REPORT

Objective: Extract text (title and article) from URLs, compute specific textual metrics, and save the output in a structured Excel file.

How I Approached the Solution-

- **Inputs**:
    1. Input.xlsx containing URL_ID and URL.
    2. Output Data Structure.xlsx defining the required metrics.
- **Outputs**: An Excel file (named Output.xlsx) with calculated metrics for each URL.

Workflow-

1. **Read Inputs**:
    - Using pandas to load the input data and output structure.
2. **Extract Text**:
    - Fetch web pages using requests.
    - Parse HTML content with BeautifulSoup to extract titles and paragraphs.
3. **Clean and Process Text**:
    - Remove links, special characters, and extra spaces using re.
    - Tokenize text into sentences and words using nltk.
    - Filter out non-alphanumeric tokens and stopwords.
4. **Compute Metrics**:
    - Metrics like word count, syllables per word, polarity score, Fog Index etc are calculated using logic and helper functions.
    - Positive and negative word counts are based on predefined lists.
5. **Save Results**:
    - Save results to a new Excel file using pandas.

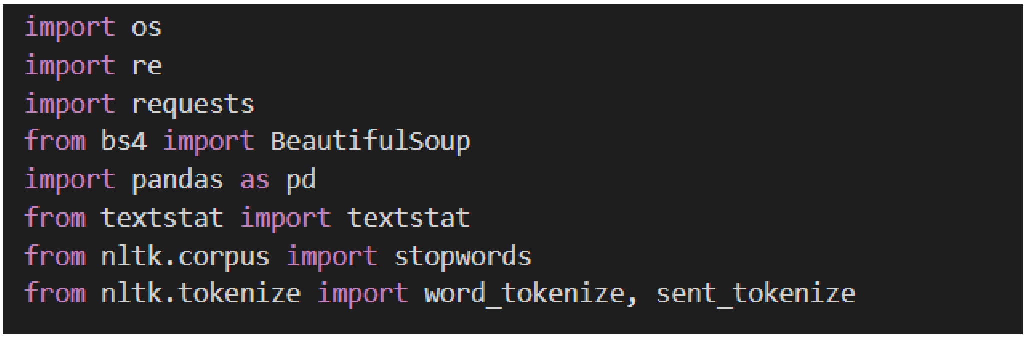
How to Run the .py File to Generate Output

Prerequisites

1. I have written code in google collab and ensure version is upto date.
2. **Dependencies**: Install the required Python libraries.

Instructions

1. **Prepare Input Files**:
    - Place Input.xlsx and Output Data Structure.xlsx in the same directory as the script.
2. **Install Dependencies**:



```
import os
import re
import requests
from bs4 import BeautifulSoup
import pandas as pd
from textstat import textstat
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize
```

    - This command ensures i have all libraries like requests, BeautifulSoup, pandas, nltk, and textstat.
3. **Run the code**:
    - In google collab.
4. **Check the Output**:
    - The output file Output.xlsx will be generated in the same directory.

Dependencies -

Here's the required dependencies:

Library	Purpose

requests	Fetches HTML content from URLs.
bs4	Parses HTML to extract specific elements like titles and paragraphs.
pandas	Handles input/output data in Excel format.
nltk	Tokenizes text into sentences/words and processes linguistic features.
textstat	Calculates linguistic metrics like syllables and readability scores.
openpyxl	Enables seamless reading/writing of Excel files.

I have written code and explained each snippet using comments .

By-

BABANDEEP SINGH