**Instructions Documentation**

**By: - Ayush kumar Singh**

**Mail: -** [**ayushnov27@gmail.com**](ayushnov27@gmail.com%20)

**Linkedin: -** [**https://www.linkedin.com/in/ayush-kumar-singh-a59844206/**](https://www.linkedin.com/in/ayush-kumar-singh-a59844206/)

# **Approached of solution:**

## **Data Extraction**

I began by reading the input.xlsx file using the **pandas** library. I iterated over each row to extract the URLs. Then, using **Selenium** with the **Chrome WebDriver**, I fetched the HTML content of each URL.

**Key Observations:**

* Every HTML page contained an **<article>** tag, which held the primary extractable content.
* Content before the header and after the footer was unnecessary, so it was excluded.
* I removed specific unwanted headers, such as ["Contact Details", "Project Snapshots", "Project Website URL", "Summarize"].

I used **BeautifulSoup (bs4)** to extract only the relevant text within the tags, including both parent and child tags recursively.

Finally, I saved the extracted text from each page into separate text files, organized under a folder named **“extracted\_files.”**

## **Data Analysis**

I utilized the **pandas** and **nltk** libraries to perform text analysis. First, I applied the stop words list and common regular expression substitutions to remove any punctuation (such as ?, !, ,, .) from the text before proceeding with the analysis.

The processed files were then stored in a folder named "filtered\_files."

Next, I read all the filtered text files and added them to a column in a pandas DataFrame for further processing.For the analysis, most of the variables were calculated directly using the formulas from the text\_analysis.docx document.

However, the following variables were calculated differently:

|  |  |
| --- | --- |
| COMPLEX WORD COUNT | --Rather than calculating syllable count in the traditional way outlined in the text\_analysis.docx, I utilized the **CMU Pronouncing Dictionary available in the nltk library** for a more accurate syllable count. For words not found in the dictionary, I used a custom function sourced from [Stack Overflow](https://stackoverflow.com/questions/46759492/syllable-count-in-python).  --To optimize the process, I implemented an **LRU\_cache** and a global dictionary to prevent redundant calculations. |
| SYLLABLE PER WORD | created a dictionary and leveraged the **CMU Pronouncing Dictionary in the nltk library** for more precise syllable counts. For words missing from the dictionary, I used the custom function mentioned above. |
| PERSONAL PRONOUNS | I used regular expressions (**re**) to match personal pronouns in the extracted files. Since the filtered files had already been cleaned of pronouns  I skipped any occurrences that were preceded by "the," "a," or "an" and were not standalone. |

## **Output Data Structure**

## Finally, we removed all the temporary columns and reordered the remaining columns to match the required format specified in the Output Data Structure.xlsx file.

## **Running the .py file to generate output**

The **.py** script is implemented using a class format and utilizes **argparse** to handle input arguments. Additionally, the progress of the process can be monitored through the use of **tqdm.**

The following arguments should be provided:

|  |  |
| --- | --- |
| **Argument** | **Description** |
| input\_file | The path to the input Excel file to be processed. |
| output\_file | The path to the output Excel file where results will be saved. |
| stop\_words\_directory | The directory that contains the stop word files to be used during text processing. |
| masterDictionary\_directory | The directory containing the master dictionary with positive and negative words for processes. |
| extracted\_files\_directory | The directory where the extracted text files are stored. |
| filter\_extracted\_directory | The directory where filtered text files will be saved after processing. |
| --default\_chrome\_profile\_path | The path to the default Chrome profile, required if the Selenium Chrome driver is not installed. (Optional) |

* Open the directory where the .py file is located.
* Ensure that the required files are present in their respective locations.
* Call the function from the terminal using the following command:

|  |
| --- |
| **python data\_extraction\_and\_nlp.py "G:\Input.xlsx" "G:\output.xlsx" "G:\20211030 Test Assignment\StopWords" "G:\20211030 Test Assignment\MasterDictionary" "G:\extracted\_files" "G:\filtered\_files" --default\_chrome\_profile\_path "C:\chrome\profile"** |

**Explanation:**

* **"G:\ Input.xlsx":** Path to the input Excel file.
* **" G:\output.xlsx":** Path to the output Excel file where the processed data will be saved.
* **"G:\20211030 Test Assignment\StopWords":** Directory containing stop word files.
* **"G:\20211030 Test Assignment\MasterDictionary":** Directory containing the master dictionary with positive and negative words.
* **" G:\extracted\_files":** Directory containing extracted text files.
* **" G:\filtered\_files":** Directory to save filtered text files.
* **"C:\path\to\chrome\profile" :** Default chrome profile path

## **Dependencies required**

* selenium
* pandas
* openpyxl
* beautifulsoup4
* nltk
* tqdm
* webdriver-manager