COURSE NAME
FALL SEMESTER 20XX

INSTRUCTOR: DR. WENDY WRITER
no_reply@example.com

04 September 20XX

Part 1: Data Preprocessing & Preparation (25 Points)

1. The lecture emphasized ”Garbage In, Garbage Out (GIGO)” in the context of data preprocessing.

(a) Explain what GIGO means for machine learning model performance. (5 points)

- "Garbage In, Garbage Out (GIGO)" means that if the input data to a machine learning model is noisy, incorrect, or poorly formatted, then the model's output will also be unreliable. No matter how advanced the algorithm, it cannot learn meaningful patterns from flawed input.

(b) List three common issues found in real-world data that require preprocessing and briefly explain why each can be problematic for an ML model. (6 points)

- Three common issues:

- **Missing values**: ML algorithms generally can't handle missing data directly, which can result in biased predictions or training errors.
- **Outliers**: These can distort model performance, especially in algorithms like linear regression, which are sensitive to extreme values.
- **Categorical variables**: Algorithms need numerical inputs. Unencoded categories (e.g., 'Red', 'Green') must be converted using label encoding or one-hot encoding.

2. Describe two common strategies for handling missing data. For each strategy, mention a scenario where it might be appropriate and a potential drawback. (8 points)

- **Mean/Median Imputation**: Replacing missing values with the column's mean/median is useful for numerical data. It works well if the data is missing at random. Drawback: may reduce variability or introduce bias.
- **Deletion**: Removing rows with missing values can be appropriate if only a small percentage is missing. Drawback: reduces dataset size and may bias results if the missingness isn't random.

3. Why is feature scaling (e.g., Normalization or Standardization) important for certain machine learning algorithms? Name one algorithm that is sensitive to feature scales and one that is generally not. (6 points)

- **Feature Scaling:** Scaling ensures that features contribute equally to the model. Without scaling, features with larger numerical ranges dominate.
- **Sensitive algorithm**: K-Nearest Neighbors (KNN)
- **Not sensitive**: Decision Trees

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

4. Explain the primary purpose of splitting your dataset into Training, Validation, and Test sets. What is the role of each set? (9 points)

- **Training set**: Used to train the model.
- **Validation set**: Used to tune hyperparameters and evaluate model performance during development.
- **Test set**: Used once at the end to evaluate final performance. Ensures an unbiased assessment.

5. What is ”overfitting” in machine learning?

(a) Describe what happens to the model’s performance on the training data and on unseen (test) data when a model overfits. (4 points)

- When a model overfits, it performs well on the training data but poorly on unseen test data because it has memorized specific patterns or noise instead of learning generalizable features.

(b) Why is using a separate test set crucial for detecting overfitting? (4 points)

- A separate test set is important because it provides an unbiased way to detect whether the model has learned real patterns or just memorized training examples.

6. What is a ”loss function” in the context of supervised model training? Why does the model training process aim to minimize it? (6 points)

- A loss function measures how far off the model's predictions are from the true values. The training process attempts to minimize this value so the model becomes better at predicting outcomes.

7. Briefly explain the concept of ”Feature Engineering.” Provide one example of how creating a new feature from existing ones could potentially improve a model’s predictive power. (7 points)

- Creating new features from existing ones to enhance model performance. Example: Combining height and weight to calculate BMI, which may better correlate with health outcomes than either alone.

Part 3: Model Validation Techniques (25 Points)

8. What is the main limitation of using a single hold-out validation set for evaluating model performance and tuning hyperparameters? (5 points)

- A single hold-out set may not represent the overall data distribution well. It can lead to misleading performance evaluations and poor hyperparameter tuning.

9. Describe the process of K-Fold Cross-Validation.

(a) How does K-Fold Cross-Validation address the limitation mentioned in the previous question? (5 points)

- K-Fold splits the dataset into K parts. The model is trained on K-1 parts and validated on the remaining one. This process repeats K times, ensuring each fold is used once for validation. This provides a more reliable estimate of model performance.

(b) If you perform 5-Fold Cross-Validation, how many times is a model trained? (3 points)

- For 5-Fold CV, the model is trained 5 times (once per fold).

10. What is ”External Validation” in the context of machine learning? Why is it considered a more robust test of a model’s generalizability compared to internal validation techniques like cross-validation on the original dataset? (7 points)

- External validation involves testing the model on a completely new dataset (from a different source or time period). It ensures the model generalizes beyond the data it was trained and validated on internally.

11. What is meant by ”data leakage” in a machine learning pipeline, particularly in the context of preprocessing (e.g., scaling) and data splitting? Why is it important to avoid it? (5 points)

- This occurs when information from outside the training dataset (e.g., target variable or test set stats) is used to train the model. For example, applying scaling before splitting the data can cause leakage. It can falsely inflate model performance and lead to failure in real-world applications.

Part 4: Model Deployment Concepts (20 Points)

12. Even if you are not coding the deployment yourself, it’s important to understand the concepts. What is the primary goal of ”model deployment”? (4 points)

- To integrate a trained ML model into a production environment so it can make predictions on new, real-world data and add value.

13. The TF session mentioned saving and loading trained models (e.g., using pickle). Why is this step essential before a model can be deployed or shared? (5 points)

- This step allows us to store the trained model so it doesn't need to be retrained every time. It ensures the same model version can be reused, shared, or deployed consistently.

14. Imagine you have built a fantastic model to predict whether a customer will click on an online advertisement. Describe a scenario where ”Batch Predictions” would be a suitable deployment pattern, and another scenario where ”Real-time Predictions” (via an API) would be more appropriate. (6 points)

- Batch: A retail company runs predictions every night on all customers to decide who should receive promotional emails. Predictions aren’t time-sensitive.
- Real-time: An online platform uses an API to predict whether a user will click an ad based on current activity. Response needs to be instant.

15. What is the ”Works on My Machine” problem in software and ML development? Briefly explain how a technology like Docker (even if we don’t code it) aims to solve this. (5 points)

- This refers to code or models working on a developer’s machine but failing elsewhere due to different environments. Docker solves this by packaging the code, dependencies, and environment settings into a container that runs identically across systems.