**Documentation for Sentiment and Risk Analysis Script**

This script is designed to fetch coffee-related news, perform sentiment and risk analysis, and output results to CSV files. It specifically focuses on detecting supply chain risks related to the coffee industry.

**Setup**

**Dependencies**

Ensure the following Python libraries are installed:

* time
* openai
* pandas
* requests
* transformers
* datetime

You can install missing dependencies with pip install <library\_name>.

**Workflow**

**1. API Keys and Configuration**

* openai.api\_key: Set your OpenAI API key to use GPT models for risk analysis.
* NEWS\_API\_KEY: Replace this with your News API key to fetch news articles.

**2. Keywords**

The script uses an expanded list of **risk-related keywords** to detect disruptions in the coffee supply chain. These keywords cover a variety of topics, such as weather events, geopolitical issues, labor strikes, and transportation delays.

**Functions**

**1. fetch\_coffee\_news()**

Fetches coffee-related news from the News API.

* **Query:** Includes keywords like coffee production, coffee prices, and filters for specific coffee-producing countries.
* **Returns:** A Pandas DataFrame containing:
  + Source
  + Title
  + Description
  + Content
  + Published At
  + URL
* Saves the fetched news as coffee\_related\_news\_data.csv.

**2. analyze\_sentiment(text)**

Uses Meta's LLaMA model for sentiment analysis.

* **Input:** Text content from news articles.
* **Output:**
  + Sentiment classification (e.g., Positive, Neutral, Negative).
  + Sentiment score (a value indicating sentiment intensity).

**3. analyze\_risk(text)**

Uses OpenAI GPT to analyze supply chain risks in the text.

* **Input:** Text content from news articles.
* **Output:**
  + Detailed risk analysis.
  + Risk score (0 to 1).
  + Boolean indicating whether any risk-related keywords were detected.

**4. categorize\_risk\_level(risk\_score)**

Categorizes the risk level based on the risk score.

* **High Risk:** Score ≥ 0.7
* **Medium Risk:** Score between 0.4 and 0.7
* **Low Risk:** Score < 0.4

**5. combine\_sentiment\_and\_risk(sentiment\_score, risk\_score)**

Combines sentiment and risk scores into a single decision matrix. Examples:

* "High Risk - Negative Sentiment"
* "Moderate Risk - Negative Sentiment"
* "Low Risk - Neutral Sentiment"

**6. process\_and\_analyze\_sentiment\_and\_risk(news\_df)**

Processes fetched news articles and performs sentiment and risk analysis on each article.

* **Input:** DataFrame of news articles.
* **Output:**
  + A DataFrame containing:
    - Sentiment classification and score.
    - Risk analysis, score, and level.
    - Combined analysis results.
  + Saves results to sentiment\_and\_risk\_analysis\_results.csv.
  + Saves risky articles to risky\_articles.csv.

**Main Function**

**Steps:**

1. Fetch coffee-related news using fetch\_coffee\_news().
2. Save fetched news to coffee\_related\_news\_data.csv.
3. Analyze sentiment and risks using process\_and\_analyze\_sentiment\_and\_risk().
4. Save:
   * Full results to sentiment\_and\_risk\_analysis\_results.csv.
   * Risky articles to risky\_articles.csv.
5. Print a summary of risky articles.

**Outputs**

1. **coffee\_related\_news\_data.csv**
   * Contains the fetched news articles with source, title, description, content, etc.
2. **sentiment\_and\_risk\_analysis\_results.csv**
   * Provides detailed sentiment and risk analysis for each article.
3. **risky\_articles.csv**
   * Includes only the articles with a risk score > 0.5 or containing risk-related keywords.

**Error Handling**

* Handles network issues during API requests.
* Retries on OpenAI rate limit errors (waits 20 seconds).
* Logs errors encountered during sentiment or risk analysis.

**Usage**

1. Replace API keys for OpenAI and News API at the beginning of the script.
2. Run the script using python <script\_name>.py.
3. Results will be saved in the current directory as CSV files