**EX 3 – ARTIFICIAL INTELLIGENCE**

**Part 1 : Non-Personalized**

1. **The data.**

For non-personalized recommender system, we will calculate the weighted average rating for each book.

The data we need is: books and ratings. In the file books.csv: book\_id and title. And in the file ratings.csv: book\_id and rating.

(The Non-personalized algorithm recommends according to the rating. All users have the same recommendations.)

For the following questions in the exercise, we’ll also need more information on the users to target the recommendations (like place and age).

1. **Get simply recommendation.**

To get the k best recommendations,

- We first, calculated the number of voters for each book: v

- We calculated the average rating for each book: R

- We fixed the minimum number of voters, like in the Tirgul: (quantile(90)) : m

- We got the average rating of all books: C

- Finally, we calculate the weighted average rating:

**WR =**

-The function returns the k books with the highest value of WR.

The 10 recommended books are:



1. **Get simply place recommendations.**

We’d like to do the same, but we want to target the recommendations according to the living location of the user.

The 10 recommended books for Ohio are:



1. **Get simply age recommendation**

We want to target the recommendations according to the age of the user.

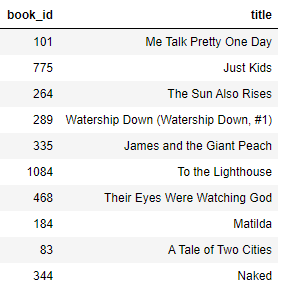
The 10 recommended books for a 28-year-old (21 to 30) user are:



**Part 2: Collaborative filtering**

Everything is in the code.

As an example, here’s the recommendations we got for user 1:



**Part 3: Contact based filtering**

1. **Features.**

The features we used to work with are: **language, tags, original title and authors**.

We tried different features to choose the best ones. It seems logical that if we want a recommendation for Twilight or Harry Potter, or The Hunger Games, it will recommend us the other books of the Saga. That’s why the title’s book is important.

Moreover, usually, every author has its own writing style. We can see it through their different books. And if I liked a book of an author, I would like to get recommendations for its other books.

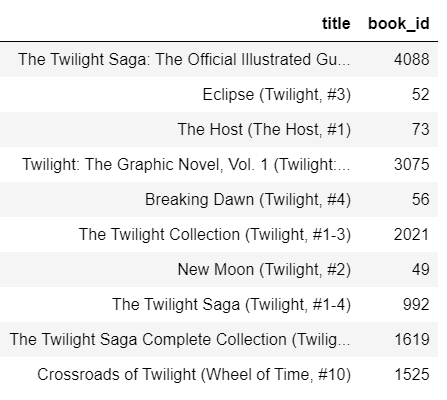
Language has also its logical impact. The way it’s been written, the language the lector speaks and language can also mean culture. French books are different from American books.

Tags give us some hints about the books: like the genre. It’s also important for the recommendation.

We also tried with the publication year but it adds noises. It limits us. The year is taken as a string, so it’s either equal or different. That is not what we want.

1. **Recommendations for Twilight.**

We got the following recommendations for Twilight (different Twilight existing so for the first one):



**Part 4: Evaluations**

1. **Table of evaluations.**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Precision\_k | ARHR | RMSE |
| Cosine | 0.08 | 0.6466666666666666 | 0.9176794695882072 |
| Euclidian | 0.008 | 0.08 | 0.9168899103744533 |
| Jaccard | 0.08 | 0.6266666666666666 | 0.9187216947055077 |

1. **Explanation of those results.**

We get a weak precision\_k for every similarity. The reason to this is that the test file is very small so it can’t give us a good precision. We don’t have enough information.

We could have got better results for ARHR, but we didn’t for the same reasons. If we had more samples in our test file, the ARHR would be better. We still get better results than precision\_k because we take into account the position of the books.

Moreover, precision\_k and ARHR, use only the top 10 recommendations that have been given.

RMSE takes into account the predicted results and compare it to the actual one. Only the difference between the rankings matters.

Those results show us that our rankings are good and that we succeeded to find the right rank of the recommendation (ARHR high).