Q1. Explain the difference between supervised, unsupervised, and reinforcement learning with examples.

Answer:

- **Supervised Learning**: The model is trained on labeled data, meaning both inputs and their corresponding outputs are known.
  - Example: Predicting house prices based on features (size, location, etc.), spam detection in emails.
- **Unsupervised Learning**: The model works with unlabeled data and tries to find hidden structures or patterns.
  - Example: Customer segmentation using clustering, dimensionality reduction with PCA.
- **Reinforcement Learning (RL)**: The model (agent) learns by interacting with an environment, receiving rewards or penalties for actions.
  - Example: Training a robot to walk, AlphaGo learning to play Go.

Q2. What is overfitting in machine learning? Discuss its causes and methods to prevent it.

Answer:

- **Overfitting** occurs when a model learns both the patterns and the noise of the training data, resulting in excellent training performance but poor generalization to new/unseen data.
- **Causes of Overfitting:**
  1. Model complexity is too high (e.g., too many parameters).
  2. Insufficient training data.
  3. Too many training epochs (in deep learning).

- **Methods to Prevent Overfitting:**
    1. Use more training data.
    2. Apply **regularization** (L1, L2).
    3. Reduce model complexity (prune decision trees, use fewer layers).
    4. Use **dropout** in neural networks.
    5. Apply **cross-validation** to evaluate model robustness.

Q3. Compare bagging and boosting in ensemble learning. Provide one real-world application for each.

Answer:

- **Bagging (Bootstrap Aggregating):**
    - Trains multiple models on random subsets of the data in parallel.
    - Final prediction is an average (for regression) or majority vote (for classification).
    - Reduces variance and prevents overfitting.
    - **Example:** Random Forest used in credit risk analysis.
- **Boosting:**
    - Models are trained sequentially, where each new model focuses on correcting errors of the previous one.
    - Produces a strong learner by combining weak learners.
    - Reduces bias and variance.
    - **Example:** XGBoost used in fraud detection systems.

Q4. What is the bias-variance tradeoff, and why is it important in machine learning?

Answer:

- **Bias**: Error introduced by assuming a simplified model that may not capture the true complexity of data. High bias → underfitting.
- **Variance**: Error introduced by sensitivity to small fluctuations in training data. High variance → overfitting.
- **Tradeoff**:
  - Increasing model complexity decreases bias but increases variance.
  - Decreasing model complexity reduces variance but increases bias.
- **Importance**: A good machine learning model must strike a balance between bias and variance to minimize total error and achieve good generalization.

Q5. Why is the F1-score often preferred over accuracy in classification tasks with imbalanced datasets? Explain with an example.

Answer:

- **Accuracy** = (Correct Predictions / Total Predictions). It works well when classes are balanced, but fails when classes are imbalanced.
- **Precision** = (True Positives / Predicted Positives)
- **Recall** = (True Positives / Actual Positives)
- **F1-score** = Harmonic mean of Precision and Recall.
- **Why F1-score is better:**
  - In imbalanced data, a model may predict all samples as the majority class and still achieve high accuracy.

- Example: In fraud detection, if 99% transactions are genuine and 1% fraudulent, predicting all as genuine gives 99% accuracy but 0% recall for frauds.
- F1-score balances **false positives** and **false negatives**, making it more suitable for imbalanced problems.