GenAI Engineer Exercise

LLM Evaluation & Prompt Engineering Home Assignment

Cybersecurity Threat Classification Challenge

Time Allocation: 3-4 hours

Submission Format: Python notebook (.ipynb) or Python script with documentation

Background

You've been provided with a dataset of 132 cybersecurity threat descriptions that need to be classified into 10 MITRE ATT&CK categories (https://attack.mitre.org/). Your task is to evaluate how well different Large Language Models perform this classification task, optimize their performance through prompt engineering, and create a framework for systematic evaluation.

If a threat cannot be confidently classified into any of the 10 provided categories, the model should return "Other" as the classification.

Provided Materials

Google Sheets document with 2 sheets:

1. **Sheet 1: Threats** - 132 threat descriptions and labels
2. **Sheet 2: Categories** - 10 MITRE tactics categories with descriptions

Assignment Tasks

Part 1: Initial LLM Evaluation

Create a Python script using **LangChain** (or similar framework like LlamaIndex, Instructor) that:

1. **Implements a classification pipeline** with structured output
    - Use Pydantic models or similar for structured responses
    - Ensure outputs include both classification and reasoning
    - Include safeguards against prompt injection attempts in threat descriptions

- Support single-input testing (ability to classify a new threat description on demand)

2. **Tests at least 2 different models**:
   - OpenAI Via API
   - Claude Haiku or Sonnet (via AWS Bedrock or Anthropic API)
   - Open-source alternative (e.g., Mistral, Llama)
3. **Captures key metrics**:
   - Accuracy (overall and per-category)
   - Response time per classification
   - Token usage and estimated cost

Deliverable: Basic evaluation script showing baseline performance with structured outputs

Part 2: Prompt Engineering & Optimization

Experiment with different prompt strategies to improve performance:

Suggested approaches to try:

- Zero-shot vs. few-shot learning (include examples)
- Chain-of-thought reasoning
- Breaking down the decision process into steps
- Role-playing ("You are a cybersecurity expert...")
- Structured reasoning templates

Deliverable: Comparison showing performance of different prompt strategies

Part 3: Evaluation Framework

Build an evaluation system that:

1. **Generates a classification report**:
   - Confusion matrix
   - Precision, recall, F1 per category

- Identification of systematic errors

2. **Provides cost-performance analysis**:
    - Accuracy vs. API costs comparison
    - Cost per classification across models
    - Recommendations for cost-effective deployment

Deliverable: Evaluation code that outputs classification report and cost analysis