

Assignment: The Disaster Evolution & Response Pipeline

Objective: Build an automated pipeline to scrape, analyze, and compare a **historical** disaster with a **current** disaster of the same type to evaluate changes in "Media Saturation" and "Humanitarian Impact."

Task 1: Target Selection & Scraping

Students must build a robust Python scraper to extract deep-tier data from the [Global Disaster Alert and Coordination System \(GDACS\)](#).

- **1.1 Data Selection:** Choose **one** disaster category (Earthquakes, Tsunamis, Floods, Cyclones, Volcanoes, Droughts, or Forest Fires) and choose two different regions.
 - **1.2 The "Source of Truth" Scraper:** Identify two specific events from those two regions:
 - **Event A:** A "Historical" event (at least 5–8 years old).
 - **Event B:** A "Recent" event (within the last 12 months).
 - **1.3 Deep-Tab Extraction:** Your scraper must navigate to the specific report pages and extract:
 - **Summary Tab:** Magnitude, Alert Level (Green/Orange/Red), and Country.
 - **Impact Tab:** Population exposed (e.g., people in 100km buffer), Vulnerability score (INFORM index), and secondary risks.
 - **Media Tab:** Total number of news articles and social media mention counts.
-

Task 2: Temporal & Entity Analysis

Once the raw data is collected, students must "clean" it and apply Natural Language Processing (NLP) to extract intelligence.

- **2.1 Response Delta Calculation:** Calculate the time difference between the "System Alert" (the moment GDACS detected the event) and the "Media Peak" (when news article volume reached its maximum).
 - *Formula:* $\Delta T = T_{\text{MediaPeak}} - T_{\text{SystemAlert}}$
- **2.2 Entity Recognition (NER):** Using a library like [spaCy](#) or a keyword-match list, identify the **Entities** involved.
 - *Task:* Extract names of NGOs (Red Cross, UNICEF), Governments, and Private Agencies mentioned in the "Media" tab headlines, number of deaths, losses and relief funds.
- **2.3 Sentiment Volatility:** Run a sentiment analysis on the news headlines for both events.
 - *Task:* Compare the "Tone" of the historical disaster vs. the current one. Has the reporting become more analytical or more alarmist?

Task 3: Comparative Insights & Intelligence

Students must derive non-obvious insights by correlating the "Impact" data with the "Media" data.

- **3.1 The "Forgotten Crisis" Index:** Create a ratio of **News Volume to Population Impact**.
 - *Goal:* Determine which disaster received "over-coverage" and which was "under-reported."
- **3.2 Vulnerability Benchmark:** Compare the **Coping Capacity** (found in the Impact tab) of the two regions.
 - *Insight:* Did a higher Coping Capacity in the current event lead to a lower Alert Level despite a higher Magnitude?

[Bonus: Students can show interesting insights by incorporating more information from other data sources.]

Task 4: Interactive Visualization & Reporting

The final deliverable is a functional Python script and a concise "Intelligence Brief."

- **4.1 The "Disaster Pulse" Dashboard:** Create a visualization (using Streamlit, Plotly, or Tableau) featuring:
 - **A Dual-Timeline:** Overlaying news volume growth for Event A vs. Event B.
 - **A Resilience Radar:** A radar chart showing Magnitude, Population Exposure, Media Coverage, and Vulnerability.

(Note: Students are encouraged to add other insights based on data availability)

- **4.2 Final Report (500 words):** * Explain the technical challenges of scraping the "Impact" tables.
 - Summarize the "Evolution" of the disaster: Is the world responding faster today? Is the media focus aligning with the areas of highest vulnerability?

Submission Requirements

1. **Python Script/Notebook:** Must be well-documented (comments explaining the scraping logic).
2. **Cleaned Dataset:** A CSV containing the data from both events with consistent headers.
3. **Visual Dashboard:** Either a link to a live app or high-resolution screenshots of the comparisons.