Instructions Documentation

# 1. How the Solution Was Approached

* **Step 1:** We first extracted the main article texts and titles for each URL in Input.xlsx and saved each article as a text file named <URL\_ID>.txt in the articles folder. Only the title and main article content were extracted.
* **Step 2:** We performed text analysis (sentiment, readability, and other linguistic metrics) as specified in Text Analysis.docx for each article file.
* **Stopwords Handling:** All stopword files in the StopWords folder (including "GenericLong") are loaded and merged into a single set for consistent cleaning across the pipeline.
* **Positive/Negative Words:** Positive and negative word lists are loaded from your provided dictionary files, with any stopwords excluded.
* **Text Processing & Metrics:** For each article file, text is cleaned and tokenized. Metrics are computed as defined (positive score, negative score, polarity, subjectivity, average sentence length, complex words, Fog index, syllables, personal pronouns, etc.).
* **Output Structure:** The script reads Output Data Structure.xlsx to ensure that the results are saved in the exact column order required for evaluation, and joins the calculated metrics with the original Input.xlsx columns (such as URL\_ID, URL, etc).
* **Error Handling:** All files are read with errors='ignore' to bypass any encoding issues that were encountered during testing, as many dictionary files and stopword files contain non-UTF-8 characters.

# 2. How to Run the .py File to Generate Output

**Before you run**

* Ensure your environment has all required dependencies (see below).
* Place your files in the following structure (update drive/folder names if needed):

D:/Blackcoffer/

├── Input.xlsx

├── Output Data Structure.xlsx

├── articles/

│ ├── 1.txt

│ ├── 2.txt

│ └── ...

├── StopWords-20250702T160606Z-1-001/

│ └── StopWords/

│ ├── StopWords\_Generic.txt

│ ├── StopWords\_GenericLong.txt

│ ├── StopWords\_Names.txt

│ └── (other stopwords .txt files)

├── MasterDictionary-20250702T160600Z-1-001/

│ └── MasterDictionary/

│ ├── positive-words.txt

│ └── negative-words.txt

├── Data\_Analysis.py # (your script)

**To Run:**

1. Open a terminal (cmd or PowerShell).
2. Navigate to the folder containing Data\_Analysis.py.
3. Type the following and press Enter:

python Data\_Analysis.py

1. The script will display progress messages. When finished, your output will be saved as Output.xlsx in your D:/Blackcoffer/ folder.

# 3. Dependencies Required

Install these with pip if they are not already installed:

pip install pandas numpy nltk openpyxl

* **pandas**: For working with Excel and tabular data.
* **numpy**: For numerical operations.
* **nltk**: For natural language processing (sentence and word tokenization).
* **openpyxl**: For Excel file support in pandas.

# 4. Specific Changes/Choices Made to Counter Errors

* **UnicodeDecodeError:**  
  Some stopword and wordlist files contain non-UTF-8 characters.  
  **Solution:** All file reading (for stopwords, positive/negative words) uses the errors='ignore' parameter:

with open(fpath, 'r', encoding='utf-8', errors='ignore') as f:

This skips any problematic bytes and prevents the script from crashing.

* **File/folder existence:** Used os.makedirs('articles', exist\_ok=True) (if relevant) to avoid errors if the folder exists.
* **Robust matching and joining:** Used the Input.xlsx mapping to ensure that every output row is matched correctly with the input columns as required.

# Support / Troubleshooting

* **If you get "file not found" errors**: Double-check all file/folder paths at the top of the script and ensure files are present.
* **If you get "UnicodeDecodeError"**: Confirm you have errors='ignore' in all open() calls reading dictionary/stopwords files.
* **If you get a missing module error**: Run pip install ... as shown above.

**How to Adapt**

* If you move files/folders, update the \*\_PATH and folder variables at the top of your script.
* You can add new stopwords or dictionary files to the appropriate folders; the script will automatically pick them up.
* The output Excel will always follow the structure given in Output Data Structure.xlsx.