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

Supervised Learning: The algorithm is trained on labeled data, meaning both input and the correct output are provided. The model learns to map inputs to outputs. Example: Predicting house prices using features like size and location.  
  
Unsupervised Learning: The algorithm works on unlabeled data. It tries to find hidden structures or patterns without explicit guidance. Example: Grouping customers into clusters based on purchasing behavior.

## 2. What is overfitting, and how can you prevent it?

Overfitting: When a model performs well on training data but poorly on unseen data because it has memorized noise or irrelevant details instead of learning general patterns.  
  
Prevention Methods:  
- Cross-validation (to test on multiple subsets)  
- Regularization (L1, L2 penalties)  
- Dropout in neural networks  
- Early stopping during training  
- Using more training data or data augmentation

## 3. What is the bias-variance tradeoff?

Bias: Error due to overly simple models (e.g., assuming linear when data is nonlinear). Leads to underfitting.  
  
Variance: Error due to overly complex models that learn noise in the training set. Leads to overfitting.  
  
Tradeoff: The challenge is to find a balance where the model is complex enough to capture important patterns but simple enough to generalize well.

## 4. What is gradient descent, and why is the learning rate important?

Gradient Descent: An optimization algorithm that updates model parameters by moving in the opposite direction of the gradient of the loss function to minimize errors.  
  
Learning Rate: Controls how big each step is during optimization.  
- Too high → the model may overshoot the optimal solution and fail to converge.  
- Too low → the model may converge very slowly or get stuck in local minima.

## 5. Explain L1 vs L2 regularization.

L1 Regularization (Lasso): Adds the sum of absolute values of coefficients as a penalty. This can shrink some coefficients to exactly zero, effectively performing feature selection.  
  
L2 Regularization (Ridge): Adds the sum of squared values of coefficients as a penalty. It reduces the size of coefficients but usually keeps all features, making it more stable.  
  
Use Case: L1 is good for sparse models, L2 is good when all features may contribute.

## 6. What is PCA, and why is it used?

Principal Component Analysis (PCA): A dimensionality reduction technique that transforms high-dimensional data into a smaller set of uncorrelated variables called principal components. These components capture the maximum variance in the data.  
  
Uses:  
- Reducing computational cost in ML models  
- Visualizing high-dimensional data in 2D or 3D  
- Removing noise while keeping most of the information

## 7. How does a Random Forest work?

A Random Forest is an ensemble of decision trees. Each tree is trained on a random subset of data and features.  
  
For prediction:  
- Classification: The forest predicts based on the majority vote of trees.  
- Regression: It predicts the average output from all trees.  
  
Advantages: Reduces overfitting compared to a single decision tree, handles missing values well, and works well on both categorical and numerical data.

## 8. What is the difference between CNNs and RNNs?

CNN (Convolutional Neural Networks):  
- Best for spatial data like images.  
- Uses filters (kernels) to detect patterns like edges, shapes, textures.  
- Strong at tasks like image recognition, object detection.  
  
RNN (Recurrent Neural Networks):  
- Designed for sequential data like text or time series.  
- Maintains a hidden state to remember previous information.  
- Good for tasks like language modeling, speech recognition, and sequence prediction.

## 9. What is the attention mechanism in deep learning?

The attention mechanism allows models to focus on the most relevant parts of the input sequence when making predictions.  
  
Instead of treating all inputs equally, it assigns different weights to different tokens or features.  
  
Example: In machine translation, when translating a word, the model can attend more to the relevant words in the source sentence.  
  
Importance: It improves context understanding in NLP and is the foundation of Transformer models like BERT and GPT.

## 10. What is model optimization and hyperparameter tuning?

Model Optimization: Improving performance by reducing errors, speeding up training, or making the model more efficient. This can include weight initialization, gradient clipping, and using better optimizers like Adam, RMSProp, or SGD with momentum.  
  
Hyperparameter Tuning: Process of finding the best configuration for hyperparameters (learning rate, batch size, number of layers, dropout rate, etc.).  
  
Methods:  
- Grid Search (testing all combinations)  
- Random Search (sampling random combinations)  
- Bayesian Optimization (guided search using probability)  
- Automated methods like Hyperband or Optuna  
  
Impact: Proper tuning can significantly improve accuracy, training stability, and model generalization.