1. Approach to the Solution:

The Python script utilizes various libraries and techniques to perform web scraping, text analysis, and data processing

Web Scraping: Utilizes the requests and BeautifulSoup libraries to fetch webpage content and extract article information from specified URLs.

* Text Analysis:
  + Cleaning Text: Removes unnecessary elements and stopwords from the extracted article text.
  + Sentiment Analysis: Calculates sentiment scores by counting positive and negative words within the cleaned text.
  + Readability Analysis: Determines readability metrics like average sentence length, percentage of complex words, Fog Index, etc., to assess text complexity.
* Data Processing: Organizes the extracted data and calculated metrics into a structured DataFrame.
* Exporting Results: Generates a CSV file containing the analyzed data.

2. Running the .py File to Generate Output:

To execute the Python script and generate the output CSV file:

* Open PyCharm or any Python IDE.
* Ensure all required libraries (dependencies) like pandas, requests, beautifulsoup4, and nltk are installed.
* Place the script in a Python file (e.g., analysis\_script.py) within a PyCharm project or any directory.
* Modify the input Excel file's name and location within the script if needed.
* Run the script. This can be done within PyCharm by right-clicking the script and choosing "Run" or by using the terminal/command prompt and navigating to the script's directory and executing python analysis\_script.py.

3. Dependencies:

The script relies on several external libraries and files:

* Python Libraries:
  + pandas
  + requests
  + BeautifulSoup (from bs4)
  + nltk
* External Files/Folders:
  + Input.xlsx: Excel file containing URLs.
  + StopWords-20231226T073510Z-001/StopWords: Folder containing text files with stopwords.
  + MasterDictionary-20231226T073510Z-001/MasterDictionary: Folder containing positive and negative word text files.