Basics of Data Analytics : Final homework Movie recommendation

In this final homework, you will progressively build different movie recommender systems based on ratings that users have given to different movies and on movie informations. These recommender systems will be based on :

1. average ratings of movies
2. weighted average ratings of movies
3. average ratings of movies conditioned by information about the users
4. movie similarity (using a nearest-neighbour approach on ratings)
5. user similarity (using a nearest-neighbour approach on ratings)
6. movie similarity (based on movie informations)

1 Part 1 : rating-based recommender systems

In this part, you will work with three datasets (coming from 3 csv files) :

1. a movie dataset (file `movies.csv`) : gives informations about the different movies that will be considered (id, title, genre(s))
2. a user dataset (file `users.csv`) : gives informations about the users (id, gender, age, occupation, zip code). See README file for details
3. a rating dataset (file `ratings_train.csv`) : gives the ratings given by users to movies (user id, movie id, rating, date)

You will also use another file to make recommendations to some users and evaluate these recommendations (file `ratings_test.csv`). In this file, you are given pairs ('user id', 'movie id'). For each of these pairs, you will have to predict the rating that would give the user 'user id' to the movie 'movie id'. We don't know this rating (because basically user 'user id' has not seen movie 'movie id' yet), but the idea of a recommender system is to predict this rating. If the predicted rating is high, the movie could be recommended to the user. You will use different techniques to predict all the ratings of the test file (as enumerated above). I will give you a way to evaluate the accuracy of your predictions.

1.1 Preliminar analysis of the datasets

Before designing recommender systems, you will first perform some analysis on the given datasets. Answer the following points. You can use some plots (even if not asked) to enrich your answers

1. You should see in the movie dataset that the release year of each movie is given at the end of the Title (inside brackets). Modify this dataset so that the release date no more appears in the Title column but instead in a new separate column 'Year'
2. What is the movie released in 1995 that was best rated by users ? How many ratings has this movie ?
3. Same question but only consider movies with at least 10 ratings from users
4. Provide some information about the average number of ratings of movies depending on their release date (using 4 different periods for the release date : before 1940, between 1940 and 1959, between 1960 and 1979, and after 1980)

5. What is the average rating in the dataset ?
6. What is the average rating given by user 148 ?
7. What is the distribution of ratings for the movie 'Toy Story' ?
8. Provide some information (numerical or visual) about the average number of ratings of movies with respect to the occupation of users and their gender.
9. Do you think that the ratings given by users to movies are impacted by the age of users ?
10. Find some movies that are more impacted by the gender of users. In other words, for which movies the difference between the ratings given by Male users and given by Female users is the most important ?

1.2 First recommender system : based on average rating

In this part, you are going to design a first (very basic) recommender system. The idea is the following :
the prediction of the rating given by user u to movie m is just the average rating of movie m amongst all users that have rated m.

This technique is very simple, the predicted rating only depends on the movie m, not on the user u. This means that the same rating will be predicted for movie m for any users (that has not seen m yet). This technique can only be accurate if no diversity exists in the users (in other words, if all users share the same opinion about all movies). It is of course not the case in reality. But you will use this basic technique as a first baseline.

Question : Implement this recommender system, and predict the rating of all pairs ('user id', 'movie id') of the test dataset.

You can evaluate the quality of your predictions as explained below. For all pairs ('user id', 'movie id') of the test dataset, I know the true rating that 'user id' gave to 'movie id' (but you don't !). But you can use the function `evaluation_function` that I gave you to evaluate your predictions. This function computes the mean squared error between the truth and your predictions (`https://en.wikipedia.org/wiki/Mean_squared_error`). For example, if you predict ratings 3 and 4 and the truth was 4 and 2, the mean squared error of these 2 predictions is :

$$\frac{1}{2}((3-4)^2 + (2-4)^2) = \frac{5}{2}.$$

The lower the error, the more accurate are the predictions.

To use the function `evaluation_function`, follow these steps :
— put the file `eval_student_38.pyc` (or `_37` if you use Python 3.7) into your working directory
— type `from eval_student_38 import evaluation_function` in your notebook or Spyder terminal (or `_37` if you use Python 3.7)
— store your predictions inside a Series (named `pred` for instance)
— type `evaluation_function(pred)` and you'll get the prediction error

Note that the two first steps need to be done only once (for the evaluation of your predictions in other parts of this TP, you don't need to do the two first steps again)

Question : what is the prediction error made by this first reccomender system ? Remember this value, you'll have to compare it with errors of other srategies (in the next sections).

1.3 Reccomendation based on weighted average rating of movies

In the previous section, you have created recommendations based on the average rating of movies. We have pointed out that this kind of recomendation might not be very acccurate as users don't share same opinions about movies. There is also another issue with this, that we will highlight here.

Question : Which movie has the highest average rating ? How many users has rated this movie ?

If we only compute the average rating of a movie, the rating does not take into account the number of users that rated the movie. So a movie with just one rating of 5 will have the best average rating, better than for instance a movie with 2000 users that rated it 5 and 10 that rated it 4. But for sure, we trust more an average rating obtained with more than 2000 ratings than one obtained with only a few ratings. You should already have faced this situation when looking at restaurants or hotels in TripAdvisor. You trust more the general opinion of a restaurant or an hotel if it has been rated by many users.

To take this into account, a weighted rating can be used. Here, we will use a weighted average rating proposed by IMDb (the famous movie database). This weighted average rating $WR(m)$ of movie m is computed as follows :

$$WR(m) = \frac{v}{v+n} \times R + \frac{n}{v+n} \times C, \text{where :}$$

— v represents the number of ratings for movie m
— R represents the average rating of movie m (as in the previous subsection)
— n represents the 75-percentile value of number of ratings for the different movies of the dataset (i.e. a number such that only 25% of movies have more ratings than this number)
— C is the average rating of all movies of the dataset

Questions :
— write a function that computes the weighted rating of a movie given its id `movie_id` and a rating matrix (as the one you have, with all the ratings given by users to the different movies)
— what is the movie with the highest weighted average rating ? You should see a difference with the average rating.
— use this weighted rating to predict the ratings of all pairs in the test dataset. (exactly the same as before, but the average rating of movies is replaced by the weighted average).
— compute the prediction error for these new predictions. Have you improved your predictions ?

1.4 Recommendation based on average rating of movies and the gender of users

We will try to improve our recommender system here with a new strategy. You should have seen in the first section that the gender of a user has an influence on the rating of movie. We will use here this information to improve our system as follows : **to predict the rating given by user u to movie m, we have to compute the average rating of movie m but only ratings given by users of the same gender as u. In other words, if u is a male, we will use the average rating given by male users, and vice-versa**.

Questions :
— predict the ratings of all pairs in the test dataset using this strategy
— compute the prediction error for these new predictions. Have you improved your predictions ?

1.5 Recommendation based on movie similarity

In this section, we will use a movie similarity measure, together with a nearest-neighbour approach to make reccomendations. The principle of this approach to predict the rating that user u would give to movie m ($rating(m, u)$) is the following :

1. Choose a number of neighbours k to consider for the nearest-neighbour approach (start with $k = 1$ and then change)
2. Compute the similarity between m and all the movies rated by u
3. Select the set of k movies (rated by u) that are the most similar to m : $\{n_1, \ldots, n_k\}$
4. Predict $rating(m, u)$ as : $\frac{1}{k} \sum_{i=1}^{k} rating(n_i, u)$

In other words, the predicted rating given by user u to movie m is the average rating that gave user u to the k movies that are most similar to m (amongst the movies that u has rated).

Movie sililarity : the most important point in this approach is how to compute the similarity between two movies. I explain this below.
The similarity between 2 movies $m1$ and $m2$ is based on the correlation between the ratings given to these 2 movies by the users. First, we need to find the set of users that rated **both** movies $m1$ and $m2$. Then create a matrix that contains two colums (one for the ratings of $m1$ given by these users and another for the ratings of $m2$ given by these users). This matrix should be indexed by the users id (i.e. a row should contain the ratings of a **same user** to $m1$ and $m2$). Then the similarity between $m1$ and $m2$ is the correlation coefficient beween the 2 columns of the created matrix. Note that we can decide to fix a minimum number of users that rated both movies to consider the similarity. If only a few users rated both, we can decide to fix the similarity to a very small value (or a nan).

Below you'll find a small example to illustrate this. Let's consider that we have only 4 users and 7 movies in our dataset and that we have the following ratings given by users :

	m1	m2	m3	m4	m5	m6	m7
User 1	9	3	7	5	3		
User 2	8	3		5	7		8
User 3	4	6	2	7		8	3
User 4	3	7	3	8	4		2

The similarity between $m1$ and $m2$ is the correlation coefficient between $[9, 8, 4, 3]$ and $[3, 3, 6, 7]$ (as all users have rated both movies). It is equal to -0.989 (obtained with Python). This means that these 2 movies are not similar at all (strong negative correlation). The higher the correlation, the most similar are movies.
The similarity between $m5$ and $m7$ is the correlation coefficient between $[7, 4]$ and $[8, 2]$ (only 2 users have rated both movies). In this case, I can decide to put the similarity to -1 if we decide that 2 users is not enough to evaluate similarity between 2 movies.

Questions :

1. Write a function that computes the similarity between two movies of your dataset with the following rule : if less than 40 users rated both movies, the similarity should be -1, otherwise it is the correlation coefficient as explained above
2. Write a function that predicts the rating given by a user u to a movie m with the nearest-neighbor approach explained above (with parameter k)
3. Predict the ratings of all pairs (user id, movie id) of the test dataset and evaluate your predictions for different values of k.
4. Can you improve the quality of predictions for some values of k ?

Bonus : you can try this after the rest if you have time
In the method that we have just applied, we have only considered positive correlations when searching for the most similar movies. We can also consider that if two movies are strongly negatively correlated, this can bring us an information. For example, let us imagine that the three movies that are most correlated to a movie m have these correlation coefficients : $1, 0.9, -0.8$. User u has rated these 3 movies $4, 4, 2$. As m is negatively correlated to the third movie, we can expect that u will have different opinions about m and this third movie. As he rated 2 to this third movie, we can expect that he will give a good rating to m (4 for instance, we can consider to reverse the ratings symmetrically to 3, as the ratings go from 1 to 5). So the predicted rating here will be the average between 4, 4 and 4.
Question : Implement this extension and compare the results of this new method with the previous one.

1.6 Recommendation based on users similarity

In this section, we will apply a similar technique than just before, but with a similarity between users. The principle of this approach to predict the rating that user u would give to movie m ($rating(m, u)$) is the following :

1. Choose a number of neighbours k to consider for the nearest-neighbour approach (start with $k = 1$ and then change)
2. Compute the similarity between u and all the users that have rated movie m
3. Select the set ok k users (that have rated m) that are the most similar to u : $\{n_1, \ldots, n_k\}$
4. Predict $rating(m, u)$ as : $\frac{1}{k}\sum_{i=1}^{k} rating(m, n_i)$

In other words, the predicted rating given by user u to movie m is the average rating that gave to the movie m the k users that are most similar to u (amongst the users that rated m).

Users similarity As before, the most important point here is to compute a similarity between users. Here we will use a similarity measure based on the euclidean distance (so it is a dissimilarity measure actually) To compute the euclidean distance between 2 users u and v,

— find the set of movies that both u and v have rated
— if this set contains less than 20 movies, then set this distance to a very high number (10000 for instance)
— else just compute the average euclidean distance between the ratings that gave u and v to this set of movies (that they both rated)

For instance, the euclidean distance between `User 1` and `User2` on the small exemple above is :

$$\frac{1}{4}((9-8)^2 + (3-3)^2 + (5-5)^2 + (3-7)^2) = \frac{17}{4}$$

Questions :

1. Write a function that computes the similarity between two users of your dataset
2. Write a function that predicts the rating given by a user u to a movie m with the nearest-neighbor approach on the users (with parameter k)
3. Predict the ratings of all pairs (user id, movie id) of the test dataset and evaluate your predictions for different values of k.
4. Can you improve the quality of predictions for some values of k ?

2 Recommendation based on movie similarity from the synopsis of movies

See file `TP_Section2.py` for instructions.