**REPORT**

**TEXT EXTRACTION AND ANALYSIS**

The project aims to automate the extraction of textual data from specified URLs and conduct a comprehensive analysis to calculate metrics related to sentiment, readability, and textual complexity.

**Approach to the Solution:**

1. Setting Up the Environment:

* Ensure Python is installed on your system.
* Install necessary Python libraries: beautifulsoup4, requests, pandas, and nltk, essential for web scraping, data manipulation, and natural language processing.

1. Data Extraction:

* Utilizes BeautifulSoup and requests for fetching and parsing HTML, extracting main article text.
* Custom function extract\_article\_text navigates DOM to isolate textual content, focusing on relevant elements while excluding headers and footers.

1. Text Analysis:

* Employs nltk for tokenizing text into sentences and words, crucial for metric calculations.
* Filters out stop words using compiled lists to refine analysis.
* Computes sentiment scores using predefined lists of positive and negative words against filtered text.

1. Metric Calculation:

* Calculates sentiment scores, readability scores (Fog Index), average sentence length, word count, and other metrics.
* Defines custom functions for tasks like syllable counting to determine text complexity.

1. Output Compilation:

* Compiles metrics into a pandas DataFrame, structured per output requirements.
* Exports DataFrame as Output.csv, facilitating easy analysis.

**Instructions for Running the Script:**

1. Dependencies: Install all dependencies using pip:

* pip install beautifulsoup4 requests pandas nltk

1. Running the Script:

* Ensure the script Code\_TextAnalysis.py and Input.xlsx (containing URLs) are in an accessible directory.
* Run the script via command line or IDE:
* python Code\_TextAnalysis.py

1. The script processes URLs from Input.xlsx, extracts and analyzes text, and outputs results to Output.csv.

**Conclusion**

This project showcases the integration of web scraping, data processing, and natural language processing techniques to automate the analysis of web content. Following the provided instructions ensures a smooth execution of the script, allowing for efficient data extraction and analysis with minimal manual intervention. The flexibility of the script and the comprehensiveness of the analysis make this solution adaptable to various content analysis needs.