**BLACKCOFFER**

**Task Instructions:**

Objective:

The objective of this task is to extract and analyze text metrics from a list of URLs. The metrics include sentiment analysis, complexity measures, and various other textual features.

Approach:

Installation of Libraries:

In this Python script, we are leveraging several essential libraries for web scraping, text analysis, and natural language processing. The **'requests'** library is used to make HTTP requests and retrieve web pages without errors. The **'BeautifulSoup'** library assists in parsing and navigating HTML content, enabling efficient web scraping. For data manipulation and analysis, the 'pandas' library is employed, providing powerful data structures and functions. The **'TextBlob'** library aids in processing textual data, offering tools for sentiment analysis, part-of-speech tagging, and more.

Additionally, we utilize the **'nltk'** library, a comprehensive toolkit for natural language processing. To tokenize text into words and sentences, we employ **'word\_tokenize'** and **'sent\_tokenize'** from **'nltk.tokenize**.' The **'nltk.corpus'** module, specifically the **'cmudict'** submodule, is used to access the Pronouncing Dictionary. These resources are essential for tasks like phonetic analysis.

To ensure the availability of necessary linguistic data, we download essential components using the **'nltk. download'** function. The script downloads required NLTK resources (**punkt**, **averaged\_perceptron\_tagger,** and **cmudict**) for tokenization, part-of-speech tagging, and syllable counting.

Specifically, we obtain the **'punkt'** tokenizer models, the **'averaged\_perceptron\_tagger'** for part-of-speech tagging, and the 'cmudict' for the Carnegie Mellon Pronouncing Dictionary. These libraries collectively empower us to perform a wide range of web scraping, data analysis, and natural language processing tasks in our Python script. And finally the commonly known **‘pandas’**.

Stop words:

A given folders of stopwords have been consolidated and saved in a file named "**STOPWORDSSS.txt**."And it is used with the with clause in the name of custom\_stopwords for accessing the words easily. which reads a list of custom stopwords from a text file.

Web Scraping Function:

Defines a function **extract\_article\_text**(url) that extracts the title and text content of an article from a given URL using BeautifulSoup.

Syllable Counting Function:

Defines a function **count\_syllables**(word) that counts the syllables in a word using the CMU Pronouncing Dictionary.

Personal Pronoun Counting Function:

Defines a function **count\_personal\_pronouns**(text) that uses TextBlob to count personal pronouns in a given text.

Text Metrics Calculation Function:

Defines a function calculate\_text\_metrics(**text, custom\_stopwords**) that computes various text metrics such as **sentiment scores, average sentence length, percentage of complex words, FOG index, etc.**

Read Data from Excel:

Reads data from an Excel file using pandas, assuming the Excel file has columns 'URL\_ID' and 'URL'.

Processing Data and Calculating Metrics:

Iterates through each row of the DataFrame, extracts the article text from the corresponding URL, and calculates text metrics using the calculate\_text\_metrics function.

Creating Output DataFrame:

Creates a new Data Frame (**output\_df**) with calculated metrics and information from the input Data Frame.

Save Output to Excel:

Saves the output DataFrame to a new Excel file.

In summary, the script is designed to analyze text metrics of articles specified in an Excel file, extract relevant information from the web, and save the calculated metrics to another Excel file. It uses various libraries for web scraping, text processing, and data manipulation.

**How to run the .py file to generate output:**

**Set Up Environment:**

Ensure you have Python installed on your system. You can download it from python.org.

Install Required Packages:

Requests, beautifulsoup4, pandas, textblob, nltk

Download NLTK Resources:

After installing nltk, you need to download additional resources. If you haven't done this already, run the following commands:

nltk.downloader punkt

nltk.downloader averaged\_perceptron\_tagger

nltk.downloader cmudict

Prepare Input Data:

Make sure you have the input Excel file (Input.xlsx) in the specified format with 'URL\_ID' and 'URL' columns.

Run the Script:

Open a command prompt and navigate to the directory containing your script.

Run the script using the following command:

python your\_script.py

Check Output:

The script will generate an output Excel file (Output.xlsx) in the same directory.

Open the output file to view the calculated metrics.

If you're using an integrated development environment (IDE) like VSCode, PyCharm, or Jupyter Notebook, you can also run the script directly within the IDE.

Make sure to replace your\_script.py with the actual name of your Python script file.

Make sure you have the file STOPWORDSSS.txt. The .txt file is attached with the submission.