Assignment: Affection status visualizer

Learning goals

In this assignment, you:

1. learn to analyse natural text.
2. improve your data manipulation skills in Python.

Preparations

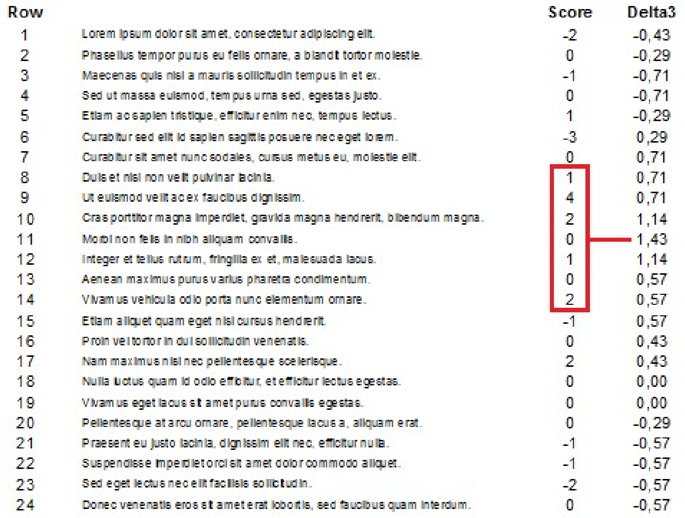
For this assignment, you need to install NLTK package and download the related modules: https://www.nltk.org/. Hint: as you use the library, use **nltk.download('popular')** statement if **nltk.download()** fails.

Assignment

Your task is to write a Python sentiment analysis tool that visualizes the changes in the affective state of a single text (e.g. a novel).

This assignment is of exploratory nature and, ideally, you end in a proof-of-principle demonstration of the method.

The following example illustrates the idea:

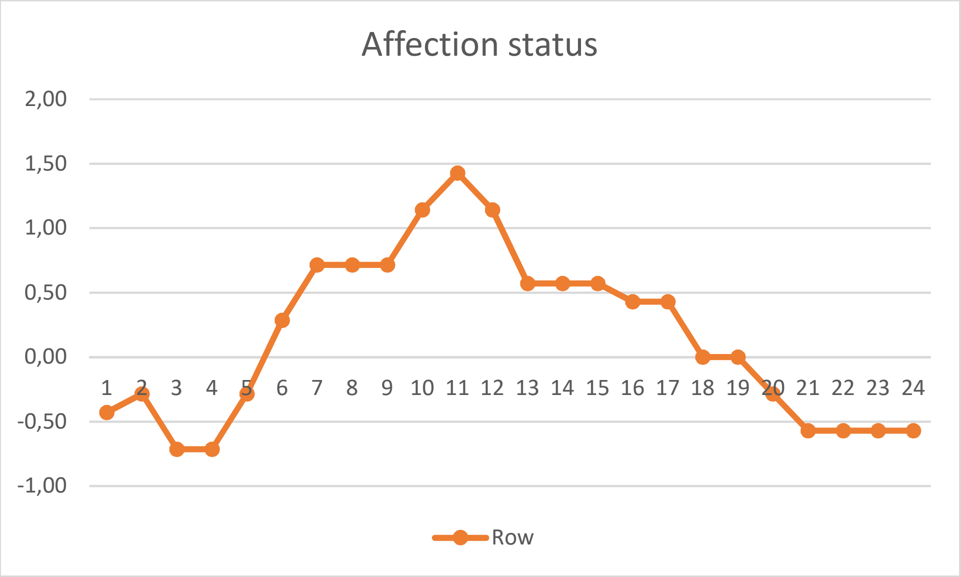


Row		Score	Delta3
1	Lorem ipsum dolor sit amet, consectetur adipiscing elit.	-2	-0,43
2	Phasellus tempor purus eu felis ornare, a blandit tortor molestie.	0	-0,29
3	Maecenas quis nisl a mauris sollicitudin tempus in et ex.	-1	-0,71
4	Sed ut massa euismod, tempus urna sed, egestas justo.	0	-0,71
5	Etiam ac sapien tristique, efficitur enim nec, tempus lectus.	1	-0,29
6	Curabitur sed elit id sapien sagittis posuere nec eget lorem.	-3	0,29
7	Curabitur sit amet nunc sodales, cursus metus eu, molestie elit.	0	0,71
8	Duis et nisi non velit pulvinar lacinia.	1	0,71
9	Ut euismod velit ac ex faucibus dignissim.	4	0,71
10	Cras porttitor magna imperdiet, gravida magna hendrerit, bibendum magna.	2	1,14
11	Morbi non felis in nibh aliquam convallis.	0	1,43
12	Integer et tellus rutrum, fringilla ex et, malesuada lacus.	1	1,14
13	Aenean maximus purus varius pharetra condimentum.	0	0,57
14	Vivamus vehicula odio porta nunc elementum ornare.	2	0,57
15	Etiam aliquet quam eget nisi cursus hendrerit.	-1	0,57
16	Proin vel tortor in dui sollicitudin venenatis.	0	0,43
17	Nam maximus nisl nec pellentesque scelerisque.	2	0,43
18	Nulla luctus quam id odio efficitur, et efficitur lectus egestas.	0	0,00
19	Vivamus eget lacus sit amet purus convallis egestas.	0	0,00
20	Pellentesque at arcu ornare, pellentesque lacus a, aliquam erat.	0	-0,29
21	Praesent eu justo lacinia, dignissim elit nec, efficitur nulla.	-1	-0,57
22	Suspendisse imperdiet orci sit amet dolor commodo aliquet.	-1	-0,57
23	Sed eget lectus nec elit facilisis sollicitudin.	-2	-0,57
24	Donec venenatis eros sit amet erat lobortis, sed faucibus quam interdum.	0	-0,57

For each unit of a text (e.g. a row), the positive and negative words can be detected, and an affection status score (e.g. the number of positive words minus the number of negative words) calculated.

Next, a sliding average of the scores within a set radius can be computed for each line. In the example above, a radius of 3 lines is used. As an example, the sliding average (Delta3) for line 11 equals to 1,43. It is simply the arithmetic mean of the seven values in the window.

A visualization of the analysis shows how the affection status of the text dynamically changes:



Feel free to experiment and try to make the performance as good as possible (on the intuitive basis, as the true accuracy is hard to verify). Play with various measures of the affection status, various units of text, and various visualization techniques (either graphical or text-based).

Apply your tool for at least one real text (preferably a novel) and display the results. Project Gutenberg is a great source of online texts to experiment with.

Deliverables

Your deliverable should include both the Python codes and your evaluation on the applicability of your tool, supported by the relevant results.

Submit your work preferably in HTML format. The deliverable should contain the information specified in the points 1 to 3 above.