**AKNOWLEDGEMENT**

Before diving into the formalities, I would like to mention that I am very grateful for this opportunity.

There are some things to consider. I have overridden a couple of instructions here:

* I was instructed to provide only certain necessary files in the submission. But for ease of running the project on another system, I am submitting the entire directory.
* It is better to split the project into separate files to reduce ambiguity and understand the workflow for anyone who is accessing the project.

**APPROACH TO THE SOLUTION**

The entire workflow of the project could be broken down into two main chunks:

1. Scraping Data
2. Text Analysis on data
3. SCRAPING THE DATA

For web scraping I have used scrapy here solely because of the volume of data to be scrapped. On comparison to Selenium or BeautifulSoup , Scrapy is exponentially faster and can directly extract info without loading the site. It scrapes directly from the page source.

On the first attempt, I was able to successfully scrape 88 pages out of 100, But there were some issues to be handled with the other sites. Before the next attempt, the other websites are separated and scraping is performed with different variations of selectors. In this attempt out of the remaining 12 I was able to successfully scrape 9 pages.

The remaining 3 pages had some vague issues. Out of that two pages are manually scraped and added. And one page “blackassign0036” is not accessible in the web.

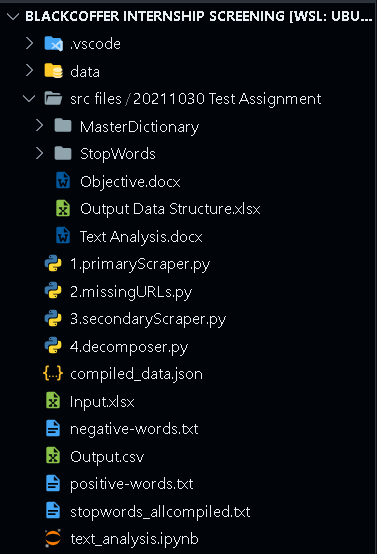
All the data is initially scraped into a json file. Json is used solely because of ease of accessibility. Later the files are decomposed into their separate text files based on their “URL\_ID”. These files are kept under the “data” directory.

1. TEXT ANALYSIS ON DATA

The analysis part is pretty straightforward. First the stopwords are all compiled into a single file to make the job easier, and then used in the project. The positive and negative words are saved as dictionaries with their weights given appropriately.Then other columns like positive score, negative score, polarity score, subjectivity score, word count, average words in a sentence, fog index are all easily computed based on their given definitions.

A difficulty I faced was at complex word feature and more specifically the syllable part. But thankfully it got resolved by using the cmu dictionary module in nltk. And all other required columns are found accordingly. The dataframe is then cleaned, formatted and then merged with URLs correctly using the URL\_ID as primary key.

**RUNNING THE PROJECT ON A LOCAL MACHINE**



It is preferred to follow the file directory structure as given in the image.

The important thing to follow is that the following files must be saved in the same directory to avoid path issues.

* 1.primaryScraper.py
* 2.missingURLs.py
* 3.secondaryScraper.py
* 4.decomposer.py
* compiled\_data.json
* Input.xlsx
* negative-words.txt
* positive-words.txt
* stopwords\_allcombined.txt
* text\_analysis.ipynb

1. Run is “1.primaryScraper.py”. This code scrapes the websites for the required info.
2. Run the “2.missingURLs.py”. This code checks for the errors occurred in certain sites due to which scraping could not be done successfully.
3. Run the “3.secondaryScraper.py”. This code resolves the issues and scrapes the data successfully from those websites
4. Run the “4.decomposer.py”. This code stores the data scraped into neatly containerized separate and individual text documents.
5. Before proceeding to text analysis, all the stopwords must be combined into a single file to make the process easier. The “stopwords\_allcompiled.txt” is this file.
6. Run all the code blocks in the “text\_analysis.ipynb”.
7. The required “Output.csv” file can be obtained after this in the parent directory.

**DEPENDENCIES REQUIRED**

These are the various python libraries and modules required.

1. scrapy
2. pandas
3. numpy
4. json
5. nltk – tokenizer
6. nltk – corpus
7. nltk – punkt
8. nltk – cmu\_dict

Since all the methods and modules we are using are very basic and are very foundational features, There isn’t much inter dependency. But still, it is better to keep all of them to the latest stable releases.