# Approach:

1. Data Extraction:
   * Used the requests library to fetch HTML content from the given URLs.
   * Utilized BeautifulSoup to parse and extract relevant text data, focusing on the article title and text.
   * Saved each articles extracted text to separate text files with names corresponding to the URL\_ID.
2. Text Analysis:
   * Applied various text analysis techniques as outlined in the provided "Text Analysis.docx" file.
   * Calculated sentiment scores, readability metrics, word counts, and other specified variables using appropriate formulas.
   * Used regular expressions, NLTK library, and other Python functions for specific tasks like counting personal pronouns and calculating syllables.
3. Output Generation:
   * Organized the results in the specified format and stored them in a CSV or Excel file.

Instructions to Run the .py File:

1. Dependencies:
   * Ensure you have the required Python libraries installed, such as requests, beautifulsoup4, nltk, and any others specified in the script.
   * Install NLTK packages using nltk.download() to acquire necessary resources (e.g., stopwords).
2. Python Script:
   * Save the provided Python script (your\_script.py) to your local machine.
3. Input Data:
   * Place the provided Input.xlsx file in the same directory as your Python script.
4. Execution:
   * Open a terminal or command prompt.
   * Navigate to the directory containing your Python script.
5. Run the Script:
   * Execute the script using the command: python your\_script.py
6. Output:
   * The script should generate an output file (CSV or Excel) containing the calculated variables in the specified order.

# Instructions to Run the .py File:

Python Script:

Save the provided Python script (Blackcoffer Data Extraction and NLP.py) to your local machine.

Input Data:

Place the provided Input.xlsx file in the same directory as your Python script.

Input Files: NOTE: CHANGE PATH FOR STOPWORD FILE

stopwords\_auditor = load\_stopwords(os.path.join(script\_directory, 'StopWords\_Auditor.txt'))

stopwords\_currencies = load\_stopwords(os.path.join(script\_directory, 'StopWords\_Currencies.txt'))

stopwords\_dates\_numbers = load\_stopwords(os.path.join(script\_directory, 'StopWords\_DatesandNumbers.txt'))

stopwords\_generic = load\_stopwords(os.path.join(script\_directory, 'StopWords\_Generic.txt'))

stopwords\_generic\_long = load\_stopwords(os.path.join(script\_directory, 'StopWords\_GenericLong.txt'))

stopwords\_geographic = load\_stopwords(os.path.join(script\_directory, 'StopWords\_Geographic.txt'))

stopwords\_names = load\_stopwords(os.path.join(script\_directory, 'StopWords\_Names.txt'))

Execution:

Open a terminal or command prompt.

Navigate to the directory containing your Python script.

Run the Script:

Execute the script using the command: python Blackcoffer Data Extraction and NLP.py)

Output:

The script should generate an output file (CSV or Excel) containing the calculated variables in the specified order.

#Dependencies requirement:

necessary libraries for web scraping, data manipulation, and natural language processing. Here is a brief explanation of each import:

1. requests: Used for making HTTP requests to fetch web page content.

2. BeautifulSoup: A library for pulling data out of HTML and XML files. Its commonly used for web scraping.

3. pandas: A powerful library for data manipulation and analysis. It provides data structures like DataFrames that are suitable for working with structured data.

4. nltk.tokenize: The word\_tokenize and sent\_tokenize functions from the NLTK library are used for tokenizing text into words and sentences, respectively.

5. nltk: The Natural Language Toolkit is a library for working with human language data. It includes various modules and corpora for tasks like tokenization, stemming, and more.

6. stopwords: A list of common words (e.g., "the", "and", "is") that are often removed during text processing to focus on meaningful words.

7. string: Provides a collection of string constants, such as ASCII characters, punctuation characters, etc.

Make sure you have these libraries installed before running your script. Additionally, consider including instructions for installing these dependencies in your submission documentation.