Text-Based Sentiment Analysis Using Python

Objective:

The project implements a simple, rule-based sentiment analysis system for Amazon cell phone and accessory reviews using only Python's built-in libraries. The goal is to classify reviews as positive, negative, or neutral, providing foundational experience in data preprocessing, thematic analysis, and sentiment classification.

Explanation:

- **Data Loading & Preprocessing:**
  The dataset is loaded from a JSON file and parsed into Python data structures. Only essential fields (reviewerID, asin, overall rating, and reviewText) are retained to streamline analysis. Text preprocessing removes punctuation and stop words using a custom stop word list, without relying on external libraries.
- **Thematic Analysis:**
  Reviews are split into positive and negative categories based on their ratings. Basic frequency analysis identifies key words and phrases associated with each sentiment. This forms the basis for constructing a sentiment lexicon and assigning word weights.
- **Rule Based Sentiment Analysis:**
  A scoring system is defined where words frequently appearing in positive reviews are given positive weights, and those common in negative reviews receive negative weights. Each review's sentiment score is computed by summing the weights of its words. Thresholds are set to classify reviews as positive, negative, or neutral based on their cumulative scores.

Key Features:

- No external libraries (like pandas or NumPy) are used; only built-in modules such as json, re, and statistics are allowed.
- Manual stop word removal and basic text cleaning are performed.
- Rule-based approach ensures explainable and transparent sentiment classification.
- Results are saved in a human-readable format for further analysis or validation.

Conclusion:

This project demonstrates the fundamentals of text data processing and sentiment analysis using Python's core capabilities. It provides a strong foundation for more advanced natural language processing work.