Project Title:
Mood of the News: Sentiment Analysis of Online News Articles

Team Members:
Jingwen Wang, jwang426@usc.edu, jingwenwang7179708388, 7179708388

Problem Statement / Research Question

In this project, I want to study the overall mood of online news during a certain time period. I will look at how sentiment differs across sections such as World, Business, and Technology. News can affect how people see the world, so I want to measure whether news in different sections is more positive, negative, or neutral.

Data Collection Plan

To study the "mood" of online news, I will collect the actual text from BBC articles, because sentiment analysis models work on text. From each article, I will extract the title, the first few paragraphs, the category (such as World, Business, or Technology), and the publication date. This text will let me run sentiment scoring using tools like VADER or TextBlob. The category and date help me compare mood across sections and see how sentiment changes over time. For each article, I will send an HTTP request using the `requests` library and parse the HTML with `BeautifulSoup`. From the HTML, I will extract the article URL, title, section/category, publication date, and part of the article text (such as the first few paragraphs). I will save the raw HTML files and intermediate JSON/CSV files in the `data/raw/` folder, following the required project structure.

Data Cleaning and Preprocessing

After collecting the data, I will clean it in a separate script. I will remove HTML tags, extra spaces, and handle missing fields (for example, missing author information). I will also standardize the date formats, remove duplicate articles based on URL or title, and drop articles that are too short or have almost no text. The cleaned data will be saved in the `data/processed/` folder as a structured CSV or JSON file, which will be used for analysis.

Planned Analysis

Using libraries such as NumPy and Pandas, I will first compute basic statistics. For example, I will look at the number of articles in each section, the distribution of title length and article length, and how many articles appear over time. Then I will apply a simple sentiment analysis tool (such as VADER or TextBlob) to get a sentiment score for each article based on its text. With these scores, I will compare the average sentiment across sections and also check how sentiment changes over time.

Planned Visualizations

I plan to use Matplotlib (and possibly Seaborn) to make plots that show my results. These may include:

- Bar charts showing the number of articles in each section
- Bar charts comparing average sentiment scores across sections

- Histograms of article sentiment scores
- Time-series plots of average daily sentiment, if the data allows