

# Data Extraction and NLP

Project Completion Report: Instruction in Detail

22.04.2023

---

DEBJIT SARKAR

## Overview

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


## Code Description:

The code I provided scrapes data from URLs in an excel file, saves the title and article text in separate text files, performs sentiment analysis using the master dictionary approach, and measures readability metrics for the text documents.

Here is a brief summary of what the code does:

1. Import necessary packages: requests, BeautifulSoup, pandas, os, nltk, re.
2. Mount Google Drive and set working directory.
3. Read input Excel file.
4. Loop through each URL in the input file.
5. Try to get response from the URL and parse the HTML using BeautifulSoup.
6. Extract the title and article text from the HTML.
7. Write the title and article text to separate text files.
8. Set directories for stopwords, sentiment analysis, and text documents.
9. Load stopwords and create a list of documents to analyze.
10. Load positive and negative words from the master dictionary files.
11. Calculate positive and negative scores, polarity score, and subjectivity score for each document.
12. Measure readability metrics for each document, including average sentence length, percentage of complex words, and Fog Index.
13. Overall, the code performs web scraping, sentiment analysis, and readability analysis on text documents.

The code performs several tasks related to text analysis. The details are provided below:



**Web scraping:** It reads input data from an Excel file ('Input.xlsx'), which contains a list of URLs and URL IDs. Then, it uses the requests library to send HTTP requests to each URL and retrieves the response using the BeautifulSoup library to parse the HTML content of the page. The title and text content of the page are extracted from the parsed HTML using BeautifulSoup, and then the title and article text are stored as separate text files for further analysis.

**Sentiment analysis:** The Master Dictionary approach is used to perform sentiment analysis on the text documents. A list of stopwords is first generated from files stored in the 'StopWords' directory. Then, for each text file in the 'TitleText' directory, the text is tokenized, and the stop words are removed. Next, positive and negative words are identified by comparing the filtered text against two lists of positive and negative words stored in files named 'positive-words.txt' and 'negative-words.txt', respectively, in the 'MasterDictionary' directory. Finally, various scores related to the sentiment of the text (positive score, negative score, polarity score, and subjectivity score) are computed and stored in separate lists.

**Readability metrics:** The code computes several readability metrics for each text file in the 'TitleText' directory. These include the average sentence length, percentage of complex words, Fog Index, complex word count, and average syllable word count. The text is first preprocessed to remove unwanted characters and split into sentences. The number of words and sentences are counted, and stop words are removed from the text. The number of complex words is identified by counting the number of words with more than two syllables. The Fog Index is computed using a formula that takes into account the average sentence length and the percentage of complex words. Finally, various metrics related to the complexity of the text (average sentence length, percentage of complex words, Fog Index, complex word count, and average syllable word count) are computed and stored in separate lists.

Overall, this code combines several techniques from natural language processing to perform web scraping, sentiment analysis, and readability analysis on a set of text documents. The output of the code is a set of files stored in various directories containing the extracted data and computed scores. These files can be further processed and analyzed to gain insights into the text content and structure of the documents.

