**Text Analysis Instructions Documentation**

**1. Approach to the Solution**

The solution for this assignment is divided into several steps, each addressing a specific task as follows:

1. **Text Preprocessing and Tokenization**:
   * The text is cleaned by removing stop words, punctuation, and unnecessary characters to ensure that only meaningful words remain for analysis.
   * Tokenization is performed to split the text into individual words and sentences, which helps in calculating metrics like word count, sentence length, and syllable count.
2. **Sentiment Analysis and Derived Variables**:
   * The code loads the Positive and Negative dictionaries from the MasterDictionary to calculate **Positive Score** and **Negative Score**.
   * Using these scores, **Polarity Score** and **Subjectivity Score** are computed. Polarity helps determine if the text is positive or negative, while Subjectivity indicates if the text is objective or subjective.
3. **Readability Analysis (Gunning Fog Index)**:
   * The **Gunning Fog Index** is calculated to determine readability based on sentence complexity and sentence length.
   * Complex words are identified as those containing more than two syllables.
4. **Text Metrics Calculations**:
   * Various metrics are calculated to provide insights into the text structure:
     + **Average Words per Sentence** is determined by dividing the total number of words by the number of sentences.
     + **Complex Word Count** is calculated by identifying words with more than two syllables.
     + **Total Word Count** is calculated after removing stop words and punctuation.
     + **Personal Pronouns Count** is calculated using regex to match specific pronouns while avoiding false positives (like the country “US”).
     + **Average Word Length** is computed by averaging the length of words in the text.
5. **Final Output**:
   * The derived metrics are displayed as output, providing a comprehensive analysis of the text's sentiment, readability, and structure.

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

**Steps to Run:**

1. Ensure all dependencies are installed (see Section 3).
2. Place the .py file containing the code (e.g., text\_analysis.py) and any required folders (such as StopWords and MasterDictionary) in the same directory.
3. Open a terminal or command prompt in this directory.
4. Run the script using the following command:

bash

Copy code

python text\_analysis.py

**Expected Output:**

The script will display the calculated metrics in the terminal. If you want to save the output to a file, you can modify the code to write results to a .txt or .csv file, or redirect the output in the terminal:

bash

Copy code

python text\_analysis.py > output.txt

**3. Dependencies Required**

To run the script successfully, you need the following dependencies:

* **Python** (3.6 or higher)
* **NLTK** (Natural Language Toolkit) for text processing and tokenization:
  + Install NLTK by running:

bash

Copy code

pip install nltk

* + Download required NLTK resources (stop words and tokenizers) in the script by including:

python

Copy code

import nltk

nltk.download('punkt')

nltk.download('stopwords')

* **Regex** (part of Python's standard library) for handling personal pronoun extraction.

Ensure the StopWords folder and MasterDictionary files are accessible in the directory to avoid any file path issues during runtime.