ASSIGNMENT Mining Complex Data

Deadline: Friday, the 10th of June, 2016

Recommendation systems have spread like wildfire, in particular with online shopping websites such as Amazon. Nowadays, they are on the way to be deployed in many other contexts, such as social networks or culture. This assignment is the following of the exam: you're asked to develop a new system for automatically recommending reading in an online bookstore.

The main objective is to implement a (very small) proof of concept for a "emotion-based" recommander system. Below are some suggestions in order to address this issue:

1. Select a limited set of (small) documents from the Internet, the length of each being no more than a couple of sentences. To this end, you can quickly browse the free Gutenberg online repository: `https://www.gutenberg.org`. Books such as "Dracula" by B. Stoker or "The Call of the Wild" by J. London look really promising for finding emotion-oriented texts (small hint: use ctrl+F to quickly spot emotion-oriented words). You can go straight to the "Plain text UTF-8" file in order to find the plain text you can copy and paste.

2. Now that you have extracted your dataset, you can go through the usual preprocessing steps provided by the TM package. Follow what you have proposed during the exam and make the most relevant choices for this specific task.

3. Compute the emotions conveyed by your texts. To this end, you can calculate a score based on a set of words clearly related to emotions[1]. You can find some useful lists in the following website: `http://karlamclaren.com/emotional-vocabulary-page/`. The easiest way consists in considering each list as a pseudo-document and computing the cosine with the texts of your dataset.

4. Compute the recommandation based on the "emotional" profile of the current user. You can assume that this profile is already known.

5. Finally, add a new feature of your choice. It may be, for instance: a) automatic evaluating of user profile, b) expansion of the emotion-oriented words by using a knowledge base, c) proposing a different way for calculating the recommandation, or any other idea that can improve your system.

Please, don't forget to extensively **comment** your code so that I can easily understand your reasoning, and make sure that I will be able to test the code on my own computer.

Send the R file(s) and all the material I need to load your project to the following email address:

`julien.velcin@univ-lyon2.fr`.

[1] Note that you might address this issue with supervised classification, but the solution proposed here is more simple.