1. First approach

1.1. What is already existing in term of applications ?

There are lots of books recommendations applications which also offer the ability to create its own personal library.

BUT, if we want :

- a french site,
- with lots of international referenced books (paper and electronic),
- which allows to give my opinion,
- which allows to export my personal created library
- and which gives me personal recommendations…

…there only one site left: Babelio.

However, its recommendation system seems to be not appreciated by users, because for example, it seems attaching too much importance on the books read (prior to the rating given !). But others examples of potential incorrect algorithm entries are given…

1.2. What is already existing in term of database ?

Some sites like TiteLive own a great books data base…but they sell their database !

The BNF DataBase is free, but only deals with french books…

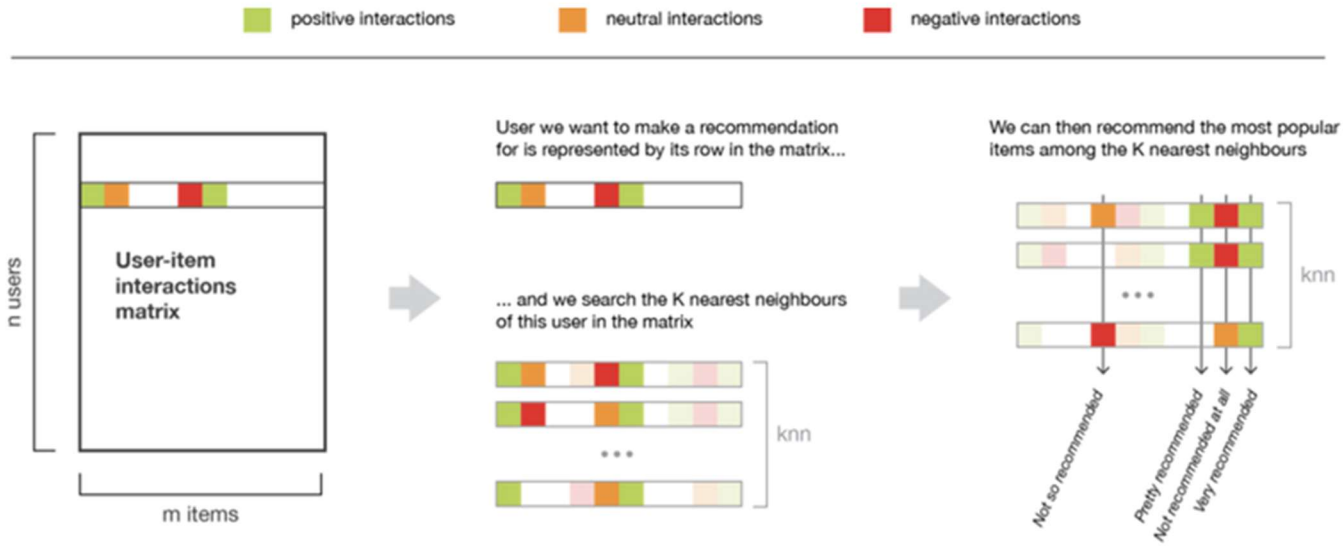
So no good idea except web scrapping on web site such as Google Books…

1.3. How works a recommendations algorithm ?

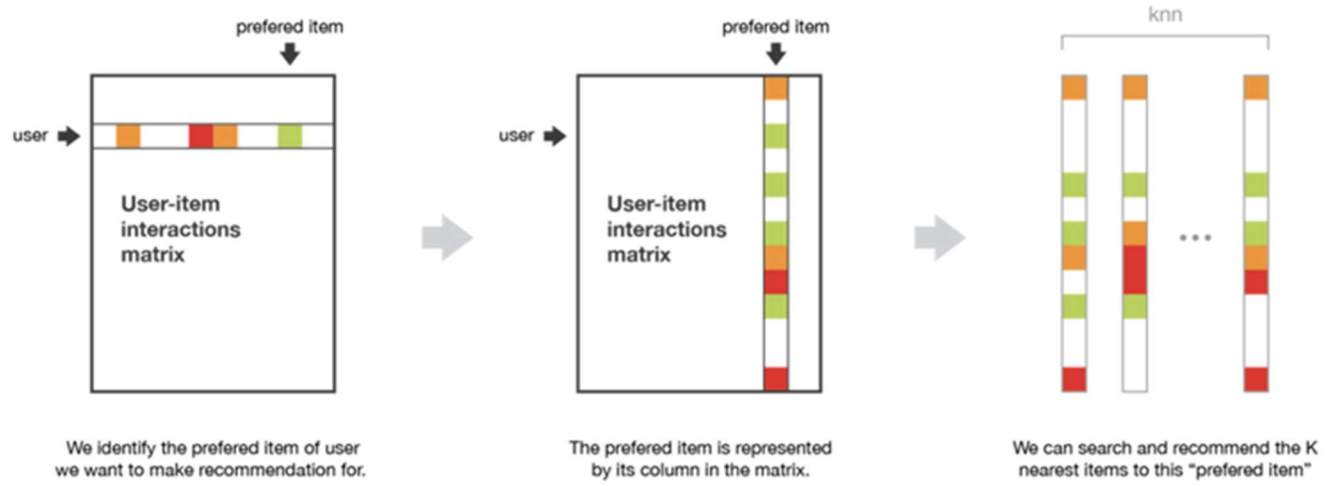
Similarity calculations seems to be the heart of this subject. Algorithms which use those similarities seems to be :

- ***Collaborative filtering*** (user-user, item-item, or with associative rules) which are based on interactions recorded between users and items : ratings given to read books.

New recommendations are based on similarity between users



Or new recommendations are based on similarity between items (books in our case) :



- ***Content analysis*** uses additional information about users and/or items. This times books characteristics can be used (gender, author, page number…), and also user's characteristics (its reading tastes…).

2. Our dreamed recommendation algorithm

We would like a customizable recommendation system which :

- Could take into acount a library or a literary critic whose books opinions are close to mine.
- Could take into acount only specified other readers for which we estimate that their opinion is close to our.
- Could return in each cases, two ratings: one with considering only my personal ratings, and the other considering also "similar" readers ratings…

We also would like our application has the ability to propose an estimated rating when I scan a ISBN book (in a library for example). This estimation should be from my personal ratings only in priority.

For user, we would like to ask the minimum of entries :

- book title and author
- associated rating
- URLs of library, a literary critic, other readers in which the reader trusts

3. How

3.1. In term of database

On books dedicated web site (Amazon, Good Reads…), no real solutions because often they no more provide their API. Web scrapping will then be needed (already begun on Google Bookg and Good Reads).

But data set can be found in machine learning dedicated web site, as for example Book Crossing, Kaggle (results of Good Reads web site scrapping for example…) or some GitHub.

The difficulty is to found data base with users ratings provided.

3.2. In term of algorithm

No real idea for the moment, but only the following first conclusions (https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada) :

Content based methods can also be neither user nor item centred: both informations about user and item can be used for our models, for example by stacking the two features vectors and making them go through a neural network architecture.

Content based methods

- ***highest bias (highest personalization)***
- lowest variance (lowest robustness)

Either we use a SVD decomposition, in order to learn some specific user's coefficients:

⇨ ***Mix of collaborative model***, in which ALL ratings are considered (not only the one of the users):

$$(X,Y) = \underset{X,Y}{argmin} \frac{1}{2} \sum_{(i,j)\in E} [(X_i)(Y_j)^T - M_{ij}]^2 + \frac{\lambda}{2}(\sum_{i,k}(X_{ik})^2 + \sum_{j,k}(Y_{jk})^2)$$

M : interaction users-items matrix (known)
X : matrix with one line per user, with its own ratings (to be learned)
Y : matrix with one line per items, with its own characteristics (to be learned)

The algorithm will try to find X and Y such that : $M \cong X \cdot Y^T$

Then a new recommendation is performed by multiplying a specific user characteristics X_i by a specific book characteristics Y_j.

⇨ ***With content model***, in which instead of guessing Y_j, we put content values (both books features AND user features)

$$X_i = \underset{X_i}{argmin} \frac{1}{2} \sum_{(i,j) \in E} [(X_i)(Y_j)^T - M_{ij}]^2 + \frac{\lambda}{2} (\sum_k (X_{ik})^2)$$

Or we can use machine learning algorithm to directly learn ratings, like ***Random Forest.***

Or we can use ***deep learning*** algorithm.

<u>Some other references :</u>

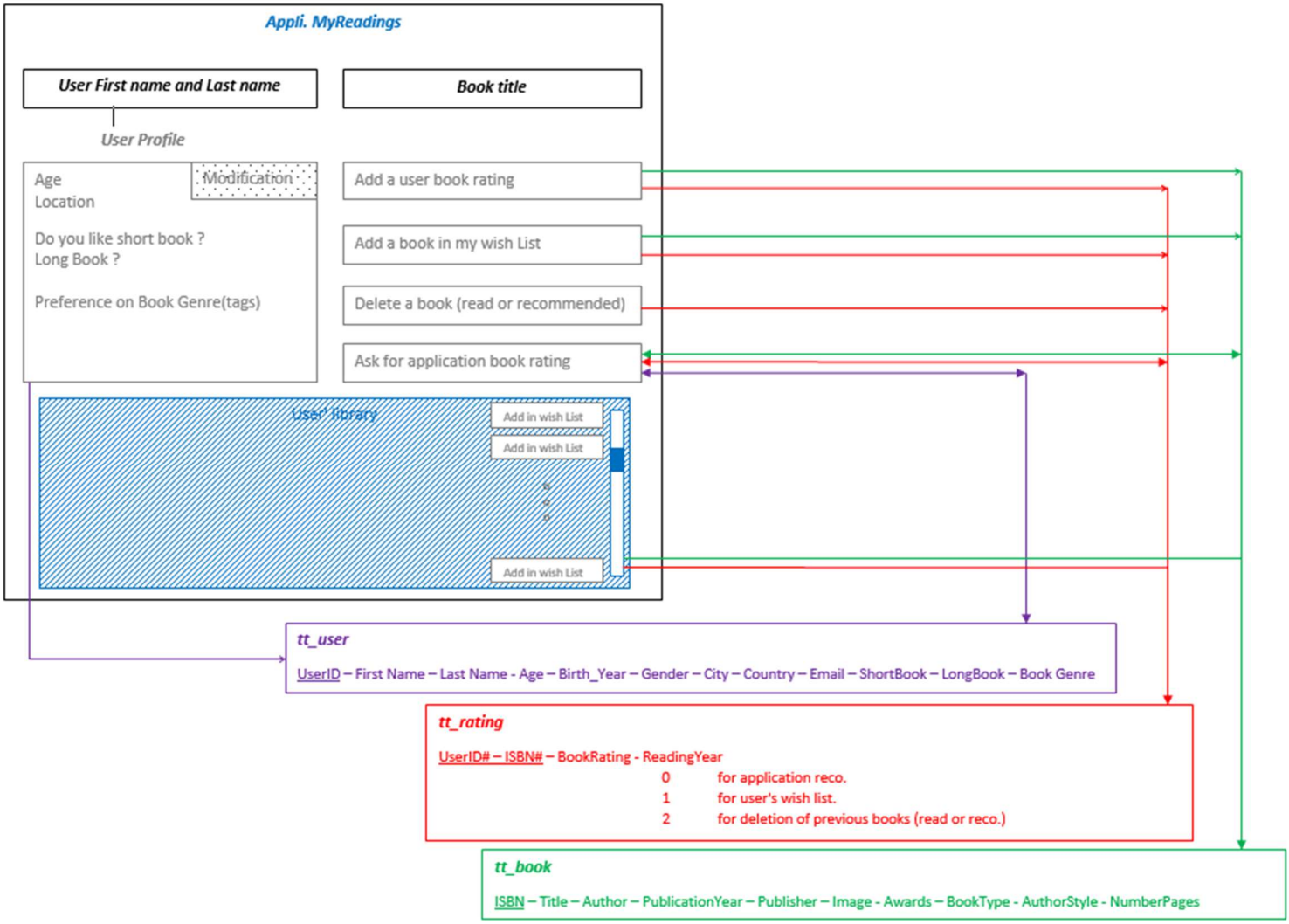
- Recommender System with Machine Learning ans Artificial Intelligence" – Jyotir Moy Chatterjee; Sachi Nandan Mohanty; Sarika Jain; Ahmed A. Elngar; Priya Gupta

- Information and Recommender Systems" – Elsa Negre

4. Technical challenge

Application :

The project time limitation, our dreamed recommendation algorithm proposes too much functionalities. The following has been retained as the minimum to reach:



Data :

First of all : how to represent our data ? How to represent the books and the Users characteristics ? May be Word To Vec transformation (https://www.analyticsvidhya.com/blog/2019/07/how-to-build-recommendation-system-word2vec-python/).

Then what kind of data we have to use to be relevant ? But in this case, what we will be able to web scrap !

Algorithms :

Lots of recommendation algorithms are existing, which one to use ? Lots of similarity measures exists, which one to use ? How to succeed the personalisation of recommendations ?