Practical Questions (Code + Output)

Submission Guidelines

1. Code Submissions: Complete Python files with comprehensive comments, explaining each step. Ensure scripts are tested and functional before submission.
2. Documentation: Professional formatting with proper citations and references. Include explanation of design decisions, code structure, and execution steps.
3. Project: Submit a GitHub repository containing the complete codebase, documentation, and example datasets (if applicable).
4. All work must be original with proper attribution.
5. Code must be functional and tested.
6. Professional presentation quality is expected.
7. Adhere to coding best practices and maintain readability.

Questions

1. Hugging Face Model Integration Script
   - Write a Python script that:
     - Loads a Hugging Face model and tokenizer
     - Performs inference on sample input
     - Includes robust error handling and logging
     - Demonstrates performance optimization (batching, GPU usage if available)

2. Prompt Optimization Framework
   - Develop a framework that automatically adapts prompts for:
     - Summarization
     - Text classification
     - Creative text generation
     - The framework should detect the task type and adjust prompt style, length, and structure accordingly.

3. Automated Data Exploration with pandas
   - Create a reusable function or class that:
     - Generates statistical summaries for each column
     - Detects and reports missing values, duplicates, and outliers
     - Recommends preprocessing steps
     - Outputs a well-structured report

4. Multi-Panel Visualization Dashboard
   - Build a dashboard using matplotlib and seaborn that:
     - Supports different data types

        - Automatically selects suitable chart types
        - Displays multiple panels for comparison
        - Includes legends, titles, and annotations for clarity

5. NLTK-Based Text Analysis System
    - Develop a script that:
        - Preprocesses a corpus (tokenization, stopword removal, lemmatization)
        - Analyzes sentiment trends over time
        - Detects frequently occurring topics or keywords
        - Calculates basic readability metrics

6. pandas Data Transformation Pipeline
    - Implement a configurable pipeline that:
        - Handles missing values with various strategies
        - Detects and removes outliers
        - Performs feature scaling (normalization, standardization)
        - Logs each transformation step

7. Visualization Automation Tool
    - Create a script that:
        - Accepts any dataset as input
        - Generates a complete exploratory data analysis (EDA) report
        - Automatically selects visualization types
        - Includes statistical summaries alongside plots

8. AI API Performance Benchmarking
    - Design a benchmarking suite that:
        - Compares at least two AI APIs (e.g., Hugging Face, Ollama)