MovieLens Project

Berkalp Altay

January 12, 2022

Contents

1. Introduction

This project aims to build a recommendation system using the MovieLens 10M dataset.

1.1. Dataset

In this MovieLens dataset, there are about 10 Million movie ratings with 6 different categories associated with each rating.

These 6 categories are the following:

- Title of the movie
- Movie ID
- Genre of the movie
- User ID
- Rating (from 0.5 to 5 in increments of 0.5 with 5 being the best rating possible)
- Timestamp for the date and time of the rating

There are approximately 70000 unique users and 10000 unique movies.
The genres include Comedy, Adventure, Drama, and so on.
Also, the release date of a movie is included in parentheses at the end of a title.

1.2. Evaluation Criteria

In the next sections, the data will be explored and recommendation systems will be built using the data. The recommendation system will be evaluated using the Root Mean Squared Error (RMSE).

The RMSE formula is as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\hat{y}_i^2 - y_i^2}$$

In this formula, $\hat{y}_i$ represents the predicted ratings based on the recommendation system while y_i represents the actual ratings from the dataset.
The lower the RMSE, the better the model is. For this project, an RMSE lower than 0.86490 is the ultimate target. This ultimate RMSE evaluation will be based on a validation set that will be used only for model evaluation at the end in Section `3. Results`.
Using the validation set, the Regularized Movie & User & Release Year & Timestamp Model, the final model in this project, reaches an RMSE of **0.8648589**. This meets the ultimate target.

2. Analysis

2.1. Data Exploration

The 10 million movie ratings are initially divided into two datasets: *edx* and *validation* with the *edx* dataset containing approximately 90% of the data while the *validation* dataset contains the remaining 10%.
The rows of both the *edx* and *validation* datasets are entries for different ratings.

Also, both the *edx* and *validation* datasets have 6 columns.

Table 1: Columns

columns	column_explanations
userId	Unique identifier for each user
movieId	Unique identifier for each movie
title	Title of a movie
genres	Genre of a movie - multiple genres are concatenated with a | sign such as Comedy|Romance
rating	Rating given by a specific user to a specific movie
timestamp	Date and time associated with each rating

The *validation* set will be used in Section `3. Results` as a final hold-out test set for only reporting the RMSE of the final model. Therefore, a training set and a test set seem necessary while building and testing models. To that end, the *edx* set has been divided into *edxTrain* and *edxTest* sets. The *edxTest* set has been created as the same size as that of the *validation* set. So, both the *edxTest* and *validation* sets have approximately 1 million movie ratings (i.e. both have 10% of the original MovieLens 10M dataset) while the *edxTrain* set has approximately 8 million movie ratings.

2.2. Data Preparation and Data Visualization

2.2.1. Data Preparation Phase

After the data exploration phase, we see that the title column contains the release year for each movie. For example, the title for Pulp Fiction is listed as `Pulp Fiction (1994)`.
Also, the timestamp is not in a human-readable format.
First, the release year for each movie will be extracted from the title column and added to *edxTrain* and *edxTest* sets as a column.
Second, the timestamp column will be converted to a human-readable form and the year will be extracted. This will provide the year in which the rating was given.

Third, the RMSE formula for the evaluation purposes will be defined as per the definition given in Section `1.2. Evaluation Criteria`.

After these changes, we can see some rows of the *edxTrain* and *edxTest* sets below.

Table 2: First 3 rows of edxTrain set

userId	movieId	rating	timestamp	title	genres	release_year	rating_year
1	185	5	838983525	Net, The (1995)	Action|Crime|Thriller	1995	1996
1	329	5	838983392	Star Trek: Generations (1994)	Action|Adventure|Drama|Sci-Fi	1994	1996
1	355	5	838984474	Flintstones, The (1994)	Children|Comedy|Fantasy	1994	1996

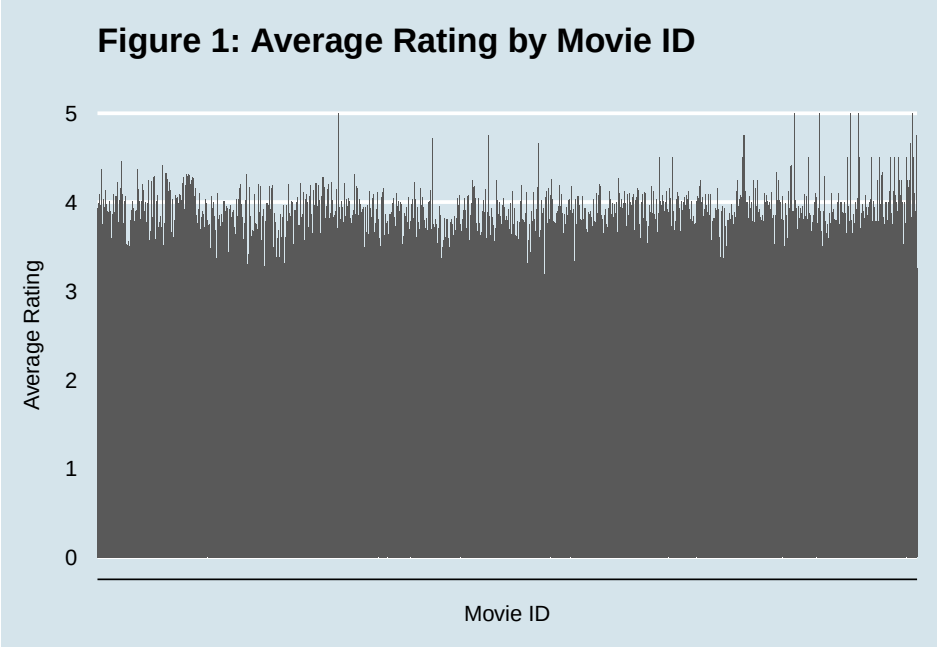
Table 3: First 3 rows of edxTest set

userId	movieId	rating	timestamp	title	genres	release_year	rating_year
1	122	5	838985046	Boomerang (1992)	Comedy|Romance	1992	1996
1	292	5	838983421	Outbreak (1995)	Action|Drama|Sci-Fi|Thriller	1995	1996
1	316	5	838983392	Stargate (1994)	Action|Adventure|Sci-Fi	1994	1996

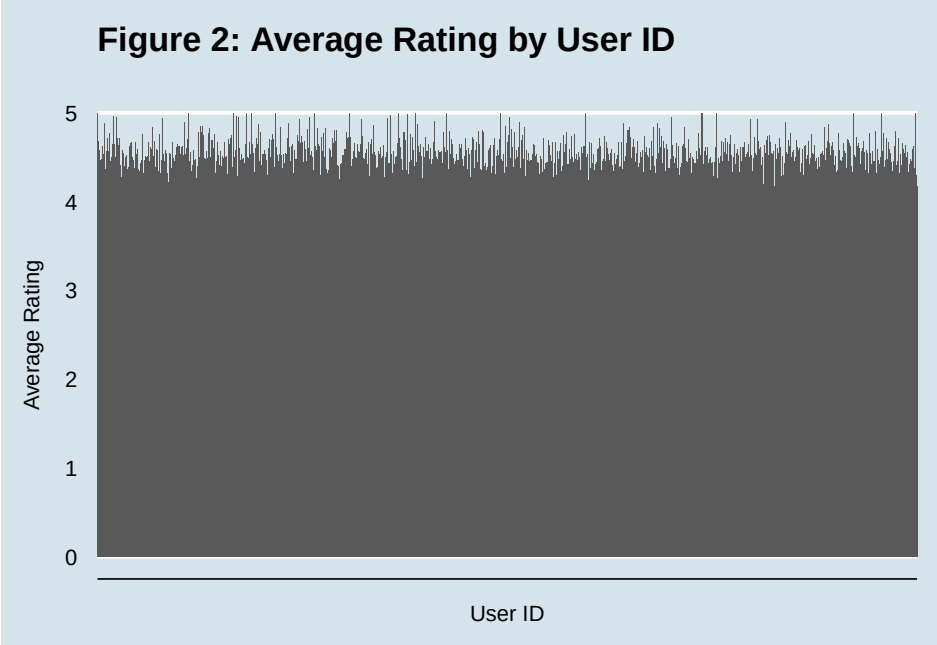
2.2.2. Data Visualization

2.2.2.1. Average Rating

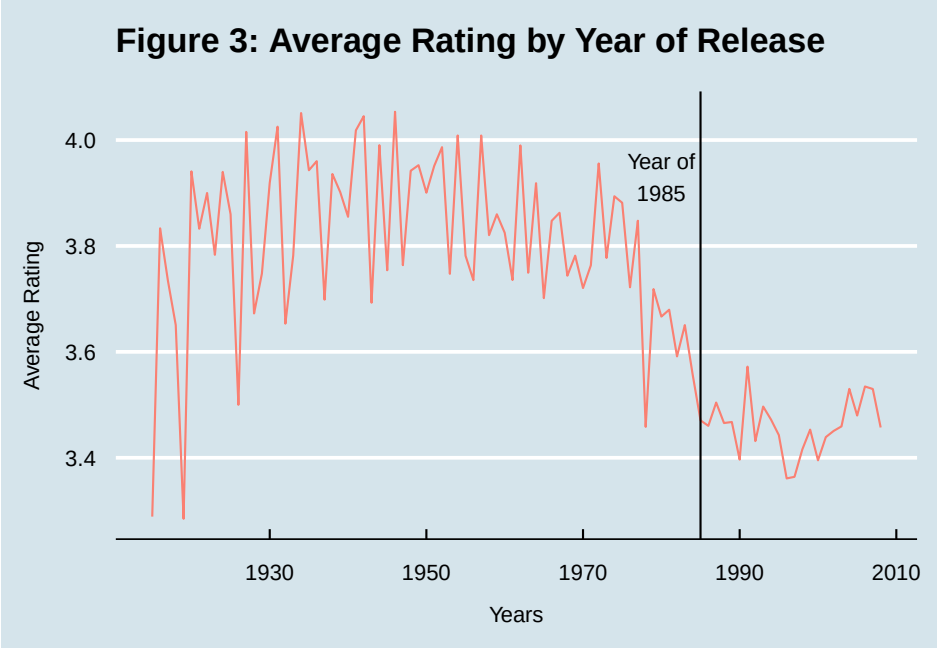
First, we notice that every movie has a different average rating. Below, we can see this looking at Figure 1 created using the *edx* set.



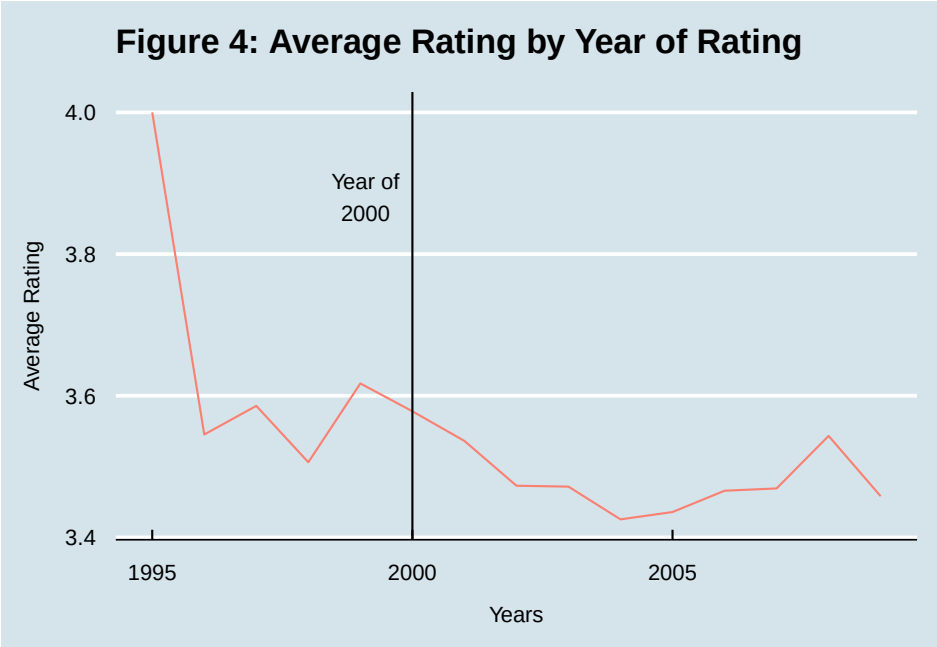
Second, we notice that every user has a different average rating. Below, we can see this looking at Figure 2 created using the *edx* set.



Third, we notice that the movies released between around 1930 and 1985 seem to have higher ratings than the ones released after 1985. Below, we can see this looking at Figure 3 created using the *edx* set.



Fourth, we notice that the movies rated before 2000 seem to have higher ratings than the ones released after 2000. Below, we can see this looking at Figure 4 created using the *edx* set.



As seen in the Figures above, there is significant variability in ratings by movie ID, user ID, year of release, and year of rating.

2.2.2.2. Frequency

First, not all movies are rated as frequently as others. In Tables 4 and 5 below, you can see top 5

and bottom 5 movies in the *edx* set by the frequency of ratings, respectively. For example, in the *edx* set, `Pulp Fiction` was rated 31362 times while `100 Feet` was rated only once.

Table 4: Top 5 Movies by Rating Frequency

title	count
Pulp Fiction (1994)	31362
Forrest Gump (1994)	31079
Silence of the Lambs, The (1991)	30382
Jurassic Park (1993)	29360
Shawshank Redemption, The (1994)	28015

Table 5: Bottom 5 Movies by Rating Frequency

title	count
1, 2, 3, Sun (Un, deuz, trois, soleil) (1993)	1
100 Feet (2008)	1
4 (2005)	1
Accused (Anklaget) (2005)	1
Ace of Hearts (2008)	1

Second, not all users rate as frequently as others. In Tables 6 and 7 below, you can see top 5 and bottom 5 user IDs in the *edx* set by the number of times they have rated, respectively. For example, in the *edx* set, the user with an ID number of 59269 rated 6616 times while the user with an ID number of 62516 rated only 10 times.

Table 6: Top 5 Users by Rating Frequency

userId	count
59269	6616
67385	6360
14463	4648
68259	4036
27468	4023

Table 7: Bottom 5 Users by Rating Frequency

userId	count
62516	10
22170	12
15719	13
50608	13
901	14

2.3. Modeling Approach

In this section, various models will be built. In all these models, *edxTest* set will be used with the RMSE function for evaluation of the model since the evaluation of the final model using the *validation* set will be conducted in the `3. Results` Section.

2.3.1. Baseline Mean Model

In this model, we assume that every movie has the same rating and that same rating is the average (i.e. mean) rating of all movies.

The formula for the Baseline Mean Model:

$$Y = \mu + \epsilon$$

where μ stands for the average rating for all movies and ϵ stands for independent errors sampled from the same distribution centered at zero.

Therefore, this model assigns a rating of about **3.512417** to every movie.

The RMSE for the Baseline Mean model using the *edxTest* set is around **1.059487**. We can interpret this result as missing the actual rating stars by around 1.06 stars. This is far from perfect and also misses the ultimate target of **0.86490**.

2.3.2. Movie Model

In this model, we start taking movies into account. In particular, we know and use the fact that not all movies are rated the same, as seen in Figure 1 in Section `2.2.`. Therefore, we account for the variability of ratings due to movies themselves.

The formula for the Movie Model:

$$Y_m = \mu + b_m + \epsilon$$

where μ stands for the average rating for all movies and b_m stands for the average rating for movie m, also called the movie effect while ϵ stands for independent errors sampled from the same distribution centered at zero.

The RMSE for the Movie Model using the *edxTest* set is around **0.9434865**. While this is a substantial improvement from the RMSE of **1.059487** for the Baseline Mean Model, it misses the ultimate target of **0.86490** and can be further improved.

2.3.3. Movie & User Model

In this model, we take both movies and users into account when it comes to predicting ratings. In particular, we know and use the fact that not all users give the same ratings.

The formula for the Movie & User Model:

$$Y_{m,u} = \mu + b_m + b_u + \epsilon$$

where μ stands for the average rating for all movies, b_m stands for the average rating for movie m, also called

the movie effect, b_u stands for the user effect while ϵ stands for independent errors sampled from the same distribution centered at zero.

The RMSE for the Movie & User Model using the *edxTest* set is around **0.8655475**. While this is a great improvement from the RMSE of **1.059487** for the Baseline Mean Model and the RMSE of **0.9434865** for the Movie Model, it misses the ultimate target of **0.86490** and can be further improved.

2.3.4. Movie & User & Year Model

While the ratings are greatly influenced by users and movies, there seems to be some variation in ratings by the year of release. In Figure 3 in Section `2.2.`, we can see that people gave higher ratings to movies released between around 1930 and 1985 than the ones released from 1985 onward.

The formula for the Movie & User & Year Model:

$$Y_{m,u,y} = \mu + b_m + b_u + b_y + \epsilon$$

where μ stands for the average rating for all movies, b_m stands for the average rating for movie m, also called the movie effect, b_u stands for the user effect, b_y stands for the effect of the year of release while ϵ stands for independent errors sampled from the same distribution centered at zero.

The RMSE for the Movie & User & Year Model using the *edxTest* set is around **0.8651656**. Although this comes closer to the ultimate target of **0.86490** than the RMSE of **0.8655475** for the Movie & User Model, it can still be improved.

2.3.5. Movie & User & Year & Timestamp Model

While the ratings are greatly influenced by users and movies, there seems to be some variation in ratings by the year of rating (i.e. year of Timestamp). In Figure 4 in Section `2.2.`, we can see some variation in ratings by the year of rating by a user.

The formula for the Movie & User & Year & Timestamp Model:

$$Y_{m,u,y,t} = \mu + b_m + b_u + b_y + b_t + \epsilon$$

where μ stands for the average rating for all movies, b_m stands for the average rating for movie m, also called the movie effect, b_u stands for the user effect, b_y stands for the effect of the year of release, b_t stands for the effect of the year of rating (i.e. timestamp effect) while ϵ stands for independent errors sampled from the same distribution centered at zero.

The RMSE for the Movie & User & Year & Timestamp Model using the *edxTest* set is around **0.8650907**. Although this comes closer to the ultimate target of **0.86490** than the RMSE of **0.8651656** for the Movie & User & Year Model, it can still be improved.

2.3.6. Regularized Movie & User & Release Year & Timestamp Model

While the last model is a good improvement, we can improve the results further. In Tables 4 and 5, we see that some movies are rated more frequently than others. Likewise, some users rate more frequently than others while some years of release and years of rating have more movie ratings than others. In this case, a smaller frequency of rating can introduce bias into the model. For example, a movie might be rated 5.0 but have only one rating. In this case, our prediction of a 5.0 rating introduces a bias into our model.
To that end, we can modify the RMSE to penalize large estimates coming from small samples.

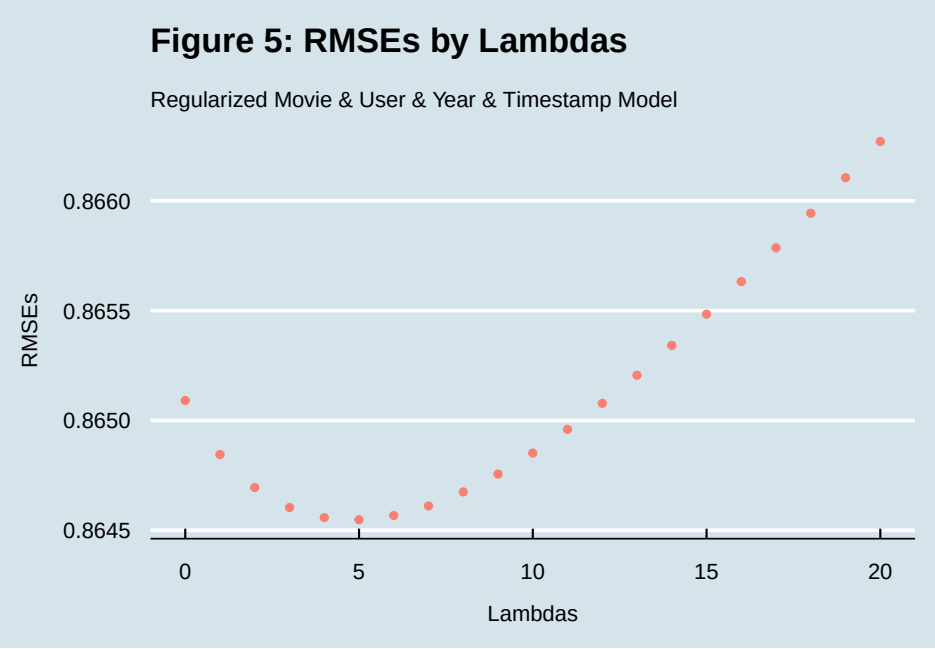
The formula for the RMSE with regularization:

$$\frac{1}{N}\sum_{m,u,y,t}(y_{m,u,y,t}-\mu-b_m-b_u-b_y-b_t)^2+\lambda(\sum_m b_m^2+\sum_u b_u^2+\sum_y b_y^2+\sum_t b_t^2)$$

The term on the left hand side of the plus sign is the mean squared error and the term on the right hand side is a penalty term for larger b's. So, we have to pick b's to minimize this equation.

Multiple values of lambda can be tried to find the best solution. To tune lambda this way, only the *edxTest* set will be used since the *validation* set is reserved only for the final evaluation of the model.

While more lambda values offer greater tuning for the regularization, more lambda values result in more computationally expensive processes. Therefore, the sequence of numbers from 0 to 20 in increments of 1 is chosen as the lambda values to strike a balance between better tuning and computational expense.
The RMSE values obtained for these lambda values are given in Figure 5 below.



As seen in the `RMSEs by Lambdas` graph above, the smallest RMSEs occurs when lambda is equal to 5. Using 5 as the lambda and the *edxTest* set, the RMSE for the Regularized Movie & User & Year & Timestamp Model is around **0.8645478** beating the ultimate target of **0.86490**.

3. Results

Now, the *validation* set can be used to evaluate the Regularized Movie & User & Year & Timestamp Model, the final model. The best lambda that results in the minimum RMSE for the model was obtained using the model on the *edxTest* set in Section `2.3.6.` and will be used for the evaluation purpose of the *validation* set.

The Regularized Movie & User & Year & Timestamp Model has a RMSE of **0.8648589** which beats the ultimate target of **0.86490**.
For reference, the RMSE of all models using the *validation* set can be found below:

Table 8: RMSEs of All Models Using Validation Set

Model	RMSE
Baseline Mean Model	1.0612018
Movie Model	0.9439998
Movie & User Model	0.8659094
Movie & User & Year Model	0.8655625
Movie & User & Year & Timestamp Model	0.8654879
Regularized Movie & User & Year & Timestamp Model	0.8648589

As seen in Table 8 above, the ultimate model was built by incorporating new features in every step of the model building process.

The model performance ultimately improved from an RMSE of **1.0612018** (i.e. Baseline Mean Model) to the final RMSE of **0.8648589**.

4. Conclusion

In this project, the MovieLens 10M dataset was sucessfully used to build a recommendation system model.

This report introduced the MovieLens dataset, explored its properties, introduced newly developed features from existing ones and visualized the data.

Then, a model was built bit by bit from the rudimentary approach of assigning the overall movie average to all movies to a regularized model that incorporated movies, users, years of release for the movies and the years of ratings by users that decreased the error between a predicted rating and an actual rating.

The model improved the most when the model incorporated movieId and userId. Adding the year of release (i.e. Year component of the models) and the year of rating (i.e. Timestamp component of the models) led to a slight improvement. Regularization also helped in this regard as it penalized big estimates of effects from small sample sizes.

All in all, the final model achieved the goal of reaching an RMSE below **0.86490**.

Some limitations of this project is that it did not employ methods such as matrix factorization or artificial neural networks. Such methods can be used to discern relationships among data that are not readily recognizable. Therefore, any future work can further improve on this report by employing such methods.