Part 1: Data Preprocessing & Preparation (25 Points)

1. (a) What GIGO Means for ML Performance:

GIGO (Garbage In, Garbage Out) means that poor quality data will result in poor model performance. Machine learning models rely on meaningful input data to learn patterns. If the input data is noisy, incomplete, or incorrect, the model's predictions will be unreliable.

(b) Three Common Real-World Data Issues:

1. **Missing Values:** ML models cannot process null values; they reduce learning quality.
2. **Outliers:** Can skew models like linear regression, leading to incorrect predictions.
3. **Inconsistent Formats:** Strings like "N/A", "null", and blanks must be standardized or cleaned.

2. Two Strategies for Handling Missing Data:

1. **Imputation (e.g., mean/median/mode):**
   - Suitable when missingness is small and random.
   - Drawback: Introduces bias; may mask real trends.
2. **Deletion (row or column):**
   - Useful when only a few rows/columns are affected.
   - Drawback: Reduces dataset size; may lose valuable info.

3. Importance of Feature Scaling:

- Some ML algorithms (like KNN, SVM, Gradient Descent models) are sensitive to feature magnitudes. Features with large ranges can dominate learning.
- **Sensitive Algorithm:** Logistic Regression
- **Not Sensitive:** Random Forest

Part 2: Model Training, Testing, & Overfitting (30 Points)

4. Role of Training, Validation, and Test Sets:

- **Training Set:** Used to teach the model.
- **Validation Set:** Used to tune hyperparameters and prevent overfitting.
- **Test Set:** Used for final evaluation on unseen data.

5. Overfitting:

(a) Overfit models perform extremely well on training data but poorly on unseen data due to memorization.

(b) A separate test set ensures unbiased evaluation of model generalization. It simulates new data.

6. What is a Loss Function:

- Measures the difference between predicted and actual outcomes.
- The training process tries to **minimize** it to improve model accuracy.

7. Feature Engineering:

- Creating new features from existing ones to improve prediction.
- **Example:** From 'Height' and 'Weight' create 'BMI' = Weight / Height^2

Part 3: Model Validation Techniques (25 Points)

8. Limitation of Single Hold-Out Validation:

- Model performance may vary depending on which data gets held out. Less robust estimate of performance.

9. K-Fold Cross-Validation:

(a) Splits the dataset into K parts. Trains on K-1 parts and validates on 1. Rotates K times.

- Solves hold-out issues by giving every sample a chance to be in validation.

(b) In 5-Fold CV, model is trained 5 times (once for each fold).

10. External Validation:

- Evaluates model on a completely **new and independent dataset**.
- Better for testing generalizability, avoids bias from training data influence.

11. Data Leakage:

- When information from test or validation sets is used during training.
- **Example:** Scaling all data before splitting.
- It falsely boosts performance and must be avoided.

Part 4: Model Deployment Concepts (20 Points)

12. Goal of Model Deployment:

- To make the trained model usable in real-world apps for generating predictions (e.g., APIs, dashboards).

13. Saving/Loading Trained Models:

- Essential to avoid retraining and ensure reproducibility.
- Enables fast deployment or sharing with teams/users.

14. Batch vs Real-Time Predictions:

- **Batch:** Predict churn for all customers once per day.
- **Real-time:** Predict if a user will click an ad **as they interact**.

15. "Works on My Machine" Problem:

- Code works on one machine but not another due to env differences.
- **Docker** solves this by creating a consistent containerized environment.
