SAT 4650 – Applied Computing with Python

Lab 2: Text processing and basic healthcare data analysis

Objective

Apply Python string operations, sequences, dictionaries, sets, and comprehensions to analyze a simple healthcare-related text dataset.

Before you start - File reading

In this lab, you will work with a text file. To read a text file and store its contents into a string variable, use the following code pattern:

```
fhandle = open("filename.txt", "r")

text = fhandle.read()

fhandle.close()
```

Make sure the text file you will work with is located in the same folder as your python script or notebook.

Dataset

You are given a file named **electronic_healthcare_record.txt** that contains short healthcare-related notes such as patient instructions, symptoms, and care descriptions.

Problem 1 – Text cleaning and transformation (Guided steps) – 1.75 marks

In this problem, you will clean and normalize healthcare text data step by step. Follow these steps in order:

1. Read the file into a string
2. Use the file-reading pattern shown above
3. Store the content in a variable named `raw_text`
4. Convert the text to lowercase
5. Create a new string variable called `clean_text`
6. Remove punctuation i.e., characters such as `. , ! ? ' " ; : ( ) - _` (*Hint: you may use multiple or chained* `replace()` *calls*)
7. Replace line breaks with spaces (*Hint: Google the difference between* `split()` *and* `split(“ “)` *and note that you might have multiple spaces between two words, not just one*)
8. Replace `\n` with a single space
9. Print the final cleaned version of the text

At the end of this problem, you should have a single lowercase string with no punctuation and no line breaks, suitable for further analysis.

Problem 2 – Healthcare term analysis (Unguided) – 1.75 marks

Healthcare providers often want to understand which symptoms or terms appear most frequently in patient notes. Using the cleaned text from problem 1, write a python program that:

1. Identifies individual words in the text
2. Counts how many times each word appears
3. Removes duplicate words
4. Produces a final collection of unique words
5. Displays/prints:
    a. The total number of unique words
    b. The frequency of each word
    c. A list of unique words with the first letter capitalized

For full grades, you must use at least one dictionary AND at least one set. You may choose whether to use loops OR list/set/dictionary comprehensions. No step-by-step instructions are provided for this problem – you must decide how to structure your solution based on what you learned in class.

Problem 3 – 0.5 marks

In one paragraph, discuss whether the instructions were clear or could have been improved, whether any steps felt unnecessary or confusing, and whether additional examples, hints, or explanations would have been helpful. Describe which parts of the problem contributed most to your learning and explain what you would like to see added, removed, or changed in a future version of this lab to make the problem or its python-based solution more effective.

Submission Requirements

Please submit the following on Canvas:

1. Your python source file and screenshots of your code output for problems 1 and 2.
2. One paragraph solution for problem 3.