Netflix Challenge: Movie rating prediction

Description

The goal of this project is to **develop a recommendation algorithm for movies**.

The data for this project comprises **910,190 ratings** (on a **1 to 5 scale**) that were given by **6,040 users** to **3,706 movies**. Next to the ratings, the data contains some information on the users (**gender**, **age**, and **profession**) and information on the movies (**title and year of release**). You have to develop an algorithm that, based on this data, predicts as good as possible what rating a particular user will give to a particular movie.

Instructions

Use your @tudelft.nl email address to make an account on kaggle.com. Once your account has been created, you can go to the Kaggle page and start downloading the data. Further information are provided in the Kaggle page and below.

Be aware that the Kaggle ID that you choose will later be used in your report.

You can develop your algorithms in either **Java or Python**, but we only offer TA support in Python. For your convenience, we will provide Python-code (in Python 3.6) that reads in all data, runs a very simple prediction algorithm, and writes the results into a submission file.

Files and data

Users.csv: Information about the users

This file contains four columns: (1) a column indicating the index of the user; (2) a column indicating whether the user is male or female; (3) a column indicating the age of the users (if known); and (4) a column containing a number that indicates the profession of the user.

Movies.csv: Information about the movies

This file contains three columns: (1) a column indicating the index of the movie; (2) a column containing the year the movie was released; and (3) a column containing the title of the movie.

Ratings.csv: User-movie ratings

This file contains three columns: (1) a column indicating the index of the user who gave the rating; (2) a column indicating the index of the movie that was rated; and (3) a column containing the rating that the user gave to the movie on a scale from 1 to 5.

Predictions.csv: List of user-movie ratings that needs to be predicted
The file contains two columns: (1) a column indicating the index of the user who gave the rating and (2) a column indicating the index of the movie that was rated. For these user-movie combinations, your algorithm needs to predict the ratings. These predicted ratings need to be written into the submission.csv file.

Submission.csv: Example of a submission file

The file that you need to write contains two columns: (1) a column indicating the corresponding row in predictions.csv ranging from 1 to 90,019 and (2) a column indicating the predicted ratings. Note that this file uses a comma as column separator (all other files use a semi-colon instead); this is the Kaggle default.

Algorithm performance

To be able to gauge how well your algorithm works, we have held out a total of 90,019 user-movie ratings. You are supposed to **make predictions for these ratings**; your predictions will be compared with the true user-movie ratings. To this end, we will measure the **root mean squared error** (RMSE) of your algorithm. Given that P are your predicted values and O are your observed values the RMSE of the recommendation algorithm is computed as:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (P_i - O_i)^2}{n}}$$

While developing your algorithms, you can measure their RMSE by writing your predictions to a CSV-file and uploading the resulting CSV-file to the Kaggle-in Class website. The website computes the RMSE of your algorithm on a random subset of the 90,019 held-out ratings, and puts your result on the public leaderboard. This allows you to compare your progress with that of other students. **No more than 15 submissions can be done per day to avoid overfitting.**

The final score of your algorithm will be computed after the end of the competition based on the predicted ratings that were not used to compute the score on the leaderboard. This procedure is chosen because it ensures that final scores are not the result of overfitting.

Assessment

The challenge requires each participating student to write a report about his/her predictor.
To start off, you can make use of the Recommender System lab posted on Brightspace. This lab provides you with the basic ideas (collaborative filtering and latent factors), however solving it does not contribute to the challenge yet. You will have to go beyond the basic algorithms (as in the Recommender System lab) to qualify for a bonus point for a challenge.
The idea is to try out several different algorithms to improve your predictor. The criteria to assess the report of your predictor are provided in the report rubric.

Please note that (despite the ranking) **only your report is evaluated and not your code**. If you implemented an algorithm that turned out not to work as expected, please make sure to describe this algorithm in your report as well. If you have developed ideas about why certain algorithms do work and why other algorithms do not, please explain these ideas in your report as well.
We will have a look at your code though to verify whether the claims in your report correspond to your code.
Also, **please make sure to report your Kaggle name** so that it is clear which entry on the leaderboard your predictor corresponds to. We will have

Project reports and code should be made on **individual basis**. The cover sheet of your report should include your name, kaggle ID and student number.

The report submissions need to be done as submissions in the corresponding assignment on Brightspace. Late submissions are not possible.