### **Objective of the Sentiment Analysis**

### **1. Problem Statement**

The goal of this project is to find out whether online news articles are written in a positive, negative, or neutral tone. Given a list of article URLs, the idea is to collect their content and then analyze how they feel — are they happy, sad, or just neutral? This helps understand the sentiment behind large volumes of news quickly and automatically.

### **2. My Approach**

We broke this project into a few clear steps:

#### **Step 1: Input File**

We started with an Excel file (Input.xlsx) that had two columns:

* URL\_ID: A unique ID for each article
* URL: The web link to the article

#### **Step 2: Scraping Article Content**

To get the actual news content, we used the **newspaper3k** Python library. It helped us:

* Open each URL
* Download the article
* Extract just the text (not ads or images)

We added the article content into a new column called **Content**.

#### **Step 3: Sentiment Analysis**

We used a simple rule-based method:

* We had two text files: one with **positive words** and one with **negative words**.
* For each article, we counted how many positive and negative words were present.
* Based on the counts:  
  + If there were more positive words → we marked it **Positive**
  + If more negative words → marked as **Negative**
  + If both were similar or no strong words → we called it **Neutral**

#### **Step 4: Output Files**

We saved two output Excel files:

* One with the article content: output\_with\_content.xlsx
* Another with sentiment results: Sentiment\_Output.xlsx

### **3. Tools & Libraries Used**

* Python
* Pandas (for handling Excel files)
* newspaper3k (for scraping news content)
* openpyxl (for writing to Excel files)

### **4. Results / Summary**

* All article URLs were successfully scraped (except maybe a few with broken links).
* The sentiment analysis gave us a quick idea of how each article felt.
* The final result was organized neatly in Excel and easy to read.

### 

### **5. Challenges Faced**

* Some websites didn’t allow scraping or had blocked access.
* The newspaper library had some dependency issues, especially with lxml, which we fixed by installing lxml[html\_clean].
* Articles with very short content or no clear sentiment words were difficult to classify.