IT WORKSHOP MINI PROJECT

Topic: Text Analyser

Group Members:

Ruchika Pandharikar	BT19CSE009
Aditi Vishwakarma	BT19CSE031
Samruddhi Selukar	BT19CSE033
Neha Kalbande	BT19CSE047
Aditi Sahu	BT19CSE057

Under the guidance of **Dr. Mayuri Digalwar**

INDEX

INTRODUCTION

What Is Text Analysis?

Text Analysis is the automated process of examining text by allowing you to find the most frequent phrases and frequencies of words. It also counts number of words, characters, sentences and syllables.

Analysing these texts by hand is time-consuming, tedious, and ineffective, especially if you deal with large amounts of data every day. Walmart, for example, receives 200 billion rows of transactional data in just a few weeks! Imagine if they wanted to process this data with human agents: it would be impossible.

Text analysis helps you deal with information overload. It never gets tired, bored, or changes its criteria, and can analyse hundreds and thousands of pieces of data in just a few seconds. it's really simple to get started with. Not only will they help you save a lot of time and resources when analysing data, but it will also allow your teams to focus on more pressing (and motivating) tasks.

How to Use Text Analysis with Python

Python is the most popular programming language today, especially in the field of scientific computing, as it is a highly intuitive language when compared to others such as Java. It's more concise, so it takes less time and effort to carry out certain operations. Finally, the syntax and code readability make it efficient, easy to process, and easy to learn. All these perks make Python the perfect option to build a model for text analysis.

Text analysis is a technique that allows companies to automatically extract and classify text data, such as tweets, emails, support tickets, product reviews, and survey responses.

Text Analysis Techniques

There are basic and more advanced text analysis techniques, each used for different purposes. First, learn about the simpler text analysis techniques you might use each one.

Word Frequency

Word frequency is a text analysis technique that measures the most frequently occurring words or concepts in a given text using the numerical statistic.

Text Extraction

Text extraction is another widely used text analysis technique that extracts pieces of data that already exist within any given text. You can extract things like keywords, prices, company names, and product specifications from news reports, product reviews, and more.

Why is Text Analysis Important?

Text analysis can stretch its wings across a range of texts depending on the results you desire. It can be applied to:

- **Whole documents**: obtains information from a complete document or paragraph: e.g., the overall sentiment of a customer review.
- **Single sentences**: obtains information from specific sentences: e.g., more detailed sentiments of every sentence of a customer review.
- **Sub-sentences**: obtains information from sub-expressions within a sentence: e.g., the underlying sentiments of every opinion unit of a customer review.

WORKING

Text Analyzer is simply used as a text analyzing tool for counting words, characters, special characters (like @, #, $, %, &) in your text that you can type.

Here we made a simple text analyzer.

Language: Python

Library used while making text analyzer are:

- Tkinter
- Matplotlib
- Pandas

Let’s get familiar with the library

Tkinter:

Tkinter is the Python interface to the Tk GUI toolkit shipped with Python. Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Matplotlib:

Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays. It provides an object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, WxPythonotTkinter. It can be used in Python and IPython shells, Jupyter notebook and web application servers also.

Pandas:

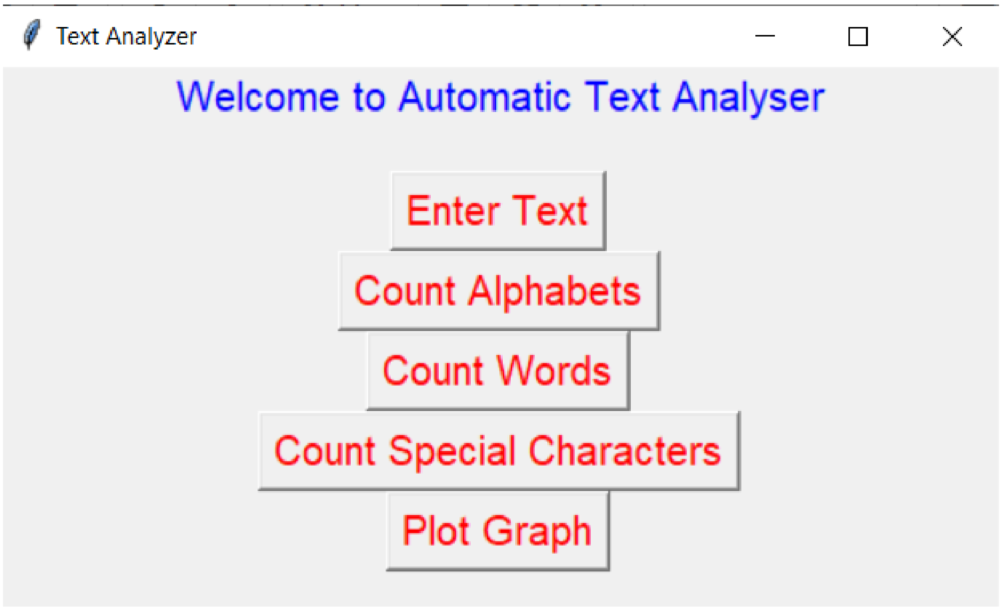
Pandas is defined as an open-source library that provides high-performance data manipulation in Python.

Pandas provide two data structures, which are supported by the pandas library**:**

- **Series, and**
- **DataFrames.**

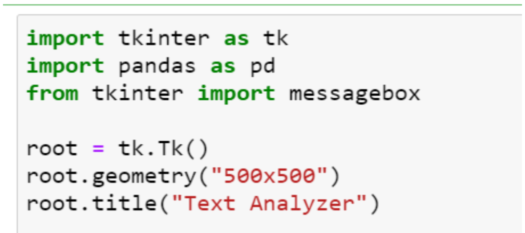
Now let's see how we used these libraries to make text analyzer more effective.

With the help of tkinter we designed a following window



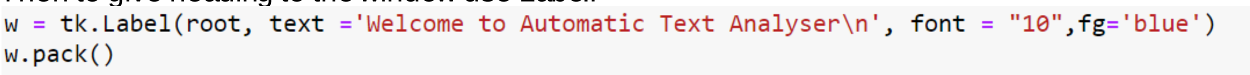
To create a window of 500x500 dimensions and with title text analyzer do the following

Import tkinter will import the tkinter module , root.geometry is defining the dimensions of the window . And root.tiltle is giving title to our window.



```
import tkinter as tk
import pandas as pd
from tkinter import messagebox

root = tk.Tk()
root.geometry("500x500")
root.title("Text Analyzer")
```

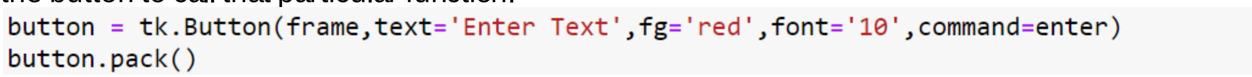
Then to give heading to the window use Label:



```
w = tk.Label(root, text ='Welcome to Automatic Text Analyser\n', font = "10",fg='blue')
w.pack()
```

And want to give specific color and font size to the text use font and fg.

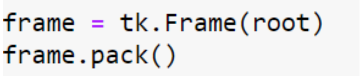
Then to call particular function of text analyzer we made buttons. So, user have to click on the button to call that particular function.



```
button = tk.Button(frame,text='Enter Text',fg='red',font='10',command=enter)
button.pack()
```

Here text =’Enter Text’ is conveying the purpose of that button and command=enter will call function enter ().

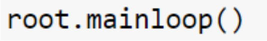
And for grouping all of these Frame is used



```
frame = tk.Frame(root)
frame.pack()
```

The Frame widget is very important for the process of grouping and organizing other widgets in a somehow friendly way. It works like a container, which is responsible for arranging the position of other widgets.

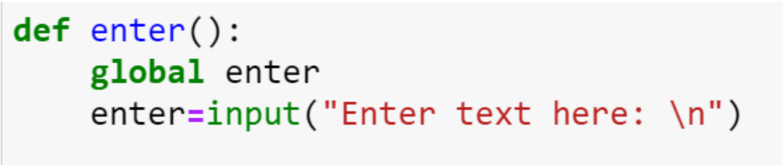
And at the end of program enter the main event loop to take action against each event triggered by the user.



```
root.mainloop()
```

There is different function which will do different work of text analyzer. Let’s have a look on those functions.

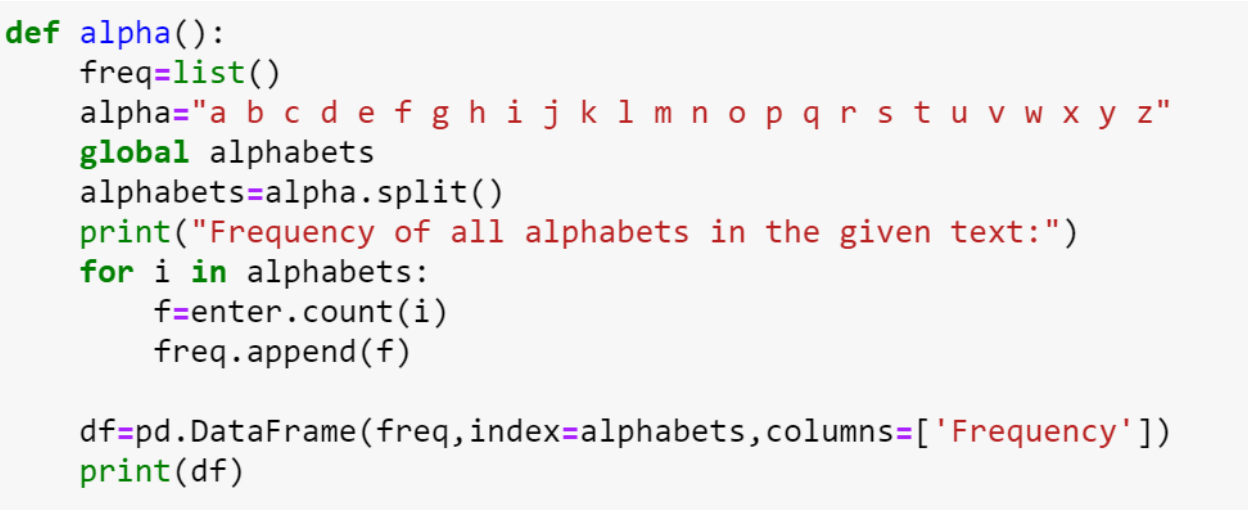
<u>enter function</u>: It will allow user to input the text which they want to analyze.



```
def enter():
    global enter
    enter=input("Enter text here: \n")
```

With the help of input () function user can enter the text which will be stored in enter variable which is global means available throughout the program.

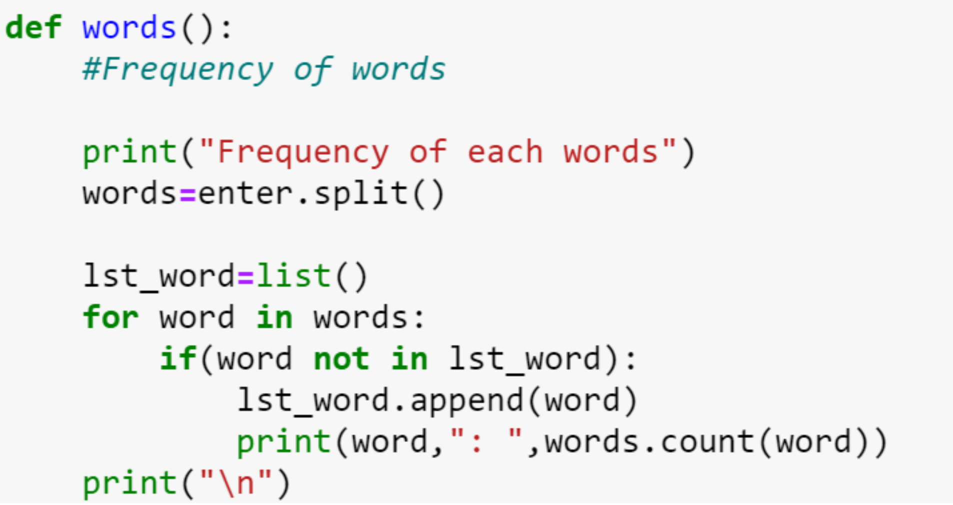
<u>alpha Function:</u> This function will help us to find the frequency of alphabets present in the text entered in enter variable and make a list (named as alphabets) containing all alphabets in lower case.



```
def alpha():
    freq=list()
    alpha="a b c d e f g h i j k l m n o p q r s t u v w x y z"
    global alphabets
    alphabets=alpha.split()
    print("Frequency of all alphabets in the given text:")
    for i in alphabets:
        f=enter.count(i)
        freq.append(f)

    df=pd.DataFrame(freq,index=alphabets,columns=['Frequency'])
    print(df)
```

In this function we will take each alphabet print in alphabet list and count the number of occurrences of that particular alphabets in text entered with the help of count () function and append it in freq list. At the end with the help of pandas library we will create a dataframe containing freq list with heading as “Frequency”.

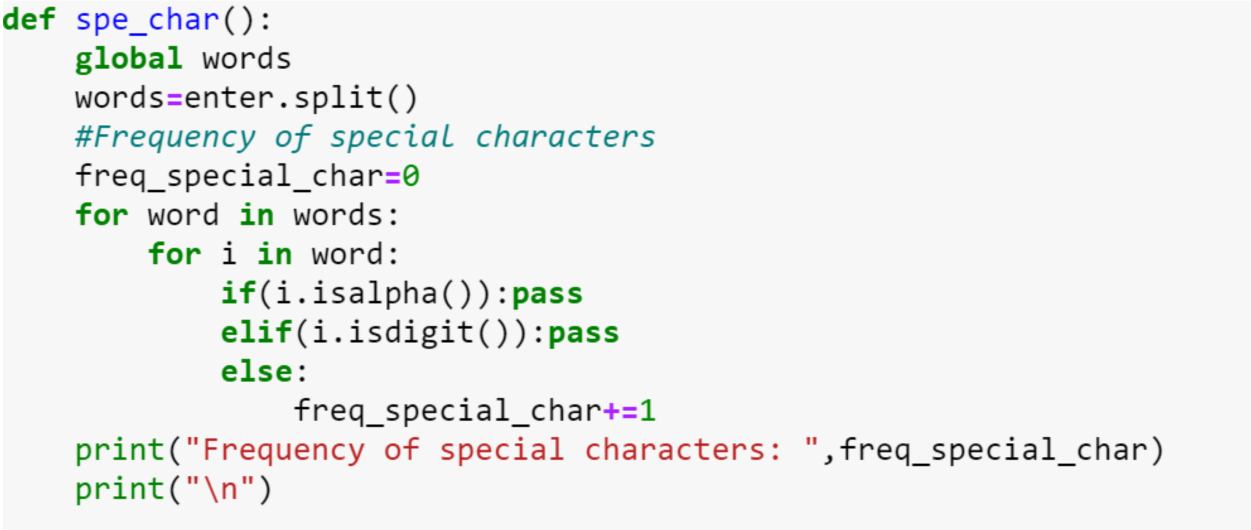
Words Function: This function will count the frequency of each word in the entered text and will print this frequency.



```
def words():
    #Frequency of words

    print("Frequency of each words")
    words=enter.split()

    lst_word=list()
    for word in words:
        if(word not in lst_word):
            lst_word.append(word)
            print(word,": ",words.count(word))
    print("\n")
```

As we can see above words function will first split the entered text and make a list of it. Then we will make another list (named as lst_word) which we keep in account the words whose frequency is already been checked. In for loop we will take each element of words list and check if it is not in lst_word then we will add it into it and print the frequency of that word in the entered text.

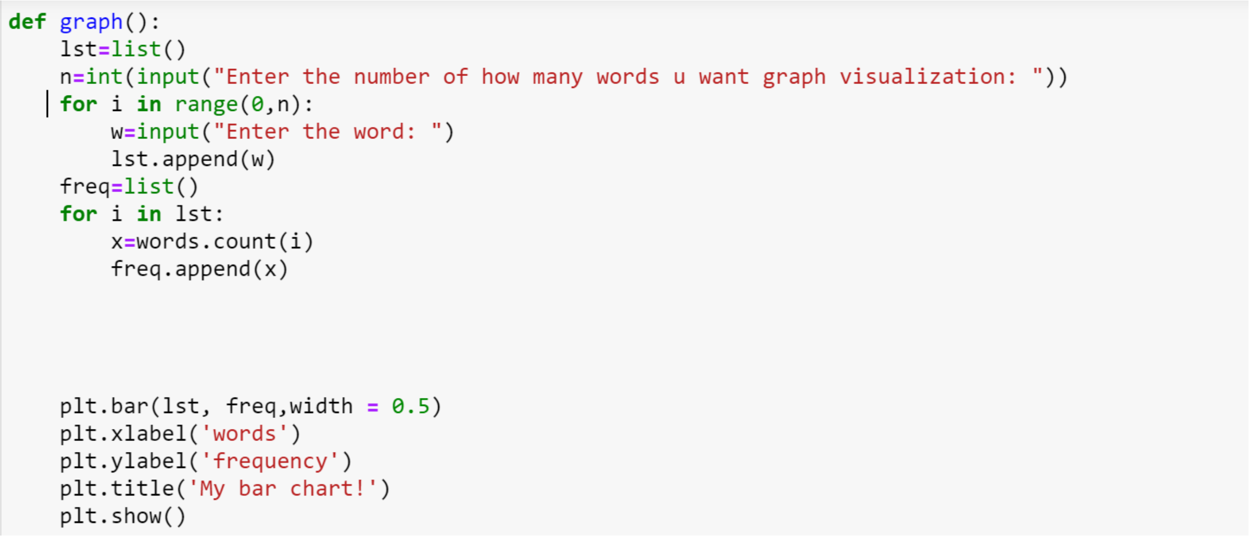
spe_char Function: This function is used to count the total occurrence of all special character in entered text.



```
def spe_char():
    global words
    words=enter.split()
    #Frequency of special characters
    freq_special_char=0
    for word in words:
        for i in word:
            if(i.isalpha()):pass
            elif(i.isdigit()):pass
            else:
                freq_special_char+=1
    print("Frequency of special characters: ",freq_special_char)
    print("\n")
```

Here, as we can see we will first take one word for the entered text then we starting taking character for that particular word. After this we will check if the selected character is alphabets or not. If it is not an alphabet then we will check if it is digit or not. If it is not digit then it means that this character is a special character. Hence, we will increase the count of special character by 1.

graph Function: This function will draw the graph for word versus its frequency.



```
def graph():
    lst=list()
    n=int(input("Enter the number of how many words u want graph visualization: "))
    for i in range(0,n):
        w=input("Enter the word: ")
        lst.append(w)
    freq=list()
    for i in lst:
        x=words.count(i)
        freq.append(x)



    plt.bar(lst, freq,width = 0.5)
    plt.xlabel('words')
    plt.ylabel('frequency')
    plt.title('My bar chart!')
    plt.show()
```

Here we will first enter the number of words for which graph is to be draw and then take input of these words in the lst list. Now we will again count the frequency of entered words and store it in the freq list. At the end with the help of matplot library we will plot a bar graph between lst list and freq list. Now we will give labels to x-axis as "words" and y-axis as "frequency" and give it the title "My bar chart!".

Code:

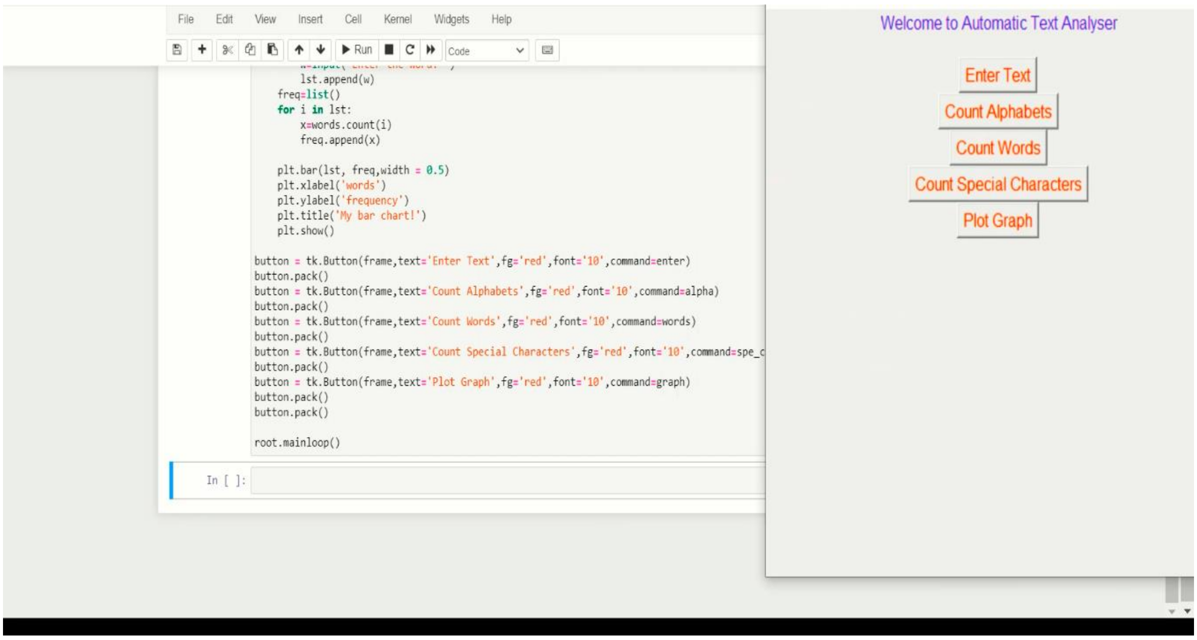
```
import pandas as pd

import matplotlib.pyplot as plt

import tkinter as tk

from tkinter import messagebox


root = tk.Tk()

root.geometry("500x500")

root.title("Text Analyzer")


w = tk.Label(root, text ='Welcome to Automatic Text Analyser\n', font = "10",fg='blue')

w.pack()


frame = tk.Frame(root)

frame.pack()


def enter():

    global enter
    enter=input("Enter text here: \n")

def alpha():
    freq=list()
    alpha="a b c d e f g h i j k l m n o p q r s t u v w x y z"
    global alphabets
    alphabets=alpha.split()
    print("Frequency of all alphabets in the given text:")
    for i in alphabets:
        f=enter.count(i)
        freq.append(f)
    df=pd.DataFrame(freq,index=alphabets,columns=['Frequency'])
    print(df)

def words():
    #Frequency of words

    print("Frequency of each words")
    words=enter.split()

    lst_word=list()
    for word in words:
        if(word not in lst_word):
            lst_word.append(word)
            print(word,": ",words.count(word))
    print("\n")
def spe_char():
    global words
    words=enter.split()
    #Frequency of special characters
    freq_special_char=0
    for word in words:
        for i in word:
            if(i.isalpha()):pass
            elif(i.isdigit()):pass
            else:
                freq_special_char+=1
    print("Frequency of special characters: ",freq_special_char)
    print("\n")

def graph():
    lst=list()
    n=int(input("Enter the number of how many words u want graph visualization: "))
    for i in range(0,n):
        w=input("Enter the word: ")
        lst.append(w)
    freq=list()
    for i in lst:
        x=words.count(i)
        freq.append(x)

    plt.bar(lst, freq,width = 0.5)
    plt.xlabel('words')
    plt.ylabel('frequency')
    plt.title('My bar chart!')
    plt.show()

button = tk.Button(frame,text='Enter Text',fg='red',font='10',command=enter)
button.pack()
button = tk.Button(frame,text='Count Alphabets',fg='red',font='10',command=alpha)
button.pack()
button = tk.Button(frame,text='Count Words',fg='red',font='10',command=words)
button.pack()
button = tk.Button(frame,text='Count Special
Characters',fg='red',font='10',command=spe_char)
```

```
button.pack()
button = tk.Button(frame,text='Plot Graph',fg='red',font='10',command=graph)
button.pack()
button.pack()

root.mainloop()
```

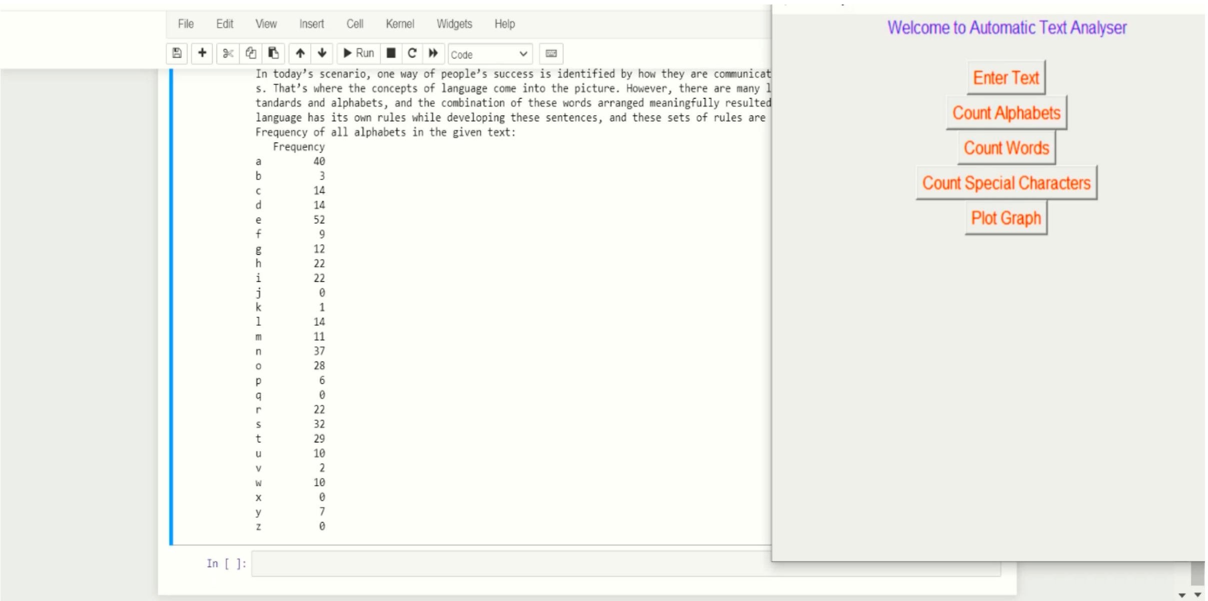
INPUT: We have taken a sample paragraph as input :--

In today’s scenario, one way of people’s success is identified by how they are communicating and sharing information with others. That’s where the concepts of language come into the picture. However, there are many languages in the world. Each has many standards and alphabets, and the combination of these words arranged meaningfully resulted in the formation of a sentence. Each language has its own rules while developing these sentences, and these sets of rules are also known as grammar.

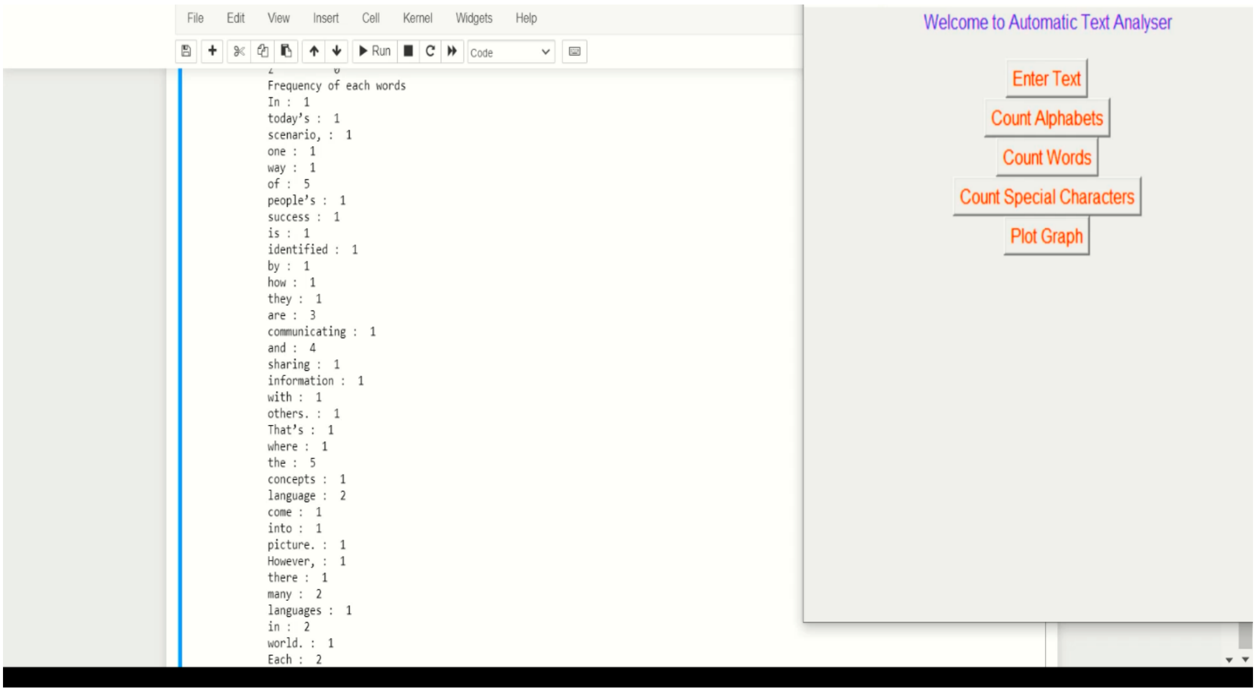
OUTPUT:



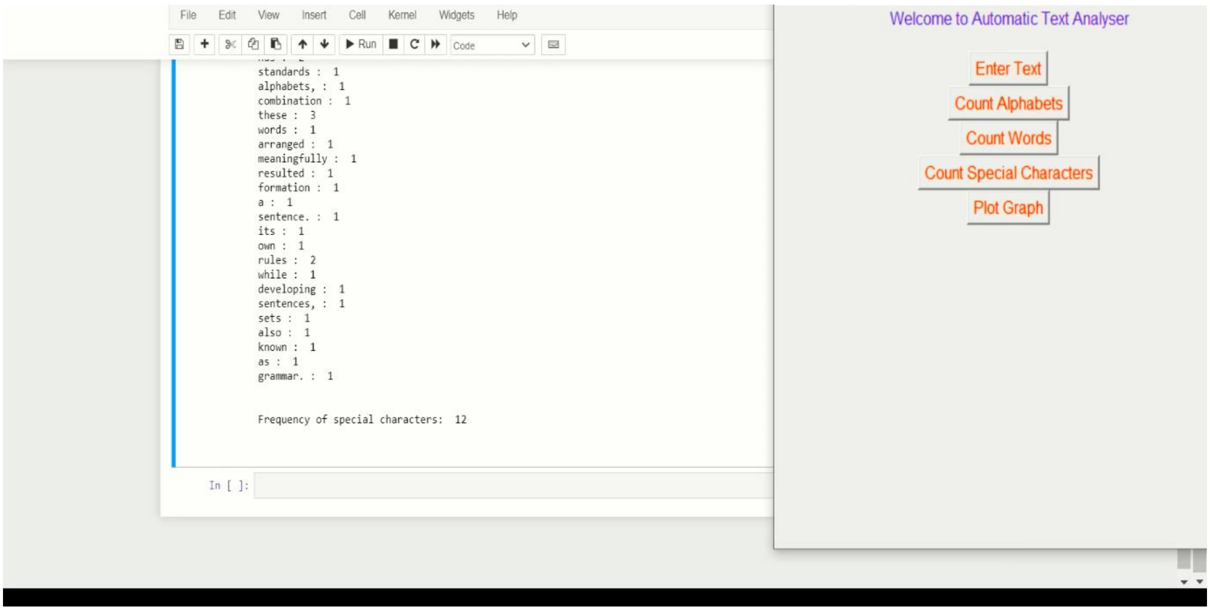
```
        lst.append(w)
    freq=list()
    for i in lst:
        x=words.count(i)
        freq.append(x)

    plt.bar(lst, freq,width = 0.5)
    plt.xlabel('words')
    plt.ylabel('frequency')
    plt.title('My bar chart!')
    plt.show()

button = tk.Button(frame,text='Enter Text',fg='red',font='10',command=enter)
button.pack()
button = tk.Button(frame,text='Count Alphabets',fg='red',font='10',command=alpha)
button.pack()
button = tk.Button(frame,text='Count Words',fg='red',font='10',command=words)
button.pack()
button = tk.Button(frame,text='Count Special Characters',fg='red',font='10',command=spe_c
button.pack()
button = tk.Button(frame,text='Plot Graph',fg='red',font='10',command=graph)
button.pack()
button.pack()

root.mainloop()
```



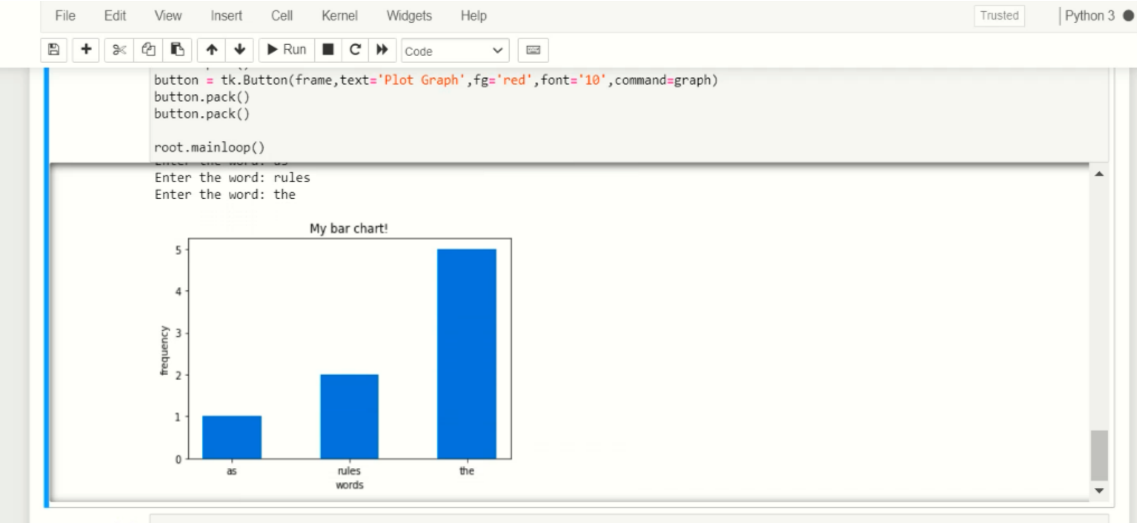
```
In today’s scenario, one way of people’s success is identified by how they are communicat
s. That’s where the concepts of language come into the picture. However, there are many l
tandards and alphabets, and the combination of these words arranged meaningfully resulted
language has its own rules while developing these sentences, and these sets of rules are
Frequency of all alphabets in the given text:
   Frequency
a         40
b          3
c         14
d         14
e         52
f          9
g         12
h         22
i         22
j          0
k          1
l         14
m         11
n         37
o         28
p          6
q          0
r         22
s         32
t         29
u         10
v          2
w         10
x          0
y          7
z          0
```



```
z         0
Frequency of each words
In :  1
today’s :  1
scenario, :  1
one :  1
way :  1
of :  5
people’s :  1
success :  1
is :  1
identified :  1
by :  1
how :  1
they :  1
are :  3
communicating :  1
and :  4
sharing :  1
information :  1
with :  1
others. :  1
That’s :  1
where :  1
the :  5
concepts :  1
language :  2
come :  1
into :  1
picture. :  1
However, :  1
there :  1
many :  2
languages :  1
in :  2
world. :  1
Each :  2
```



```
standards :  1
alphabets, :  1
combination :  1
these :  3
words :  1
arranged :  1
meaningfully :  1
resulted :  1
formation :  1
a :  1
sentence. :  1
its :  1
own :  1
rules :  2
while :  1
developing :  1
sentences, :  1
sets :  1
also :  1
known :  1
as :  1
grammar. :  1


Frequency of special characters:  12
```



```
button = tk.Button(frame,text='Plot Graph',fg='red',font='10',command=graph)
button.pack()
button.pack()

root.mainloop()
```

```
Enter the word: rules
Enter the word: the
```

Link for demonstration video of text analyzer:

https://drive.google.com/folderview?id=1idD8fM-b35AZaD5cwaksoj3VvaZEQVkj

CONCLUSION

This text analyser is used for counting alphabets, words, particular words, special characters and plotting graph of words with their particular frequency of any text.

Text analysis Applications:

You can extract specific information, like keywords, names, or company information from thousands of emails, or categorize survey responses by sentiment and topic.

Subsequently, we use text analytics to help companies find hidden customer insights and be able to easily answer questions about their existing customer data.

Hence, text analyser helps in:

1) Knowledge Management:

When we are not able to find any particular information Quickly is always a challenge when managing large text Documents. This problem is solved by the text analysis.

2) Customer Care Service:

When we have data of valuable information such as surveys, customer reviews so the graph plotting method in the analyser helps us to get the overall response rapidly and effectively.

3) Content Enrichment:

While it’s true that working with text content still require. A bit of human effort, text analytics techniques make a significant difference when It comes to being able to more effectively manage large volumes of information, Organize and summarize the available content.

REFERENCES:

From Video lectures and program files provided

https://en.wikipedia.org/wiki/Matplotlib

https://www.geeksforgeeks.org/text-analysis-in-python-3/

https://www.tutorialspoint.com/matplotlib/in

https://polestarllp.com/text-analytics

https://www.geeksforgeeks.org/python-gui-tkinter/

https://www.learnpython.org/en/Pandas_Basics#:~:text=Pandas%20is%20a%20high%2Dlevel,observations%20and%20columns%20of%20variables.

https://monkeylearn.com/text-analysis/

https://towardsdatascience.com/getting-started-with-text-analysis-in-python-ca13590eb4f7