**INSTRUCTIONS**

**Steps Involved:**

**Step 1: Install & Import the necessary libraries:**

Install the libraries by going to terminal and writing these commands:

* pip install pandas
* pip install numpy
* pip install requests
* pip install beautifulsoup4
* pip install contractions
* pip install nltk

After installing the libraries, import them

**Step 2: Load the Input.xlsx file**:

Load the ‘Input.xlsx’ file using pandas .

**Step 3**: **Find the elements class:**

First I opened one of the website url and looked for the classes in which the article title and article text are present. After opening the website right-click on the article title and select the ‘Inspect’ option to see the class. Upon checking I found most of the website’s title and text belong to same class except for one website.

**Step 4**: **Extracting the Article title and Article text using Requests and BeautifulSoup.**

Send an HTTP request to the URL and check if the response status code is 200 (OK). Parse the HTML content of the page using BeautifulSoup. Extract the article title from the HTML using specific classes. Removing the References present at the end of each article under 'pre' elements with class 'wp-block-preformatted'. Extract article text candidates using specific classes. The URL link 14 has its content under class 'tdb-block-inner td-fix-index' in 14th index and for rest of URL where article is present in '0th; index. Append the extracted data to the empty 'data' list that was created above. If URL is not found display this statement.

**Step 5: Fix the contraction from the text:**

Fixing the contractions that are present in the Article text using contraction library. Contractions library for fixing contractions for e.g.

"I'm" is a contraction of "I am."

"you're" is a contraction of "you are."

"can't" is a contraction of "cannot."

**Step 6:** Removing digits from the Article text using Regular Expression (re).

**Step 7:** Identifying all the punctuations from the Article text using regular expression.

**Step 8**: Convert the texts to lower case andTokenize the words using nltk.word\_tokenize for text analysis.

**Step 9:** Load the custom stopwords from the folder and add them up with the nltk stopwords so that we can r remove them from the article text while calculating positive and negative words.

**Step 10:** Load the positive and negative words from the ‘MakeDictionary’ folder and store them into a dictionary with key ‘positive’ for positive words and key ’negative’ for negative words except for the words that are already present in the custom stopwords. Create a function that will iterate through tokenized words and match with positive and negative words and give ‘1’ point to positive score whenever positive word is found and ‘-1’ to negative score whenever a negative word is found .

**Step 11:** Find the Polarity score by subtracting the negative score of each article from positive score of each article and add 0.000001.

**Step 12:** Find the Word\_Count by removing the nltk stopwords and punctuations .

**Step 13:** Calculate Subjectivity\_Score by adding positive score and negative score and divide it by cleaned word counts.

**Step 14:** Define a function to Count the number of syllables present in the article text by setting a condition to exclude ‘es’ and ‘ed’ from the words. Use this function inside the function for counting complex words by setting a threshold and count the Complex words.

**Step 15:** Calculate the number of sentences present in each article by using nltk.sent\_tokenize.

**Step 16:** Calculate the total number of words present in each article by using nltk.word\_tokenize after removing punctuations and without removing nltk stopwords.

**Step 17:** Calculate average word length by defining a function that will iterate through each character of a tokenized word from each article and then divide the total characters by total words.

**Step 18:** Calculate average sentence length by dividing the number of words with number of sentences.

**Step 19:** Calculate percentage of complex words by dividing complex word count with number of words.

**Step 20:** Calculate FOG index by adding average sentence length and percentage of complex words and then multiplying with 0.4

**Step 21:** Calculate number of words per sentence by dividing number of words with number of sentence.

Step 22: Find personal pronouns by defining a function that will use regular expression and iterate through each word in the article and match with the list that contains personal pronouns provided to find the pronouns ‘I’, ‘we’, ‘my’, ‘ours’, ‘us’ except ‘US’.

**Step 22**: Drop the columns that are not needed in the output file using pandas.

**Step 23:** Provide the location where you want to save the output excel file and save it using pandas to\_excel function.

**Note:** Change the file path of the ‘Input.xlsx’, ‘custom stopwords’, ‘MasterDictionary’ accordingly.

**Instructions to run the ‘Data Extraction and NLP.py’ python file:**

Open Python app.

Import subprocess.

#Specify the path to your script file.

script\_path=r ” ”

# Run the Python script using the subprocess module.

subprocess.run([‘python’, script\_path])