श्रमोऽनवरत चेष्टाय

NIT·PATNA

National Institute of Technology Patna

PYTHON PROJECT

Web Page Word Frequency Analyzer

Analyze the frequency of words on a web page and display the top 10 most common words.

SUBMITTED TO :

Dr Akshay Deepak sir

Date – 21/11/2024

SUBMITTED BY:

Suryash Nigam

Sushant Kumar

Anoop Kumar Dwivedi

Objective: Analyze the frequency of words on a web page and display the top 10 most common words.

Instructions:

- Use urllib to retrieve the content of a web page.
- Use BeautifulSoup to extract text from paragraphs (<p> tags) in the page.
- Clean the extracted text by removing punctuation and converting it to lowercase.
- Split the text into words and count the frequency of each word using a dictionary.
- Print the top 10 most frequent words and their counts.

Project Overview :

The Web page word Frequency Analyzer works by interacting with a web page and processing its textual content to count the frequency of each word .

The steps involve include:

- Retrieving the content of web page using urllib.
- Extracting and processing the text using BeautifulSoup to prase HTML .
- Cleaning the text by removing punctuation and converting to lowercase.

- Splitting the text into words and counting their frequency using a dictionary .

- Displaying the to 10 most frequent words and their counts.

3. Tools and Technologies Used:

- **python version 3.12(64 bit**): the programming language used for implementing the solution.

- **urllib:** A Python library used to open and read URLs, retrieve the content of a web page.

- **BeautifulSoup:** A library used for parsing HTML and extracting the text from the <p> tags in the web page.

- **collections.Counter:** A Python data structure to count the frequency of elements in an iterable, which is used to count word frequencies.

- **string**: A built-in Python library used to handle punctuation and clean the text.

4. Approach:

The approach to solving the assignment follows a step-by-step method:

- **4.1 Fetching the Web Page Content:**

The first step involves retrieving the content of a web page. The urllib.request.urlopen() function is used to send an HTTP GET request to the URL provided by the user and fetch the HTML content of the page.

4.2 Extracting Text from the Web Page:

- After retrieving the HTML content, the BeautifulSoup library is used to parse the HTML structure. The text inside <p> tags (which typically contains the main content of the page) is extracted and stored in a string.

4.3 Cleaning and Processing the Text:

- The extracted text is cleaned to ensure that it can be processed effectively:

- All the text is converted to lowercase to ensure uniformity and avoid treating words with different cases as different.

- Punctuation marks are removed using str.maketrans() to ensure that punctuation does not interfere with word splitting.

- The cleaned text is split into individual words by using the split() function, which breaks the text into a list of words.

4.4 Counting Word Frequency:

- Once the text is cleaned and split into words, a Counter object from the collections module is used to count how many times each word appears in the text.

4.5 Displaying the Top 10 Most Frequent Words:

- The most_common() method of the Counter class is used to get the top 10 most frequent words along with their frequency counts. The results are then printed to the screen.

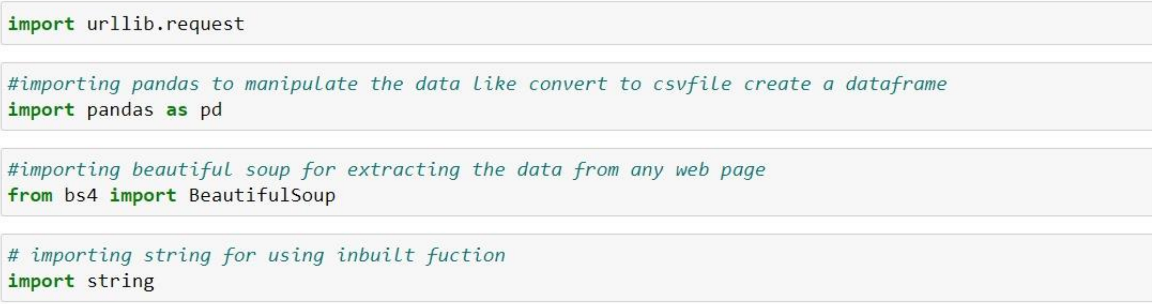
```
import urllib.request
from bs4 import BeautifulSoup
import string
from Collections import counter

# Specify the URL of the web page to analyze

url = "https://www.bbc.com/news"
# Replace with the desired URL

# Retrieve content of the web page

response = urllib.request.urlopen(url)
html = response.read()

# Extract text from paragraphs in the HTML

soup = BeautifulSoup(html, 'html.parser')
paragraphs = soup.find_all('p', class_ = "sc-b8778340-4")
text = ' '.join([p.get_text() for p in paragraphs])

# print(text)

# Clean the text by converting it to lowercase and removing
punctuation

text = text.translate(str.maketrans('', '', string.punctuation))  #
Remove punctuation
text = text.lower()  # Convert to lowercase
```

```
# split each word

words = text.split()
print(words)
my_dict = {}

for word in words:
    if word in my_dict:
        my_dict[word] += 1
    else:
        my_dict[word] = 1

print(my_dict)

# Display the top 10 most common words and their counts

Words_count = Counter(my_dict)

#we count only top10 frequents words we put 10 as index

top_10_words = words_count.most_common(10)

#print top 10 most common words

for word,count in top_10_words:
       print(f’”{word}:{count}”)
```

Code explanation :

- ## Importing all useful libraries

Firstly we import all the necessary libraries urllib, BeautifulSoup , string , Collections.



```
import urllib.request
```

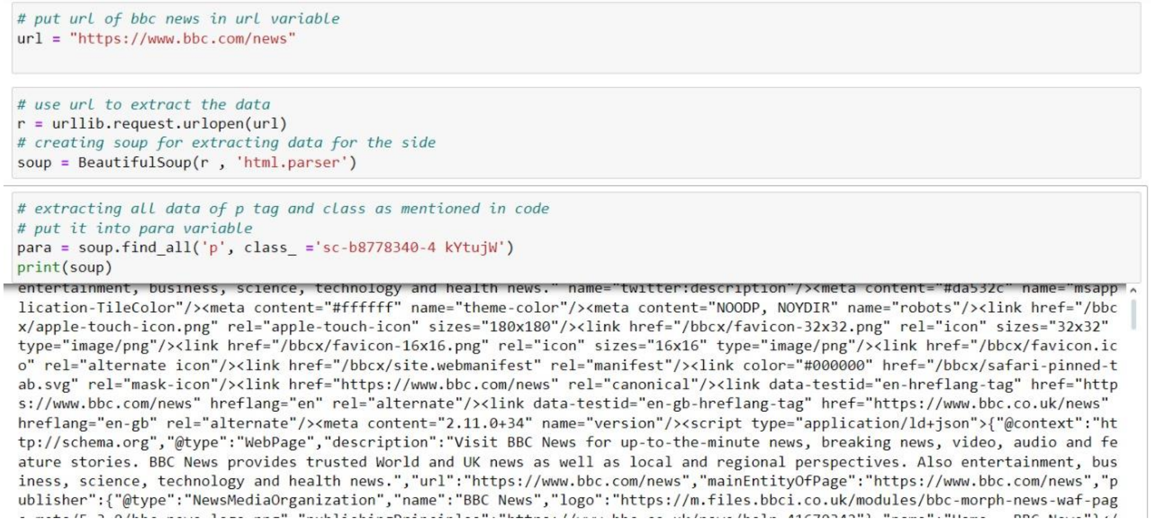
```
#importing pandas to manipulate the data like convert to csvfile create a dataframe
import pandas as pd
```

```
#importing beautiful soup for extracting the data from any web page
from bs4 import BeautifulSoup
```

```
# importing string for using inbuilt fuction
import string
```

- Retrieve url from urllib3 and extract

This function takes a URL as input , and send an HTTP request to fetch content of the page , and extract all the text inside <p> tags using BeautifulSoup.



```
# put url of bbc news in url variable
url = "https://www.bbc.com/news"
```

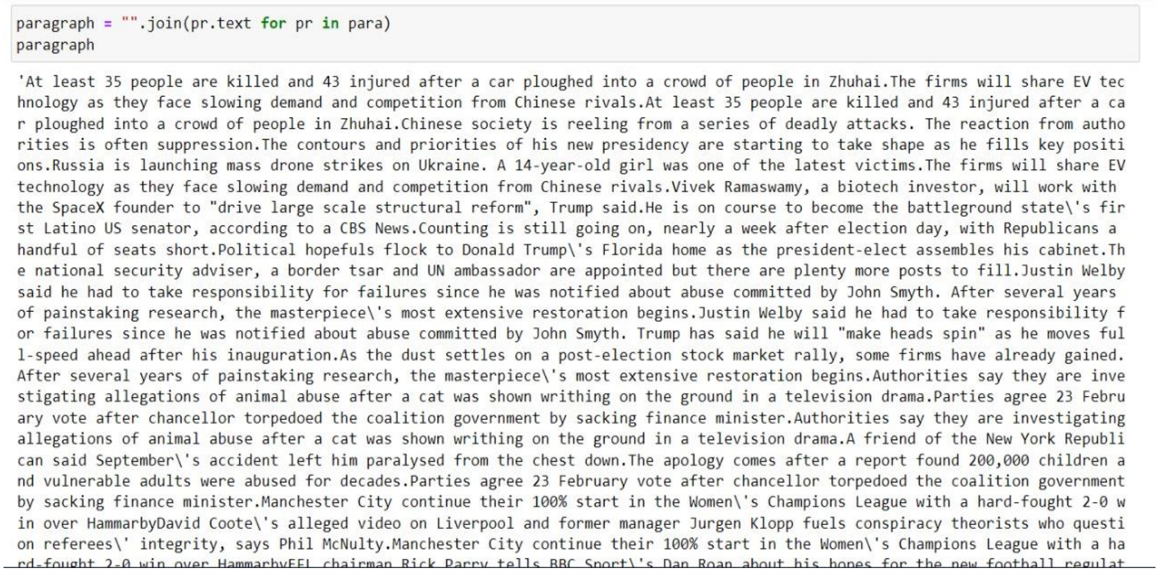
```
# use url to extract the data
r = urllib.request.urlopen(url)
# creating soup for extracting data for the side
soup = BeautifulSoup(r , 'html.parser')
```

```
# extracting all data of p tag and class as mentioned in code
# put it into para variable
para = soup.find_all('p', class_ ='sc-b8778340-4 kYtujW')
print(soup)
```

```
entertainment, business, science, technology and health news." name="twitter:description"/><meta content="#da532c" name="msapp
lication-TileColor"/><meta content="#ffffff" name="theme-color"/><meta content="NOODP, NOYDIR" name="robots"/><link href="/bbc
x/apple-touch-icon.png" rel="apple-touch-icon" sizes="180x180"/><link href="/bbcx/favicon-32x32.png" rel="icon" sizes="32x32"
type="image/png"/><link href="/bbcx/favicon-16x16.png" rel="icon" sizes="16x16" type="image/png"/><link href="/bbcx/favicon.ic
o" rel="alternate icon"/><link href="/bbcx/site.webmanifest" rel="manifest"/><link color="#000000" href="/bbcx/safari-pinned-t
ab.svg" rel="mask-icon"/><link href="https://www.bbc.com/news" rel="canonical"/><link data-testid="en-hreflang-tag" href="http
s://www.bbc.com/news" hreflang="en" rel="alternate"/><link data-testid="en-gb-hreflang-tag" href="https://www.bbc.co.uk/news"
hreflang="en-gb" rel="alternate"/><meta content="2.11.0+34" name="version"/><script type="application/ld+json">{"@context":"ht
tp://schema.org","@type":"WebPage","description":"Visit BBC News for up-to-the-minute news, breaking news, video, audio and fe
ature stories. BBC News provides trusted World and UK news as well as local and regional perspectives. Also entertainment, bus
iness, science, technology and health news.","url":"https://www.bbc.com/news","mainEntityOfPage":"https://www.bbc.com/news","p
ublisher":{"@type":"NewsMediaOrganization","name":"BBC News","logo":"https://m.files.bbci.co.uk/modules/bbc-morph-news-waf-pag
```

- Combining all paragraph into one paragraph

After extracting all the paragraph we combine all the paragraph and convert into the one paragraph using join() function.

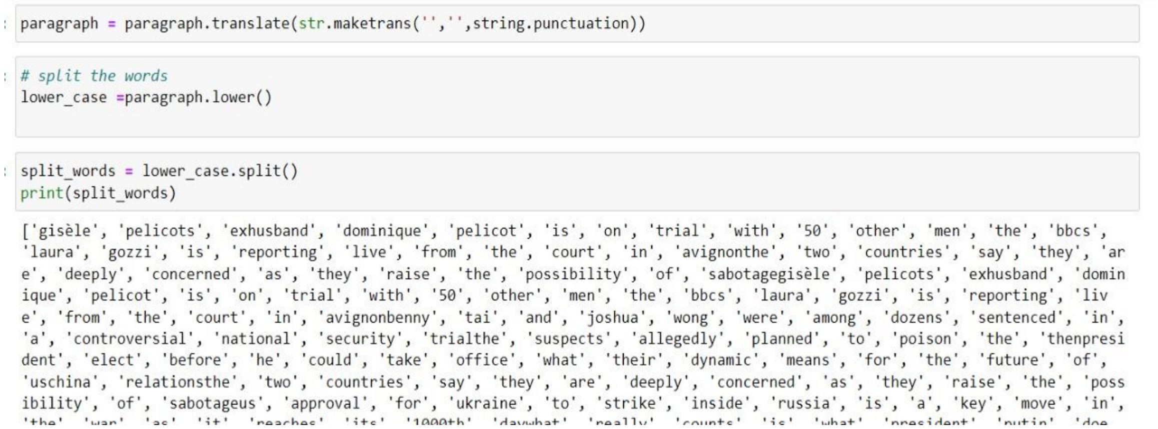


```
paragraph = "".join(pr.text for pr in para)
paragraph
```

```
'At least 35 people are killed and 43 injured after a car ploughed into a crowd of people in Zhuhai.The firms will share EV tec
hnology as they face slowing demand and competition from Chinese rivals.At least 35 people are killed and 43 injured after a ca
r ploughed into a crowd of people in Zhuhai.Chinese society is reeling from a series of deadly attacks. The reaction from autho
rities is often suppression.The contours and priorities of his new presidency are starting to take shape as he fills key positi
ons.Russia is launching mass drone strikes on Ukraine. A 14-year-old girl was one of the latest victims.The firms will share EV
technology as they face slowing demand and competition from Chinese rivals.Vivek Ramaswamy, a biotech investor, will work with
the SpaceX founder to "drive large scale structural reform", Trump said.He is on course to become the battleground state\'s fir
st Latino US senator, according to a CBS News.Counting is still going on, nearly a week after election day, with Republicans a
handful of seats short.Political hopefuls flock to Donald Trump\'s Florida home as the president-elect assembles his cabinet.Th
e national security adviser, a border tsar and UN ambassador are appointed but there are plenty more posts to fill.Justin Welby
said he had to take responsibility for failures since he was notified about abuse committed by John Smyth. After several years
of painstaking research, the masterpiece\'s most extensive restoration begins.Justin Welby said he had to take responsibility f
or failures since he was notified about abuse committed by John Smyth. Trump has said he will "make heads spin" as he moves ful
l-speed ahead after his inauguration.As the dust settles on a post-election stock market rally, some firms have already gained.
After several years of painstaking research, the masterpiece\'s most extensive restoration begins.Authorities say they are inve
stigating allegations of animal abuse after a cat was shown writhing on the ground in a television drama.Parties agree 23 Febru
ary vote after chancellor torpedoed the coalition government by sacking finance minister.Authorities say they are investigating
allegations of animal abuse after a cat was shown writhing on the ground in a television drama.A friend of the New York Republi
can said September\'s accident left him paralysed from the chest down.The apology comes after a report found 200,000 children a
nd vulnerable adults were abused for decades.Parties agree 23 February vote after chancellor torpedoed the coalition government
by sacking finance minister.Manchester City continue their 100% start in the Women\'s Champions League with a hard-fought 2-0 w
in over HammarbyDavid Coote\'s alleged video on Liverpool and former manager Jurgen Klopp fuels conspiracy theorists who questi
on referees\' integrity, says Phil McNulty.Manchester City continue their 100% start in the Women\'s Champions League with a ha
rd-fought 2-0 win over HammarbyEFL chairman Rick Parry tells BBC Sport\'s Dan Roan about his hopes for the new football regulat
```

- Clean Text

After combing all the text firstly converts the extracted text to lowercase, remove punctuations , and splits the text into words.



```
paragraph = paragraph.translate(str.maketrans('','',string.punctuation))
```

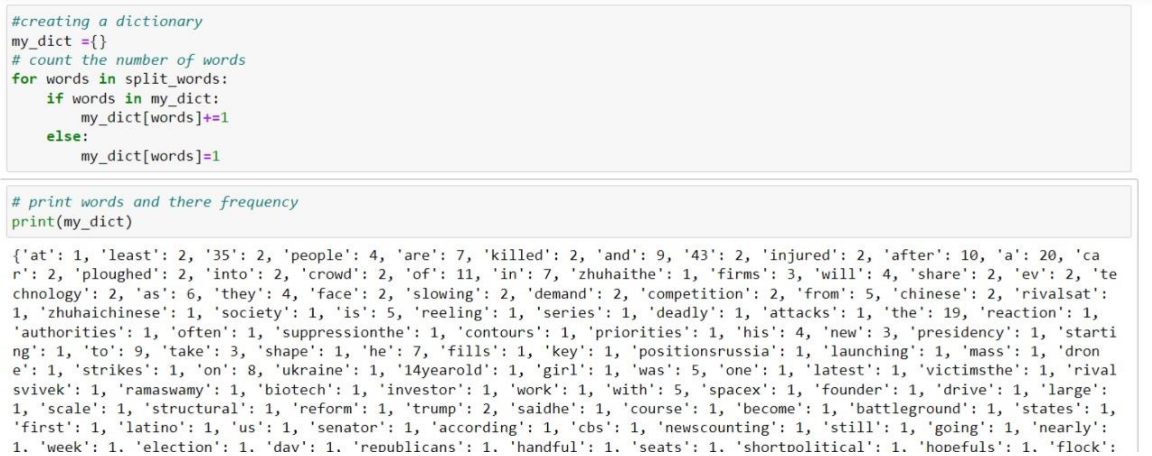
```
# split the words
lower_case =paragraph.lower()
```

```
split_words = lower_case.split()
print(split_words)
```

```
['gisèle', 'pelicots', 'exhusband', 'dominique', 'pelicot', 'is', 'on', 'trial', 'with', '50', 'other', 'men', 'the', 'bbcs',
'laura', 'gozzi', 'is', 'reporting', 'live', 'from', 'the', 'court', 'in', 'avignonthe', 'two', 'countries', 'say', 'they', 'ar
e', 'deeply', 'concerned', 'as', 'they', 'raise', 'the', 'possibility', 'of', 'sabotagegisèle', 'pelicots', 'exhusband', 'domin
ique', 'pelicot', 'is', 'on', 'trial', 'with', '50', 'other', 'men', 'the', 'bbcs', 'laura', 'gozzi', 'is', 'reporting', 'liv
e', 'from', 'the', 'court', 'in', 'avignonbenny', 'tai', 'and', 'joshua', 'wong', 'were', 'among', 'dozens', 'sentenced', 'in',
'a', 'controversial', 'national', 'security', 'trialthe', 'suspects', 'allegedly', 'planned', 'to', 'poison', 'the', 'thenpresi
dent', 'elect', 'before', 'he', 'could', 'take', 'office', 'what', 'their', 'dynamic', 'means', 'for', 'the', 'future', 'of',
'uschina', 'relationsthe', 'two', 'countries', 'say', 'they', 'are', 'deeply', 'concerned', 'as', 'they', 'raise', 'the', 'poss
ibility', 'of', 'sabotageus', 'approval', 'for', 'ukraine', 'to', 'strike', 'inside', 'russia', 'is', 'a', 'key', 'move', 'in',
'the', 'war', 'as', 'it', 'reaches', 'its', '1000th', 'daywhat', 'really', 'counts', 'is', 'what', 'president', 'putin', 'doe
```

- Count Words Frequency (words):

We create a empty dictionary than count the frequency of each word and than put into the each word and their frequency as key value pair and print the dictionary my_dict().



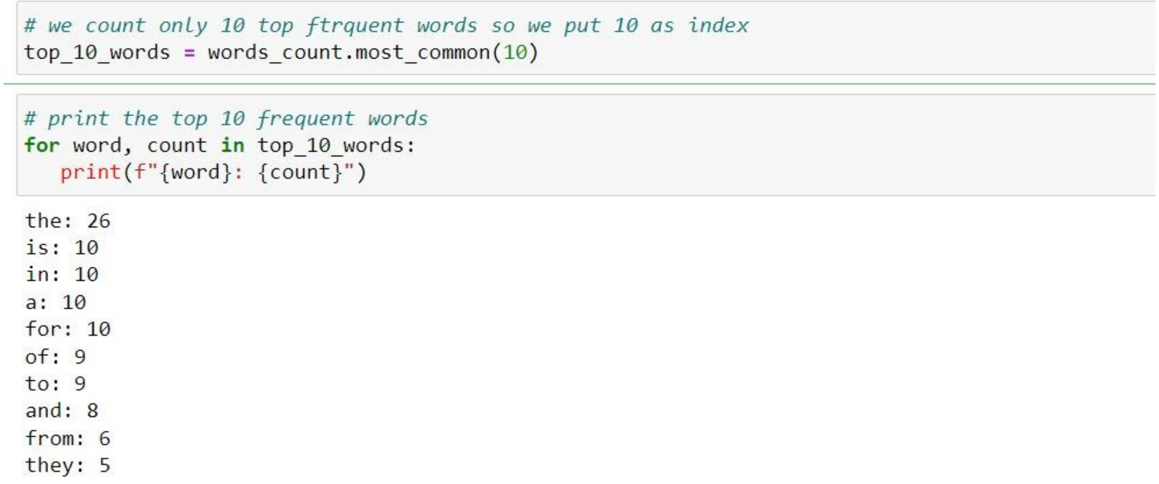
```
#creating a dictionary
my_dict ={}
# count the number of words
for words in split_words:
    if words in my_dict:
        my_dict[words]+=1
    else:
        my_dict[words]=1
```

```
# print words and there frequency
print(my_dict)
```

```
{'at': 1, 'least': 2, '35': 2, 'people': 4, 'are': 7, 'killed': 2, 'and': 9, '43': 2, 'injured': 2, 'after': 10, 'a': 20, 'ca
r': 2, 'ploughed': 2, 'into': 2, 'crowd': 2, 'of': 11, 'in': 7, 'zhuhaithe': 1, 'firms': 3, 'will': 4, 'share': 2, 'ev': 2, 'te
chnology': 2, 'as': 6, 'they': 4, 'face': 2, 'slowing': 2, 'demand': 2, 'competition': 2, 'from': 5, 'chinese': 2, 'rivalsat':
1, 'zhuhaichinese': 1, 'society': 1, 'is': 5, 'reeling': 1, 'series': 1, 'deadly': 1, 'attacks': 1, 'the': 19, 'reaction': 1,
'authorities': 1, 'often': 1, 'suppressionthe': 1, 'contours': 1, 'priorities': 1, 'his': 4, 'new': 3, 'presidency': 1, 'starti
ng': 1, 'to': 9, 'take': 3, 'shape': 1, 'he': 7, 'fills': 1, 'key': 1, 'positionsrussia': 1, 'launching': 1, 'mass': 1, 'dron
e': 1, 'strikes': 1, 'on': 8, 'ukraine': 1, '14yearold': 1, 'girl': 1, 'was': 5, 'one': 1, 'latest': 1, 'victimsthe': 1, 'rival
svivek': 1, 'ramaswamy': 1, 'biotech': 1, 'investor': 1, 'work': 1, 'with': 5, 'spacex': 1, 'founder': 1, 'drive': 1, 'large':
1, 'scale': 1, 'structural': 1, 'reform': 1, 'trump': 2, 'saidhe': 1, 'course': 1, 'become': 1, 'battleground': 1, 'states': 1,
'first': 1, 'latino': 1, 'us': 1, 'senator': 1, 'according': 1, 'cbs': 1, 'newscounting': 1, 'still': 1, 'going': 1, 'nearly':
1, 'week': 1, 'election': 1, 'day': 1, 'republicans': 1, 'handful': 1, 'seats': 1, 'shortpolitical': 1, 'hopefuls': 1, 'flock':
```

- Find top 10 most common words:

This function uses collections , counter , to count the frequency of each word and then returns the 10 most common words.



```
# we count only 10 top ftrquent words so we put 10 as index
top_10_words = words_count.most_common(10)
```

```
# print the top 10 frequent words
for word, count in top_10_words:
   print(f"{word}: {count}")
```

```
the: 26
is: 10
in: 10
a: 10
for: 10
of: 9
to: 9
and: 8
from: 6
they: 5
```

CONCLUSION

This project demonstrates how to retrieve and process web content to analyze the frequency of words . By using libraries like urllib and BeautifulSoup , we can easily extract text from web pages , clean the text , and perform simple text analysis . the resulting tool can be useful for web content analysis and content summarization.

Thank you...

Group members :

Anoop Kumar Dwivedi : 2446007

Sushant Kumar : 2446052

Suryash Nigam : 2446054