**Programming with Python for Data Sciences**

**Project Title: Develop an SEO Tool to Analyze Live Web Pages**

Done by

**T.S. Vijay**

**S191165200005**

**Synopsis**

Search Engine Optimization (SEO) is an important aspect of a Web page to gain importance for a search engine to be able to display it earlier in the search list. The optimization is based on a lot of factors such as title, description, header tags and keyword density. Different search engines will have their own mechanisms that calculate the score of a keyword on the page and thus work out its ranking in the search order.

The course project is to develop a generic SEO toolset which will help a Web developer analyze live Web pages for keywords and other components of the page which contribute to SEO. We should be able to provide the keywords of interest and understand its density pattern across the various components of the HTML page. The Web page analysis can analyze how many ever Web pages the user inputs and the results are saved as reports in spread sheets with graph.

The project would not only give exposure to a real life problem solving using Python, but would also make learners understand about Internet and Search Engines and how they relate to each other.

**HOW IT WORKS**

The user is prompted to input a website URL (Uniform Resource Locator). After successful verification that the URL is valid, the user is prompted to enter one or many keywords, with a space between them to be searched. The code written is programmed to search through the text of the web page and find out all occurrences of the keyword entered. It displays the value to the user with the corresponding keyword entered and also writes the information into an excel sheet and a text file. The user is then displayed three choices.

1. To search again in the same web page
2. To change the URL and search
3. Exit

The user may choose to continue or terminate.

**EXPLAINING THE CODE**

**LIBRARIES USED:**

**Web Scraping**

The project is centered on this topic. Web Scraping in python is done through two libraries namely urlllib and BeautifulSoup 4.

* urllib.request
  + This is the library that contains the method urlopen which scraps the web page form the internet.
  + It downloads the whole web page as a whole blandly and it is not easily readable to humans, and this is where BeautifulSoup comes in.
* bs4
  + This is the library that contains the method which beautifies the html code of the downloaded website.
  + Its main function is to enable the programmer to read the code and discard any unwanted information.

**Spreadsheet creator**

In order to visualize the data with ease and store the information permanently, xlsxwriter is the perfect tool. It is the library which contains a variety of powerful methods which help to write the information to the spreadsheet and create graphs.

**Regular expressions**

Regular expressions is a powerful tool which helps in identifying a particular piece of information in the data obtained. It is used to verify the URL which the user inputs. The methods are contained in the library called re.

**THE CODE:**

* The entire code is encapsulated inside a for loop in order to continue the program when the user inputs his/her wish after searching.
* The user asked to input the URL of the web page which is achieved using the input() function.
* The URL is cross-checked using the regular expressions code. It is done using the methods, compile() and match().

Code used:

pattern = re.compile('http[s]?://[a-z]{2,4}\.(.+)\.(.{2,4})')

matchobject = pattern.match(url)

* If the URL is valid, the program continues and is written to the spread sheet and a text file, else, a message is displayed to the user to input the URL properly. This is done using if…else block.

Code used:

(inside a for loop)

if (matchobject):

#writing to the file and xlsx

fo.writelines(url)

fo.writelines(' ')

worksheet.write(wr,wc,url)

wc+=1

break

else :

print('Invalid url address. Please enter absolute url address.')

print('Example: https://www.google.com/')

#writing to file

fo.writelines(url)

fo.writelines(' ')

continue

* The web page is downloaded (code present above) first in the urlopen() function and then beautified using BeautifulSoup() function.

Code used

#converting it into readable form

holder = urlopen(url)

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

* All the script and style tags are removed along with the contents inside those tags using extract() function. Then the text in the remaining tags alone are extracted using the get\_text() function. The text is then converted into lowercase.

Code used:

#removing script and style tags

for script in soup('script',{'style':'script'}):

script.extract()

#getting text

text=soup.get\_text().strip()

* The user is now prompted to input the keywords to be searched. The input can contain many keywords with space as a separation.
* If the input is multiple words, the words are put into a list as individual words using the split() function.

Code used:

words = input('Enter the word you want to search : ')

words=words.lower()

word\_list=words.split()

* This list is then cross checked with the text from the web page. This is done using for loops and subsequently the count of the occurrence of the key words are stored.
* The word with the corresponding count is written in the spreadsheet using worksheet.write() function. It is also written into the text file.
* The user is then asked if he/she wants to continue searching in the same website or change the URL and search in that website. This is done using for loop.
* The process in a loop if the user chooses to continue else the program repeats itself terminates and closes the workbook.
* This can be repeated multiple number of times.

**SOURCE CODE**

import re

from bs4 import BeautifulSoup as bs

from urllib.request import urlopen

import xlsxwriter

#opening a file to store the data

fo = open('bckup.txt','a')

#creating workbook for xlsxwriter

workbook=xlsxwriter.Workbook('data.xlsx')

worksheet=workbook.add\_worksheet()

worksheet.write('A1','URL')

worksheet.write('B1','WORD')

worksheet.write('C1','FREQUENCY')

#creating two variables for xlsx

wr=1

wc=0

#to plot charts for multiple urls

next\_chart=0

#to repeat the process with a new url

endless=10000000000000

#checking if the url is correct

for rep in range(endless):

#count for frequency

count = 0

for x in range(endless):

#obtaining input from the user

url = input('Enter the url you want to search in : ')

#regex to make sure that the url is absolute

pattern = re.compile('http[s]?://[a-z]{2,4}\.(.+)\.(.{2,4})')

matchobject = pattern.match(url)

if (matchobject):

#writing to the file and xlsx

fo.writelines(url)

fo.writelines(' ')

worksheet.write(wr,wc,url)

wc+=1

break

else :

print('Invalid url address. Please enter absolute url address.')

print('Example: https://www.google.com/')

#writing to file

fo.writelines(url)

fo.writelines(' ')

continue

#converting it into readable form

holder = urlopen(url)

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

#removing script and style tags

for script in soup('script',{'style':'script'}):

script.extract()

#getting text

text=soup.get\_text().strip()

#print(text)

text\_list=[]

for i in range(endless):

#obtaining the words to be searched

words = input('Enter the word you want to search : ')

words=words.lower()

word\_list=words.split()

#writing to file

fo.writelines(words)

fo.writelines(' ')

for wordx in word\_list:

worksheet.write(wr,wc,wordx)

wc+=1

#searching for the words in the file

text\_lower = text.lower()

text\_lower = text\_lower.split()

for lowertext in text\_lower:

if wordx in lowertext:

count+=1

else :

continue

print(wordx, 'has occurred ', count,' time(s).')

#writing to file and xlsx

fo.writelines(wordx)

fo.writelines(' ')

worksheet.write(wr,wc,count)

wr+=1

wc = 1

#writing to file

fo.writelines(str(count))

fo.writelines(' ')

#to renew the loop

count = 0

yn = input(""" If you want to search again in the same url, press 1.

If you want to change the url and search again, press 2.

If you want to exit, press 3.

Enter your wish : """)

if yn == '1':

#writing to file

fo.writelines(yn)

fo.writelines(' ')

wc=1

continue

else :

break

if yn == '2' :

#count becomes zero

count=0

#writing to file

fo. writelines(yn)

fo.writelines(' ')

wc=0

continue

else :

#writing to file

fo. writelines('end\n')

chart = workbook.add\_chart({'type': 'line'})

chart.set\_x\_axis({'name': 'Word'})

chart.set\_y\_axis({'name': 'Frequency'})

chart.add\_series({

'categories': '=Sheet1!$B${}:$B${}'.format(2,wr),

'values': '=Sheet1!$C${}:$C${}'.format(2,wr)

})

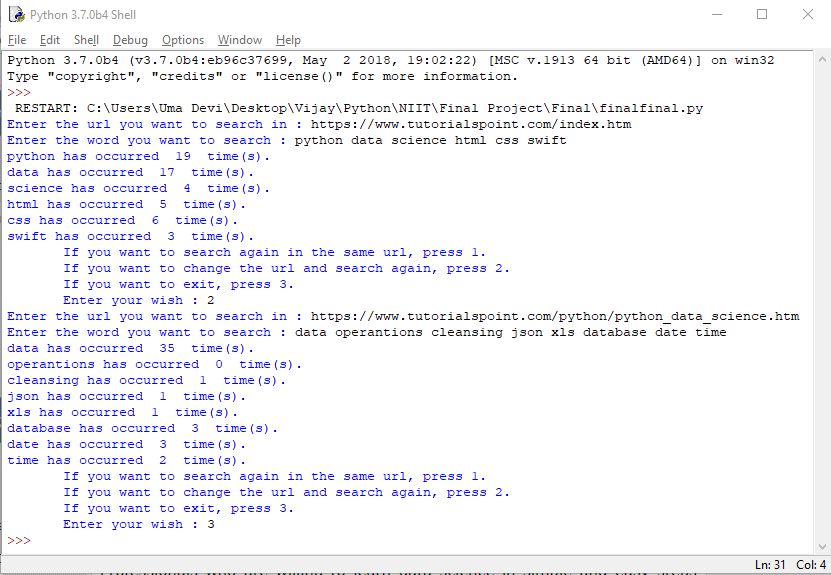
worksheet.insert\_chart('E2', chart)

break

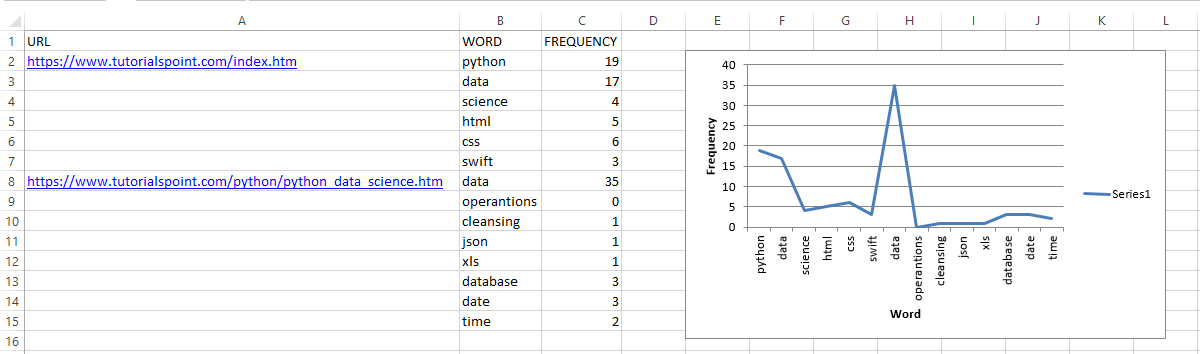
fo.close()

workbook.close()

**USER INTERFACE**



**SPREADSHEET AND GRAPH**



**CONSTRAINTS**

* Some websites such as udemy have restricted the download of web pages.
* This program cannot be performed in a batch mode as it only accepts one URL at a time.
* This is works only in the python interface. A user need to know how to open the python file and run the program.

**USES**

* This is the mechanism of a search engine tool. It helps to identify all relevant websites related to a topic.
* It is very helpful in finding out the keywords in a website quickly and easily.
* It comes in very handy when the user is trying to check if a website is relevant to the topic they are researching.