Data Extraction and text analysis

Blackcoffer Assignment for assessment

Name: Hursh Karnik

Phone: +91 7738390361  
Email: [hurshkarnik5603@gmail.com](mailto:hurshkarnik5603@gmail.com)

Objective:

1. **Objective**

The objective of this assignment is to extract textual data articles from the given URL and perform text analysis to compute variables that are explained below.

1. **Data Extraction**

Input.xlsx

For each of the articles, given in the input.xlsx file, extract the article text and save the extracted article in a text file with URL\_ID as its file name.

While extracting text, please make sure your program extracts only the article title and the article text. It should not extract the website header, footer, or anything other than the article text.

**NOTE: YOU MUST USE PYTHON PROGRAMMING TO EXTRACT DATA FROM THE URLs. YOU CAN USE BEATIFULSOUP, SELENIUM OR SCRAPY, OR ANY OTHER PYTHON LIBRARIES THAT YOU PREFER FOR DATA CRAWLING.**

1. **Data Analysis**

For each of the extracted texts from the article, perform textual analysis and compute variables, given in the output structure excel file. You need to save the output in the exact order as given in the output structure file, “Output Data Structure.xlsx”

**NOTE: YOU MUST USE PYTHON PROGRAMMING FOR THE DATA ANALYSIS**

1. **Variables**

The definition of each of the variables given in the “Text Analysis.docx” file.

Look for these variables in the analysis document (Text Analysis.docx):

1. POSITIVE SCORE
2. NEGATIVE SCORE
3. POLARITY SCORE
4. SUBJECTIVITY SCORE
5. AVG SENTENCE LENGTH
6. PERCENTAGE OF COMPLEX WORDS
7. FOG INDEX
8. AVG NUMBER OF WORDS PER SENTENCE
9. COMPLEX WORD COUNT
10. WORD COUNT
11. SYLLABLE PER WORD
12. PERSONAL PRONOUNS
13. AVG WORD LENGTH

1. **Output Data Structure**

**Output Variables:**

1. All input variables in “Input.xlsx”
2. POSITIVE SCORE
3. NEGATIVE SCORE
4. POLARITY SCORE
5. SUBJECTIVITY SCORE
6. AVG SENTENCE LENGTH
7. PERCENTAGE OF COMPLEX WORDS
8. FOG INDEX
9. AVG NUMBER OF WORDS PER SENTENCE
10. COMPLEX WORD COUNT
11. WORD COUNT
12. SYLLABLE PER WORD
13. PERSONAL PRONOUNS
14. AVG WORD LENGTH

Check out the output data structure spreadsheet for the format of your output, i.e. “Output Data Structure.xlsx”.

1 Explain how you approached the solution  
Ans

**Step 1: Importing Necessary Libraries**

First, we import all the necessary libraries. These include libraries for web scraping (requests and BeautifulSoup), data handling (pandas), natural language processing (nltk), and regular expressions (re). We also ensure the required NLTK data is downloaded.

**Step 2: Setting Up Directories**

We create a directory named "TitleText" if it doesn't already exist. This is where we'll save the text files containing the scraped titles and articles from the URLs.

**Step 3: Reading URLs and Scraping Content**

We read the URLs from an Excel file named Input.xlsx. For each URL:

* We make a request to fetch the page content.
* We use BeautifulSoup to parse the HTML content.
* We extract the title (usually found in an <h1> tag) and the main text (usually found in <p> tags).
* We save the title and text into a file named with the URL ID in the "TitleText" directory.

**Step 4: Loading Stop Words**

We load all stop words from a directory named "StopWords". Stop words are common words that don't contribute much to the analysis (like "and", "the", etc.). These words are stored in a set for easy lookup.

**Step 5: Tokenizing and Filtering Text**

We read each text file from the "TitleText" directory, tokenize the text into words, and filter out the stop words. The filtered tokens for each file are stored in a list called docs.

**Step 6: Loading Sentiment Words**

We load positive and negative words from a directory named "MasterDictionary". These words are used to calculate sentiment scores. Positive and negative words are stored in separate sets.

**Step 7: Calculating Sentiment Scores**

For each document in docs, we:

* Count the positive and negative words.
* Calculate the positive score, negative score, polarity score (difference between positive and negative scores), and subjectivity score (sum of positive and negative scores divided by the total word count).

**Step 8: Measuring Readability**

For each text file, we:

* Calculate the average sentence length by dividing the number of words by the number of sentences.
* Identify complex words (words with more than two syllables).
* Calculate the percentage of complex words, Fog Index (a readability metric), and average syllable count per word.

**Step 9: Word Count and Average Word Length**

We calculate the total word count and the average word length for each text file by removing punctuation and stop words.

**Step 10: Counting Personal Pronouns**

We count the occurrences of personal pronouns ("I", "we", "my", "ours", "us") in each text file.

**Step 11: Writing Results to Output**

We read an Excel template named Output Data Structure.xlsx to get the structure of our output. We remove rows corresponding to non-existent URLs (those that returned a 404 error). We then write the calculated metrics (sentiment scores, readability metrics, word count, etc.) to the DataFrame and save it as a CSV file named Output\_Data.csv.

2 How to run the .py file to generate output

Ans 1. Install all the required libraries

2.If text folder already there delete it

3. Through cmd navigate to the folder and type python filename.py

4. if any python code editor just run the file

3 List all dependencies

Ans

here is the list of dependencies:

1. requests
2. beautifulsoup4
3. pandas
4. nltk
5. os (built-in)
6. re (built-in)

**Additional NLTK Data**

* punkt
* stopwords

**Specific Files and Directories**

1. Input.xlsx
2. TitleText directory
3. StopWords directory
4. MasterDictionary directory
5. Output Data Structure.xlsx