Thought Process of My Code

**Research Topic:** “Jailbreak-as-a-Service” Attacks: Measuring and Mitigating Crowdsourced LLM Exploits.

**Core Components:**

**JailbreakFeatureExtractor:**

* Extracts 8 key characteristics from prompts that may indicate jailbreak attempts
* Uses semantic similarity (SentenceTransformers) and pattern matching
* Features include role impersonation, hypothetical language, obfuscation, etc.

**JailbreakClassifierV2:**

* Improved classifier using weighted feature scores
* Categorizes prompts into 5 main threat categories with thresholds
* Combines multiple features to calculate category scores

**ActiveLearner:**

* Implements active learning by identifying uncertain classifications
* Stores borderline cases for human review
* Can retrain the system based on human feedback

**AdversarialTester:**

* Tests classifier robustness by applying perturbations to prompts
* Checks if modified prompts can bypass detection

**JailbreakClassifier (original):**

* Rule-based classifier using regex patterns
* Comprehensive taxonomy of jailbreak techniques
* Multi-level pattern matching

**JailbreakPromptTester:**

* Main class that orchestrates the testing process
* Loads prompts, runs tests, and saves results
* Handles model interactions and human feedback processing

**Key Features:**

* Multi-Method Detection:
* Combines rule-based (regex) and machine learning (feature-based) approaches
* Uses semantic similarity to known jailbreak patterns
* Scores prompts across multiple dimensions

**Active Learning:**

* Identifies uncertain classifications for human review
* Incorporates human feedback to improve detection
* Maintains a master log of reviewed prompts

**Comprehensive Taxonomy:**

* Covers 6 main categories of jailbreak techniques
* Includes 15+ specific techniques with patterns and examples
* Handles both obvious and subtle attack vectors

**Robust Testing Framework:**

* Tests prompts against actual AI models
* Generates detailed reports with classifications
* Handles encoding issues and large prompt sets

**Usage Flow:**

* Initialize the tester with an AI model (default: GPT-4)
* Load prompts from CSV (with filtering options)

**Run tests which:**

* Classify prompts using both methods
* Get model responses
* Identify uncertain cases for review
* Process human feedback to improve the classifier
* Generate reports and save results

**Improvements Over Basic Classifiers:**

* Combines multiple detection approaches for better accuracy
* Adaptive learning from human feedback
* Detailed taxonomy of attack techniques
* Confidence scoring and uncertainty detection
* Robust handling of prompt variations and obfuscation