# Web Scraping and NLP Project: Article Analysis

## Project Description

This project involves extracting textual data from a list of article URLs and performing natural language processing (NLP) to derive meaningful insights from the content. The primary objective is to analyze articles for various linguistic and readability metrics such as sentiment polarity, subjectivity, complexity, and more. This not only showcases proficiency in web scraping techniques using Python but also demonstrates analytical skills through textual data processing.

## Problem Statement

Given a collection of URLs pointing to news or blog articles, the task is to automatically extract the article title and body text, excluding any irrelevant content like headers, footers, ads, or sidebars. Once the clean text is extracted, the goal is to perform comprehensive text analysis to generate metrics like sentiment scores, fog index, average word length, and more. The results should be saved in a structured format for further analysis or reporting.

## Steps to Accomplish the Goal

* 1. Read the list of URLs from the input Excel file.
* 2. Use Python libraries such as BeautifulSoup and requests to scrape the content of each URL.
* 3. Clean the extracted HTML to retain only the article title and main content.
* 4. Save the clean article text using the URL\_ID as the file name.
* 5. Perform NLP tasks on each text file to calculate the required variables like:
* - Sentiment Analysis (Positive, Negative, Polarity, Subjectivity Scores)
* - Readability Metrics (Fog Index, Complex Word %, Avg Sentence Length)
* - Linguistic Features (Word Count, Syllables, Avg Word Length, etc.)
* 6. Output the results in a structured Excel or CSV format.

## Tools and Technologies Used

* Python Programming Language
* BeautifulSoup and requests for web scraping
* NLTK and TextBlob for natural language processing
* Pandas and NumPy for data handling
* OpenPyXL for Excel file operations
* Regex for text pattern identification and cleaning