**Python Tasks**

**Q-1) Scrap any website to find out five top repeated words and perform the following:**

**a- how many times the word has been repeated.**

**b- you have to plot a graph of the output.**

import requests

from bs4 import BeautifulSoup as bs

import operator

from collections import Counter

import matplotlib.pyplot as plt

def start(url):

blacklist = ['[document]','noscript','header','html','meta','head', 'input','script','style']

wordlist=[]

source=requests.get(url).text

soup=bs(source,'html.parser')

text= soup.find\_all(text=True)

for t in text:

if t.parent.name not in blacklist:

word=t.lower().split()

wordlist.append(format(word))

clean\_wordlist(wordlist)

# print(wordlist)

def clean\_wordlist(wordlist):

clean\_list =[]

for word in wordlist:

symbols = '!@#$%^&\*()\_-+={[}]|\;:"<>?/., '

for i in range (0, len(symbols)):

word = word.replace(symbols[i], '')

if len(word) > 0:

clean\_list.append(word)

create\_dictionary(clean\_list)

def create\_dictionary(clean\_list):

word\_count = {}

for word in clean\_list:

if word in word\_count:

word\_count[word] += 1

else:

word\_count[word] = 1

# print(word\_count)

# for key, value in sorted(word\_count.items(), key = operator.itemgetter(1)):

# print ("% s : % s " % (key, value))

c = Counter(word\_count)

# returns the most occurring elements

top = c.most\_common(5)

# print(top)

plotGraph(top)

def plotGraph(top):

keyword=[]

count=[]

for t in top:

keyword.append(t[0])

count.append(t[1])

# ax = fig.add\_axes([0,0,1,1])

plt.bar(keyword,count,align='center',alpha=0.5)

plt.title('Top 5 words on the website and their count')

plt.xlabel("Words")

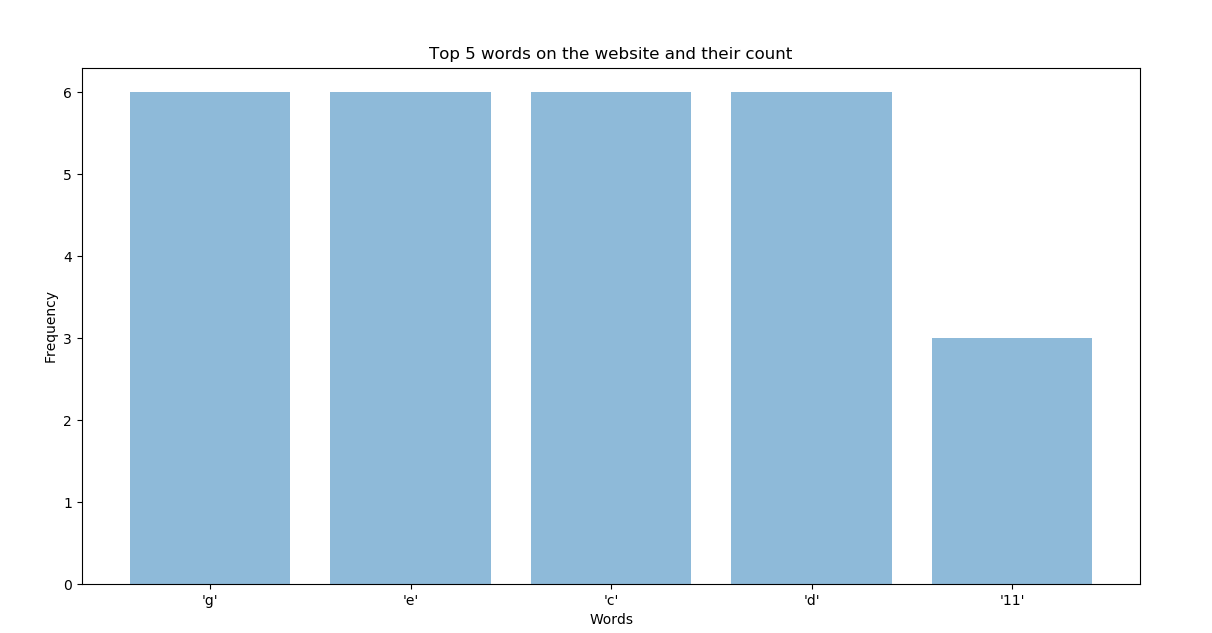
plt.ylabel("Frequency")

plt.show()

if \_\_name\_\_ == '\_\_main\_\_':

url=str(input("Enter your url: "))

start(url)

url=’ <https://sng.nu/en/song/Concrete_Blonde/Joey/357034>’

**Q-2)** **Get input from the users and it is recommended to be in voice command and you**

**need to do a sentimental analysis of the given input [ hint: use textblob library ].**

import speech\_recognition as sr

from textblob import TextBlob

sample\_rate = 48000

#Chunk is like a buffer. It stores 2048 samples (bytes of data)

#here.

#it is advisable to use powers of 2 such as 1024 or 2048

chunk\_size = 2048

#Initialize the recognizer

r = sr.Recognizer()

# mic = s\_r.Microphone(device\_index=1)

with sr.Microphone(device\_index = 1, sample\_rate = sample\_rate,

chunk\_size = chunk\_size) as source:

#wait for a second to let the recognizer adjust the

#energy threshold based on the surrounding noise level

r.adjust\_for\_ambient\_noise(source)

print("Say Something")

#listens for the user's input

audio = r.listen(source)

try:

text = r.recognize\_google(audio)

print("you said: " + text)

blob=TextBlob(text)

print(format(blob.sentiment))

print("In Polarity, 0 indicates neutral, +1 indicates a very positive sentiment and -1 represents a very negative sentiment. ")

print("In subjective, 0.0 is very objective and 1.0 is very subjective.")

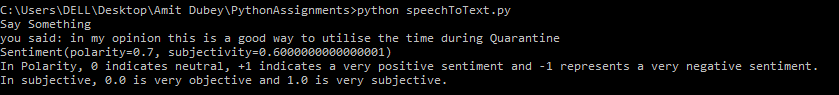
#error occurs when google could not understand what was said

except sr.UnknownValueError:

print("Google Speech Recognition could not understand audio")

except sr.RequestError as e:

print("Could not request results from Google Speech Recognition service; {0}".format(e))



**Q-3) Find different locations of fruit farms in India and save then into an excel sheet. Write a program in python in which you have to pin different locations on map. You also have to make pie chart of fruit farms according to the states. After doing all this we will initiate this idea in second phase.**

import csv

import requests

from bs4 import BeautifulSoup

import pandas as pd

import matplotlib.pyplot as plt

from geopy.geocoders import Nominatim

import folium

#url to scrape table

url = 'https://www.mapsofindia.com/indiaagriculture/fruits-map/'

response = requests.get(url)

html = response.content

heading=[]

soup = BeautifulSoup(html,'html.parser')

table = soup.find('table', attrs={'class': 'tableizer-table'})

tableHeader=table.find\_all('th')

for x in tableHeader:

heading.append(x.text)

# print(heading)

list\_of\_rows = []

for row in table.findAll('tr')[1:]:

list\_of\_cells = []

for cell in row.findAll('td'):

text = cell.text.replace('&nbsp;', '')

list\_of\_cells.append(text)

list\_of\_rows.append(list\_of\_cells)

outfile = open("./fruit.csv", "w")

writer = csv.writer(outfile)

writer.writerow(heading)

writer.writerows(list\_of\_rows)

outfile.close()

#ploting pie-graph and marker on map

df = pd.read\_csv ('fruit.csv')

state\_name = df[heading[1]]

production = df[heading[2]]

colors = ["#1f77b4", "#ff7f0e", "#2ca02c", "#d62728", "#8c564b"]

plt.title("Production of fruits in different states")

plt.pie(production , labels=state\_name , colors=colors,

autopct='%1.1f%%', shadow=True, startangle=140)

plt.show()

dict={}

for x in state\_name:

print("\n",x,"\n")

loc = geolocator.geocode(x+','+ country)

dict[x]=loc.latitude,loc.longitude

m=folium.Map(location=list(dict['Gujarat']),zoom\_start=5,tiles='Stamen Terrain')

for x in dict:

folium.Marker(

location=list(dict[x]),

popup=x,

icon=folium.Icon(color='red')

).add\_to(m)

m.save('index.html')

**Output:**



