# 1.Text Extraction and Analysis Assignment Approach Solution

## Objective

The primary objective of this assignment is to extract textual data from articles given in a URL and perform text analysis to compute various variables

To download the folders which are stopwords and MasterDictonary are sharable by this below google drive link.

https://drive.google.com/drive/folders/1ltdsXAS_zaZ3hI-q9eze_QCzHciyYAJY?usp=sharing

## Data Extraction

The first step of this assignment is data extraction, The data is provided in an excel file named Input.xlsx , Each row in the file corresponds to an article and the goal is to extract the article text and save it in a separate text file. The filename for each article is its URL_ID

The extraction process is designed to only capture the article title and the article text, excluding the website header, footer, or any other elements.

## Code Implementation

The code for this project is written in Python and makes use of several libraries including glob, pandas, nltk, and textstat .

The glob library is used to iterate over each text file in the current directory, For each file, the code opens it and initializes variables for the title, text a flag to indicate if we are reading text content, and a counter for consecutive empty lines.

After the data extraction reads the file line by line. If a line starts with “Title:”, the titie is extracted, If a line starts with “Text:”, the text extraction begins. The text is continuously appended until two consecutive empty lines are encountered, at which point the text readings is stopped.

**Text Analysis**

After the extraction and cleaning of the text data, the next steps involve text analysis to compute various variables such as positive score and negative score, and polarity score.

**Stop Words Removal**

The first step in this process is to load the stop words from files in the "StopWords" folder. These stop words are common words that do not contribute much to the meaning of a sentence and are often removed before text analysis.

**Text Cleaning**

The text data is then cleaned by converting it to lowercase, replacing full stops with a special character, removing extra spaces, alphanumeric and numeric values, special characters, and stop words. The cleaned text is then added to the DataFrame

**Sentiment Analysis**

The sentiment analysis involves calculating the positive and negative scores for each text, this is done by tokenizing the text, and for each token, checking if it is in the list of positive or negative words. The positive and negative scores are then added to the DataFrame.

This score gives an indication of the overall sentiment of the text. A positive polarity score indicates a positive sentiment. While a negative polarity score indicates a negative sentiment.

**Polarity Score** = (Positive Score – Negative Score)/ ((Positive Score + Negative Score) + 0.000001)

This completes the text analysis part of the assignment. The DataFrame now contains the extracted and cleaning text data.

The next variables is to calculate is Subjectivity Score

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

This score gives an indication of the subjectivity of the text, A higher subjectivity score indicates a more subjective text, the overall analysis done on the "Cleaned_Text" column.

**Average Sentence Length**

**Average Number of Words Per Sentence =** the total number of words / the total number of sentences

Based on these formulae the average number of words per sentences is calculated by tokenized the text into sentences, counting the total number of words and sentences, and dividing the total words by the total sentences it involves the "Text" column it does not any cleaning operations on that column. So that easily makes to count the tokens based on the formula.

**Percentage of Complex Words**

The percentage of complex words is calculated by tokenizing the text into words, counting the words with more than 2 syllables (assumed to be complex) based on the formula to calculate the percentage of complex words

**Percentage of Complex words** = the number of complex words / the number of words

**Gunning Fox Index**

The Gunning Fox index is a readability test designed to measure the understandability of a text. It is calculated using the formula

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

**Complex word count**

The complex word count is calculated by tokenizing the “Text” column the text into words and counting the words more than 2 syllables(assumed to be complex).

**Word Count**

The word count is calculated by tokenizing the text into words,

removing the stop words (using stopwords class of nltk package).

removing any punctuations like ? ! , . from the word before counting.

This applied on the “Text” column in the DataFrame.

**Syllable Count Per Word**

The syllable count per word is calculated on the “Text” column in the DataFrame.

Count the number of Syllables in each word of the text by counting the vowels present in each word. We also handle some exceptions like words ending with "es","ed" by not counting them as a syllable.

**Personal Pronouns Count**

The personal pronouns count is calculated using regular expression pattern that matches the personal pronouns “I”, “We”, “my”, “ours”, and “us”. The function

Calculates personal pronouns count finds all matches in the text and returns the count of matches. This function is then applied to the “Text” column in the DataFrame.

**Average Word Length**

The average word length is calculated by tokenizing the text into words, calculating the total number of characters in each word, and dividing the total chatacters by the total words. The function calculate average word length. Performs these calculating and is applied to the “Text” column in the DataFrame

**Avg Word Length**=Sum of the total number of characters in each word/Total number of words

After all these computations, the DataFrame extracted_df now contains the all variables in a DataFrame format.

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

To run a Python script that requires specific libraries and files, you can follow these steps:

**Install the Required Libraries**: Python libraries can be installed using pip, which is a package manager for Python. You can install a library by opening your terminal/command prompt and typing pip install library name. Replace “library_name” with the name of the library you want to install. For example, to install pandas, textstat,  you would type pip install pandas.

**Import the Libraries in Your Script**: Once the libraries are installed, you can use them in your script by adding an import statement at the top of your script. For example, to use pandas in your script, you would add the line import pandas as pd at the top of your script. All libraries are mentioned in the top of the code .py file

**Download the Required Files**: If your script requires specific files, you’ll need to download these files and save them in a location that your script can access. If

the files are provided through Google Drive, you can download them by navigating to the file in Google Drive, right-clicking on the file, and selecting "Download".

To utilize files from Google Drive for your project, begin by mounting your Google Drive onto your working environment. This is a standard procedure in platforms like Google Colab. Once mounted, you can access your files using the path '/content/drive/My Drive/'. For instance, if your files are stored in a Google Drive folder named 'YourFolderName', the file paths will be '/content/drive/My Drive/YourFolderName/Input.xlsx' for your input file, '/content/drive/My Drive/YourFolderName/StopWords.txt' for your stopwords file, and '/content/drive/My Drive/YourFolderName/MasterDictionary.csv' for your master dictionary. Ensure to replace 'YourFolderName' with the actual name of your Google Drive folder. This setup allows your script to interact with the files stored on Google Drive directly. Remember to authorize access when prompted, and adjust the file extensions to match your actual files.

**Run Your Script**: Once the libraries are installed, the files are downloaded, and the file paths in your script are updated, you can run your script. Open a terminal/command prompt, navigate to the directory containing your script, and type python script_name.py to run your script. Replace "script_name.py" with the name of your script.

Please note that these instructions assume that you have Python and pip installed on your computer. If you haven't installed Python yet, you can download it from the official website: https://www.python.org/downloads/

Also, the exact command to install packages can vary depending on your operating system and Python installation. For example, you might need to use pip3 instead of pip, or python -m pip on some versions of Windows. If you're

using a Jupyter notebook, you might need to prefix the command with an exclamation mark, like !pip install pandas.

**3.A list of all dependencies required in script uses and how to install them:**

1. **beautifulsoup4 & requests:** These libraries commonly used in python for web scraping.
2. **glob**: This is a standard Python library and doesn't need to be installed separately.
3. **pandas**: This is a powerful data manipulation library. You can install it using pip:

- !pip install pandas
- !pip install numpy

3. **string**: This is a standard Python library and doesn't need to be installed separately.
4. **re**: This is a standard Python library for regular expressions and doesn't need to be installed separately.
5. **nltk**: This is a library for natural language processing. You can install it using pip:

- pip install nltk

After installing nltk, you also need to download the 'punkt' and 'stopwords' datasets. You can do this in Python:

```
import nltk
nltk.download('punkt')
nltk.download('stopwords')
```

**6.textstat**: This is a library for calculating statistics from text. It can be installed using pip:

- !pip install textstat

Please note that you need to have Python and pip installed on your computer to run these commands. If you haven't installed Python yet, you can download it from the official website: https://www.python.org/downloads/