import requests # Importing requests to extract content from a url

from bs4 import BeautifulSoup as bs # Beautifulsoup is for web scrapping...used to scrap specific content

import re

import os

import urllib.request

import urllib.parse

import urllib.error

from bs4 import BeautifulSoup

import ssl

import json

# For ignoring SSL certificate errors

ctx = ssl.create\_default\_context()

ctx.check\_hostname = False

ctx.verify\_mode = ssl.CERT\_NONE

import nltk

from nltk.corpus import stopwords

import matplotlib.pyplot as plt

from wordcloud import WordCloud

# creating empty reviews list

iphone\_reviews=[]

#forest = ["the","king","of","jungle"]

#url="https://www.amazon.in/All-New-Kindle-Display-Built-Light/product-reviews/B07FRJTZ4T/ref=dpx\_acr\_txt?showViewpoints=1"

#url = "https://www.amazon.in/All-New-Kindle-reader-Glare-Free-Touchscreen/product-reviews/B0186FF45G/ref=cm\_cr\_getr\_d\_paging\_btm\_3?showViewpoints=1&pageNumber="

# response = requests.get(url)

#soup = bs(response.content,"html.parser")# creating soup object to iterate over the extracted content

#reviews = soup.findAll("div",attrs={"class","a-fixed-right-grid view-point"})# Extracting the content under specific tags

#reviews

for i in range(1,4):

ip=[]

url="https://www.amazon.in/All-New-Kindle-Display-Built-Light/product-reviews/B07FRJTZ4T/ref=dpx\_acr\_txt?showViewpoints="+str(i)

#url = "https://www.amazon.in/All-New-Kindle-reader-Glare-Free-Touchscreen/product-reviews/B0186FF45G/ref=cm\_cr\_getr\_d\_paging\_btm\_3?showViewpoints=1&pageNumber="

response = requests.get(url)

soup = bs(response.content,"html.parser")# creating soup object to iterate over the extracted content

reviews = soup.findAll("div",attrs={"class","a-fixed-right-grid view-point"})# Extracting the content under specific tags

reviews

for i in range(len(reviews)):

ip.append(reviews[i].text)

iphone\_reviews=iphone\_reviews+ip # adding the reviews of one page to empty list which in future contains all the reviews

iphone\_reviews

#os.chdir('D:\Python codes\Text mining NLP')

# writng reviews in a text file

with open("iphone.txt","w",encoding='utf8') as output:

output.write(str(iphone\_reviews))

# Joinining all the reviews into single paragraph

ip\_rev\_string = " ".join(iphone\_reviews)

# Removing unwanted symbols incase if exists

ip\_rev\_string = re.sub("[^A-Za-z" "]+"," ",ip\_rev\_string).lower()

ip\_rev\_string = re.sub("[0-9" "]+"," ",ip\_rev\_string)

ip\_rev\_string

# words that contained in iphone 7 reviews

ip\_reviews\_words = ip\_rev\_string.split(" ")

stop\_words = stopwords.words('english')

with open("D:\\Python codes\\Text mining NLP\\sw.txt","r") as sw:

stopwords = sw.read()

stopwords = stopwords.split("\n")

temp = ["this","is","awsome","Data","Science"]

[i for i in temp if i not in "is"]

ip\_reviews\_words = [w for w in ip\_reviews\_words if not w in stopwords]

# Joinining all the reviews into single paragraph

ip\_rev\_string = " ".join(ip\_reviews\_words)

# WordCloud can be performed on the string inputs. That is the reason we have combined

# entire reviews into single paragraph

# Simple word cloud

wordcloud\_ip = WordCloud(

background\_color='black',

width=1800,

height=1400

).generate(ip\_rev\_string)

plt.imshow(wordcloud\_ip)

# positive words # Choose the path for +ve words stored in system

with open("D:\\Python codes\\Text mining NLP\\positive-words.txt","r") as pos:

poswords = pos.read().split("\n")

poswords

poswords = poswords[36:]

# negative words Choose path for -ve words stored in system

with open("D:\\Python codes\\Text mining NLP\\negative-words.txt","r") as neg:

negwords = neg.read().split("\n")

negwords = negwords[37:]

# negative word cloud

# Choosing the only words which are present in negwords

ip\_neg\_in\_neg = " ".join ([w for w in ip\_reviews\_words if w in negwords])

wordcloud\_neg\_in\_neg = WordCloud(

background\_color='black',

width=1800,

height=1400

).generate(ip\_neg\_in\_neg)

plt.imshow(wordcloud\_neg\_in\_neg)

# Positive word cloud

# Choosing the only words which are present in positive words

ip\_pos\_in\_pos = " ".join ([w for w in ip\_reviews\_words if w in poswords])

wordcloud\_pos\_in\_pos = WordCloud(

background\_color='black',

width=1800,

height=1400

).generate(ip\_pos\_in\_pos)

plt.imshow(wordcloud\_pos\_in\_pos)