MOVIE LENS PROJECT REPORT

HARVARD UNIVERSITY
Professional Certificate in Data Science
Capstone Project # 1

Andrés Amaya Chaves
June 22nd, 2021

MACHINE LEARNING MODEL FOR PREDICTION OF MOVIE RATINGS IN AN ONLINE MEDIA SERVICE USING THE MATRIX FACTORIZATION METHOD

INTRODUCTION

The present report is about a hands-on project that requires the application of previous knowledge and expertise about Data Science, mainly those in Machine Learning.

For this project, a dataset called “MovieLens” was used; it is part of a public set of datasets from the GroupLens research lab of the Department of Computer Science and Engineering at the University of Minnesota (see https://grouplens.org/ (https://grouplens.org/)).

In this dataset, the GroupLens research lab documents a large number of observations about the movie ratings given by the users of a large online media service. The present project employs a Matrix Factorization machine learning algorithm to make predictions on the movie ratings over the alluded dataset.

In the GroupLens webpage, multiple versions of the mentioned dataset can be found, here it is used that one called “ml-10m”, which contains near 10 million ratings. In a bit more detail we can say that “MovieLens” dataset contains:

- **10 000 054** ratings, ranging in integers between 0 and 5, and
- **95 580** genre tags applied to
- **10677** movies by
- **69878** users

“MovieLens” has a total of **60 000 324** data, sizing about 401 Mb, unzipped.

The “MovieLens” dataset was split into a validation set, with a 10% size of the total entries, and a train-test dataset with the remaining 90%, for the development, test, and application of the machine learning algorithm.

The training-test data set was also split, this time into training and test sets. The **movie bias** in ratings, the **user’s bias**, and the **genre bias by users** were calculated. Then the predictions were calculated using those biases.

Two **MATRIX FACTORIZATION METHOD** models were tested. The better one was used to predict movie ratings over the validation set, from the combination of **User ID**, **Movie ID**, and the **set of genre tags** associated with each movie.

The goal of this capstone project is to reinforce, synthesize and apply the concepts learned all along with the *Professional Certificate Program*, as well as to have a rewarding experience with the development of a machine learning algorithm large real-world dataset.

METHODS

The gathering of the raw data was performed with a ‘download.file’ statement, then the ‘fread’ with nested ‘readLines’ and ‘unzip’ functions took out the _*.dat_ file concerning the userId, movieId, rating, and timestamp information. A similar procedure was applied to the information concerning the movie title and associated genres. The left_join function glued all that information together.

The createDataPartition function from the caret library was used to split the raw data, 10% to the validation set, and the remaining 90% to the training-test set. After the initial 90-10% splitting, the training-test data frame was also split between test and training sets in 20% and 80% of the data, respectively.

In Stage 1, over the training set, it was calculated the general bias about every single movie, then called “**movie bias**”, which measures how deviated is, in average, the rating for each movie; the ‘movie_bias’ data frame, was built with that information for later use in various steps.

In Stage 2, a “**user bias**” was built, which tells how much, on average, a user’s rating differs from the general mean.

Then, in Stage 3 (“CALCULATE USER BIAS REGARDING EVERY GENRE”) the **"user-genre bias** was calculated, which tells in detail, how much a user increases or decreases his rating given for the movies tagged with every genre, on average.

Stage 3 is the core of the built models and required 6 steps to be done; the Matrix ‘A’ is created in that stage, which contains the relationship of every movie in the ‘subTrain’ set with the genre tags, then, matrix ‘A’ was used to build a data frame that reveals how biased is each user regarding every genre. As a result, the code creates a data frame called “b_g”, which contains the specific bias that every user has regarding every genre in the dataset.

Next, in Stage 4, two **MATRIX FACTORIZATION METHOD** models are developed with the **training set** and used to accomplish the predictions over the **test set**; both, in each observation add that calculated “**user-genre bias**” to the overall mean rating plus the “**movie bias**” and “**user bias**”.

The difference between the models is that the ‘whole **user-genre bias**’, in the first one, is obtained **summing up** all genres bias from each user (taken from the “b_g” data frame), while in the second it comes from calculating the **mean** of the genre single-bias for each user. As we will see, the second one performs better.

There is a data frame, called “D”, that is a milestone in the models because it takes each entry in the input set, where the model reads the ‘**userId**’, looks for it in the “b_g” dataframe to gather the **user-genre biases**, identifies the movie’s associated genres, and then multiply them, outputting only the user-genre biases that concern to that specific movie.

That latter procedure makes it possible to ‘communicate’ two large data sets that have very different sizes and output a single genre bias number for each entry of the dataset where the prediction occurs.

Then, a column with a unique bias number per user is attached to the original input set; that number is calculated according to the model, as described above.

In Stage 5, model 2 (“mean of all genre bias for each user”) is used to make the predictions over the **validation set**, as it has a lower RMSE than model 1.

RESULTS

Once each model was trained and tested over the proper sets, a histogram was graphed with the ‘qplot’ function. In each of the next 3 Figures, a 3-graph set shows how the “movie bias”, “user bias” and “genre bias” distribute along with the corresponding dataset, applying the corresponding model.

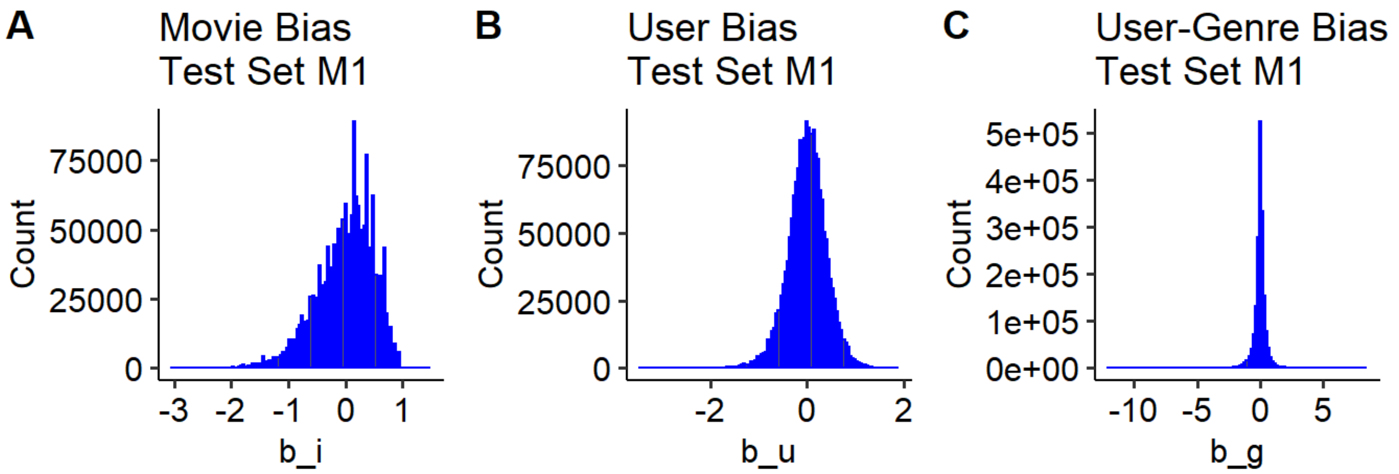


Figure 1. Histograms showing bias distributions in Test Set for Model 1.

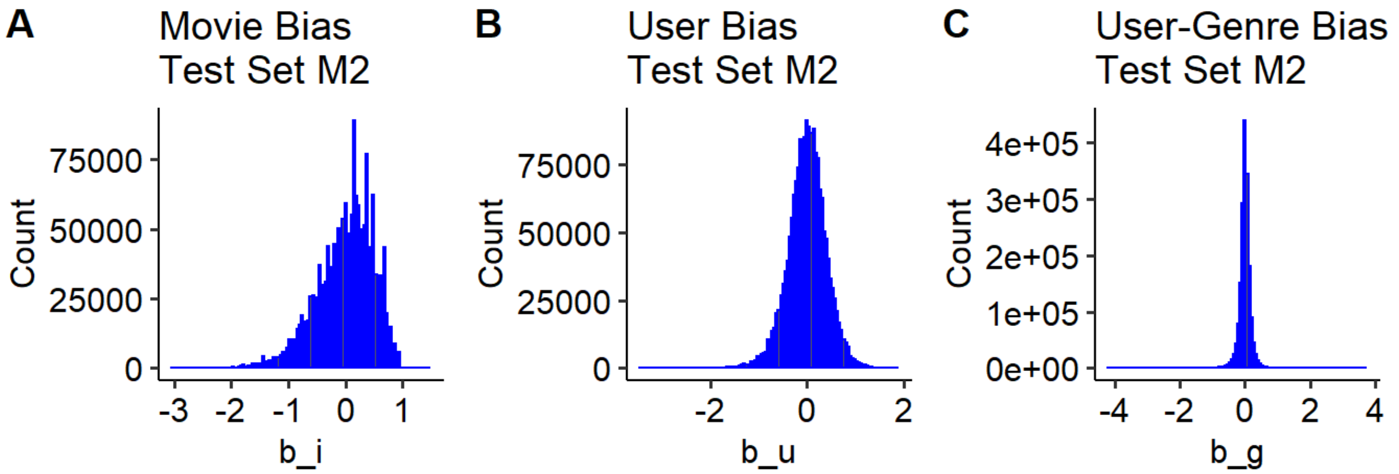


Figure 2. Histograms showing bias distributions in Test Set for Model 2.

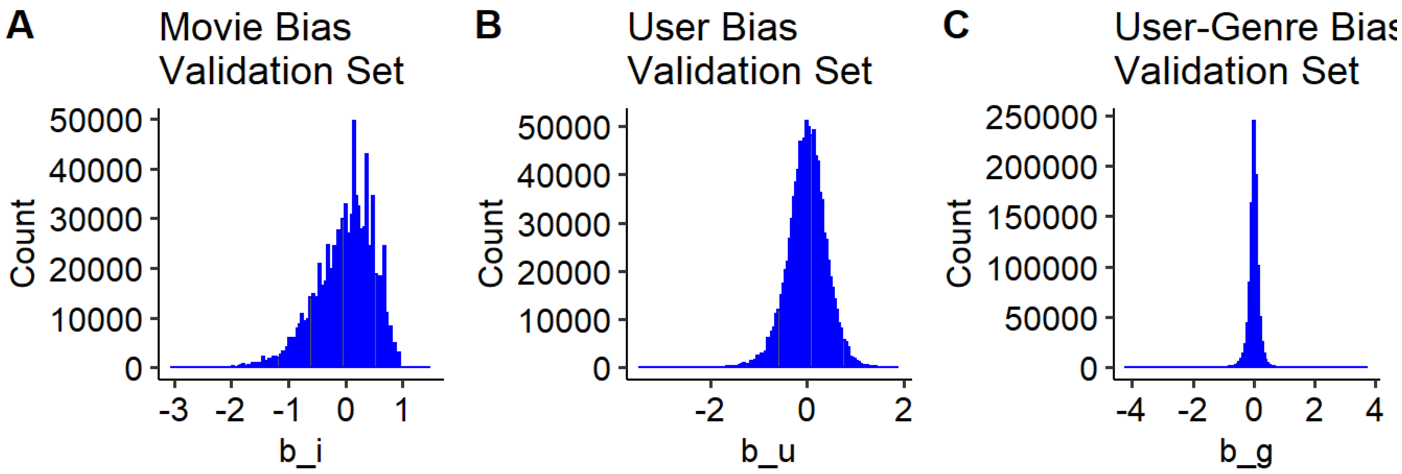


Figure 3. Histograms showing bias distributions in Verification Set for the picked model.

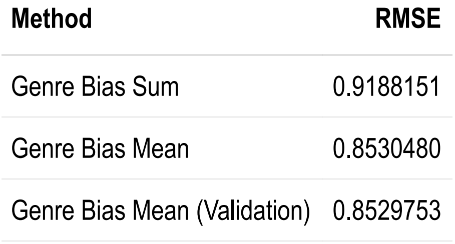
In the latter figures, we can observe that the users of the online media service have a tiny degree of bias about the movie genres; the vast majority of users is very close to the absence of bias in that sense, therefore, we can state that **the set of genres with which a movie is tagged is not so much determinant to the rating they give for a movie. However, it is indeed a variable that helps to improve the accuracy of the model**.

From Figure 1A to Figure 2A, and from Figure 1B to Figure 2B, b_i and b_u distributions do not vary at all because both models are trained with the same b_i and b_u values. Since the difference of those models is the way they calculate b_g, Figure 1C and Figure 2C do indeed change, although their shapes resemble very much, the 'Count' values differ.

Distributions in Figure 3 for all three variables change in magnitude in comparison with Figure 2, because the Validation Set is smaller than the Test Set; but they do not vary in shape, because both datasets were built from samples of the original 10-million dataset downloaded from the *GroupLens* webpage, and both Figures presents the components which addition ends up in predictions that come from exactly the same model.

In Table 1, below, we can see the output of the **rmse_results** data frame, which summarizes the **root-mean-square error (RMSE)** results, that come from the comparison between the actual ratings in the corresponding set and the predicted values for those ratings, product of the application of each model. The first row is from **Model 1** for **Test Set**. The second row is from **Model 2** for **Test Set**. The last row is from the unique model used for the **validation set**.

Table 1. RMSE results for the corresponding sets and models.



Method	RMSE
Genre Bias Sum	0.9188151
Genre Bias Mean	0.8530480
Genre Bias Mean (Validation)	0.8529753

Thus, the developed model predicts the movie ratings with a root-mean-square error **(RMSE)** of **0.8529753**, computed over the validation set.

With the resulting error of the models, we can infer that, over the validation set, the developed model is enough accurate to predict a movie rating within a ± 0.8530 interval error, with high probability.

If we recall that the range of ratings can be between 0 and 5, such error represents about 17% of the possible range. Although that is not a very high accuracy, it is a quite acceptable error for this model.

NOTES

• There are some points in the code in which it was necessary to raise a bit the memory limit using the memory.limit() function, to ensure R does not stop while running the code. The function is already included in the code, in the following points:

- In step 3.5.2, before building the 'b_ug' data frame. Raised memory to 7000 Mb.
- In step 3.6, before building the 'b_g' data frame. Raised memory to 7000 Mb.
- In step 4.1.1, before calculating the 'D' data frame. Raised memory to 8500 Mb.
- In step 4.2.1, before calculating the 'D' data frame. Raised memory to 8500 Mb.
- In step 5.2.1, before calculating the 'V' data frame. Raised memory to 8500 Mb.

• The fact that a mov254ie does not have any tags related to genres, was taken as another movie genre.

• The file "Amaya_MovieLensProject.r", attached to the submitted package, contains the code of the algorithm in this report; it includes 412 lines and weighs 13 Kb.

CONCLUSIONS

• The effect over a movie rating, regarding its combination of the genre tags, is better modeled by calculating the mean of the single user-genre bias rather than summing them up, as the results of this MATRIX FACTORIZATION METHOD models suggest.

• We can state that **the set of genres with which a movie is tagged is not so much determinant to the rating they give for a movie. However, it is indeed a variable that helps to diminish the error of the model**.

• The developed model can predict movie ratings with the precision that yields from the fact that specifically over the created validation set, the predictions output a root-mean-square error **(RMSE)** over the validation set of **0.8529753**.

• This model can help the ONLINE MEDIA SERVICES to make precise predictions about the movie ratings that its users will give, based on previous behavior. This translates into more accurate suggestions to users, that would permit them to offer a better service, which could cause a rise in new users.