PROJECT INSTRUCTIONS

1. Approach to the Solution-

Data Extraction (data_extraction.py):
The **Objective** of the Data Extraction was to extract the contents of the provided HTML page.

Steps-
1. Read Input: Read the list of URLs from Input.xlsx**.**
2. Scrape Content: Use BeautifulSoup to scrape the article titles and texts from each URL.
3. Save Extracted Texts: Save the extracted article texts in separate text files named with their respective URL_ID.(this involved using a for loop and creating a directory to store the directories extracted)

Data Analysis (data_analysis.py):

The **Objective** of the Data Analysis was to analyze the data that was extracted using the provided functions in the code.

1.**create_sentiment_dictionary**: Loads and returns sets of positive and negative words from provided text files.
2.**perfom_sentiment_analysis**: Calculates and returns a positive score, negative score, polarity score, and subjectivity score using tokenized words and sentiment dictionaries.
3.**form_list_output_scores**: Appends the dictionary of metrics for each article to a cumulative list.
4.**count_syllables_per_word**: Calculates and returns the average number of syllables per word in the text.

5.**count_syllables**: Counts and returns the number of syllables in a given word.
6.**complex_word_count**: Counts and returns the number of complex words (words with more than two syllables) in the text.
7.**clean_text**: Cleans the text by removing stopwords and punctuation, and tokenizing it into words.
8.**total_cleaned_words_cnt**: Returns the total number of cleaned words in the text.
9.**analyze_readibility**: Calculates and returns average sentence length, percentage of complex words, and Fog Index.
10.**avg_words_per_sentence**: Calculates and returns the average number of words per sentence in the text.
11.**count_personal_pronouns**: Counts and returns the number of personal pronouns in the text.
12.**calculate_avg_word_length**: Calculates and returns the average word length in the text.
13. **main**: Main function that processes all articles, cleans the text, performs sentiment analysis, computes readability metrics, and saves results to a CSV file.

2. Procedure-

1. The first step is to run the data_exteaction.py file in the device which generates the output that is a file directory inside the project folder about the articles extracted using Beautiful Soup.
2. Now we have to run the data_analysis.py file which will perform the data cleaning and perform the various functions on the text data that was extracted in the data extraction.py file. This step also returns the output Excel file that will required for the output data structure format as provided in the assignment.

3. Dependencies-

1.data_extraction.py-

Python Libraries required-

- **pandas:** For reading data from Excel file.
- **requests:** For making HTTP requests to fetch the HTML content of web pages.
- **BeautifulSoup:** For parsing HTML content and extracting text from web pages.

2.data_analysis.py-

Python Libraries required-

- **pandas**: For data manipulation and reading/writing Excel and CSV files.
- **numpy**: For numerical operations.
- **nltk**: For natural language processing tasks such as tokenization and stopwords.
- **requests**: For making HTTP requests (if used in data extraction).
- **beautifulsoup4**: For parsing HTML content (if used in data extraction).
- **openpyxl**: For reading and writing Excel files.
- **scikit-learn**: For additional data processing utilities

External Files-

1. **Input.xlsx**: The Excel file containing the list of URLs.
2. **Master_Dictionary**:

- positive-words.txt: A text file containing positive words for sentiment analysis.
- negative-words.txt: A text file containing negative words for sentiment analysis.

3.stopwords:

- StopWords_Auditor.txt: Stopwords file.
- StopWords_Currencies.txt: Stopwords file.
- StopWords_DatesandNumbers.txt: Stopwords file.
- StopWords_Generic.txt: Stopwords file.
- StopWords_GenericLong.txt: Stopwords file.
- StopWords_Geographic.txt: Stopwords file.
- StopWords_Names.txt: Stopwords file.