**Code Report: Analyzing News Articles for Cement Sector**

**Objective:** The code processes news articles related to the cement sector, evaluates their relevance based on associated tags, and generates a relevance score. The relevance score is derived using a generative AI model and, if necessary, a fallback sentiment analysis approach.

**Components and Workflow:**

1. **Imports and Configuration:**
   * Libraries: pandas, google.generativeai, textblob, and google.colab.
   * Configures the Gemini generative AI model with an API key for accessing Google’s AI services.
2. **File Upload:**
   * Uploads a CSV file containing news articles and associated tags using files.upload() from Google Colab.
3. **Data Loading:**
   * Loads the uploaded CSV file into a pandas DataFrame.
   * Displays the DataFrame’s columns and the first few rows to understand its structure.
4. **Gemini Model Setup:**
   * Initializes the Gemini generative AI model with a specific configuration for generating content.
   * Defines a system prompt guiding the model to evaluate news articles for their relevance to the cement sector.
5. **Content Analysis Function (analyze\_content):**
   * Constructs a prompt for the generative AI model to assign a relevance score between -10 and 10 based on the news content and tags.
   * Includes retry logic (3 attempts) in case of model response issues.
   * Fallback to sentiment analysis using TextBlob if the AI model fails to provide a valid score.
6. **Batch Processing:**
   * Processes the DataFrame in batches (size defined by BATCH\_SIZE).
   * Starts processing from a specified batch (start\_batch), which is set to begin after batch 15.
   * For each batch, evaluates news articles and tags, computes relevance scores, and stores the results.
   * Includes a pause (60 seconds) between batches to manage API rate limits and prevent overloading.
7. **Results Compilation:**
   * Collects the results of relevance scoring for each news article.
   * Creates a new DataFrame (enhanced\_df) with the processed results.
8. **Saving and Downloading:**
   * Saves the enhanced dataset as a CSV file named "enhanced\_cement\_sector\_news\_events.csv".
   * Downloads the CSV file to the local environment using files.download().