics:

Accuracy, Precision, Recall, F1 Score

Confusion Matrix

2. Regression Metrics:

Mean Squared Error (MSE)

R-squared

3. Other Metrics:

ROC-AUC for binary classification.

Common Challenges in ML

1. Overfitting:

The model learns noise instead of the signal.

Solution: Use regularization, cross-validation.

2. Underfitting:

The model fails to capture underlying patterns.

Solution: Use more complex models or features.

3. Bias-Variance Tradeoff:

Bias: Error due to overly simplistic assumptions.

Variance: Error due to model sensitivity to fluctuations in training data.

4. Data Imbalance:

When one class dominates in the dataset.

Solution: Use techniques like oversampling, undersampling, or weighted loss functions.

Feature Engineering

1. Feature Selection: Choose the most relevant features for the model.

2. Feature Extraction: Create new features from raw data.

3. Normalization/Scaling: Adjust data to a standard range for better model performance.

Common Tools and Libraries

1. Python Libraries:

Scikit-learn

TensorFlow

PyTorch

XGBoost

2. Visualization Tools:

Matplotlib

Seaborn

3. Data Processing:

Pandas

NumPy

Workflow of an ML Project

1. Define the Problem: Understand the objective.

2. Collect Data: Gather and preprocess data.

3. Exploratory Data Analysis (EDA): Understand data trends and patterns.

4. Feature Engineering: Prepare data for training.

5. Model Selection: Choose the best algorithm.

6. Training: Fit the model to the training data.

7. Evaluation: Test and tune the model on unseen data.

8. Deployment: Use the trained model in production.

Applications of Machine Learning

1. Healthcare: Disease prediction, personalized medicine.

2. Finance: Fraud detection, credit scoring.

3. Retail: Recommendation systems, demand forecasting.

4. Transportation: Autonomous vehicles, route optimization.

Machine learning continues to evolve, empowering numerous fields with intelligent systems capable of solving complex problems.

Advanced Concepts in Machine Learning

1. Feature Importance and Selection

Feature Importance: Measures how much each feature contributes to the model's prediction.

Example: Tree-based algorithms (e.g., Random Forest, XGBoost) provide feature importance scores.

Feature Selection Techniques:

Filter Methods: Based on statistical tests (e.g., correlation, chi-square).

Wrapper Methods: Use a search algorithm (e.g., forward selection, backward elimination).

Embedded Methods: Perform feature selection as part of the model training (e.g., Lasso regression).

2. Hyperparameter Tuning

Hyperparameters: Configuration settings for a model that are not learned during training (e.g., learning rate, number of layers).

Tuning Methods:

Grid Search: Exhaustively searches over a parameter grid.

Random Search: Randomly samples parameter combinations.

Bayesian Optimization: Uses probabilistic models to find the best parameters.

Tools: Scikit-learn, Optuna.

3. Cross-Validation

Splits data into multiple subsets to evaluate model performance.

Common Methods:

K-Fold Cross-Validation: Divides data into K parts and rotates the validation set.

Leave-One-Out Cross-Validation (LOOCV): Uses one data point as a test set.

Stratified K-Fold: Ensures class distribution remains constant in each fold.

4. Dimensionality Reduction

Reducing the number of features while retaining important information.

Techniques:

Principal Component Analysis (PCA): Projects data into lower dimensions.

t-SNE: Visualizes high-dimensional data in 2D or 3D.

Autoencoders: Neural network-based reduction methods.

5. Ensemble Learning

Combines multiple models to improve performance.

Techniques:

Bagging: Combines independent models (e.g., Random Forest).

Boosting: Sequentially corrects errors of weak models (e.g., AdaBoost, Gradient Boosting).

Stacking: Combines predictions of several models using a meta-model.

6. Neural Networks

Types:

Feedforward Neural Networks: Basic structure for supervised learning tasks.

Convolutional Neural Networks (CNNs): Used for image processing.

Recurrent Neural Networks (RNNs): Designed for sequential data like text or time series.

Activation Functions:

Sigmoid, ReLU, Tanh, Softmax.

7. Regularization

Prevents overfitting by adding penalties to the model.

Types:

L1 Regularization (Lasso): Shrinks coefficients to zero.

L2 Regularization (Ridge): Penalizes large coefficients.

Dropout: Randomly disables nodes in neural networks during training.

8. Optimization Algorithms

Methods to minimize loss functions during training.

Common Algorithms:

Gradient Descent: Adjusts weights iteratively.

Stochastic Gradient Descent (SGD): Uses a random subset of data for updates.

Adam Optimizer: Combines momentum and adaptive learning rates.

Emerging Topics in Machine Learning

1. Transfer Learning

Reuses pre-trained models for new tasks.

Common in deep learning (e.g., using models like BERT for NLP or ResNet for images).

2. Generative Models

Create new data instances.

Examples:

Generative Adversarial Networks (GANs): Used for image synthesis.

Variational Autoencoders (VAEs): Used for data compression and generation.

3. Federated Learning

Decentralized learning from distributed datasets.

Ensures data privacy by training models locally.

4. Explainable AI (XAI)

Focuses on making ML models interpretable.

Tools:

SHAP (Shapley Additive Explanations)

LIME (Local Interpretable Model-Agnostic Explanations)

5. Active Learning

Model queries the most informative samples for labeling.

Reduces the need for extensive labeled data.

6. Few-Shot Learning

Trains models with very few labeled examples.

Common in NLP and image recognition tasks.

Data Considerations in ML

1. Data Preprocessing

Handling missing values: Imputation or removal.

Encoding categorical variables: One-hot encoding, label encoding.

Scaling: Standardization (z-score), Min-Max scaling.

2. Data Augmentation

Expands dataset by applying transformations (e.g., rotation, flipping images).

3. Handling Class Imbalance

Resampling techniques:

Oversampling (e.g., SMOTE).

Undersampling.

Class weights: Assign more importance to minority classes.

4. Data Drift

Changes in data distribution over time.

Solution: Periodic model retraining.

Real-World Deployment Challenges

1. Scalability

Handling large datasets and high inference loads.

Tools: Apache Spark, distributed training frameworks.

2. Monitoring and Maintenance

Track model performance over time.

Detect data drift or degradation in accuracy.

3. Latency

Optimize models for low inference times.

Tools: TensorRT, ONNX Runtime.

4. Security and Privacy

Protect models from adversarial attacks.

Ensure compliance with regulations like GDPR.

Popular Machine Learning Frameworks

1. TensorFlow: Developed by Google, supports deep learning and ML.

2. PyTorch: Widely used for research and production.

3. Scikit-learn: Best for traditional ML models.

4. Keras: High-level API for neural networks.

5. ONNX: For interoperable ML models across frameworks.

Machine learning is dynamic and continuously advancing, offering endless opportunities for solving complex problems. Understanding these concepts is crucial for any practitioner or enthusiast aiming to create impactful solutions.

Further Machine Learning Concepts and Insights

1. Advanced Model Evaluation Techniques

Resampling Methods:

Bootstrap Sampling: Random sampling with replacement to estimate the variability of a model.

Nested Cross-Validation:

Ensures hyperparameter tuning does not leak information into the test set, providing unbiased model evaluation.

Learning Curves:

Visualize model performance over different training set sizes to diagnose underfitting or overfitting.

2. Probabilistic Models

Models that incorporate uncertainty in predictions.

Examples:

Naive Bayes: Assumes feature independence, used in text classification.

Bayesian Networks: Graphical models representing probabilistic relationships.

---

3. Optimization in ML

Batch Size:

Small batch sizes lead to noisier but faster updates.

Larger batches provide smoother updates but require more computation.

Learning Rate Scheduling:

Adjust learning rate during training (e.g., step decay, cosine annealing).

Momentum:

Accelerates gradient descent by using past gradients to smooth updates.

---

4. Handling Time Series Data

Special Considerations:

Time-dependent features (e.g., lag, rolling averages).

Stationarity: Ensure constant mean and variance over time.

Common Models:

ARIMA (Autoregressive Integrated Moving Average).

LSTMs (Long Short-Term Memory Networks): Capture sequential dependencies.

Prophet: Used for forecasting with trend and seasonality.

---

5. Natural Language Processing (NLP)

Key Techniques:

Tokenization: Splitting text into words, phrases, or sentences.

Word Embeddings: Vector representations of words (e.g., Word2Vec, GloVe).

Sequence Models: Use RNNs, GRUs, or Transformers for text sequences.

State-of-the-Art Architectures:

BERT (Bidirectional Encoder Representations from Transformers).

GPT (Generative Pre-trained Transformer).

---

6. Computer Vision

Image Processing Steps:

Data Augmentation: Enhance images (e.g., cropping, flipping).

Transfer Learning: Use pre-trained models like ResNet, VGG.

Advanced Techniques:

Object Detection: Identifying objects in an image (e.g., YOLO, Faster R-CNN).

Image Segmentation: Pixel-level categorization (e.g., U-Net).

---

7. Handling Big Data in ML

Scalable ML:

Distributed computing frameworks: Apache Spark MLlib, Dask.

Incremental Learning:

Train models on streaming data without retraining from scratch.

Dimensionality Reduction for Big Data:

Use distributed implementations of PCA or t-SNE.

---

8. Interpretability in ML

Importance in Real-World Applications:

Understanding how models make decisions is critical in healthcare, finance, and law.

Model-Agnostic Techniques:

Partial Dependence Plots (PDP): Show feature impact.

SHAP and LIME: Explain individual predictions.

Interpretable Models:

Decision Trees, Linear Regression.

---

9. Automated Machine Learning (AutoML)

Automates the end-to-end ML workflow.

Popular Tools:

H2O AutoML, AutoKeras, Google Cloud AutoML.

Benefits:

Reduces manual effort, allowing focus on business problems.

---

10. Handling Noisy Data

Noise refers to random variations or errors in data.

Techniques:

Data Cleaning: Removing or correcting erroneous entries.

Robust Models: Algorithms like Random Forests and SVMs are less sensitive to noise.

---

11. Cost-Sensitive Learning

Focuses on minimizing cost associated with errors, especially in imbalanced datasets.

Examples:

Assign higher penalties to misclassifying minority class examples.

---

12. Continual and Online Learning

Models adapt to new data without retraining from scratch.

Example:

Online Gradient Descent.

---

13. Generative Models for Creativity

Applications:

Music composition, image synthesis, and text generation.

Examples:

GANs for art generation.

GPT models for story writing.

---

14. Ethics in Machine Learning

Addressing biases in data and models.

Ensuring fairness, accountability, and transparency in decision-making.

Avoiding unintended consequences, such as discrimination.

---

15. Reinforcement Learning (RL)

Advanced Techniques:

Q-Learning: Finds optimal actions without a model of the environment.

Deep Reinforcement Learning: Combines RL with neural networks (e.g., DQN).

Applications:

Robotics, resource management, game-playing.

---

16. Zero-Shot and Few-Shot Learning

Zero-Shot Learning: Recognizes new classes without labeled examples by leveraging knowledge transfer.

Few-Shot Learning: Learns from a small number of labeled examples.

---

17. Model Deployment Techniques

Common Deployment Frameworks:

Flask, FastAPI for APIs.

TensorFlow Serving, ONNX Runtime for scalable model inference.

Edge Deployment:

Deploy models on edge devices for real-time applications.

---

18. Key Challenges in Machine Learning

Scalability: Handling large-scale data and computation.

Model Drift: Degradation of model performance due to changing data distributions.

Ethical Concerns: Ensuring unbiased and fair predictions.

Machine learning is a rapidly evolving field. Mastery of these advanced concepts ensures a deeper understanding and better problem-solving capabilities for real-world applications.

Machine Learning Interview Questions with Answers

1. What is the difference between supervised, unsupervised, and reinforcement learning?

Supervised Learning: The model learns from labeled data.
Example: Predicting house prices based on labeled datasets of houses.

Unsupervised Learning: The model identifies patterns in unlabeled data.
Example: Clustering customers based on purchasing behavior.

Reinforcement Learning: The model learns by interacting with the environment and receiving rewards or penalties.
Example: Training a robot to walk.

---

2. What is overfitting, and how can it be prevented?

Answer:
Overfitting occurs when a model performs well on training

data but poorly on unseen data.

Prevention Methods:

Regularization (L1 or L2 penalties).

Use more training data.

Reduce model complexity (e.g., fewer layers or nodes).

Cross-validation to tune hyperparameters.

Dropout for neural networks.

---

3. What is the difference between bias and variance?

Answer:

Bias: Error due to overly simplistic assumptions in the model (underfitting).

Variance: Error due to sensitivity to small fluctuations in training data (overfitting).

Trade-off: Achieving a balance minimizes total error.

---

4. What is the difference between bagging and boosting?

Answer:

Bagging: Combines predictions from multiple independent models to reduce variance. Example: Random Forest.

Boosting: Sequentially trains models, where each corrects the errors of the previous. Example: Gradient Boosting.

---

5. What is cross-validation, and why is it used?

Answer:
Cross-validation splits data into training and testing sets multiple times to evaluate model performance.

Types: K-fold, Stratified K-fold.

Purpose: Ensures the model generalizes well on unseen data.

---

6. What are precision, recall, and F1 score?

Answer:

Precision: Proportion of true positives among predicted positives.

Recall: Proportion of true positives among actual positives.

F1 Score: Harmonic mean of precision and recall.
Formula:

F1 = 2 * (Precision * Recall) / (Precision + Recall)

---

7. How do you handle imbalanced datasets?

Answer:

Resampling techniques (over-sampling minority or under-sampling majority).

Use of algorithms like SMOTE.

Penalized models (e.g., class-weight adjustments in Logistic Regression).

Use evaluation metrics like ROC-AUC or F1 score instead of accuracy.

---

8. What is the curse of dimensionality?

Answer:
As dimensionality increases, the data becomes sparse, making it harder for models to find patterns.

Solutions:

Dimensionality reduction (e.g., PCA, t-SNE).

Feature selection based on importance.

---

9. Explain Regularization and its types.

Answer:
Regularization prevents overfitting by adding a penalty to the loss function.

L1 Regularization (Lasso): Shrinks coefficients to zero (feature selection).

L2 Regularization (Ridge): Shrinks coefficients without making them zero.

---

10. What is a confusion matrix?

Answer:
A table summarizing prediction results:

True Positive (TP): Correct positive predictions.

True Negative (TN): Correct negative predictions.

False Positive (FP): Incorrectly predicted as positive.

False Negative (FN): Missed positive predictions.

---

11. What is gradient descent?

Answer:
Gradient descent optimizes the model by minimizing the loss

function.

Variants:

Batch Gradient Descent.

Stochastic Gradient Descent (SGD).

Mini-batch Gradient Descent.

---

12. What is the difference between softmax and sigmoid functions?

Answer:

Sigmoid: Outputs values between 0 and 1; used for binary classification.

Softmax: Outputs probabilities for multiple classes (sums to 1); used for multi-class classification.

---

13. What are eigenvalues and eigenvectors?

Answer:
Eigenvalues and eigenvectors are used in PCA for dimensionality reduction.

Eigenvectors determine the direction of new axes.

Eigenvalues measure the magnitude of variance along these axes.

---

14. How does a Random Forest work?

Answer:
Random Forest combines predictions from multiple decision trees:

Bagging is used for creating independent trees.

Final output is determined by majority voting (classification) or averaging (regression).

---

15. What is the difference between SVM and Logistic Regression?

Answer:

Logistic Regression: Linear model for binary classification; predicts probabilities.

SVM: Finds a hyperplane to maximize the margin between classes. Works well with high-dimensional data.

---

16. What is the role of activation functions in neural networks?

Answer:
Activation functions introduce non-linearity, enabling the model to learn complex patterns.

Common Types:

ReLU: Avoids vanishing gradients.

Sigmoid: For binary outputs.

Softmax: For multi-class outputs.

---

17. What are CNNs, and how do they work?

Answer:
Convolutional Neural Networks (CNNs) are designed for image data.

Use convolutional layers to detect features (e.g., edges, textures).

Pooling layers reduce dimensionality while retaining important features.

---

18. What is the difference between batch normalization and dropout?

Answer:

Batch Normalization: Normalizes inputs to each layer, speeding up training and reducing overfitting.

Dropout: Randomly deactivates neurons during training to prevent overfitting.

---

19. How do you select the best model for a problem?

Answer:

Evaluate using metrics like accuracy, precision, recall, F1 score, or ROC-AUC.

Use cross-validation for robust performance estimation.

Consider interpretability and computational efficiency.

---

20. What is transfer learning?

Answer:
Transfer learning uses a pre-trained model on a new task.

Example: Using a model trained on ImageNet for classifying medical images.

Benefits: Requires less data and reduces training time.

------

21. What is the difference between a generative and a discriminative model?

Answer:

Generative Models: Learn the joint probability distribution and generate data.
Examples: Naive Bayes, GANs (Generative Adversarial Networks).

Discriminative Models: Learn the decision boundary directly by modeling .
Examples: Logistic Regression, SVM.

---

22. What is the vanishing gradient problem? How can it be solved?

Answer:

Vanishing Gradient Problem: Gradients become very small during backpropagation, especially in deep networks, leading to slow or no learning.

Solutions:

Use activation functions like ReLU or Leaky ReLU.

Use techniques like batch normalization.

Initialize weights properly (e.g., Xavier or He initialization).

Use advanced architectures like LSTMs for sequential data.

---

23. What is a hyperparameter? How is it different from a model parameter?

Answer:

Model Parameters: Learned during training (e.g., weights, biases in neural networks).

Hyperparameters: Set before training and control the model's behavior (e.g., learning rate, number of layers, regularization strength).

Tuning Hyperparameters:

Grid Search.

Random Search.

Bayesian Optimization.

---

24. What is PCA, and why is it used?

Answer:
Principal Component Analysis (PCA) is a dimensionality reduction technique that projects data onto principal components (orthogonal axes).

Why Use PCA?:

Reduce dimensionality while retaining most variance.

Improve computational efficiency.

Handle multicollinearity.

---

25. What is the difference between Gini Impurity and Entropy in decision trees?

Answer:

Gini Impurity: Measures the probability of misclassification if a data point is randomly classified.
Formula:

$$Gini = 1 - \Sigma(p_i^2)$$

Entropy: Measures the level of uncertainty or randomness in the data.
Formula:

$$Entropy = -\Sigma(p_i * \log2(p_i))$$

Gini is faster to compute, while entropy can be more informative.

---

26. Explain the working of K-Means clustering.

Answer:
K-Means is an unsupervised algorithm that partitions data into k clusters:

1. Initialize centroids randomly.

2. Assign each point to the nearest centroid.

3. Update centroids by averaging points in each cluster.

4. Repeat until convergence.

Evaluation: Use metrics like inertia (sum of squared distances) or silhouette score.

---

27. What is the difference between L1 and L2 regularization?

Answer:

L1 Regularization (Lasso): Shrinks coefficients to zero, enabling feature selection.
Regularization term: .

L2 Regularization (Ridge): Shrinks coefficients without making them zero, reducing overfitting.
Regularization term: .

---

28. What is the difference between epochs, batches, and iterations?

Answer:

Epoch: One complete pass through the entire training dataset.

Batch: Subset of the training data used to compute gradients in one iteration.

Iteration: One update of model parameters using a batch.

---

29. How does an RNN differ from a traditional neural network?

Answer:
Recurrent Neural Networks (RNNs) are designed to handle sequential data.

They have feedback loops that allow information to persist.

Applications: Time-series forecasting, language modeling.

Challenges:

Suffer from vanishing gradient problems in long sequences.

Solutions: Use LSTMs or GRUs.

---

30. What is an autoencoder, and how is it used?

Answer:
Autoencoders are neural networks used for unsupervised learning.

The encoder compresses data into a lower-dimensional representation.

The decoder reconstructs the data from this representation.

Applications: Anomaly detection, data denoising, dimensionality reduction.

---

31. What is the difference between accuracy and ROC-AUC?

Answer:

Accuracy: Measures the percentage of correct predictions but is misleading for imbalanced datasets.

ROC-AUC (Receiver Operating Characteristic - Area Under Curve): Evaluates a classifier's ability to distinguish between classes across all thresholds.

---

32. How do Gradient Boosting Machines (GBMs) work?

Answer:
GBMs build models sequentially, where each model corrects the errors of the previous ones.

Use loss functions like Mean Squared Error for regression or Log Loss for classification.

Popular implementations: XGBoost, LightGBM, CatBoost.

---

33. What is the difference between batch gradient descent and stochastic gradient descent (SGD)?

Answer:

Batch Gradient Descent: Computes gradients using the entire dataset; slower but stable.

SGD: Computes gradients using one sample at a time; faster but noisier.

Mini-batch Gradient Descent: Balances both by using small batches of data.

---

34. Explain the role of embedding layers in NLP models.

Answer:
Embedding layers convert categorical data (e.g., words) into dense, continuous vectors.

Capture semantic relationships between words.

Example: Word2Vec, GloVe, Transformers.

---

35. What is the difference between ensemble learning and stacking?

Answer:

Ensemble Learning: Combines multiple models (e.g., bagging, boosting).

Stacking: Uses predictions of multiple models as inputs to a meta-model for final prediction.

---

36. How do GANs work?

Answer:
Generative Adversarial Networks consist of two components:

1. Generator: Creates fake data.

2. Discriminator: Distinguishes real data from fake data.

They train adversarially, improving each other iteratively.

---

37. How do you evaluate clustering performance?

Answer:
Metrics for clustering include:

Silhouette Score: Measures how well a point fits within its cluster.

Inertia: Sum of squared distances within clusters.

Adjusted Rand Index (ARI): Measures similarity with ground truth labels.

---

38. What are the differences between LSTMs and GRUs?

Answer:

LSTMs: Use separate forget and input gates for long sequences.

GRUs: Combine forget and input gates for simpler computation.

GRUs are faster, while LSTMs handle complex patterns better.

---

39. What is early stopping?

Answer:
Early stopping halts training when performance on validation data stops improving, preventing overfitting.

---

40. How do you handle missing data?

Answer:

Drop missing rows/columns: If missingness is minimal.

Imputation: Replace with mean, median, or predictive values.

Model Handling: Use algorithms like XGBoost that handle missing values.

----

41. What is the curse of dimensionality, and how do you address it?

Answer:

Curse of Dimensionality: As the number of dimensions increases, data points become sparse, making it hard for models to generalize.

Solutions:

Use dimensionality reduction techniques (PCA, t-SNE, autoencoders).

Feature selection methods.

Regularization techniques to penalize irrelevant features.

---

42. What is overfitting? How can you prevent it?

Answer:

Overfitting: A model performs well on the training data but poorly on unseen data.

Prevention Methods:

Use more training data.

Apply regularization (L1, L2).

Use dropout in neural networks.

Cross-validation.

Early stopping.

---

43. What is data normalization, and why is it important?

Answer:
Normalization scales features to a specific range, often [0, 1], or standardizes them to have zero mean and unit variance.

Important for algorithms like gradient descent and distance-based models (e.g., k-NN, SVM) for faster convergence and better performance.

---

44. How do decision trees handle missing values?

Answer:
Decision trees can handle missing values by:

Splitting based on surrogate splits (secondary criteria).

Imputing missing values.

Assigning samples with missing values to the branch with the majority of the samples.

---

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

Answer:

Parametric Models: Assume a fixed number of parameters (e.g., Logistic Regression, Linear Regression).

Faster, but less flexible with complex data.

Non-Parametric Models: Do not assume a fixed number of parameters (e.g., k-NN, Decision Trees).

Flexible but require more data.

---

46. Explain the difference between bagging and boosting.

Answer:

Bagging: Reduces variance by training multiple models on different subsets of data and averaging their outputs. Example: Random Forests.

Boosting: Reduces bias by sequentially training models, each correcting the errors of the previous one. Example: AdaBoost, Gradient Boosting.

---

47. What are feature importance techniques in machine learning?

Answer:

Permutation Importance: Measures the change in model accuracy when a feature is shuffled.

SHAP Values: Measures the contribution of each feature to a prediction.

Model Coefficients: In linear models, the magnitude of coefficients indicates feature importance.

Tree-based Methods: Use split gains in models like Random Forest or XGBoost.

---

48. What are kernel functions in SVMs?

Answer:
Kernel functions transform data into a higher-dimensional space where a linear decision boundary can be applied.
Common kernels:

Linear Kernel: No transformation.

Polynomial Kernel: Captures polynomial relationships.

RBF (Radial Basis Function) Kernel: Maps data into infinite-dimensional space.

---

49. Explain the trade-off between bias and variance.

Answer:

Bias: Error due to oversimplified assumptions; high bias leads to underfitting.

Variance: Error due to model complexity; high variance leads to overfitting.

The goal is to balance bias and variance to achieve low total error (Bias-Variance Tradeoff).

---

50. What is the difference between cross-entropy loss and mean squared error?

Answer:

Cross-Entropy Loss: Used for classification tasks; measures the difference between predicted probabilities and actual labels.

Mean Squared Error (MSE): Used for regression tasks; measures the average squared difference between predicted and actual values.

---

51. How does dropout work in neural networks?

Answer:
Dropout randomly sets a fraction of neurons to zero during training, forcing the network to learn more robust features.

Helps prevent overfitting.

---

52. What is a confusion matrix? Explain its components.

Answer:
A confusion matrix evaluates the performance of classification models.
Components:

True Positives (TP): Correctly classified positive cases.

True Negatives (TN): Correctly classified negative cases.

False Positives (FP): Incorrectly classified as positive.

False Negatives (FN): Incorrectly classified as negative.

---

53. What is transfer learning?

Answer:
Transfer learning involves reusing a pre-trained model on a new but related task.

Common in deep learning (e.g., using ResNet or BERT).

Benefits: Reduces training time and improves performance on smaller datasets.

---

54. What are the advantages of XGBoost over traditional gradient boosting?

Answer:

Regularization to prevent overfitting.

Faster due to parallel processing.

Handles missing values automatically.

Support for tree pruning and early stopping.

---

55. What are the assumptions of linear regression?

Answer:

1. Linearity: The relationship between features and the target is linear.

2. Independence: Residuals are independent.

3. Homoscedasticity: Residuals have constant variance.

4. Normality: Residuals are normally distributed.

5. No multicollinearity: Independent variables are not highly correlated.

---

56. Explain the difference between bag-of-words and TF-IDF.

Answer:

Bag-of-Words (BoW): Represents text by word frequency without considering importance.

TF-IDF (Term Frequency-Inverse Document Frequency): Adjusts word frequency by penalizing common words across documents, giving more weight to unique terms.

---

57. What is model drift? How do you handle it?

Answer:
Model drift occurs when the model's performance degrades due to changes in data distribution over time.

Handling Model Drift:

Regularly retrain the model.

Use monitoring tools to track performance.

Implement adaptive learning techniques.

---

58. How do you choose the number of clusters in K-Means?

Answer:

Elbow Method: Plot inertia vs. number of clusters and find the "elbow."

Silhouette Score: Evaluate how well data points fit within their cluster.

---

59. What is gradient clipping? Why is it used?

Answer:
Gradient clipping limits the magnitude of gradients during backpropagation to prevent exploding gradients, especially in RNNs or deep networks.

---

60. How do you handle class imbalance in a dataset?

Answer:

Resampling techniques (oversampling minority class, undersampling majority class).

Use class weights in the loss function.

Use specialized algorithms like SMOTE (Synthetic Minority Over-sampling Technique).

Use metrics like F1-score, precision-recall instead of accuracy.

Amar Sharma

AI Engineer

Follow me on LinkedIn for more
informative content 📝