Text Analysis project

Objective: The objective of this assignment is to extract textual data articles from the given URL and perform text analysis to compute variables that are explained below.

Steps Involved :

To accomplish this assignment, you would typically follow these key steps:

1. Web scraping:
   - Use a library like Beautiful Soup or Scrapy to extract the textual content from the given URL.
   - Handle any authentication or dynamic content loading if necessary.

2. Text preprocessing:
   - Remove HTML tags and other non-textual elements.
   - Clean the text by removing special characters, extra whitespace, etc.
   - Tokenize the text into words or sentences as needed.

3. Text analysis:
   - Apply various natural language processing (NLP) techniques to compute the required variables. These might include:
     a. Word frequency analysis
     b. Sentiment analysis
     c. Named entity recognition
     d. Topic modelling
     e. Readability scores
     f. Text complexity measures

4. Variable computation:
   - Calculate specific metrics based on the assignment requirements. These could involve:
     a. Word counts
     b. Sentence structure analysis
     c. Use of specific language features (e.g., passive voice, complex words)
     d. Sentiment scores
     e. Subject matter categorization

5. Data storage and presentation:
   - Store the computed variables in a suitable format (e.g., CSV, JSON).
   - Create visualisations or summaries of the findings if required.

Required python Libraries :

```
import pandas as pd
import requests
from bs4 import BeautifulSoup
import openpyxl
from urllib.parse import urlparse
Import textBlob
import nltk
from nltk.tokenize import word_tokenize, sent_tokenize
from nltk.corpus import stopwords, cmudict
import textstat
import re
import os
import chardet
from collections import Counter
import string
import math
```

Importance of each modules in python

1. pandas (pd)
   - Reading and writing Excel files
   - Data manipulation and analysis
   - Creating DataFrames for structured data handling

2. requests
   - Sending HTTP requests to web pages
   - Fetching HTML content from URLs

3. beautifulsoup4 (bs4)
   - Parsing HTML content
   - Extracting specific elements from web pages
   - Removing unwanted HTML tags

4. openpyxl
   - Detailed Excel file manipulation (used internally by pandas)
   - Creating, reading, and writing .xlsx files

5. urllib.parse
   - Parsing and manipulating URL strings
   - Extracting components of URLs

6. nltk (Natural Language Toolkit)
   - Tokenization (splitting text into words or sentences)
   - Part-of-speech tagging
   - Accessing linguistic resources (e.g., stopwords)

- Syllable counting (using cmudict)

7. textstat
   - Calculating readability scores (e.g., Flesch Reading Ease, Fog Index)
   - Counting syllables, sentences, and words

8. re (Regular Expressions)
   - Pattern matching in strings
   - Complex text parsing and manipulation

9. os
   - File and directory operations
   - Handling file paths across different operating systems

10. chardet
    - Detecting the encoding of text files
    - Helping to read files with unknown encodings

11. collections
    - Using specialized container datatypes
    - Counter for efficient counting of items (e.g., word frequency)

12. string
    - Accessing string constants (e.g., punctuation)
    - String manipulation operations

13. math
    - Performing mathematical operations
    - Used in various calculations for text analysis metrics