INSTRUCTIONS

➢ DOWNLOADING AND EXECUTING THE .py FILE:

1. Download the Python file to a location on your computer.

2. Open the terminal on your computer.

3. Type “cd” followed by a space.

4. Drag the folder containing the Python file from your file explorer and drop it onto the terminal. This will automatically fill in the file path for you.

5. Press enter to execute the command.

[NOTE: IT'S JUST ONE WAY TO EXECUTE THE CODE, YOU CAN JUST COPY IT AND PASTE IT ON A CODE EDITOR OR FOLLOW SOME ANOTHER WAY AS WELL TO RUN THE CODE WITHOUT ANY DOUBT.**]**

➢ WHAT THIS CODE DOES?

the code is written to perform sentiment analysis on a list of URLs, as well as calculate various metrics related to text complexity, such as the FOG index, average word length, and number of complex words.

The program reads an Excel file called "Output Data Structure.xlsx" into a pandas dataframe, and accesses the column containing the URLs. It then iterates over the URLs and performs the following steps for each URL:

- Sends a GET request to the URL and retrieves the HTML content.
- Uses the BeautifulSoup library to extract the text content from the HTML.
- Calculates the positive and negative sentiment scores based on the number of positive and negative words in the text. The positive and negative words are stored in separate files called 4. "positive-words.txt" and "negative-words.txt".
- Calculates the polarity score, which is the difference between the positive and negative scores divided by the sum of the positive and negative scores.
- Calculates the number of words, number of sentences, and average sentence length.
- Calculates the number and percentage of complex words in the text. A complex word is defined as a word with more than two syllables.
- Calculates the FOG index, which is a measure of text complexity that takes into account the average sentence length and percentage of complex words.
- Calculates the average number of words per sentence.
- Removes stop words from the text and calculates the subjectivity score based on the remaining words.
- Counts the number of personal pronouns in the text.
- Calculates the average word length and average number of syllables per word.

➢ OUTPUT:

After implementing each URL, the output window will show a statement like "**url 'URL_ID' done**" just to make sure that the program is running properly for each URL. (**NOTE**: it is obviously not important to print the statements, it is just for convenience. can remove the print statement and it will not harm the whole code even a little bit)

After calculating the linguistic features for all the URLs, the code appends the results to several lists, which are then written to a new Excel file using openpyxl. The output Excel file contains one row for each URL, with columns for each linguistic feature and the corresponding value.

finally, the output terminal will show "**YOUR OUTPUT DATA HAS BEEN SUCCESSFULLY SAVED IN 'final_output.xlsx' EXCEL FILE IN THIS SAME FOLDER**" which means the code has been run successfully and the output has been stored in a excel file named "final_output.xlsx" in the same directory. you can now open the excel file and check the results.

[NOTE: SOME OF THE URLs IN THE EXCEL SHEET DOES NOT EXISTS, FOR THOSE, THE CODE WILL PROVIDE "**NULL**" TO EVERY VARIABLE COLUMN.**]**

➢ OUTPUT EXCEL FILE:

The output excel file named "my_data_final.xlsx" of this code is also attached with the submission. kindly check whether the outputs, I have got from this code, are correct or not.

THANK YOU.