

S.I.E.S College of Arts, Science and Commerce (Autonomous)
Sion(W), Mumbai – 400 022.

CERTIFICATE

This is to certify that **Ms. Bangi Iqra Aslam** Roll No.
SMDS2324003 has successfully completed the necessary course of experiments in the subject of **Web and Social Network Data Analytics** during the academic year **2023 – 2024** complying with the requirements for the course of **M.Sc. Data Science Part – II [Semester 4]**

Prof. In-Charge
Prof. Rajesh Yadav

Head of Department
Dr. Prof. Abuzar Ansari

Sr. No	Practical list	Date
1	Practical 1- Implementation of Page Rank using NetworkX without weighted edge	06/01/2024
2	Practical 2- Page Rank using NetworkX	06/01/2024
3	Practical 3-HITS Algorithm	06/01/2024
4	Practical 4- Write a program to implement a simple web crawler	13/01/2024
5	Practical 5- Develop a focused Crawler for Local Search	13/01/2024
6	Practical 6-Python Program on Opinion Search and Retrieval	27/01/2024
7	Practical 7-Sentiment Analysis	27/01/2024
8	Practical 8- Python program on Web Content Mining	03/02/2024
9	Practical 9- Python Program on Web Structure Mining	03/02/2024

1 /24, :08 PM

P1 PAGE RANK - Jupyter Notebook

In [2]: 1 pip install networkx

Requirement already satisfied: networkx in c:\users\diksh\anaconda3\lib\site-packages (3.1)
Note: you may need to restart the kernel to use updated packages.

In [3]: 1 import networkx as nx
2 G = nx.DiGraph()
3 [G.add_node(k) for k in ["A", "B", "C", "D", "E", "F", "G"]]

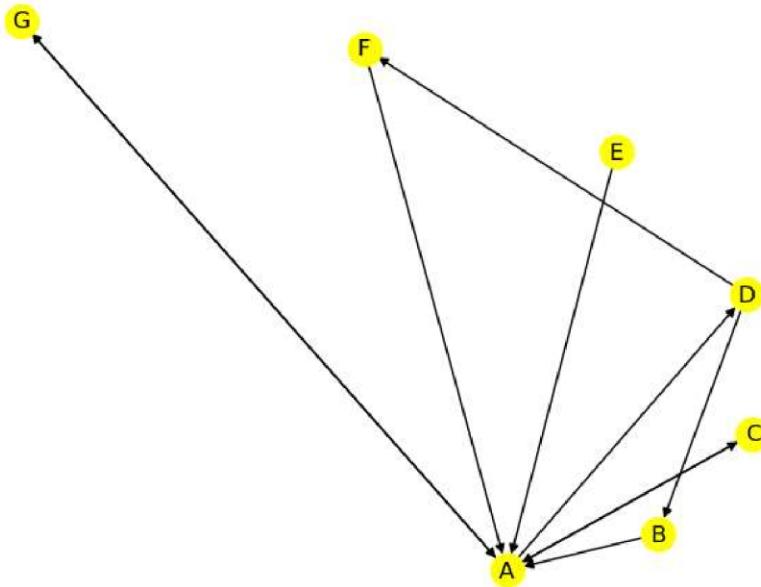
Out[3]: [None, None, None, None, None, None, None]

In [4]: 1 G.add_edges_from([('G', 'A'), ('A', 'G'), ('B', 'A'), ('C', 'A'), ('A', 'C'), ('A', 'D'), ('E', 'A'), ('F', 'A'), ('D', 'B'), ('D', 'F')])

In [5]: 1 ppr1 = nx.pagerank(G)

In [6]: 1 print("Page rank value:", ppr1)

Page rank value: {'A': 0.4080745143467559, 'B': 0.07967426232810562, 'C': 0.13704946318948705, 'D': 0.13704946318948705, 'E': 0.021428571428571432, 'F': 0.07967426232810562, 'G': 0.13704946318948705}

In [13]: 1 pos = nx.spiral_layout(G)
2 nx.draw(G, pos, with_labels = True, node_color="Yellow")

0 P

WM PRAC2

```
In [3]: import networkx as nx  
import pylab as plt
```

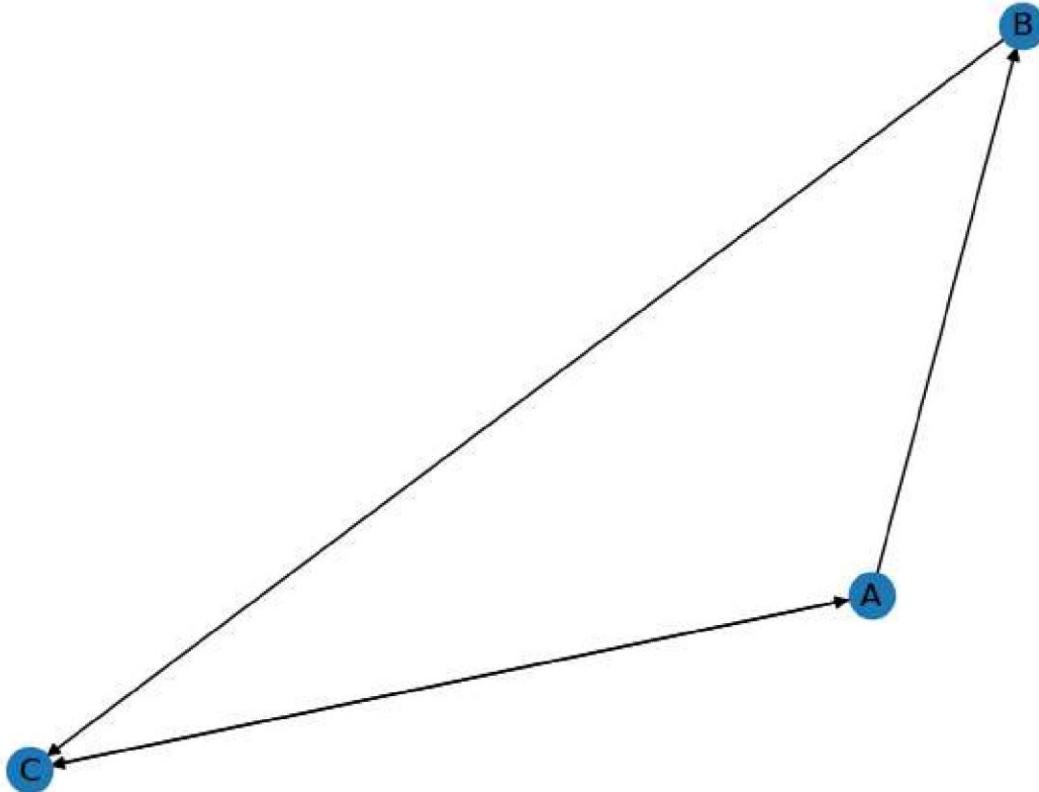
```
In [4]: D=nx.DiGraph()
```

```
In [5]: D.add_weighted_edges_from([('A','B',1),('A','C',1),('C','A',1),('B','C',1)])
```

```
In [6]: print(nx.pageRank(D))
```

```
{'A': 0.387789442707259, 'B': 0.21481051315058508, 'C': 0.3974000441421556}
```

```
In [9]: nx.draw(D, with_labels=True)  
plt.show()
```



1 P

P3 H TS Algo

In [2]: #PRAC3

import networkx as nx

In [4]: G = nx.DiGraph()

In [5]: G.add_edges_from([(1, 2), (1, 3), (2, 4), (3, 4), (4, 5)])

In [8]: authority_scores, hub_scores = nx.hits(G)

In [9]: print("Authority Scores:", authority_scores)

print("Hub Scores:", hub_scores)

```
Authority Scores: {1: 0.9308493107467511, 2: 0.034575344626624455, 3: 0.034575344626624455, 4: -1.240182114257068e-16, 5: -0.0}
```

```
Hub Scores: {1: -0.0, 2: 0.4820931936872627, 3: 0.4820931936872627, 4: 0.03581361262547467, 5: -1.284597516082113e-16}
```

```
In [ ]: # Practical 4  
# Aim : Write a program to implement a simple web crawler.
```

```
In [1]: import requests  
from parsel import Selector  
import time
```

```
In [2]: start=time.time()
```

```
In [3]: response=requests.get('http://recurship.com/')
```

```
In [4]: selector=Selector(response.text)
```

```
In [5]: href_links=selector.xpath('//a/@href').getall()
```

```
In [6]: image_links=selector.xpath('//img/@src').getall()
```

```
In [7]: print("*****Href_links*****")
```

```
*****Href_links*****
```

```
In [8]: print(href_links)
```

```
['#primary', 'http://recurship.com/', 'http://recurship.com/', 'http://recurship.com/', 'http://recurship.com/about/', 'http://recurship.com/playthinks/', 'http://recurship.com/build-a-mvp/', 'http://recurship.com/careers/', 'http://recurship.com/contact/', 'http://recurship.com/blog/category/uncategorized/', 'http://recurship.com/blog/2018/07/08/2018-7-8-sastaticket-acquires-recurship/', 'http://recurship.com/blog/author/mashhoodr/', 'http://recurship.com/blog/author/mashhoodr/', 'http://recurship.com/blog/2018/07/08/2018-7-8-sastaticket-acquires-recurship/', 'http://recurship.com/blog/2018/07/08/2018-7-8-sastaticket-acquires-recurship/', 'http://recurship.com/blog/2018/07/08/2018-7-8-sastaticket-acquires-recurship/', 'http://recurship.com/blog/2018/06/03/2018-6-4-ngrx-selectors-how-to-stop-worrying-about-your-store-structure/', 'http://recurship.com/blog/author/mashhoodr/', 'http://recurship.com/blog/author/mashhoodr/', 'http://recurship.com/blog/2018/06/03/2018-6-4-ngrx-selectors-how-to-stop-worrying-about-your-store-structure/', 'http://recurship.com/blog/2018/06/03/2018-6-4-ngrx-selectors-how-to-stop-worrying-about-your-store-structure/', 'http://recurship.com/blog/2018/06/03/2018-6-4-ngrx-selectors-how-to-stop-worrying-about-your-store-structure/', 'http://recurship.com/blog/2018/06/03/2018-6-4-ngrx-selectors-how-to-stop-worrying-about-your-store-structure/', 'http://recurship.com/blog/2018/06/03/2018-6-4-ngrx-selectors-how-to-stop-worrying-about-your-store-structure/', 'http://recurship.com/blog/2018/06/03/2018-6-4-ngrx-selectors-how-to-stop-worrying-about-your-store-structure/', 'http://recurship.com/blog/2018/06/03/2018-6-4-ngrx-selectors-how-to-stop-worrying-about-your-store-structure/', 'http://recurship.com/blog/2018/06/03/2018-6-4-ngrx-selectors-how-to-stop-worrying-about-your-store-structure/', 'http://recurship.com/blog/2018/06/03/2018-6-1-jjknwadn9ivw1gba3wxsspjlpe9grk/', 'http://recurship.com/blog/author/mashhoodr/', 'http://recurship.com/blog/author/mashhoodr/', 'http://recurship.com/blog/2018/06/03/2018-6-1-jjknwadn9ivw1gba3wxsspjlpe9grk/', 'http://recurship.com/blog/2018/06/03/2018-6-1-jjknwadn9ivw1gba3wxsspjlpe9grk/', 'http://recurship.com/blog/2018/06/03/2018-5-31-angulars-user-authentication-tool-belt/', 'http://recurship.com/blog/author/mashhoodr/', 'http://recurship.com/blog/2018/06/03/2018-5-31-angulars-user-authentication-tool-belt/', 'http://recurship.com/blog/2018/06/03/2018-5-31-angulars-user-authentication-tool-belt/', 'http://recurship.com/blog/2018/06/03/2018-5-31-angulars-user-authentication-tool-belt/', 'http://recurship.com/blog/2018/06/03/2018-5-31-xfvrq9aauskayhkd4kzp7gsbfg2bf1/', 'http://recurship.com/blog/author/mashhoodr/', 'http://recurship.com/blog/2018/06/03/2018-5-31-xfvrq9aauskayhkd4kzp7gsbfg2bf1/', 'http://recurship.com/blog/2018/06/03/2018-5-31-xfvrq9aauskayhkd4kzp7gsbfg2bf1/', 'http://recurship.com/blog/2018/06/03/2018-5-31-real-time-stream-processing-with-reactive-extensions-rx/', 'http://recurship.com/blog/author/mashhoodr/', 'http://recurship.com/blog/2018/06/03/2018-5-31-real-time-stream-processing-with-reactive-extensions-rx/', 'http://recurship.com/blog/2018/06/03/2018-5-31-real-time-stream-processing-with-reactive-extensions-rx/', 'http://recurship.com/blog/2018/06/03/2018-5-31-real-time-stream-processing-with-reactive-extensions-rx/', 'http://recurship.com/blog/2018/05/31/2018-5-31-supercharging-the-angular-cli-with-nx/', 'http://recurship.com/blog/author/mashhoodr/', 'http://recurship.com/blog/2018/05/31/2018-5-31-supercharging-the-angular-cli-with-nx/', 'http://recurship.com/blog/2018/05/31/2018-5-31-supercharging-the-angular-cli-with-nx/', 'http://recurship.com/blog/2018/05/31/2018-5-31-angular-as-a-strategy-for-collaboration-and-scale/', 'http://recurship.com/blog/author/mashhoodr/', 'http://recurship.com/blog/2018/05/31/2018-5-31-angular-as-a-strategy-for-collaboration-and-scale/', 'http://recurship.com/blog/2018/05/31/2018-5-31-angular-as-a-strategy-for-collaboration-and-scale/', 'http://recurship.com/blog/2018/05/12/keynote-five-years-of-angular/', 'http://recurship.com/blog/author/mashhoodr/', 'http://recurship.com/blog/2018/05/12/keynote-five-years-of-angular/', 'http://recurship.com/blog/2018/05/12/keynote-five-years-of-angular/', 'http://recurship.com/blog/2018/05/12/keynote-five-years-of-angular/', 'http://recurship.com/blog/2018/04/29/2018-4-29-understanding-advanced-dependancy-injection-in-angular/', 'http://recurship.com/blog/author/mashhoodr/', 'http://recurship.com/blog/2018/04/29/2018-4-29-understanding-advanced-dependancy-injection-in-angular']
```

```
ar/', 'http://recurship.com/blog/2018/04/29/2018-4-29-understanding-advanced-dependancy-injection-in-angular/', 'http://recurship.com/page/2/']
```

In [9]: `print("*****/href_links*****")`

```
*****/href_links*****
```

In [10]: `# data type of this is List print("*****Image_Links*****")`

```
['http://recurship.com/wp-content/themes/stag-blocks/images/placeholder.svg',  
 'http://recurship.com/wp-content/themes/stag-blocks/images/menu.svg', 'ht  
 tp://recurship.com/wp-content/themes/stag-blocks/images/close-button.svg', 'ht  
 tp://recurship.com/wp-content/themes/stag-blocks/images/search.svg', 'http://  
 recurship.com/wp-content/themes/stag-blocks/images/placeholder.svg', 'http://  
 2.gravatar.com/avatar/8a081ac7e6aadaabfdc51ec038867890?s=80&d=mm&r=g', 'htt  
 p://recurship.com/wp-content/themes/stag-blocks/images/placeholder.svg', 'htt  
 p://2.gravatar.com/avatar/8a081ac7e6aadaabfdc51ec038867890?s=80&d=mm&r=g', 'h  
 ttp://recurship.com/wp-content/themes/stag-blocks/images/placeholder.svg', 'h  
 ttp://2.gravatar.com/avatar/8a081ac7e6aadaabfdc51ec038867890?s=80&d=mm&r=g',  
 'http://recurship.com/wp-content/themes/stag-blocks/images/placeholder.svg',  
 'http://2.gravatar.com/avatar/8a081ac7e6aadaabfdc51ec038867890?s=80&d=mm&r=  
 g', 'http://recurship.com/wp-content/themes/stag-blocks/images/placeholder.sv  
 g', 'http://2.gravatar.com/avatar/8a081ac7e6aadaabfdc51ec038867890?s=80&d=m  
 m&r=g', 'http://recurship.com/wp-content/themes/stag-blocks/images/placehol  
 der.svg', 'http://2.gravatar.com/avatar/8a081ac7e6aadaabfdc51ec038867890?s=80  
 &d=mm&r=g', 'http://recurship.com/wp-content/themes/stag-blocks/images/placehol  
 der.svg', 'http://2.gravatar.com/avatar/8a081ac7e6aadaabfdc51ec038867890?s=80  
 &d=mm&r=g', 'http://recurship.com/wp-content/themes/stag-blocks/images/place  
 holder.svg', 'http://2.gravatar.com/avatar/8a081ac7e6aadaabfdc51ec038867890?  
 s=80&d=mm&r=g', 'http://recurship.com/wp-content/themes/stag-blocks/images/ba  
 ck.svg']
```

In [11]: `print("*****image_links*****")`

```
*****image_links*****
```

In [12]: `end=time.time()`

In [13]: `print("Time Taken in seconds: ",(end-start))`

```
Time Taken in seconds: 258.38392663002014
```

2

p 5 b i p o

```
In [ ]: # Practical 5:Develop A Focused Crawler For Local Search.
```

```
In [25]: import requests
from bs4 import BeautifulSoup
import re
```

```
In [26]: class LocalSearchCrawler:
    def __init__(self, seed_url, keyword, max_depth=3):
        self.seed_url = seed_url
        self.keyword = keyword
        self.max_depth = max_depth
        self.visited_urls = set()

    def crawl(self, url, depth=1):
        if depth > self.max_depth or url in self.visited_urls:
            return
        try:
            response = requests.get(url)
            if response.status_code == 200:
                soup = BeautifulSoup(response.text, 'html.parser')
                self.extract_information(url, soup)
                # Find and crawl links on the page
                links = soup.find_all('a', href=True)
                for link in links:
                    next_url = link['href']
                    if self.is_valid_url(next_url):
                        self.crawl(next_url, depth + 1)
                        self.visited_urls.add(url)
        except Exception as e:
            print(f"Error crawling {url}: {e}")

    def extract_information(self, url, soup):
        # Extract information from the page based on your needs # For example,
        print(f"Extracting information from {url}")

    def is_valid_url(self, url):
        # Customize this method to filter URLs based on your criteria
        return re.search(self.keyword, url) is not None
```

2

p 5 b i p o

```
In [27]: if __name__ == "__main__":
    seed_url = "http://recurship.com/" # Replace with the starting URL for your
    keyword = "blog" # Replace with a keyword that identifies local business URLs
    max_depth = 3 # Maximum depth for crawling (adjust as needed)
    local_search_crawler = LocalSearchCrawler(seed_url, keyword, max_depth)
    local_search_crawlercrawl(seed_url)
```

Extracting information from <http://recurship.com/> (<http://recurship.com/>)
 Extracting information from <http://recurship.com/blog/category/uncategorized/> (<http://recurship.com/blog/category/uncategorized/>)
 Extracting information from <http://recurship.com/blog/category/uncategorized/> (<http://recurship.com/blog/category/uncategorized/>)
 Extracting information from <http://recurship.com/blog/2018/07/08/2018-7-8-sastaticket-acquires-recurship/> (<http://recurship.com/blog/2018/07/08/2018-7-8-sastaticket-acquires-recurship/>)
 Extracting information from <http://recurship.com/blog/author/mashhoodr/> (<http://recurship.com/blog/author/mashhoodr/>)
 Extracting information from <http://recurship.com/blog/2018/06/03/2018-6-4-ngrx-selectors-how-to-stop-worrying-about-your-store-structure/> (<http://recurship.com/blog/2018/06/03/2018-6-4-ngrx-selectors-how-to-stop-worrying-about-your-store-structure/>)
 Extracting information from <http://recurship.com/blog/2018/06/03/2018-6-1-jjknwadn9ivw1gba3wxsspjlpe9grk/> (<http://recurship.com/blog/2018/06/03/2018-6-1-jjknwadn9ivw1gba3wxsspjlpe9grk/>)
 Extracting information from <http://recurship.com/blog/2018/06/03/2018-5-31-angulars-user-authentication-tool-belt/> (<http://recurship.com/blog/2018/06/03/2018-5-31-angulars-user-authentication-tool-belt/>)
 Extracting information from <http://recurship.com/blog/2018/06/03/2018-5-31-xfvrq9aauskayhkd4kzp7gsbfg2bf1/> (<http://recurship.com/blog/2018/06/03/2018-5-31-xfvrq9aauskayhkd4kzp7gsbfg2bf1/>)
 Extracting information from <http://recurship.com/blog/2018/06/03/2018-5-31-real-time-stream-processing-with-reactive-extensions-rx/> (<http://recurship.com/blog/2018/06/03/2018-5-31-real-time-stream-processing-with-reactive-extensions-rx/>)
 Extracting information from <http://recurship.com/blog/2018/05/31/2018-5-31-supercharging-the-angular-cli-with-nx/> (<http://recurship.com/blog/2018/05/31/2018-5-31-supercharging-the-angular-cli-with-nx/>)
 Extracting information from <http://recurship.com/blog/2018/05/31/2018-5-31-angular-as-a-strategy-for-collaboration-and-scale/> (<http://recurship.com/blog/2018/05/31/2018-5-31-angular-as-a-strategy-for-collaboration-and-scale/>)
 Extracting information from <http://recurship.com/blog/2018/05/12/keynote-five-years-of-angular/> (<http://recurship.com/blog/2018/05/12/keynote-five-years-of-angular/>)
 Extracting information from <http://recurship.com/blog/2018/04/29/2018-4-29-understanding-advanced-dependancy-injection-in-angular/> (<http://recurship.com/blog/2018/04/29/2018-4-29-understanding-advanced-dependancy-injection-in-angular/>)
 Extracting information from <http://recurship.com/blog/category/uncategorized/page/2/> (<http://recurship.com/blog/category/uncategorized/page/2/>)

48 P

Practical6 Jupyter Notebook

```
In [1]: 1 import nltk
2 from nltk.sentiment import SentimentIntensityAnalyzer
3 nltk.download('vader_lexicon')

[nltk_data] Downloading package vader_lexicon to
[nltk_data]     C:/Users/diksh/AppData/Roaming/nltk_data...
[nltk_data]     Package vader_lexicon is already up-to-date!
```

Out[1]: True

```
In [2]: 1 opinions=["I love this product!","The service was terrible","The customer support team was very helpful"]
```

```
In [3]: 1 def analyze_sentiment(text):
2     sia=SentimentIntensityAnalyzer()
3     sentiment_score=sia.polarity_scores(text)['compound']
4
5     if sentiment_score>=0.05:
6         return "positive"
7     elif sentiment_score<=0.05:
8         return "negative"
9     else:
10        return "neutral"
```

```
In [4]: 1 def retrieve_opinions(sentiment):
2     matching_opinions=[opinion for opinion in opinions if analyze_sentiment(opinion) == sentiment]
3     return matching_opinions
4
5 if __name__ == "__main__":
6     positive_opinions=retrieve_opinions("positive")
7
8     print("Positive opinions")
9     for opinion in positive_opinions:
10         print(f" -{opinion}")
11
12     negative_opinions=retrieve_opinions("negative")
13
14     print ("\nNegative opinions")
15     for opinion in negative_opinions:
16         print(f" -{opinion}")
```

Positive opinions

Negative opinions

-The service was terrible

In [1]: `pip install TextBlob`

```
Collecting TextBlob
  Obtaining dependency information for TextBlob from https://files.pythonhosted.org/packages/02/07/5fd2945356dd839974d3a25de8a142dc37293c21315729a41e775b5f3569/textblob-0.18.0.post0-py3-none-any.whl.metadata
    Downloading textblob-0.18.0.post0-py3-none-any.whl.metadata (4.5 kB)
Requirement already satisfied: nltk>=3.8 in c:\users\dell\anaconda3\lib\site-packages (from TextBlob) (3.8.1)
Requirement already satisfied: click in c:\users\dell\anaconda3\lib\site-packages (from nltk>=3.8->TextBlob) (8.0.4)
Requirement already satisfied: joblib in c:\users\dell\anaconda3\lib\site-packages (from nltk>=3.8->TextBlob) (1.2.0)
Requirement already satisfied: regex>=2021.8.3 in c:\users\dell\anaconda3\lib\site-packages (from nltk>=3.8->TextBlob) (2022.7.9)
Requirement already satisfied: tqdm in c:\users\dell\anaconda3\lib\site-packages (from nltk>=3.8->TextBlob) (4.65.0)
Requirement already satisfied: colorama in c:\users\dell\anaconda3\lib\site-packages (from click->nltk>=3.8->TextBlob) (0.4.6)
  Downloading textblob-0.18.0.post0-py3-none-any.whl (626 kB)
----- 0.0/626.3 kB ? eta ------
----- 10.2/626.3 kB ? eta ------
----- 30.7/626.3 kB 435.7 kB/s eta 0:00:02
----- 430.1/626.3 kB 3.9 MB/s eta 0:00:01
----- 626.3/626.3 kB 4.9 MB/s eta 0:00:00
Installing collected packages: TextBlob
Successfully installed TextBlob-0.18.0.post0
Note: you may need to restart the kernel to use updated packages.
```

In [2]: `from textblob import TextBlob`

```
def sentiment(text):
    blob=TextBlob(text)
    sentiment_polarity=blob.sentiment.polarity

    if sentiment_polarity > 0:
        return 'positive'
    elif sentiment_polarity < 0:
        return 'negative'
    else:
        return 'neutral'

if __name__ == "__main__":
    sample_text="I love python very much,its anamazing language"
    get_sentiment=sentiment(sample_text)

    print(f"Sample Text:{sample_text}")
    print(f"sentmental:{get_sentiment}")
```

Sample Text:I love python very much,its anamazing language
sentmental:positive

5

P8 Po a b content m g

boo

```
In [19]: import requests
from bs4 import BeautifulSoup
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.probability import FreqDist
import matplotlib.pyplot as plt
import nltk
nltk.download('punkt')
nltk.download('stopwords')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\diksh\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\diksh\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.
```

```
Out[19]: True
```

```
In [20]: def get_html_content(url):
    response = requests.get(url)
    if response.status_code == 200:
        return response.text
    else:
        print(f"Failed to retrieve content. Status code: {response.status_code}")
        return None
```

```
In [21]: def extract_text(html_content):
    soup = BeautifulSoup(html_content, 'html.parser')
    paragraphs = soup.find_all('p')
    text=''.join([paragraph.get_text() for paragraph in paragraphs])
    return text
```

```
In [22]: def analyze_word_frequency(text):
    tokens = word_tokenize(text)

    stop_words = set(stopwords.words('english'))
    filtered_tokens = [word.lower() for word in tokens if word.isalnum() and wo

    freq_dist = FreqDist(filtered_tokens)
    return freq_dist
```

```
In [23]: def plot_word_frequency(freq_dist):
    freq_dist.plot(20, cumulative=False)
    plt.show()
```

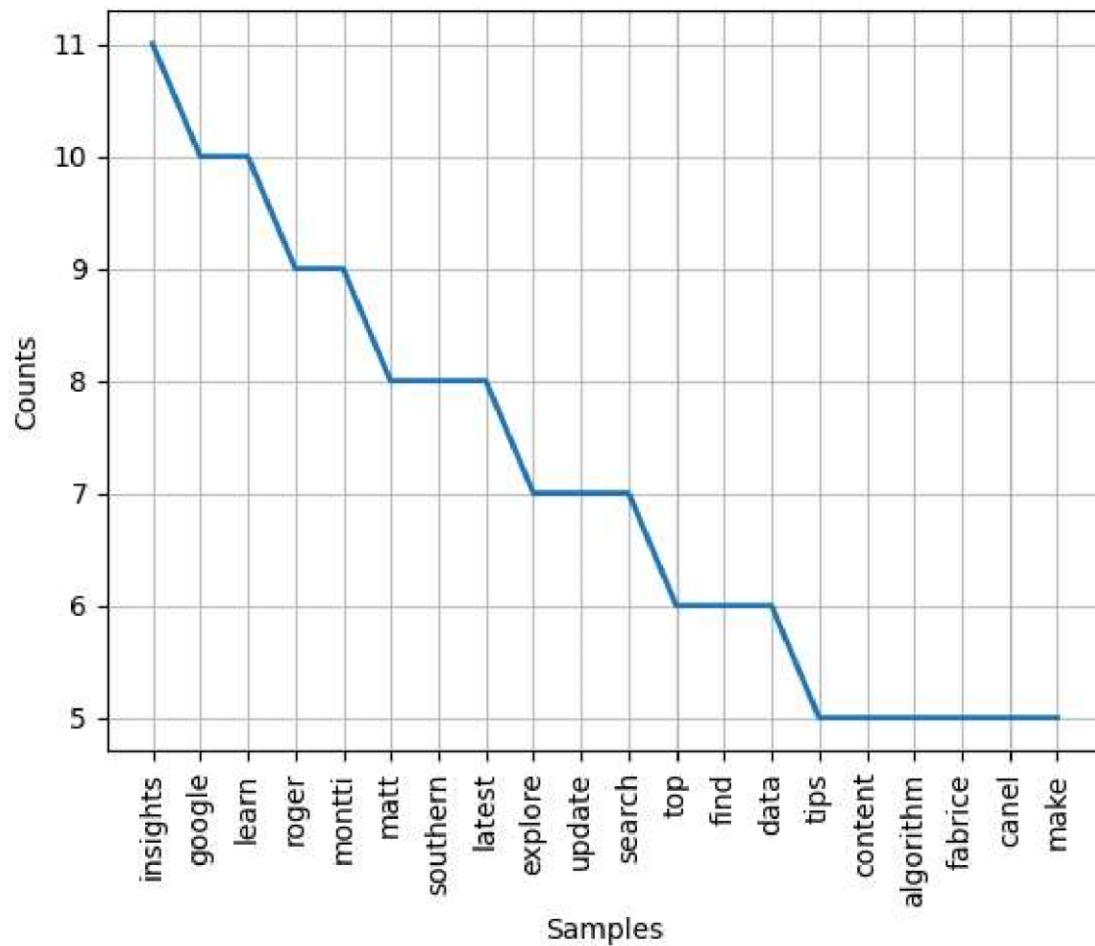
5

P8 P o a b c o n e t m g boo

```
In [24]: if __name__ == "__main__":
    target_url = 'https://www.searchenginejournal.com'

    html_content = get_html_content(target_url)
    if html_content:
        extracted_text = extract_text(html_content)

        word_frequency = analyze_word_frequency(extracted_text)
        plot_word_frequency(word_frequency)
```



7

P9 Po a

b Structure M g

boo

```
In [1]: from bs4 import BeautifulSoup
```

```
In [2]: import requests
```

```
def get_html_content(url):
    response = requests.get(url)
    if response.status_code == 200:
        return response.text
    else:
        print(f"Failed to retrieve content. Status code: {response.status_code}")
        return None
```

```
In [3]: def extract_links(html_content, base_url):
    soup = BeautifulSoup(html_content, 'html.parser')
    links = soup.find_all('a', href=True)
    absolute_links = [link['href'] if link['href'].startswith('http') else f'{base_url}{link["href"]}' for link in links]
    return absolute_links
```

```
In [4]: if __name__ == "__main__":
    target_url = 'https://www.searchenginejournal.com'
```

```
In [6]: html_content = get_html_content(target_url)
if html_content:

    base_url = '/'.join(target_url.split('/')[-3:])
    links = extract_links(html_content, base_url)

    print("Extracted Links:")
    for link in links:
        print(link)
```

Extracted Links:

<https://www.searchenginejournal.com/#main-content> (<https://www.searchenginejournal.com/#main-content>)
<https://www.searchenginejournal.com/#topmenubutton> (<https://www.searchenginejournal.com/#topmenubutton>)
<https://www.searchenginejournal.com> (<https://www.searchenginejournal.com>)
<https://www.searchenginejournal.com/category/news/> (<https://www.searchenginejournal.com/category/news/>)
<https://www.searchenginejournal.com/category/news/> (<https://www.searchenginejournal.com/category/news/>)
<https://www.searchenginejournal.com/category/news/?ver=seo> (<https://www.searchenginejournal.com/category/news/?ver=seo>)
<https://www.searchenginejournal.com/category/news/?ver=pay-per-click> (<https://www.searchenginejournal.com/category/news/?ver=pay-per-click>)
<https://www.searchenginejournal.com/category/news/?ver=social> (<https://www.searchenginejournal.com/category/news/?ver=social>)
<https://www.searchenginejournal.com/category/webinar/> (<https://www.searchenginejournal.com/category/webinar/>)

