Deadlines:

For all Groups: December 13 2022

Grading system:

Add the accumulated score to obtain your grade:

- 1 - 1
- 2 - 1
- 3 - 1
- 4.1 - 0.5
- 4.2 – 0.5
- 4.3 – 0.5
- 4.4 – 1.5
- 4.5 – 0.5
- 4.6 – 1.5
- 4.7 - 2

1. Important Distributions

The number of deer in the "Codrii" reserve is unknown. To estimate the number of deer, 50 deer are captured and tagged. Six months later, 200 deer were captured, and it is found that 8 of these were tagged. Estimate the number of deer in the Codrii Reserve using this data. Write a computer program to confirm your guess.

2. Continuous Conditional probability

Suppose you toss a dart at a circular target of radius 10 inches. Given that the dart lands in the upper half of the target, find the probability that

- it lands in the right half of the target
- its distance from the center is less than 5 inches
- its distance from the center is greater than 5 inches
- it lands within 5 inches of the point (0, 5)

3. Counting

100 people line up to take their seats in a 100 seat theater. The 1st in line lost her ticket and decided to sit in a random seat. Each remaining theatergoer sits in their

assigned seat unless it’s occupied and then sits in a random seat. What’s the probability the last person takes their assigned seat?

4. Networking

In this task you have the opportunity to solve problems closer to the real file. This implies that you have to combine your probability skills with the **natural language processing** field, aka NLP. The dataset you are going to use is a list of tweets, found here tweets.json

Your task is divided into two topics.

In the first one you have to deal with **word frequency** and finding out the most popular and interesting words.

The second topic is about **typing prediction**. A good use case for typing prediction is a smart phone. While writing messages to your beloved friends is a cumbersome work to do (many taps on your display), the phone operating system usually comes with a typing prediction software. Now while you type your text a list of suggestions pops up at every keystroke. Your task would be to write programs that do something similar, naturally in a simplified version (no need to write mobile applications).

The NLP part

One of the main issues of text processing is breaking a text into words. At the first glance it seems to be an easy job. For instance, you have the sentence Today is a sunny day. Easy enough you just use the python built in function sentence.split() (where sentence is the string). Now you have a list of words. But as you know words in a text come hand in hand with characters like **. , ! ? "** and a bunch of others. So extracting words from a sentence like Now! Tell me what is your "problem"? is suddenly more difficult.
NLP libraries hold extensive tools for text processing. In this lab you'll be only scratching the surface of NLP. The word extraction problem mentioned above can be solved with a simple function call. For python you can use the import nltk, but before that don't forget to pip install nltk.

The dataset

The data that you have in this laboratory work is a **JSON** string. It is a popular data format. One of the advantages is that it is easy to read. Also it's very easy to transpose the data into python primitives like *lists*, *dictionaries*, *strings* and *integers*. For that in your script you import json and now can use it like this result = json.loads(input_string_you_read_from_file). In the result is stored either a python **list** or a **dictionary**. That's easy.

Frequency

4.1 Popular

Write a program that prints the first 10 most frequently used **words**, and the number of times it was mentioned.

Ex:

```
the 352

a 235

at 120

. . .
```

4.2 Nouns

Write a program that prints the first 10 most frequently used **nouns**, and the number of times it was mentioned.

4.3 Proper nouns

Write a program that prints the first 10 most frequently used **proper nouns**, and the number of times it was mentioned.

4.4 Frequency

Write a program that receives a word as an input and draws a frequency bar chart. Every bar should represent the period of 1 month.

4.5 Popularity

In our dataset we also have the number of likes and retweets for every message. This can give us some insight about the tweet's popularity. Hence we can compute some sort of rating. The popularity of nouns is computed by the following formula $\text{frequency} * (1.4 + \text{normRetweet}) * (1.2 + \text{normLikes})$. The values **normRetweet** and **normLikes** are the normalized values of retweets and likes for every word. To compute the number of likes and retweets for every word you just cumulatively collect the numbers from every tweet that the word was mentioned.

Ex: There are 2 tweets that mention the noun **program**. The first tweet has **32** retweets and **87** likes. The second tweet has **42** retweets and **103** likes. The number of retweets of the word **program** is **32 + 42** and the number of likes is **87 + 103**.

Write a program that prints the first 10 most popular **nouns**. The popularity is defined by the computed rating discussed above.

Typing prediction

4.6 Suggestion

Write a program that receives as input an **uncompleted word** and prints 3 word suggestions, followed by their frequency. The suggestions should be based on the initial dataset and sorted by the word frequency, computed in the first problem.

The input can be any uncompleted word.

Ex. Input: app, Output: application (324), apple (164), appreciate (53). Where application has the highest frequency, apple the second highest etc.
Ex. Input: pro, Output: programming (196), product (176), program (103). Again programming has the highest frequency.

4.7 Suggestion occurrences

Write a program that receives as input a **word** and prints 3 word suggestions, followed by the **suggestion occurrences**.

The suggestions should be selected in the following way. You have to go through your tweets dataset and identify every occurrence of the input word. At every occurrence collect the word that follows the input word. That is the suggestion you are looking for. And also don't forget to count the number of times you get the same suggestions. Ex: input like and you find 5 occurrences of beer and 2 occurrences of love labs. Your suggestion words would be beer and labs. But beer has a priority because it occurred more times in your dataset. Your task is to select the most relevant suggestions as in the one that occurred the most.
The input can be any completed word.

Ex. Input: love, Output: programming (5), cars (2), beer (2)
Ex. Input: awesome, Output: party (10), language (4), framework (2)