TEXT ANALYSIS USING NATURAL LANGUAGE PROCESSING (NLP)

DATA COLLECTION-:

I was provided with an input file containing a list of 114 URLs along with corresponding URL IDs. To gather the necessary data, I employed the Beautiful Soup library for web scraping. This process involved extracting articles from the provided URLs by sending HTTP requests, parsing the HTML content of the pages, and subsequently retrieving the relevant textual information. & the retrieved text is then saved into .txt file with each file being named according to its corresponding URL ID."

Challenge faced -: However, I encountered an issue during the process where two of the provided URL links could not be located. As a result of this issue, the retrieved data was limited to only 112 out of the initially provided 114 URLs. This discrepancy in data retrieval was due to the absence of content on those particular URLs.

DATA ANALYSIS-:

1. **Reading Scraped Text Files:** To initiate the data analysis phase, my initial step involved reading the text files that were obtained through the web scraping process. This facilitated the beginning of the data analysis process.
2. **Stop Words Extraction -:** I proceeded to extract stopwords from a designated stopword directory which contained 7 distinct stopwords text files. Through this step, I extracted stopwords from each of these files and subsequently compiled them into a unified stopwords list.
3. **Tokenizing Text into Words and Sentences:** Subsequently, I utilized the NLTK library to tokenize the acquired text into both individual words and sentences. This tokenization process was undertaken to prepare the text for subsequent stages of analysis, ensuring its readiness for text analysis.
4. **Removing Extracted Stopwords from Tokenized Words:** In this step, the previously extracted stopwords were eliminated from the tokenized words.

This action aimed to obtain cleaned data for further analysis by eliminating words that typically hold less significant meaning within the context of text analysis.

5. **Extraction of Positive and Negative Words: F**ollowing that, I proceeded to extract both positive and negative words from the master directory provided. These extracted words were subsequently added to separate lists, namely the positive words list and the negative words list. These lists served a specific purpose in the analysis process, as they were employed to calculate positive and negative scores from the text data.
6. **Calculating Positive and Negative Scores:** In this stage, I used the previously compiled positive words and negative words lists, derived from the master directory, to compute the positive and negative scores within the text. These scores were calculated by evaluating the presence of positive and negative words in the text, offering insights into the sentiment conveyed by the text data.
7. **Calculation of Polarity Score:** Following the acquisition of positive and negative scores, I proceeded to calculate the polarity score utilizing the provided formula. The polarity score, ranging between -1 and +1, serves as an indicator of the sentiment expressed within the text. The formula used for calculating the polarity score is as follows:

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

   This score normalization approach helps in accurately representing the sentiment, considering both positive and negative aspects of the text.
8. **Word Count (Number of Cleaned Words in Text):** Once the words were tokenized, they underwent a cleaning process. This involved removing stopwords using the NLTK library and eliminating punctuations through the string library. This cleanup ensured that only meaningful words remained. Subsequently, the total count of these cleaned words was calculated.
9. **Calculating Subjectivity Score:** The subjectivity score is determined using the following formula, which takes into account the positive score, negative score, and the total number of cleaned words calculated earlier. The formula is as follows:

Subjectivity Score = (Positive Score + Negative Score) / ((Total Cleaned Words) + 0.000001)

This score computation offers an understanding whether a given text leans towards being objective or subjective.

10. **Calculating Average Number of Words per Sentence (Average Sentence Length):** Utilizing the tokenized words and sentences, I proceeded to calculate the average sentence length. This metric provides valuable insights into the structure of the text. The calculation was performed using the formula:

    Average Sentence Length = Total Number of Words / Total Number of Sentences

11. **Syllable Count per Words:** The analysis continued with the determination of syllable counts for individual words within the text. This involved assessing the number of syllables in each word by identifying the vowels present. Furthermore, exceptions such as words ending with "es" or "ed" were managed by not considering them as separate syllables. Following this, the calculation of syllables per word was executed using the formula:

    Syllables per Word = Total Syllables / Total Words

12. **Complex Word Count and Percentage of Complex Words:** Moving forward, I focused on determining the count of complex words in the text. Complex words, in this context, refer to words containing more than two syllables. This involved identifying words with syllable counts exceeding two. Subsequently, the complex word count was computed. Furthermore, I calculated the percentage of complex words within the text. This computation was achieved using the formula:

    Percentage of Complex Words = (Number of Complex Words / Total Number of Words) * 100

13. **Analysis of Readability:** A more profound analysis of the text's readability was pursued through the application of the Fog Index formula. This assessment hinged on the previously computed average sentence length and the percentage of complex words. The formula utilized for the Fog Index calculation is as follows:

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

By combining the average sentence length and the proportion of complex words, this formula produced the Fog Index score, which serves as a gauge of the text's readability. This analysis provides insights into the ease of comprehension of the text's content.

14. **Personal Pronoun Count:** Subsequently, an assessment of personal pronouns was conducted within the text. Pronouns such as "I," "we," "my," "ours," and "us" were specifically identified and tallied using regular expressions. An important exception was implemented to ensure that the country name "US" was not inadvertently included in the list of pronouns. The utilization of the following regex expression facilitated this analysis:

    personal_pronouns = ["i", "we", "my", "ours", "us"]
    pattern = r'\b(?:' + '|'.join(personal_pronouns) + r')\b'

    This approach allowed for the calculation of the occurrences of the mentioned personal pronouns, thereby contributing to an understanding of their usage within the text.

15. **Average Word Length:** As the final step of our analysis, we calculated the average word length. This measure gives us an idea of how long, on average, the words in the text are. To find this, we added up the total number of characters in all the words and divided it by the total number of words in the text. The formula is:

    Average Word Length = (Sum of Total Characters in Words) / (Total Number of Words)

DATA EXPORTED:

Following the computation of the above parameters, the resulting values were appended to a list. This list was subsequently transformed into a dataframe. The dataframe was then exported to a file named "OUTPUT FINAL DATA.xlsx."