TI2736-C: Datamining Project

https://inclass.kaggle.com/c/ti2736-c-datamining-project

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.

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 $n = 90{,}019$ true ratings r_i and the corresponding rating predictions $\hat{r}_i$, the RMSE of the recommendation algorithm is computed as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=0}^{n}(r_i - \hat{r}_i)^2}$$

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.

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.

Instructions

Use your @tudelft.nl email address to make an account on kaggle.com. You can download the following files from the Kaggle-in-Class project website:

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.

You can develop your algorithms in any programming language that you like. For your convenience, we provide. Java-code that reads in all data, runs an extremely simple prediction algorithm (namely, predict the mean rating), and writes the results into a submission file.

Assessment

Before the project deadline, you have to hand in all code that you have written as well as a small report (four pages max, excluding cover sheet) that describes the algorithm(s) you have implemented. 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. Also, please make sure that it is clear which entry on the leaderboard your submission corresponds to.

The following criteria will be used to assess your project: (1) the variety of algorithms you have tried, (2) the creativity of the solutions you have implemented, (3) the quality and efficiency of your implementations, (4) the level of understanding of the algorithms you display in your report, and (5) the quality of your final solution as reflected by the leaderboard. Please note that your final grade depends only for a small part on the performance of your final solution, so you can still receive a high grade with a poorly performing solution.

Project reports and code should be made individually or in groups of two. If you work in a group, the report should have a section detailing the contributions of each group member. The cover sheet of your report should include your name and student number.

Please hand in the project by emailing all items to datamining@abeellab.org. To make it easy for me to process all projects, please use the subject "TI2736-C Project" in your email. The deadline for handing in the project is April 10, 2015. This e-mail should have two attachments: (1) PDF with report, (2) zip file with source code