WEB SCRAPED NLP TEXTUAL ANALYSIS

Abstract

This project aims to automate the extraction and analysis of textual data from online articles using Python programming language. Utilising web scraping tools such as BeautifulSoup, systematic retrieval and saving of article content were achieved. Subsequently, text analysis was conducted to compute various linguistic and readability metrics, including polarity, subjectivity, fog index, and word complexity. The final output conformed to a predefined structure for consistent reporting. This comprehensive pipeline provides valuable insights into content sentiment and quality, enabling enhanced content evaluation and categorization. The project demonstrates the effective application of Natural Language Processing (NLP) for automated data extraction, processing, and sentiment analysis from web-based textual sources.

Keywords

Web Scraping, NLP, Sentiment analysis, analysis of readability, Data analysis, Text Analysis, Python programming, Fog Index, Data Extraction, Automated Content Evaluation

Introduction

In today's data-driven world, the abundance of unstructured information, particularly in the form of online articles and blogs, presents both an opportunity and a challenge. While these textual resources are rich in knowledge and insights, extracting meaningful data and conducting comprehensive analysis from them requires sophisticated tools and techniques. This project aims to address this challenge by developing a Python-based solution to automate the extraction and analysis of online article content, thus enabling deeper insights through Natural Language Processing (NLP).

The project simulates a real-world application of text mining by combining web scraping and NLP to evaluate a set of articles based on predefined metrics. The primary objective is to extract article titles and their content from a list of URLs, clean the data, and perform a series of analytical operations that help quantify the content's sentiment, complexity, and readability. These operations are not only important for academic exercises but are also widely used in real-life applications like content moderation, SEO optimisation, media monitoring, and opinion mining.

The solution approach involves multiple components of text processing. Initially, URLs are scraped using Python libraries such as BeautifulSoup and Requests, with care taken to avoid extracting irrelevant parts like website headers or footers. After successful data extraction, textual analysis is performed using custom logic and NLP tools such as NLTK, with key metrics such as positive and negative sentiment scores, polarity, subjectivity, and personal pronoun usage being calculated.

Beyond sentiment analysis, this project delves into linguistic complexity by calculating features such as the FOG Index, average sentence length, and syllable count per word. These indicators reflect how difficult or easy the content is to understand. Such insights are particularly useful in fields like education technology, journalism, and marketing, where understanding content readability can shape

Vipul Gupta
20296303122

audience engagement strategies.

Furthermore, the implementation is structured to comply with a strict output format as specified in the assignment instructions. The final results are compiled into a structured CSV file that presents all computed variables in a tabular format. The Python code was modularized for reusability and clarity, while also providing the flexibility to be adapted for future extensions such as keyword extraction, named entity recognition, or multilingual analysis.

This project not only strengthened core programming skills but also offered practical exposure to working with real-world data and using open-source tools to tackle common challenges in data extraction and processing. Overall, this assignment exemplifies how the integration of web scraping and NLP can transform unstructured data into valuable, structured information suitable for further analysis and decision-making.

Requirements and Dependencies

1. A computer having a python interpreter or an IDE to execute the python code.
2. Following python libraries should be installed:
   a) numpy
   b) pandas
   c) BeautifulSoup
   d) warnings
   e) nltk
   f) requests
   g) re
   h) chardet
   i) openpyxl

   To install any of these libraries use the code: “pip install library_name”.
3. Store the input file given along the assignment with the path: “Question/Input.xlsx”.
4. Form a folder with name “Extracted Data” to store the data extracted from the URL in the dataset.
5. Store the stopwords attached to the assignment with the path: “Question/StopWords/StopWords_file_name”
6. Store the positive and negative words with the path: “Question/MasterDictionary/file_name”.

Tech Contribution

The successful execution of this project depends on the effective use of multiple technologies and Python libraries that contribute to each phase of the workflow, from data extraction to advanced text analysis. Here’s a detailed list of the prominent technologies and their functions:

1. **Python Programming Language:** Python, known for its simplicity, extensive library support, and suitability for data-centric applications, was the foundation of this project. Its modular syntax and robust community-supported packages made it ideal for web scraping, text processing, and data analysis workflows.

2. **BeautifulSoup:** A Python library specifically designed for web scraping, parsed HTML content and extracted article titles and body content from URLs by traversing the HTML structure and excluding extraneous components.
3. **Requests:** The Requests library transmitted HTTP requests to URLs and retrieved corresponding HTML pages, which were then processed by BeautifulSoup.
4. **Regular Expressions (re):** Python's re module was used for text cleaning and preprocessing, removing punctuation, filtering HTML tags, identifying personal pronouns, and extracting syllables for complexity analysis.
5. **NLTK (Natural Language Toolkit):** NLTK was instrumental in linguistic and syntactic analysis, tokenizing words and sentences, removing stopwords, identifying complex words, and calculating metrics like average sentence length, word count, and FOG index, which are indicators of readability.
6. **Pandas:** Pandas was used for data analysis and processing. Pandas was used extensively for data manipulation and organisation. It read the initial dataset (Input.xlsx), stored intermediate and final results, including scores and metrics, in dataframes, and created the final output file in CSV format. NumPy processed numeric arrays and applied formulas for metrics like polarity and subjectivity scores. Predefined stopword files and sentiment lexicons were integrated to filter irrelevant terms and score sentiment, contributing to accurate analysis.

Metrics used

- **Positive Score**: This score is calculated by assigning the value of +1 for each word if found in the Positive Dictionary and then adding up all the values.
- **Negative Score**: This score is calculated by assigning the value of -1 for each word if found in the Negative Dictionary and then adding up all the values. We multiply the score with -1 so that the score is a positive number.
- **Polarity Score**: This is the score that determines if a given text is positive or negative in nature. It is calculated by using the formula:
  Polarity Score = (Positive Score – Negative Score)/ ((Positive Score + Negative Score) + 0.000001)
  Range is from -1 to +1
- **Subjectivity Score**: This is the score that determines if a given text is objective or subjective. It is calculated by using the formula:
  Subjectivity Score = (Positive Score + Negative Score)/ ((Total Words after cleaning) + 0.000001)
  Range is from 0 to +1
- **Average Sentence Length** = the number of words / the number of sentences
- **Percentage of Complex words** = the number of complex words / the number of words
- **Fog Index** = 0.4 * (Average Sentence Length + Percentage of Complex words)
- **Average Number of Words Per Sentence =** the total number of words / the total number of sentences
- **Complex words** are words in the text that contain more than two syllables.
- **number of Syllables** in each word of the text is the count of vowels present in each word. We also handle some exceptions like words ending with "es","ed" by not counting them as a syllable.

Vipul Gupta
20296303122

- To calculate **Personal Pronouns** mentioned in the text, we use regex to find the counts of the words - "I," "we," "my," "ours," and "us". Special care is taken so that the country name US is not included in the list.
- **Average Word Length** is calculated by the formula:
  Sum of the total number of characters in each word/Total number of words

Organization

1. **Requirement Analysis**
   - Reviewed the assignment objective to extract article content from URLs and compute text-based metrics.
   - Understood the variables and output structure from the provided documentation.
2. **Environment Setup**
   - Installed necessary Python libraries (beautifulsoup4, requests, nltk, pandas, etc.).
   - Created folder structure for input files, stopwords, dictionary files, and output storage.
3. **Article Extraction**
   - Fetched HTML content using requests.
   - Parsed article titles and bodies using BeautifulSoup, excluding irrelevant elements.
   - Saved cleaned articles as .txt files named by URL_ID.
4. **Text Preprocessing**
   - Cleaned text using regex and removed stopwords using NLTK and custom files.
   - Lowercased text for uniformity and consistency.
5. **Sentiment Analysis**
   - Loaded sentiment word lists.
   - Calculated positive, negative, polarity, and subjectivity scores.
6. **Readability & Complexity Metrics**
   - Used nltk for tokenization.
   - Computed metrics like FOG index, average sentence length, complex word count, and syllables per word.
7. **Additional Text Features**
   - Counted personal pronouns using regex.
   - Calculated word count and average word length.
8. **Output Compilation**
   - Structured results as per Output Data Structure.xlsx.
   - Exported final data to OUTPUT.csv.
9. **Final Submission**
   - Packaged Python script, output file, and documentation for submission as per instructions.

Vipul Gupta
20296303122

Proposed Workflow of the Project

Step 1: Import necessary libraries and the dataset

a) Import necessary libraries like numpy, pandas, BeautifulSoup, warnings, nltk, requests, re and openpyxl.
b) Import the input dataset given using pandas library.

Step 2: Data Extraction

a) Using BeautifulSoup library, construct a function to extract article text from one URL by studying any webpage given in the dataset.
b) Add the functionality of removing the HTML tags present in the text using regular expressions.
c) Loop the function created through the URL column and save the text in a text file with 'URL_UD' as its name.Also append it to the dataframe created while importing the dataset.

Step 3: Remove the stopwords of sentiment analysis

a) Determine the encoding of files given for stopwords.
b) Construct a function to extract stopwords from the files given and save them in a set. Also, lowercase the words for uniformity.
c) Construct a function to remove these stopwords from each article and apply it on the column containing article text.

Step 4: Performing sentiment analysis

a) Determine the encoding of files given for positive and negative words. Construct a function to extract these words from the files given and save them in a set. Also, lowercase the words for uniformity.
b) Remove URLs present and punctuation marks present in the text without stopwords.
c) Construct a function for calculating positive and negative score from a text and then pass article text without stopwords to the function to get scores for each row.
d) After calculating total words after cleaning of stopwords, find the polarity score and subjectivity score with the help of formulas given.

Step 5: Analysis of readability and complex word count:

a) Using nltk library, tokenize the article text and calculate total number of sentences and words in the article text. Also calculate average sentence length and average number of words per sentence using these.
b) Create a function to calculate total number of complex words keeping in mind the rules present in assignment guidelines for a complex word by not considering words ending with 'es' or 'ed'.
c) After passing article text through this function, find value of complex word count along with percentage of complex words using formula given.
d) Also calculate, Fog Index using these variables.

Step 6: Word Count

Vipul Gupta
20296303122

a) Using nltk library, tokenize the text and remove the stopwords present in article text using nltk stopwords package.
b) Also remove punctuation from the text, and calculate word count for the article text.

Step 6: Syllable Count per word

a) Using rules given in the assignment, create a function for constructing a list that gives number of syllables for each word in a sentence.
b) Apply the function through the article text column. Get the list for each article in the dataframe.

Step 7: Count Personal pronouns

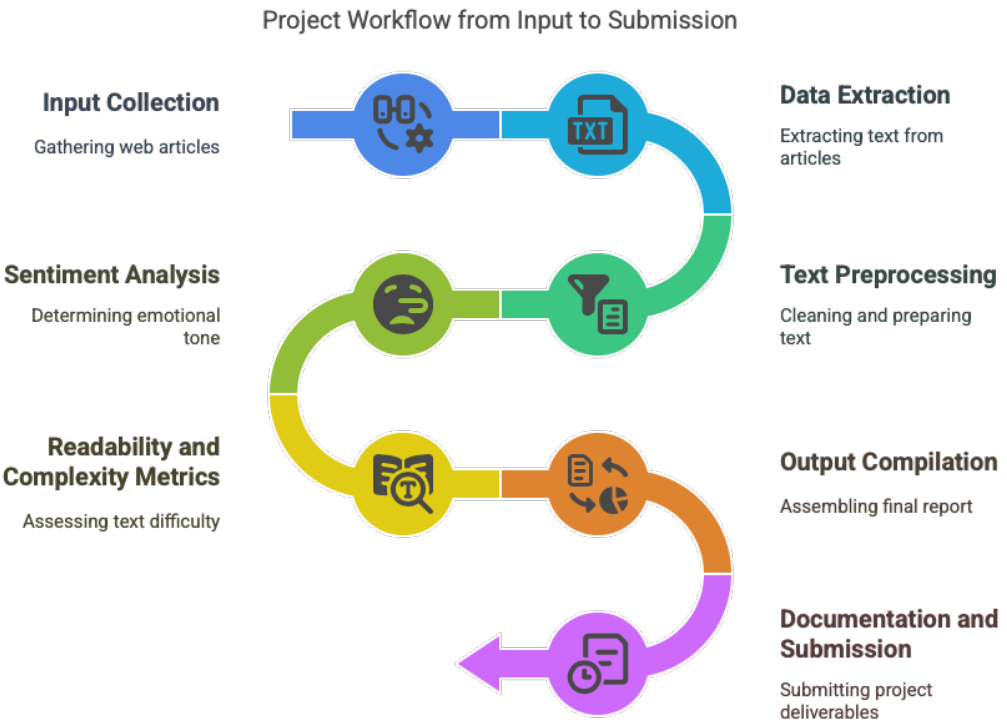
a) Remove the word 'US' for taking special care so that it doesn't occur in the list. Lowercase the text as well.
b) Using regular expressions count the number of personal pronouns.

Step 8: Average Word Length

a) Calculate the total number of characters present in the article after tokenizing the text.

Step 9: Forming dataframe as given in the output data structure.

a) Rename the required columns as given in the Output data structure.
b) Create a new dataframe and add all the columns sequentially as given in output data structure.
c) Save the dataframe created as a csv file as well using pandas library.



Vipul Gupta
20296303122

Results

The project successfully achieved its goal of extracting and analysing textual data from online articles using Python-based automation. The pipeline was developed to process a list of URLs, extract relevant text, clean and analyse it, and finally compile the results into a structured format.

The primary outcomes of this project are as follows:

1. **Accurate Text Extraction**
   - Only the article title and content were extracted from each URL using BeautifulSoup and requests, effectively removing unwanted web elements such as navigation bars, ads, and footers.
   - The extracted data was saved in separate .txt files named by URL_ID, ensuring traceability.

2. **Sentiment and Subjectivity Metrics**
   - Each article was analysed for sentiment using a custom positive and negative word list.
   - The project computed sentiment-based metrics including Positive Score, Negative Score, Polarity Score, and Subjectivity Score, enabling a deep understanding of article tone.

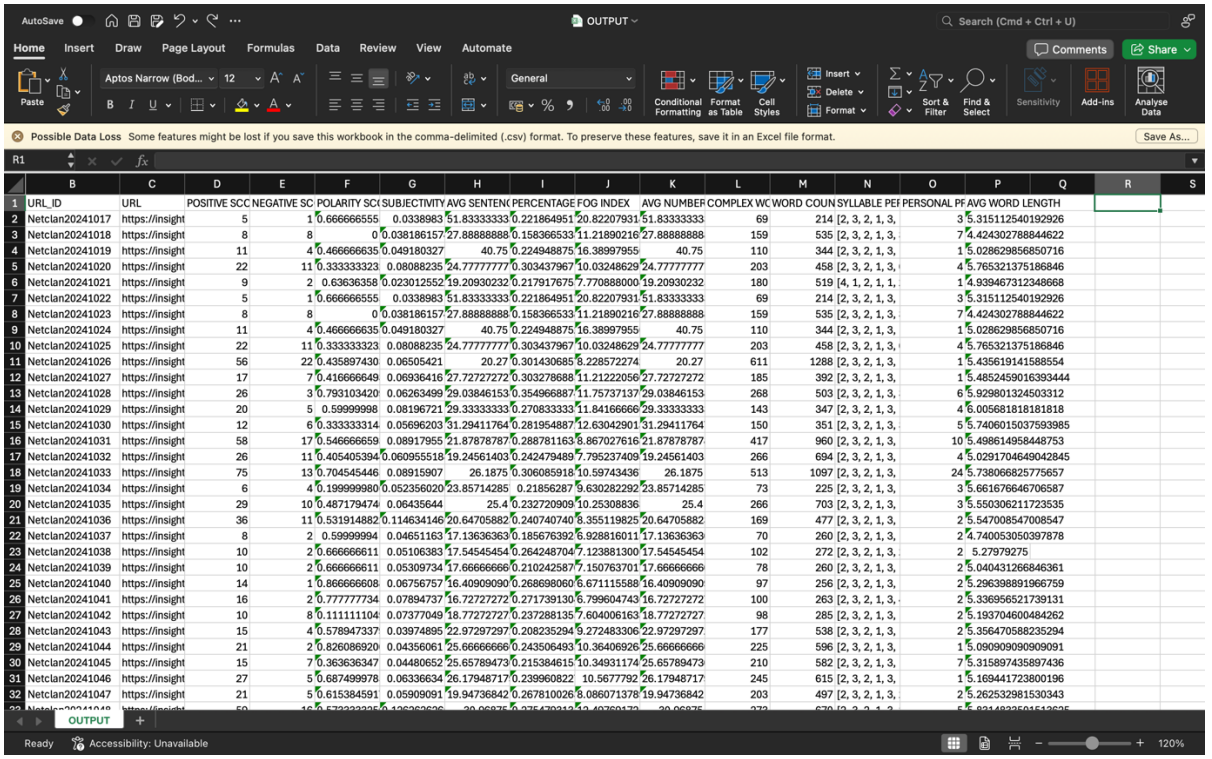
3. **Readability and Linguistic Features**
   - The articles were evaluated for complexity using standard readability formulas like the FOG Index.
   - Complex Word Count, Average Sentence Length, and Percentage of Complex Words were calculated to assess textual difficulty.

4. **Word-Level and Structural Insights**
   - Word Count, Syllables per Word, Personal Pronoun Count, and Average Word Length were calculated using regex and NLP techniques.
   - These metrics added granularity to the analysis and highlighted structural properties of the text.

5. **Final Output Compilation**
   - All analytical metrics, along with the input metadata, were compiled into a clean and organised OUTPUT.csv file.

Vipul Gupta
20296303122

Resulting CSV file:



	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	URL_ID	URL	POSITIVE SCC	NEGATIVE SC	POLARITY SC(	SUBJECTIVITY	AVG SENTEN(	PERCENTAGE	FOG INDEX	AVG NUMBER	COMPLEX WC	WORD COUN	SYLLABLE PEF	PERSONAL PF	AVG WORD LENGTH		
2	Netclan20241017	https://insight	5	1	0.666666555	0.0338983	51.83333333	0.221864951	20.82207931	51.83333333	69	214	[2, 3, 2, 1, 3,	3	5.315112540192926		
3	Netclan20241018	https://insight	8	8	0	0.038186157	27.88888888	0.158366533	11.21890216	27.88888888	159	535	[2, 3, 2, 1, 3,	7	4.424302788844622		
4	Netclan20241019	https://insight	11	4	0.466666635	0.049180327	40.75	0.224948875	16.38997955	40.75	110	344	[2, 3, 2, 1, 3,	1	5.028629856850716		
5	Netclan20241020	https://insight	22	11	0.333333323	0.08088235	24.77777777	0.303437967	10.03248629	24.77777777	203	458	[2, 3, 2, 1, 3,	4	5.765321375186846		
6	Netclan20241021	https://insight	9	2	0.63636358	0.023012552	19.20930232	0.217917675	7.770888000	19.20930232	180	519	[4, 1, 2, 1, 1,	1	4.939467312348668		
7	Netclan20241022	https://insight	5	1	0.666666555	0.0338983	51.83333333	0.221864951	20.82207931	51.83333333	69	214	[2, 3, 2, 1, 3,	3	5.315112540192926		
8	Netclan20241023	https://insight	8	8	0	0.038186157	27.88888888	0.158366533	11.21890216	27.88888888	159	535	[2, 3, 2, 1, 3,	7	4.424302788844622		
9	Netclan20241024	https://insight	11	4	0.466666635	0.049180327	40.75	0.224948875	16.38997955	40.75	110	344	[2, 3, 2, 1, 3,	1	5.028629856850716		
10	Netclan20241025	https://insight	22	11	0.333333323	0.08088235	24.77777777	0.303437967	10.03248629	24.77777777	203	458	[2, 3, 2, 1, 3,	4	5.765321375186846		
11	Netclan20241026	https://insight	56	22	0.435897430	0.06505421	20.27	0.301430685	8.228572274	20.27	611	1288	[2, 3, 2, 1, 3,	1	5.435619141588554		
12	Netclan20241027	https://insight	17	7	0.416666649	0.06936416	27.72727272	0.303278688	11.21222056	27.72727272	185	392	[2, 3, 2, 1, 3,	1	5.485245916393444		
13	Netclan20241028	https://insight	26	3	0.793103420	0.06263499	29.03846153	0.354966887	11.75737137	29.03846153	268	503	[2, 3, 2, 1, 3,	6	5.929801324503312		
14	Netclan20241029	https://insight	20	5	0.59999998	0.08196721	29.33333333	0.270833333	11.84166666	29.33333333	143	347	[2, 3, 2, 1, 3,	4	6.005681818181818		
15	Netclan20241030	https://insight	12	6	0.333333314	0.05696203	31.29411764	0.281954887	12.63042901	31.29411764	150	351	[2, 3, 2, 1, 3,	5	5.740601503593985		
16	Netclan20241031	https://insight	58	17	0.546666659	0.08917955	21.87878787	0.288781163	8.867027616	21.87878787	417	960	[2, 3, 2, 1, 3,	10	5.498614958448753		
17	Netclan20241032	https://insight	26	11	0.405405394	0.060955518	19.24561403	0.242479489	7.795237409	19.24561403	266	694	[2, 3, 2, 1, 3,	4	5.029170649042845		
18	Netclan20241033	https://insight	75	13	0.704545446	0.08915907	26.1875	0.306085918	10.59743436	26.1875	513	1097	[2, 3, 2, 1, 3,	24	5.738066825775657		
19	Netclan20241034	https://insight	6	4	0.199999980	0.052356020	23.85714285	0.21856287	9.630282292	23.85714285	73	225	[2, 3, 2, 1, 3,	3	5.661676646706587		
20	Netclan20241035	https://insight	29	10	0.487179474	0.06435644	25.4	0.232720909	10.25308836	25.4	266	703	[2, 3, 2, 1, 3,	3	5.550306211723535		
21	Netclan20241036	https://insight	36	11	0.531914882	0.114634146	20.64705882	0.240740740	8.355119825	20.64705882	169	477	[2, 3, 2, 1, 3,	2	5.547008547008547		
22	Netclan20241037	https://insight	8	2	0.59999994	0.04651163	17.13636363	0.185676392	6.928816011	17.13636363	70	260	[2, 3, 2, 1, 3,	2	4.740053050397878		
23	Netclan20241038	https://insight	10	2	0.666666611	0.05106383	17.54545454	0.264248704	7.123881300	17.54545454	102	272	[2, 3, 2, 1, 3,	2	5.27979275		
24	Netclan20241039	https://insight	10	2	0.666666611	0.05309734	17.66666666	0.210242587	7.150763701	17.66666666	78	260	[2, 3, 2, 1, 3,	2	5.040431266846361		
25	Netclan20241040	https://insight	14	1	0.866666608	0.06756757	16.40909090	0.268698060	6.671115588	16.40909090	97	256	[2, 3, 2, 1, 3,	2	5.296398891966759		
26	Netclan20241041	https://insight	16	2	0.777777734	0.07894737	16.72727272	0.271739130	6.799604743	16.72727272	100	263	[2, 3, 2, 1, 3,	2	5.336956521739131		
27	Netclan20241042	https://insight	10	8	0.111111104	0.07377049	18.77272727	0.237288135	7.604006163	18.77272727	98	285	[2, 3, 2, 1, 3,	2	5.193704600484262		
28	Netclan20241043	https://insight	15	4	0.578947337	0.03974895	22.97297297	0.208235294	9.272483306	22.97297297	177	538	[2, 3, 2, 1, 3,	2	5.356470588235294		
29	Netclan20241044	https://insight	21	2	0.826086920	0.04356061	25.66666666	0.243506493	10.36406926	25.66666666	225	596	[2, 3, 2, 1, 3,	1	5.090909090909091		
30	Netclan20241045	https://insight	15	7	0.363636347	0.04480652	25.65789473	0.215384615	10.34931174	25.65789473	210	582	[2, 3, 2, 1, 3,	7	5.315897435897436		
31	Netclan20241046	https://insight	27	5	0.687499978	0.06336634	26.17948717	0.239960822	10.5677792	26.17948717	245	615	[2, 3, 2, 1, 3,	1	5.169441723800196		
32	Netclan20241047	https://insight	21	5	0.615384591	0.05909091	19.94736842	0.267810026	8.086071378	19.94736842	203	497	[2, 3, 2, 1, 3,	2	5.262532981530343		

Conclusion

This project successfully demonstrated the integration of web scraping and Natural Language Processing (NLP) to extract, clean, and analyse online article content. By automating the entire pipeline using Python, the system was able to fetch articles from a given set of URLs, process them efficiently, and generate valuable sentiment and readability metrics.

All predefined output variables were computed and structured into a standardised CSV file, matching the expected format. The implementation showcases a practical application of text analysis using libraries such as BeautifulSoup, NLTK, pandas, and python programming. The metrics derived—such as polarity score, FOG index, word count, and personal pronoun usage—offer meaningful insights into the sentiment and complexity of the articles.

The project helped reinforce foundational concepts in web automation, linguistic analysis, and data wrangling, while also developing problem-solving and software structuring skills. Overall, it provides a solid foundation for scaling to more advanced real-world applications.

- **Future ScopeReal-time Data Processing:**
  Extend the system to operate in real time by integrating live RSS feeds or APIs for continuously updated article analysis.

- **Multilingual Support:**
  Integrate language detection and translation modules to analyse articles in multiple languages.

Vipul Gupta
20296303122

- **Advanced Sentiment Models:**
  Replace dictionary-based sentiment scoring with deep learning or transformer-based models (e.g., BERT) for more nuanced emotional understanding.

- **Named Entity Recognition (NER):**
  Add NER to extract and classify people, places, organisations, and dates mentioned in the articles.

- **Visualisation Dashboard:**
  Create a web-based or desktop dashboard using tools like Plotly, Dash, or Streamlit for dynamic visualisation of the results.

- **Content Categorization:**
  Implement topic modelling techniques like LDA to group articles by subject matter or theme.

- **SEO Optimisation Analysis:**
  Add features to assess content quality based on SEO metrics such as keyword density and readability grade level.

Vipul Gupta
20296303122