Project Overview

This project simulates the tasks you will encounter as a Machine Learning Engineer, specifically focusing on processing unstructured technical documentation from PDFs into structured, actionable data. Your goal is to develop an ML-based system that extracts procedural steps, identifies modules, and orchestrates decision flows without relying on third-party APIs (fully on-premise).

Objective

Develop a fully-contained, modular ML pipeline that:

1. Converts technical documentation (provided as PDFs) into structured data formats (JSON).
2. Identifies logical modules, procedural steps, and actionable insights from these documents.
3. Utilizes large language models (open-source LLMs such as GPT-J or Llama) in conjunction with rule-based decision systems.
4. Operates entirely on-premise, ensuring no dependency on external cloud services or APIs.
5. Includes a fallback mechanism or decision logic to handle ambiguous cases.

Requirements

1. **Data Preparation and Processing**
    - Extract and preprocess text from provided PDF documents.
    - Implement techniques for text normalization, cleaning, and segmentation.
2. **Structured Data Extraction**
    - Use NLP techniques (such as named-entity recognition, dependency parsing, semantic search, etc.) to identify and structure relevant information.
    - Output structured data clearly defining procedural steps and modules.

3. **Agentic Workflow Integration**
    - Design an AI agent capable of multi-step reasoning, selecting appropriate ML models, and integrating outputs into structured, actionable steps.
    - Ensure the agent can handle rule-based triggers for selecting models or executing fallback logic when confidence is low.
4. **Deployment and Security**
    - Provide a clear deployment strategy for running the pipeline entirely on-premise (Docker containerization is preferred).
    - Demonstrate considerations for data security, reliability, and scalability.
5. **Evaluation and Testing**
    - Include metrics to evaluate extraction accuracy, precision/recall for procedural steps, and robustness in handling edge cases.
    - Develop test cases demonstrating system functionality and limitations clearly.

Deliverables

- Git repository with clean, maintainable, and documented code.
- Dockerfile or detailed deployment instructions for on-premise operation.
- Detailed README documenting your approach, design decisions, system architecture, and any assumptions made.
- Structured output example (JSON format) for provided sample PDF documents.
- Brief report summarizing performance metrics, challenges encountered, and potential improvements.

Evaluation Criteria

- Clarity and completeness of structured outputs.
- Robustness and maintainability of the ML pipeline.
- Ability to handle ambiguous or noisy input data effectively.
- Security and practicality of the proposed deployment strategy.