**Instructions Documentation**

**1. Approach to the Solution**

This project is designed to extract, analyze, and process text data to generate various sentiment, readability, and linguistic metrics. The solution follows these steps:

1. **Data Extraction**:

* Articles were scraped from web URLs using the requests and BeautifulSoup libraries.
* Extracted content was saved as .txt files for further processing.

1. **Sentiment Analysis**:

The sentiment analysis is broken down into the following key metrics:

* **Positive Score:** A score of +1 is assigned to each positive word found in the lexicon. These are summed to give the overall positive score.
* **Negative Score:** Similarly, a score of -1 is assigned to negative words. We multiply by -1 for positive output.
* **Polarity Score**: This score determines whether the overall text sentiment is more positive or negative. It is calculated using the formula:

**Polarity Score**: This score determines whether the overall text sentiment is more positive or negative. It is calculated using the formula:

**Polarity Score =** (Positive Score − Negative Score) / (Positive Score + Negative Score) + 0.000001​

The range is from -1 (most negative) to +1 (most positive).

**Subjectivity Score:** This reflects the degree of personal opinion versus factual information in the text. The formula is:

**Subjectivity Score =** (Positive Score + Negative Score) / (Total Words after Cleaning) + 0.000001

​ This score ranges from 0 (objective) to 1 (subjective).

**2. Readability Analysis**

The readability of a text is essential for understanding how accessible the content is to the target audience. We employ several metrics to gauge text complexity.

**2.1 Average Sentence Length**

This metric is derived by dividing the total number of words by the total number of sentences. It gives an idea of sentence complexity.

**2.2 Percentage of Complex Words**

Complex words are those with more than two syllables. This metric is calculated as:

**Percentage of Complex Words =** Number of Complex Words / Total Words × 100

**2.3 Fog Index**

The Gunning Fog Index is a popular readability formula that estimates the years of formal education required to understand the text. It is calculated as:

**Fog Index =** 0.4 ×( Average Sentence Length + Percentage of Complex Words)

**3. Complex Word Identification**

Complex words are those that contain more than two syllables. The identification of such words is important for assessing the overall complexity of the text.

**Additional Metrics**:

* **Complex Word Count**: Identified words with more than two syllables.
* **Word Count**: Counted meaningful words after removing stopwords.
* **Syllable Count Per Word**: Counted vowels per word while handling exceptions like "ed" or "es."
* **Personal Pronouns**: Detected pronouns like “I,” “we,” “us,” and “my” using regex.
* **Average Word Length**: Calculated by dividing total characters by total words.

**Output File Generation**:

* All metrics were compiled into an Excel file (Output.xlsx) structured according to the template provided in Output Data Structure.xlsx.

**4. How to Run the Script**

1. **Ensure Python is Installed**:
   * Install Python 3.x from [python.org](https://www.python.org/) if not already installed.
2. **Install Required Dependencies**:

* Open a command prompt or terminal and run:

**pip install -r requirements.txt**

Ensure the requirements.txt includes:

* pandas
* requests
* beautifulsoup4
* nltk
* openpyxl

1. **Download Required Files**:

* Place the following in the same directory as the script:
* Input.xlsx: The input Excel file containing URLs and IDs.
* StopWords/: Folder with stopword lists for cleaning text.
* MasterDictionary/: Folder containing positive and negative word dictionaries.

1. **Run the Script**:

* Execute the script using the following command:

**python analysis\_script.py**

1. **Output**:

The results will be saved in:

* Output.xlsx: Contains all metrics in the required structure.
* content/: Directory with individual .txt files for each article.

**5. Dependencies Required**

* **Python Libraries**:
  + pandas: For data manipulation and Excel file operations.
  + requests: For fetching web content.
  + beautifulsoup4: For parsing HTML content.
  + nltk: For natural language processing tasks.
  + openpyxl: For creating and editing Excel files.
* **External Files**:
  + **StopWords Folder**: Contains text files with stopwords for cleaning.
  + **MasterDictionary Folder**: Contains dictionaries of positive and negative words.