Skills Network

Practice lab

+ Code + Text

Scenario: Text Analysis

Estimated time needed: **45** minutes

What is Text analysis?

Text analysis, also known as text mining or text analytics, refers to the process of extracting meaningful information and insights from textual data.

Objectives

After completing this lab, you will be able to use Python commands to perform text analysis. This includes converting the text to lowercase and then finding and counting the frequency of all unique words, as well as a specified word.

Setup

For this lab, we will be using the following data types:

- List
- Strings
- Classes and objects

Let's consider a real-life scenario where you are analyzing customer feedback for a product. You have a large dataset of customer reviews in the form of strings, and you want to extract useful information from them using the three identified tasks:

Task 1. String in lower case: You want to Pre-process the customer feedback by converting all the text to lowercase. This step helps standardize the text. Lower casing the text allows you to focus on the content rather than the specific letter casing.

Task 2. Frequency of all words in a given string: After converting the text to lowercase, you want to determine the frequency of each word in the customer feedback. This information will help you identify which words are used more frequently, indicating the key aspects or topics that customers are mentioning in their reviews. By analyzing the word frequencies, you can gain insights into the most common issues raised by customers.

Task 3. Frequency of a specific word: In addition to analyzing the overall word frequencies, you want to specifically track the frequency of a particular word that is relevant to your analysis. For example, you might be interested in monitoring how often the word "reliable" appears in the customer reviews to gauge customer sentiment about the product's reliability. By focusing on the frequency of a specific word, you can gain a deeper understanding of customer opinions or preferences related to that particular aspect.

By performing these tasks on the customer feedback dataset, you can gain valuable insights into customer sentiment

---

Part-A

Note:- In Part-A, you won't be getting any output as we are just storing the string and creating a class.

Step-1 Define a string.

"Lorem ipsum dolor! diam amet, consetetur Lorem magna. sed diam nonumy eirmod tempor. diam et labore? et diam magna. et diam amet."
Hint:- Use a variable and store the above string.

```
#Press Shift+Enter to run the code
givenstring="Lorem ipsum dolor! diam amet, consetetur Lorem magna.
```

For achieving the tasks mentioned in the scenario, We need to create a class with 3 different methods.

Step-2 Define the class and its attributes:

1. Create a class named TextAnalyzer.
2. Define the constructor `__init__` method that takes a text argument.

```
1 # Please do not run this code cell as it is incomplete and will pr
2 
3 # Let's create a class called TextAnalyzer to analyze text.
4 class TextAnalyzer(object):
5     # The __init__ method initializes the class with a 'text' para
6     # We will store the provided 'text' as an instance variable.
7     def __init__(self, text):
```

```
  Cell In[2], line 7
    def __init__(self, text):
                             ^
SyntaxError: incomplete input
```

Step-3 Implement a code to Format the text in Lowercase:

1. Inside the constructor,we will convert the text argument to lowercase using the `lower()` method.
2. Then, will Remove punctuation marks (periods, exclamation marks, commas, and question marks) from the text using the `replace()` method.
3. At last, we will Assign the formatted text to a new attribute called fmtText.

Here we will be Updating the above `TextAnalyzer` class with points mentioned above.

```
# Press Shift+Enter to run the code.
class TextAnalzer(object):

    def __init__ (self, text):
        # remove punctuation
        formattedText = text.replace('.','').replace('!','').repla

        # make text lowercase
        formattedText = formattedText.lower()

        self.fmtText = formattedText
```

Step-4 Implement a code to count the Frequency of all unique words:

- In this step, we will Implement the `freqAll()` method with the below parameters:
    1. Split the fmtText attribute into individual words using the `split()` method.
    2. Create an empty dictionary to store the word frequency.
    3. Iterate over the list of words and update the frequency dictionary accordingly.
    4. Use `count` method for counting the occurence
    5. Return the frequency dictionary.

Update the above `TextAnalyzer` class with points mentioned above.

```
#Press shift+Enter to run the code
class TextAnalyzer(object):

    def __init__ (self, text):
        # remove punctuation
        formattedText = text.replace('.','').replace('!','').repla

        # make text lowercase
        formattedText = formattedText.lower()

        self.fmtText = formattedText

    def freqAll(self):
        # split text into words
        wordList = self.fmtText.split(' ')

        # Create dictionary
        freqMap = {}
        for word in set(wordList): # use set to remove duplicates
            freqMap[word] = wordList.count(word)

        return freqMap
```

Step-5 Implement a code to count the Frequency of a specific word:

*In step-5, we have to Implement the `freqOf(word)` method that takes a word argument:

1. Create method and pass the word that need to be found
2. Get the `freqAll` method for look for count and check if that word is in the list.
3. Return the count.

Update the above `TextAnalyzer` class with points mentioned above.

```
#Press Shift+Enter to run the code
class TextAnalyzer(object):

    def __init__ (self, text):
        # remove punctuation
        formattedText = text.replace('.','').replace('!','').repla

        # make text lowercase
        formattedText = formattedText.lower()

        self.fmtText = formattedText

    def freqAll(self):
        # split text into words
        wordList = self.fmtText.split(' ')

        # Create dictionary
        freqMap = {}
        for word in set(wordList): # use set to remove duplicates
            freqMap[word] = wordList.count(word)

        return freqMap

    def freqOf(self,word):
        # get frequency map
        freqDict = self.freqAll()

        if word in freqDict:
            return freqDict[word]
        else:
            return 0
```

Now, We have successfully created a class with 3 methods

Part-B

In Part B, we will be calling the functions created in Part A, allowing the functions to execute and generate output.

Step-1 Create an instance of TextAnalyzer Class.

- Instantiate the TextAnalyzer class by passing the given string as an argument.

```
# type your code here
analyzed = TextAnalyzer(givenstring)
```

Step-2 Call the function that converts the data into lowercase

```
# Press Shift+Enter to run the code.
print("Formatted Text:", analyzed.fmtText)
```

```
Formatted Text: lorem ipsum dolor diam amet consetetur lorem magna sed diam nonumy ei
```

We have successfully converted string into lowercase.

Step-3 Call the function that counts the frequency of all unique words from the data.

```
# Press Shift+Enter to run the code.
freqMap = analyzed.freqAll()
print(freqMap)
```

```
{'consetetur': 1, 'labore': 1, 'eirmod': 1, 'lorem': 2, 'magna': 2, 'tempor': 1, 'dia
```

We have successfully calculated the frequency of all unique words in the string.

Step-4 Call the function that counts the frequency of specific word.

Here, we will call the function that counts the frequency of the word "lorem"

Print the output.**

```
# type your code here
word = "lorem"
frequency = analyzed.freqOf(word)
print("The word",word,"appears",frequency,"times.")
```

```
The word lorem appears 2 times.
```

We have successfully calculated the frequency of all specified words.

Authors

Author1
Akansha yadav

Other Contributors

Contributor with Link, Contributor No Link

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2023-05-17	0.3	Akansha yadav	Created lab under maintenance
2020-07-17	0.1	Sam	Create Lab Template
2022-11-19	0.2	Shengkai	Update Lab Template