Authors: Neeta Kumari., Bir Bhadur Gharti
Institution: St. John's University, Queens, NY 08544, USA

Objective

This project involves retrieving news articles related to "AI in Health" using a public API, cleaning the data, and then storing it in an AWS S3 bucket for later use. We used **NewsAPI.org** to fetch articles based on specific search parameters and filters. After cleaning the data to ensure consistency and usability, we uploaded it to an S3 bucket under the folder structure that makes sense for our project.

Key Steps in the Project

1. **API Setup and Fetching Data:**
    a. We interacted with the **NewsAPI.org** to fetch articles about "AI in Health" under the technology section. The API allows us to filter articles based on various parameters, such as the date, category, and page size.
    b. We used pagination to retrieve a maximum of **1000 articles** in total, by making multiple requests to the API.
2. **Cleaning the Data:**
    a. The retrieved data, consisting of news articles, was cleaned to remove duplicates and any incomplete or missing entries.
    b. We processed the data to extract key details such as the article's title, description, source, and the full content, ensuring that we could store and analyze it efficiently.
3. **Saving the Cleaned Data:**
    a. After cleaning the data, we saved it in a **CSV format**. This format was chosen for easy reading, manipulation, and analysis in tools like Excel or pandas (a Python data analysis library).
    b. We could also save the data in **JSON format**, but CSV was preferred due to its simplicity and wide usage in data processing.
4. **Uploading Data to AWS S3:**
    a. Once the data was saved in a CSV file, it was uploaded to an **AWS S3 bucket** (designated for the project). The S3 service was chosen because it provides a scalable and reliable storage solution, which is perfect for handling large amounts of data.
    b. The uploaded data was organized under a **folder structure** based on the team name and the category of the articles. For example, all data related to "AI in Health" was stored under the folder TEAM_6/AI_in_health/.

S3 Folder Structure

The folder structure on the S3 bucket was organized as follows:

```
markdown
CopyEdit
cus635-spring2025/
  TEAM_6/
    AI_in_health/
      team_6_ai_in_health.csv
```

This organization ensures that data is grouped logically by team and category, making it easy for both storage and retrieval in the future.

Challenges and Solutions

- **Challenge with API Pagination:**
  - The API limited the number of articles that could be fetched in one request, so we had to handle pagination to gather all articles. We set the page-size parameter to 200 articles per page and iterated through multiple pages to get up to 1000 articles.
  - **Solution:** We implemented a loop to fetch data across 5 pages, ensuring that we collected a substantial amount of articles for our analysis.
- **Data Cleaning:**
  - Some of the articles contained missing information, duplicates, or unnecessary HTML tags. We had to clean the data thoroughly to ensure consistency.
  - **Solution:** We created scripts that removed any articles with missing titles or descriptions, deduplicated entries, and formatted the text to make it usable for storage.
- **Uploading Data to S3:**
  - Initially, there was uncertainty around organizing the data in the S3 bucket to ensure it was structured in a way that matched the project requirements.
  - **Solution:** We organized the data into folders named after the team and category (e.g., TEAM_6/AI_in_health/). This approach made it easy to locate the correct files later on.

Technical Details

- **API Used:** NewsAPI.org
- **Programming Language:** Python
- **Libraries Used:**
  - **requests:** For interacting with the NewsAPI to fetch articles.
  - **pandas:** For organizing the data and saving it into a CSV file.
  - **boto3:** For interacting with AWS S3 to upload the cleaned data.
- **Storage:** AWS S3 Bucket (provided by the professor)
- **Data Format:** CSV (for easy data manipulation and storage)

Future Improvements

- **Increase the Number of Articles:** We could fetch more than 1000 articles by modifying the API request or by handling rate limits more effectively.
- **Data Analysis:** After storing the data, we could perform an analysis on the articles, such as identifying trends in AI in health news, summarizing article topics, or sentiment analysis on the content.
- **User Interface (UI):** We could build a simple UI to allow users to query the data, view articles, and download them.

Conclusion

This project successfully demonstrates how to interact with public APIs, retrieve large-scale textual data, clean the data, and store it in a cloud-based solution like AWS S3. By breaking down the steps and tackling challenges, we were able to complete the task efficiently and ensure that the data is well-organized and accessible for future use.