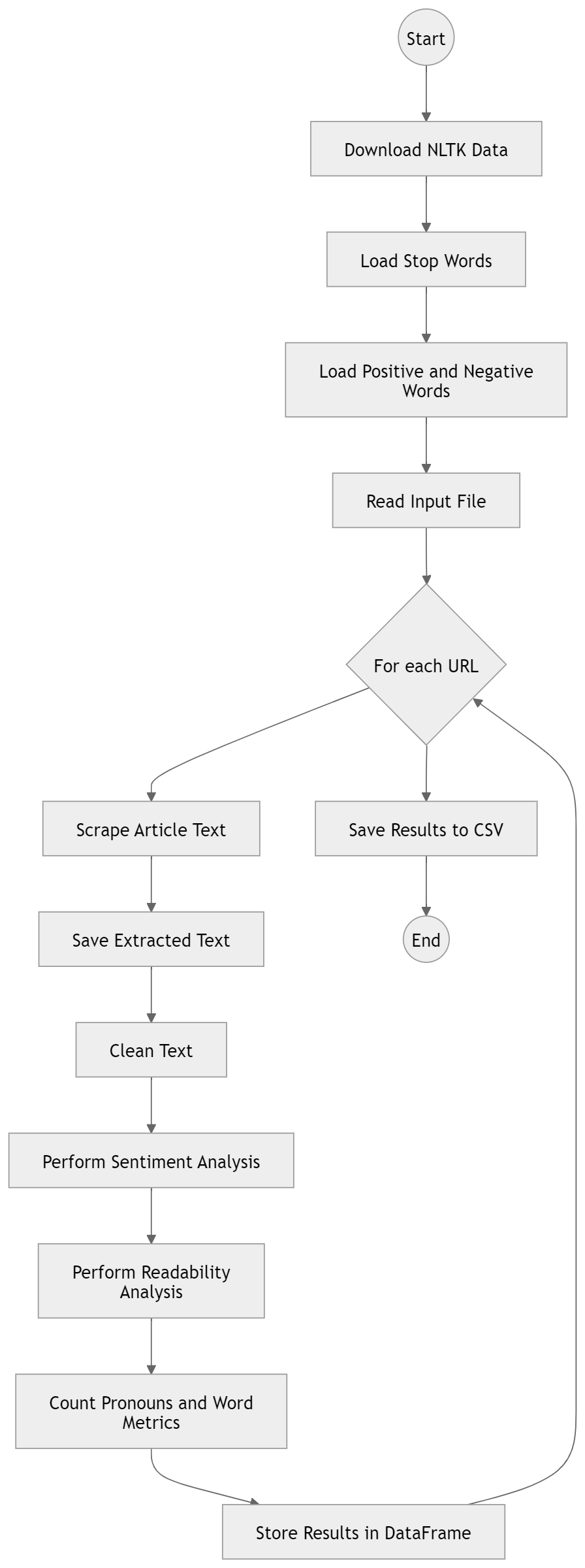
Instructions Documentation

# The Approach:



- Web Scraping  
- Text Cleaning   
- Sentiment and Readability Analysis  
- Output

**Step:1**

Used pandas to load the Input url files and identify URL and the URL\_ID column.

**Step:2**

Used BeautifulSoup library to extract the text data from the url excluding HTML tags and attributes. And saved the extracted text into the file with its URL\_ID name as instructed in Objective file of assignment.

**Step:3**

Used nltk library to clean the text after removing the stop words(provided) and naming that clean\_text as the tokens and passing to another function which will take the tokens as the input and then find all the sentiment score using the formula provided in the assignment, also we can analyze the readability using the nltk library.

**[Basically we are breaking the paragraph in the list of words using the clean\_text() local function and breaking the paragraph into the list of sentences using the nltk.sent\_tokenize() function]**

**Step:4**

Creating a empty data frame using pandas with the name of output\_df and inserting all the required column in the output data structure. After using the operators and formulas provided we made a data frame with all the fields and records.

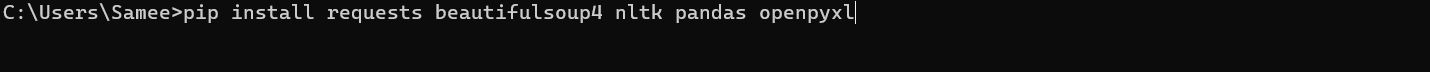
Note:- We can use another approach to append the **new\_row** into the previous made dataframe **Output\_df** in step 4 but it was not working (I think pandas version issue) so, I used concatenation property to concatenate the **output\_df** dataframe and the **new\_row** dataframe.

**Step:5**

Used a loop to repeat all the 4 step to find the same for the other links in the **Input.xlsx** file.

# 2. How to Run the .py File:

1. Dependencies:  
 - Install the required libraries by running:  
 pip install requests beautifulsoup4 nltk pandas openpyxl

  
 - The script will automatically download the necessary NLTK resources (punkt) if not already installed.  
  
3. Directory Setup:  
 - Ensure the StopWords folder (containing .txt files) and the MasterDictionary folder (containing positive-words.txt and negative-words.txt) are present in the same directory as the script.

(You can also make some modifications in the script to take path everytime from the user when running the code using input(“enter the path to Positive words”) function or you can change the path directly in the script)

  
  
4. Input File:  
 - Place an Input.xlsx file in the script's directory. This file should contain two columns: URL\_ID and URL.

(If changing the format or the directory of the Input.xlsx file you have to update the script.)  
  
5. Execution:  
 - Run the script:  
 python script\_name.py  
 - The extracted articles will be saved in the extracted\_articles folder.  
 - The results will be saved in Output.csv.

# 3. Required Dependencies:

**os**: File and folder operations (reading/creating)  
**requests**: Sending HTTP requests  
**beautifulsoup4**: Web scraping  
**nltk**: Text data cleaning  
**pandas**: DataFrame and Excel file handling  
**re**: Searching for specific words (e.g., pronouns)