Approach

STEP 1:

Initially i have scraped data from the URL provided in NLP\_task/Provided\_data/Input.xlsx file using libraries

Requests, BeautifulSoup.

Then i have done some preprocessing on the extracted text such as:

Remove numbers, remove url , remove emails and corrected spelling using libraries

re, TextBlob.

Then formed a dataframe of the scraped and preprocessed data as : NLP\_task/Web\_scraping/web\_scraped\_data.csv

STEP 2:

Imported **Scraped dataset** NLP\_task/Web\_scraping/web\_scraped\_data.csv

and **Output dataset** NLP\_task/Provided\_data/Output\_Data\_Structure.xlsx

which contains the structure of final output.

Imported some of directories for sentimental analysis

**SENTIMENTAL ANALYSIS**

## **Cleaning using Stop Words Lists**

Firstly i have combined all the Stopword list that were provided to me into single SET stopwords = load\_stop\_words(mydir) where mydir = 'NLP\_task/Provided\_data/Sentimental\_analysis\_data/StopWords'

as easy to check values from set if they are present or not.

Then simply removed the Stopwords from every rows of **Scapred Dataset**

# **Creating a dictionary of Positive and Negative word**

I have have loaded both positive and negative word text file into:

positive\_words = load\_word\_list(positivedictionaydir)

negative\_words = load\_word\_list(negativedictionaydir)

1. **Extracting Derived variables**

Perform sentiment analysis based on positive and negative word lists using function positive\_negative\_score which basically iterates through Scraped data and count positive and negative score

Then we Calculate polarity score and subjectivity score using these formulas

Polarity\_Score = (positive\_score - negative\_score) / (positive\_score + negative\_score + 0.000001)

Subjectivity\_Score = (positive\_score + negative\_score) / (words\_count + 0.000001)

Finally process all these steps on each row of Scraped Dataset and append the values in Output Dataset and export it

output\_dataset.to\_csv('NLP\_task/Sentimental\_Analysis/Output\_Data\_Structure.csv', index=False)

STEP 3:

1. Initialized dataset

dataset = pd.read\_csv('NLP\_task/Web\_scraping/web\_scraped\_data.csv')

output\_dataset = pd.read\_csv('NLP\_task/Sentimental\_Analysis/Output\_Data\_Structure.csv')

2. Imported required libraries

import pandas as pd

import re

import spacy

from nltk.corpus import cmudict

import json

from nltk.corpus import stopwords

from tensorflow.keras.preprocessing.text import Tokenizer

3. As data contained few words like acb.zyx we removed that ‘.’ with DOT do that there will be no issue while splitting the text dot\_removal\_text

4. Then we counted sentences. First I used split(‘\n’) to split every sentence because when we use the input dataset it contain ‘\n’ to split the line then I used nlp = spacy.load('en\_core\_web\_sm') to separate line more thoroughly and made a plain\_text which contains the whole text splitted into more sophisticated way containing only ‘.’ as line separator left then we split the sentences with ‘.’ and counted it.

dot\_removed\_text = dot\_removal\_text(plaintext)

total\_sen = total\_sentence(article\_text)

5. We used NLTK stopwords libraries to get cleaned text and converted into SET because it's easy to use as compared to a dictionary. Then used Regular Expressions to remove PUNCTUATIONs from the text and counted number of words in cleaned text

stopwords\_removed = remove\_stopwords(article\_text)

article\_no\_punctuation = remove\_punctuation(stopwords\_removed)

total\_words = counts(article\_no\_punctuation)

6. Then we calculated Average Word Length using formula

total\_characters / total\_words

average\_word\_length\_value = calculate\_average\_word\_length(article\_no\_punctuation)

used article\_no\_puntuation because now it's the cleaned we have acquired from above steps

7. Now we calculated the syllable count which will be a dictionary of the number of syllables in each word and to do that I have created a no\_vowels function which basically counts the number of vowels in each word and exceptions for ‘ed’ and ‘es’ were made. Then I have used the no\_vowels function in syllable count function while iterating through each word and finding its values and storing it in the dictionary and used artucle\_no\_punctutation text as it’s cleaned and used .split() to provide single words.

syllable\_words = syllable\_count\_per\_word(article\_no\_punctuation.split())

syllable\_words\_str = json.dumps(syllable\_words)

and converted it into json to append it into output\_dataset

8. Now we calculate the number of complex word which means words with more than **two syllables** so we simply used syllable\_words as an input for out function which will iterate whole dictionary and check for condition syllable\_count > min\_syllable

and count number of words

complex\_words = complex\_word\_count(syllable\_words)

9. Now we have calculated number of personal pronouns using Regular Expression

pronouns\_pattern = r'\b(I|we|my|ours|us)\b'

pronouns\_count = count\_personal\_pronouns(article\_text)

10. At last few more values were created with help of the output we gained above with formulas :

avg\_word\_length = average\_word\_length\_value

Average\_Sentence\_Length = total\_words / total\_sen if total\_sen > 0 else 0

Percentage\_of\_Complex\_words = complex\_words / total\_words if total\_words > 0 else 0

Fog\_Index = 0.4 \* (Average\_Sentence\_Length + Percentage\_of\_Complex\_words)

Finally process all these steps on each row of Scraped Dataset and append the values in Output Dataset and export it

output\_dataset.to\_csv('NLP\_task/Task\_2\_to\_8/final\_output.csv', index=False)

as our final output file

How to run the .py file

Structure your file into this format

├── Provided\_data

│ ├── Input.xlsx

│ ├── Output\_Data\_Structure.xlsx

│ └── Sentimental\_analysis\_data

│ ├── MasterDictionary

│ │ ├── negative-words.txt

│ │ └── positive-words.txt

│ └── StopWords

│ ├── StopWords\_Auditor.txt

│ ├── StopWords\_Currencies.txt

│ ├── StopWords\_DatesandNumbers.txt

│ ├── StopWords\_Generic.txt

│ ├── StopWords\_GenericLong.txt

│ ├── StopWords\_Geographic.txt

│ └── StopWords\_Names.txt

├── Sentimental\_Analysis

│ ├── Output\_Data\_Structure.csv

│ └── Sentimental\_analysis.ipynb

├── Task\_2\_to\_8

│ ├── Task\_2\_to\_8.ipynb

│ └── final\_output.csv

└── Web\_scraping

├── data\_extraction.ipynb

└── web\_scraped\_data.csv

**DEPENDENCIES**

**1: Commands to install dependencies:**

**pip install requests beautifulsoup4 pandas regex textblob spacy nltk tensorflow chardet**

**2: After installing nltk, you'll need to download additional data (like stopwords and tokenizers):**

**import nltk**

**nltk.download('punkt')**

**nltk.download('stopwords')**

**nltk.download('cmudict')**

**3: After installing spacy, you'll need to download the required language model:**

**python -m spacy download en\_core\_web\_sm**