import pandas as pd

import requests

from bs4 import BeautifulSoup

def DownloadWeb(url, filename):

response = requests.get(url)

webContent = BeautifulSoup(response.content, 'html.parser')

pretty = BeautifulSoup(webContent.prettify(), 'html.parser')

title = pretty.find('title')

title1 = title.get\_text()

body = pretty.find('body')

para = body.find('div', class\_='td-post-content tagdiv-type')

if para != None:

para1 = para.find\_all('p')

paratext1 = []

for x in para1:

paratext1.append(x.get\_text())

with open(filename, 'w', encoding='utf8') as fd:

print(title1, paratext1, file=fd)

else:

with open(filename, 'w', encoding='utf8') as fd:

print('Class not found', file=fd)

def PositiveCount(filename):

with open(filename, 'rt', encoding='utf-8') as fd1:

print(filename)

file\_content = fd1.read()

file\_content\_list = file\_content.split()

print(file\_content\_list)

with open(r"C:\Users\gurup\Downloads\positive-words.txt", 'rt', encoding='utf-8') as fd2:

positive\_file\_content = fd2.read()

positive\_file\_content\_list = positive\_file\_content.split()

positive\_word\_count = sum(x in positive\_file\_content\_list for x in file\_content\_list)

df1 = pd.read\_excel(r"C:\Users\gurup\Downloads\Output Data Structure1.xlsx", sheet\_name='Sheet1')

df1.at[index, 'POSITIVE SCORE'] = positive\_word\_count

df1.to\_excel(r"C:\Users\gurup\Downloads\Output Data Structure1.xlsx", index=False)

return positive\_word\_count

def NegativeCount(filename):

with open(filename, 'rt', encoding='utf-8') as fd1:

file\_content = fd1.read()

file\_content\_list = file\_content.split()

with open(r"C:\Users\gurup\Downloads\negative-words.txt") as fd2:

negative\_file\_content = fd2.read()

negative\_file\_content\_list = negative\_file\_content.split()

negative\_word\_count = sum(x in negative\_file\_content\_list for x in file\_content\_list)

df1 = pd.read\_excel(r"C:\Users\gurup\Downloads\Output Data Structure1.xlsx", sheet\_name='Sheet1')

df1.at[index, 'NEGATIVE SCORE'] = negative\_word\_count

df1.to\_excel(r"C:\Users\gurup\Downloads\Output Data Structure1.xlsx", index=False)

return negative\_word\_count

def polarity\_score(filename):

with open(filename, 'rt', encoding='utf-8') as fd1:

file\_content = fd1.read()

file\_content\_list = file\_content.split()

with open(r"C:\Users\gurup\Downloads\positive-words.txt", 'rt', encoding='utf-8') as fd2:

positive\_file\_content = fd2.read()

positive\_file\_content\_list = positive\_file\_content.split()

positive\_word\_count = sum(x in positive\_file\_content\_list for x in file\_content\_list)

with open(r"C:\Users\gurup\Downloads\negative-words.txt") as fd2:

negative\_file\_content = fd2.read()

negative\_file\_content\_list = negative\_file\_content.split()

negative\_word\_count = sum(x in negative\_file\_content\_list for x in file\_content\_list)

polarity\_score\_count = (positive\_word\_count - negative\_word\_count) / (

(positive\_word\_count + negative\_word\_count) + 0.000001)

df1 = pd.read\_excel(r"C:\Users\gurup\Downloads\Output Data Structure1.xlsx", sheet\_name='Sheet1')

df1.at[index, 'POLARITY SCORE'] = polarity\_score\_count

df1.to\_excel(r"C:\Users\gurup\Downloads\Output Data Structure1.xlsx", index=False)

def stop\_words(filename):

with open(filename, 'rt', encoding='utf-8') as fd1:

file\_content = fd1.read()

file\_content\_list = file\_content.split()

file\_content\_list1 = set(file\_content\_list)

with open(r"C:\Users\gurup\Downloads\StopWords\_Auditor.txt", 'rt', encoding='utf-8') as fd2:

stopword\_file\_content = fd2.read()

stopword\_file\_content\_list = stopword\_file\_content.split()

stopword\_file\_content\_list1 = set(stopword\_file\_content\_list)

with open(r"C:\Users\gurup\Downloads\StopWords\_Currencies.txt", 'rt') as fd3:

stopword\_file\_content1 = fd3.read()

stopword\_file\_content\_list2 = stopword\_file\_content1.split()

stopword\_file\_content\_list3 = set(stopword\_file\_content\_list2)

with open(r"C:\Users\gurup\Downloads\StopWords\_DatesandNumbers.txt", 'rt', encoding='utf-8') as fd4:

stopword\_file\_content2 = fd4.read()

stopword\_file\_content\_list4 = stopword\_file\_content2.split()

stopword\_file\_content\_list5 = set(stopword\_file\_content\_list4)

with open(r"C:\Users\gurup\Downloads\StopWords\_Generic.txt", 'rt', encoding='utf-8') as fd5:

stopword\_file\_content3 = fd5.read()

stopword\_file\_content\_list6 = stopword\_file\_content3.split()

stopword\_file\_content\_list7 = set(stopword\_file\_content\_list6)

with open(r"C:\Users\gurup\Downloads\StopWords\_GenericLong.txt", 'rt', encoding='utf-8') as fd6:

stopword\_file\_content4 = fd6.read()

stopword\_file\_content\_list8 = stopword\_file\_content4.split()

stopword\_file\_content\_list9 = set(stopword\_file\_content\_list8)

with open(r"C:\Users\gurup\Downloads\StopWords\_Geographic.txt", 'rt', encoding='utf-8') as fd7:

stopword\_file\_content5 = fd7.read()

stopword\_file\_content\_list10 = stopword\_file\_content5.split()

stopword\_file\_content\_list11 = set(stopword\_file\_content\_list10)

with open(r"C:\Users\gurup\Downloads\StopWords\_Names.txt", 'rt', encoding='utf-8') as fd8:

stopword\_file\_content6 = fd8.read()

stopword\_file\_content\_list12 = stopword\_file\_content6.split()

stopword\_file\_content\_list13 = set(stopword\_file\_content\_list12)

finalwords = file\_content\_list1 - stopword\_file\_content\_list1 - stopword\_file\_content\_list3 - stopword\_file\_content\_list5 - stopword\_file\_content\_list7 - stopword\_file\_content\_list9 - stopword\_file\_content\_list11 - stopword\_file\_content\_list13

NoOfFinalwords = (len(finalwords))

with open(r"C:\Users\gurup\Downloads\negative-words.txt") as fd9:

negative\_file\_content = fd9.read()

negative\_file\_content\_list = negative\_file\_content.split()

negative\_word\_count = sum(x in negative\_file\_content\_list for x in file\_content\_list)

with open(r"C:\Users\gurup\Downloads\positive-words.txt", 'rt', encoding='utf-8') as fd10:

positive\_file\_content = fd10.read()

positive\_file\_content\_list = positive\_file\_content.split()

positive\_word\_count = sum(x in positive\_file\_content\_list for x in file\_content\_list)

SubjectivityScore = (positive\_word\_count + negative\_word\_count) / (NoOfFinalwords + 0.000001)

print(SubjectivityScore)

df1 = pd.read\_excel(r"C:\Users\gurup\Downloads\Output Data Structure1.xlsx", sheet\_name='Sheet1')

df1.at[index, 'SUBJECTIVITY SCORE'] = SubjectivityScore

df1.to\_excel(r"C:\Users\gurup\Downloads\Output Data Structure1.xlsx", index=False)

df = pd.read\_excel(r'C:\\Users\\gurup\\Downloads\\Input.xlsx', sheet\_name='Sheet1')

"""col1 = pd.DataFrame(df, columns=['URL\_ID'])

col2 = pd.DataFrame(df, columns=['URL'])

url\_id\_list = col1.values.tolist()

url\_list = col2.values.tolist()

"""

for index, row in df.iterrows():

url\_id = row['URL\_ID']

url = row['URL']

filename = f'{url\_id}.txt'

DownloadWeb(url, filename)

PositiveCount(filename)

NegativeCount(filename)

polarity\_score(filename)

stop\_words(filename)

Downloaded text files sample

<https://drive.google.com/file/d/1vqeDefvqLhYwVYpVWLW_LArBdAMxHX_E/view?usp=sharing>

<https://drive.google.com/file/d/1M2QR5R1qIbNo-rucvJwPJqXurA4ORPTq/view?usp=sharing>

<https://drive.google.com/file/d/17dhAMLFrWqxteWKbtY-JZRl4oep0szhy/view?usp=sharing>

<https://drive.google.com/file/d/16F9FsmotfeTwfEwZ7vY97rB3EyIgDba3/view?usp=sharing>