Movielens Recommendation System

HarvardX Data Science Professional Certificate: PH125.9x Capstone 1

Johannes Werner

12 March, 2025

Contents

1 Introduction

In this project, a movie recommendation system is developed using the MovieLens dataset with machine learning techniques. The report describes the creation of the recommendation system at the example of the MovieLens 10M dataset (GroupLens 2009), with 10 million movie ratings.

The MovieLens dataset is split into two parts: the training set (edx) and the final holdout test set (validation). The goal is to create an algorithm that predicts the ratings for movies with high accuracy with a root mean square error (RMSE) lower than 0.86490, tested on the validation set.

The project is divided into the following steps:

- **Data exploration:** Analysis and visualization of the dataset to get an overview of the data and to see important patterns and behaviors.
- **Method development:** Implementation of various approaches and effects, starting with simple averages and progressing to more advanced techniques, described mathematically.
- **Model results:** Application of the methods on the data with their impact on the RMSE and evaluation.

Over the process, the edx dataset is used for training and testing. The validation set (final holdout set) is reserved for the final performance check of the completed algorithm.

The sections in this report describe the methods used, the results obtained, and the learnings gained during the development of the recommendation system. The report concludes with a discussion of the limitations of the model and gives recommendations for future projects and their improvements.

The report was created using R Markdown in RStudio including the use of the R programming language for statistical computing and data analysis.

2 Data Exploration

In this section an overview over the edx dataset is made. In Table 1 and Table 2 the number of rows and the structure of the data in the first 10 rows can be seen.

Table 1: Number of rows in edx Dataset

Rows
9000055

Table 2: First 10 Rows of edx Dataset

	userId	movieId	rating	timestamp	title	genres
1	1	122	5	838985046	Boomerang (1992)	Comedy|Romance
2	1	185	5	838983525	Net, The (1995)	Action|Crime|Thriller
4	1	292	5	838983421	Outbreak (1995)	Action|Drama|Sci-Fi|Thriller
5	1	316	5	838983392	Stargate (1994)	Action|Adventure|Sci-Fi
6	1	329	5	838983392	Star Trek: Generations (1994)	Action|Adventure|Drama|Sci-Fi
7	1	355	5	838984474	Flintstones, The (1994)	Children|Comedy|Fantasy
8	1	356	5	838983653	Forrest Gump (1994)	Comedy|Drama|Romance|War
9	1	362	5	838984885	Jungle Book, The (1994)	Adventure|Children|Romance
10	1	364	5	838983707	Lion King, The (1994)	Adventure|Animation|Children|Drama|Musical
11	1	370	5	838984596	Naked Gun 33 1/3: The Final Insult (1994)	Action|Comedy

2.1 Ratings

In the data, movie ratings are included. Possible ratings were in the range between 0.5 and 5. Hereby, whole numbers seem to be preferred. The distribution of the number of ratings on the different rating values can be seen in Figure 1. In Table 3, the top 10 films with the most ratings are listed.

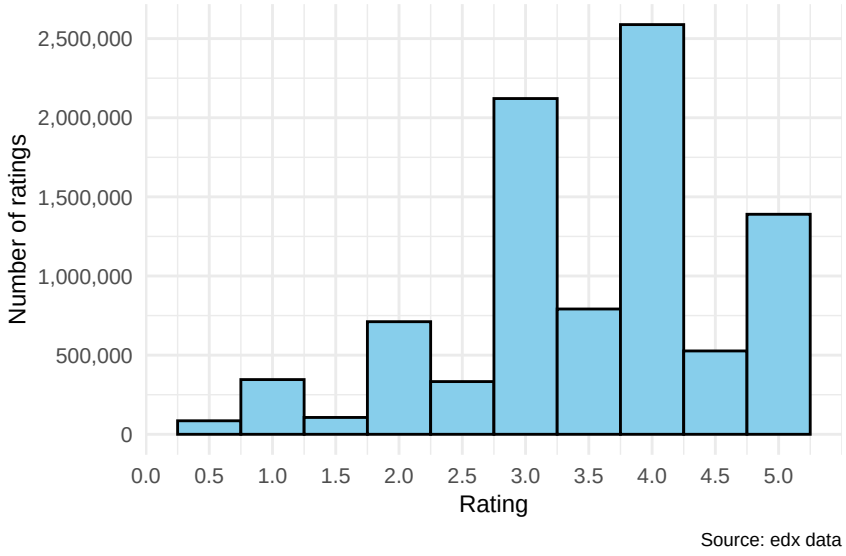

Figure 1: Distribution of film ratings

Table 3: Top 10 Films with the most ratings

movieId	title	count
296	Pulp Fiction (1994)	31362
356	Forrest Gump (1994)	31079
593	Silence of the Lambs, The (1991)	30382
480	Jurassic Park (1993)	29360
318	Shawshank Redemption, The (1994)	28015
110	Braveheart (1995)	26212
457	Fugitive, The (1993)	25998
589	Terminator 2: Judgment Day (1991)	25984
260	Star Wars: Episode IV - A New Hope (a.k.a. Star Wars) (1977)	25672
150	Apollo 13 (1995)	24284

The dataset includes ratings from 1995 to 2009. During that time, the number of ratings made changes. In Figure 2, the number of ratings are summed up per week and plotted by time. It can be seen that before 2000, the number of ratings varied a lot. After 2000, the number of ratings reached a certain level with some peaks, decreased again but stayed at a level of about 22,000 ratings per week till 2009. The peaks may possibly come from a blockbuster movie coming out at that time.

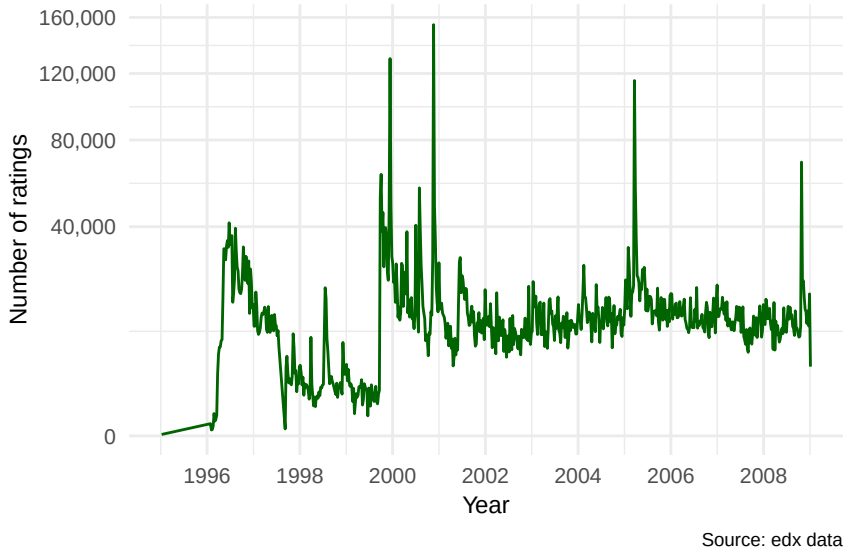

Figure 2: Number of ratings over time (weekly)

2.2 Genre related observations

Film genres are also included in the data. In Figure 3 (left), there are the top 10 first named genres related to the specific movie and its rating. So, these genres seem to be the most important in the dataset. In Figure 3 (right), the average rating per genre with the dot size representing the number of ratings can be seen. There seems to be a bias per genre, as different genres tend to get higher ratings than others. So, there seems to be a genre effect on the rating. If this can be seen here, it is also likely that on a lower level, specific films also influence the average rating.

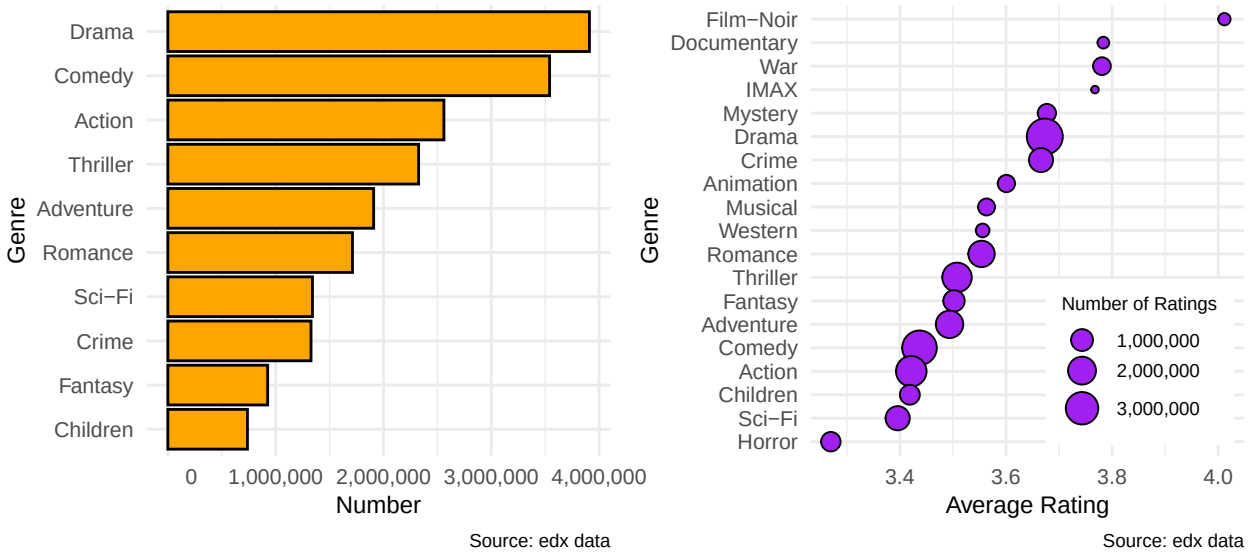

Figure 3: Top 10 film genres (left) and average rating per genre (right)

2.3 User related observations

The next two diagrams in Figure 4 show the average rating vs. number of users, respectively, the average number of ratings per user. Most of the users have an average rating overall of 3.5 - 3.8. This seems to follow a Gaussian distribution. On the right, it can be seen that there are some "power users" (around 200 - 400) who have several thousand ratings. However, most of the users only made a small number of ratings.

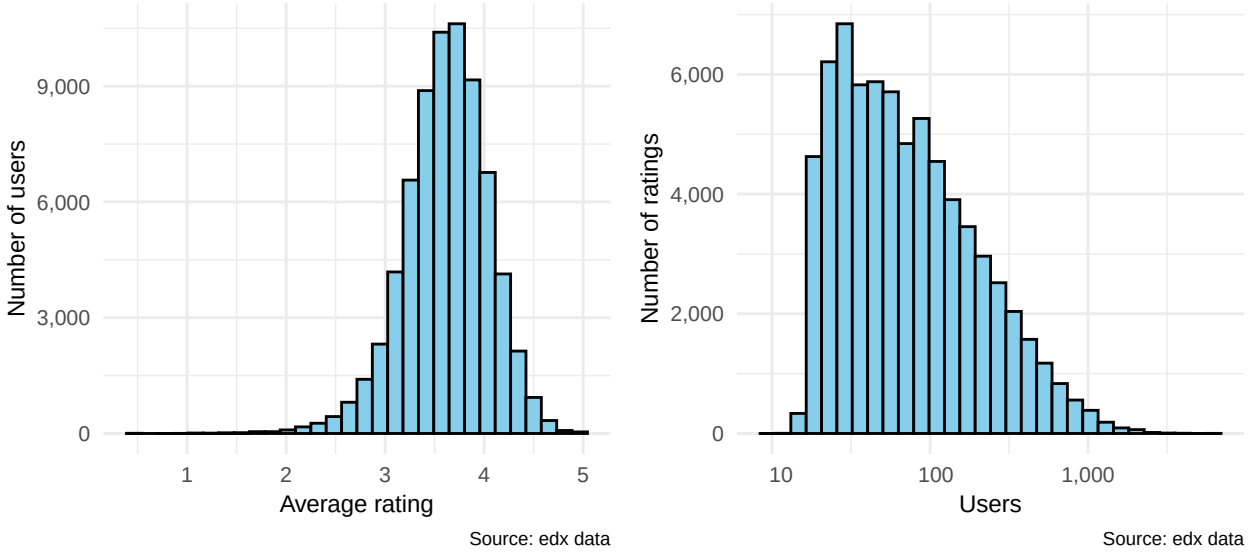

Figure 4: Average rating per user (left) and average number of ratings per user (right)

2.4 Time dependent observations

In the dataset, there are time dimensions that should be mentioned. The first one is the time of rating. Possibly, in some time period or time of the year, the average ratings are different. So, there could possibly be a time effect on the rating value. The second time dimension is the release year of the specific movie. This value can be extracted from the title in the dataset. If the average rating per release year of the film is plotted in a diagram, it can be seen that older films tend to have a higher rating than newer films. So, both effects could play a role in developing the algorithm to predict the ratings. The two diagrams can be seen in Figure 5.

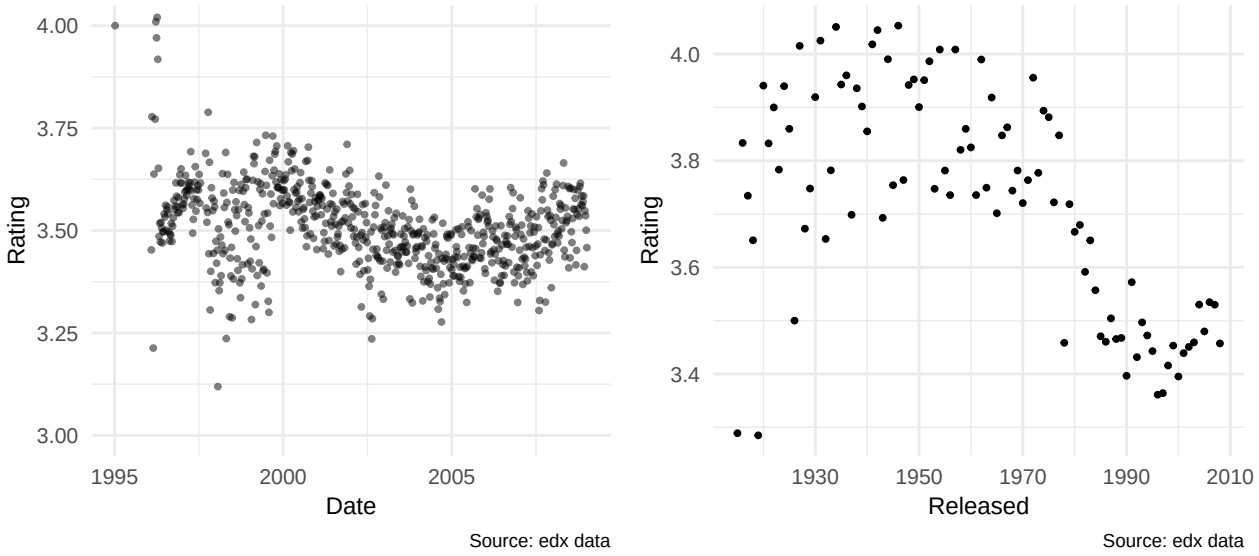

Figure 5: Date vs. weekly averaged rating (left) and release year vs. averaged rating (right)

3 Methods

In this chapter the mathematical methods used for evaluating and for building up the prediction algorithm are described step by step. It starts with the performance metric and the average rating and adds in every step another effect to improve the model.

3.1 Performance metric: RMSE

The model performance is evaluated by the "Root Mean Square Error" (RMSE), which was also used in the Netflix challenge (Koren, Bell, and Volinsky 2009), where a similar problem needed to be evaluated. The RMSE is a measure of the average deviation between predicted and actual ratings.

The RMSE is defined by the following formula (Irizarry 2019, Ch. 33.7.3):

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{u,i}\left(\hat{y}_{u,i} - y_{u,i}\right)^2}$$

Where:

- $y_{u,i}$ is the actual rating for movie i by user u
- $\hat{y}_{u,i}$ is the prediction
- N is the total number of user-movie combinations

A lower RMSE indicates better performance. The objective of this project is to achieve an RMSE below 0.86490.

3.2 Model structure

The predition model is constructed step by step, starting with the most basic form and incrementally adding additional effects. This approach allows to understand the change coming from each factor and to evaluate the contribution to the overall prediction accuracy.

Basic model

The simple model assumes that all ratings are scattered around a global mean (Irizarry 2019, Ch. 33.7.4):

$$Y_{u,i} = \mu + \varepsilon_{u,i}$$

Where:

- μ is the overall average rating across all movies and users
- $\varepsilon_{u,i}$ is the error term, representing the deviation from the mean

Movie effect model

Then, a movie-specific effect is added to account for the fact that some movies are rated higher or lower than others (Irizarry 2019, Ch. 33.7.5):

$$Y_{u,i} = \mu + b_i + \varepsilon_{u,i}$$

Where b_i represents the movie effect for movie i. This takes into account how much a movie's average rating deviates from the global mean.

User effect model

Next, a user effect to account for individual user rating biases is added (Irizarry 2019, Ch. 33.7.6):

$$Y_{u,i} = \mu + b_i + b_u + \varepsilon_{u,i}$$

Where b_u represents the user effect for user u. This takes into account how much a user's average rating deviates from the global mean.

Release year effect model

To account for potential trends or biases related to a movie's release year, a release year effect is introduced:

$$Y_{u,i} = \mu + b_i + b_u + b_r + \varepsilon_{u,i}$$

Where b_r represents the release year effect for the release year r in which the movie was released.

Genre effect model

Finally, a genre effect is implemented to capture rating patterns coming with different movie genres:

$$Y_{u,i} = \mu + b_i + b_u + b_r + b_g + \varepsilon_{u,i}$$

Where b_g represents the genre effect for genre g.

3.3 Time adjustment

Temporal changes in ratings are implemented by a time adjustment using weekly averaged ratings and generalized additive model (GAM) curves (Clark n.d.). This allows to capture and correct trends in ratings over time.

This is done by the following steps:

1. **Calculate the weekly averaged ratings.**
2. **Fit a GAM** curve to these ratings. This creates a smooth curve that represents the trend of rating with a small amount of data points.
3. **Substract the overall mean rating** from the fit curve to get the effect relative to the mean.
4. **Use the centered GAM curve** to adjust the individual effects.

Mathematically, for the movie effect b_i, this can be represented as:

$$b_i^*(t) = b_i + (f_i(t) - \mu)$$

Where:

- $b_i^*(t)$ is the time-adjusted movie effect at time t
- b_i is the original movie effect
- $f_i(t)$ is the fitted GAM function for movie i at time t
- μ is the overall mean rating

Similar adjustments are made for user effects (b_u) and release year effects (b_y).

This approach allows to capture and correct temporal trends in ratings while the structure in the effect stays the same. The time-adjusted effects (b_i^*, b_u^*, and b_y^*) are then used the following steps.

Note: Not all effects are time-adjusted. The reason for that is explained in 4.2.

3.4 Range correction

After applying time-adjustments a range correction can be performed. This makes sure that the predicted ratings lie within the valid rating range (0.5 to 5). Any predictions outside of that range are adjusted to the nearest valid rating.

3.5 Regularization

For prevention of overfitting, especially for movies or users with a small amount of ratings, regularization is applied. This penalizes large effect values, by shifting them towards 0 (deviation) ("Regularization in Machine Learning" 2025).

Therefore, the equation that includes a penalty term is minimized (cf. Irizarry 2019, Ch. 33.9.2):

$$\sum_{u,i}(y_{u,i} - \mu - b_i)^2 + \lambda \sum_i b_i^2$$

The first term is the sum of squared errors; the second term penalizes large values of b_i. The parameter λ controls the strength of the regularization.

The values of b_i that minimize this equation are (Irizarry 2019, Ch. 33.9.2):

$$\hat{b}_i(\lambda) = \frac{1}{\lambda + n_i} \sum_{u=1}^{n_i}(Y_{u,i} - \hat{\mu})$$

Where n_i is the number of ratings for movie i.

This regularization approach is applied to all aforementioned effects (movie, user, release year, and genre).

3.6 Regularization of all effects

To apply regularization to all effects together, an expanded equation that includes penalty terms for each effect is constructed:

$$\sum_{u,i}(y_{u,i} - \mu - b_i^* - b_u^* - b_r^* - b_g)^2 + \lambda(\sum_i (b_i^*)^2 + \sum_u (b_u^*)^2 + \sum_y (b_y^*)^2 + \sum_g b_g^2)$$

Where λ is the regularization parameter applied to all effects. This comprehensive regularization balances the influence of each effect and prevents any single factor from dominating the predictions.

By applying this methodology, a robust model that accurately predicts movie ratings while considering various influencing factors, accounting for temporal changes, and addressing potential overfitting issues is created. The incremental approach allows one to understand the contribution of each effect, while the time-adjustment and regularization help to improve the model's performance.

3.7 Final model

When the model is created and tuned with the train and test set out of the edx data, the RMSE is calculated by predicting the ratings in the final_holdout_test dataset with all the shown methods. This step makes sure that the model's performance is evaluated on "unknown" data, and then provides a realistic measure of its predictive accuracy.

4 Results

In this section, the methods presented in the chapter before are applied to the data step by step. After implementing each method, the RMSE is checked as if it decreases.

The data used for this is only the edx dataset. The final holdout test set is just used at the end to check the model performance and if the target is reached.

At first, the test set and the train set need to be prepared. Therefore, the data is sampled. 90% for the train set and 10% for the test set.

The most trivial model is using just the average of the ratings. The RMSE result of this can be seen in the following table.

Method	RMSE
Just the average	1.060054

4.1 Applying the effects

Movie effect

The first additional effect applied to the model is the movie effect. This is done by grouping by movieID and summarized by the mean of rating per movie relative to the average. This effect is named b_i The movie effect on the result is plotted in Figure 6 with the average deviation from the mean relative to the number of movies for which the specific deviation is valid.

The RMSE change can be seen in the table below the figure.

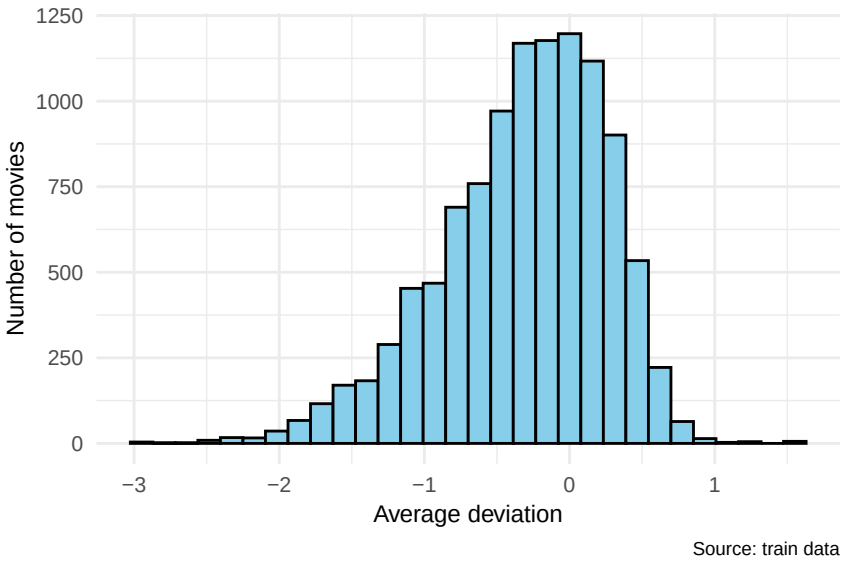

Figure 6: Movie effect

Method	RMSE
Just the average	1.0600537
Movie effect model	0.9429615

User effect

A similar approach is done for the user effect. The data is grouped by userId and summarized by the mean of rating per user relative to the average and minus b_i. This effect is named as b_u The user effect on the result is plotted in Figure 7 with the average deviation from the mean relative to the number of users for which the specific deviation is valid.

The RMSE change can be seen in the table below the figure.

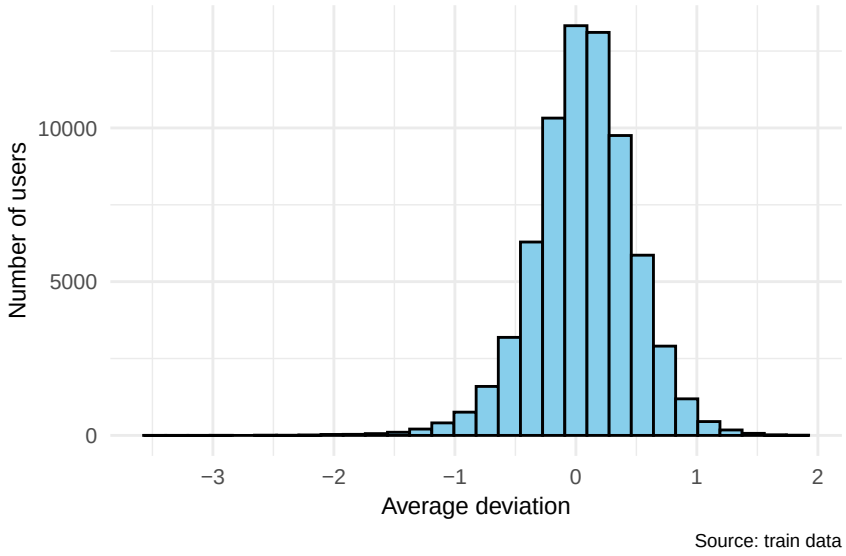

Figure 7: User effect

Method	RMSE
Just the average	1.0600537
Movie effect model	0.9429615
Movie + User effect model	0.8646844

Release year effect

For the release year effect, the data is grouped by "released", which is the Release year of the movie that was already extracted before for data exploration. The ratings are summarized by the mean of rating per release relative to the average and minus b_i and b_u. This effect is named as b_r. The average ratings per release year are again plotted in Figure 8 (left) but with a generalized additive model (GAM) smoothed curve for a better presentation of how, in general, the movie ratings by release year change. The release year effect on the result is plotted in Figure 8 (right) with the average deviation from the mean relative to the number of years for which the specific deviation is valid.

The RMSE change can be seen in the table below the figure.

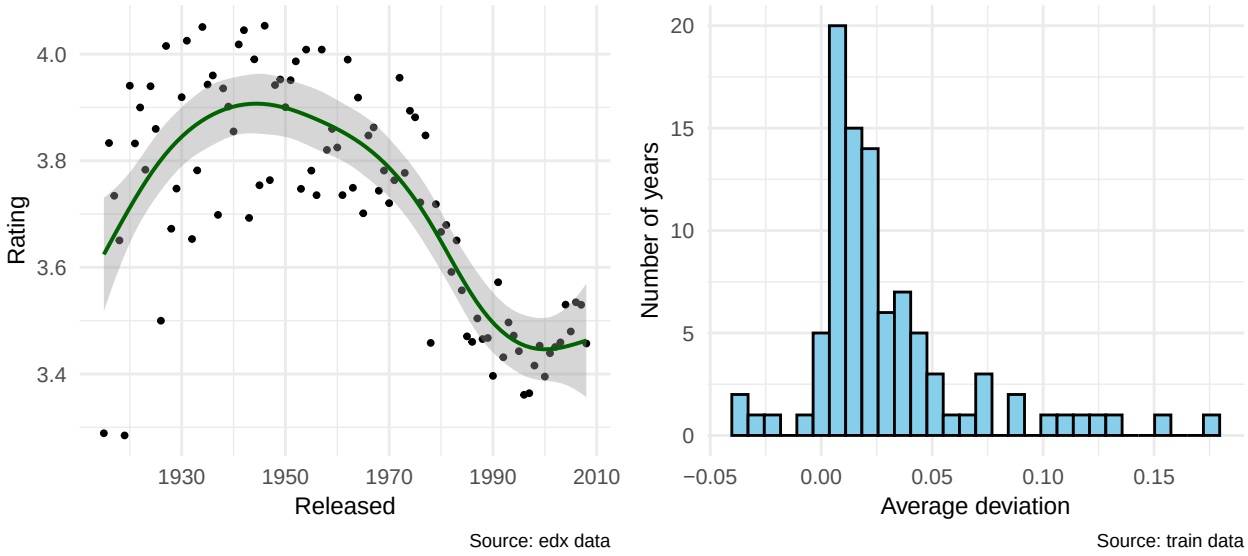

Figure 8: Release year - Average rating (left) and Release year effect (right)

Method	RMSE
Just the average	1.0600537
Movie effect model	0.9429615
Movie + User effect model	0.8646844
Movie + User effect Model + Release year effect	0.8643302

Genre effect

Next is the genre effect. The data is grouped by genres and summarized by the mean of rating per genre relative to the average and minus b_i, b_u, and b_r. This effect is named as b_g The genre effect on the result is plotted in Figure 9 with the average deviation from the mean relative to the number of genre combinations for which the specific deviation is valid. Here, every single genre combination is considered, not just the main genres.

The RMSE change can be seen in the table below the figure.

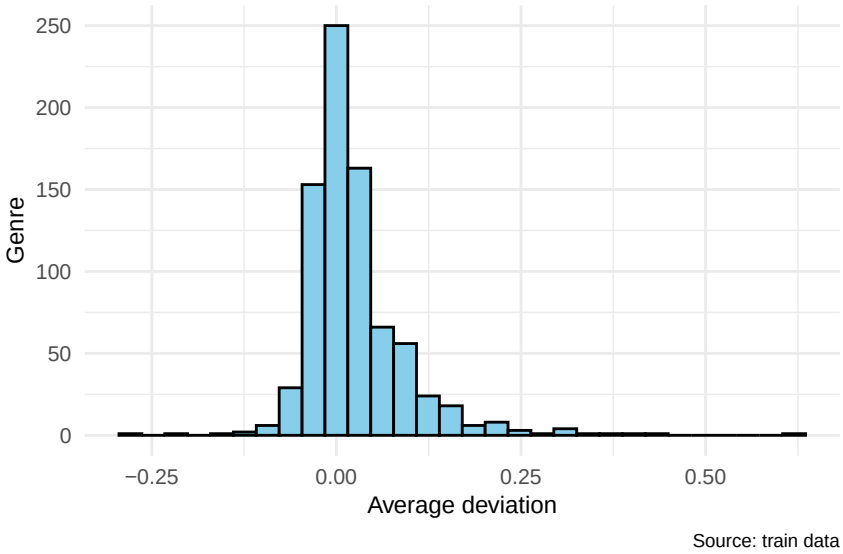

Figure 9: Genre effect

Method	RMSE
Just the average	1.0600537
Movie effect model	0.9429615
Movie + User effect model	0.8646844
Movie + User effect Model + Release year effect	0.8643302
Movie + User + Release year + Genre effect	0.8640802

4.2 Time of rating

The next effect for implementation is the time of rating. Therefore, due to having around 8.1 million ratings in the train set, a fit curve was created. For data reduction purposes, week numbers were implemented in the data, and then the average rating per week was calculated. Again, a generalized additive model (GAM) was used to create a smooth curve for the data. The weekly averaged data points and the fit curve can be seen in Figure 10 (compare with Figure 5).

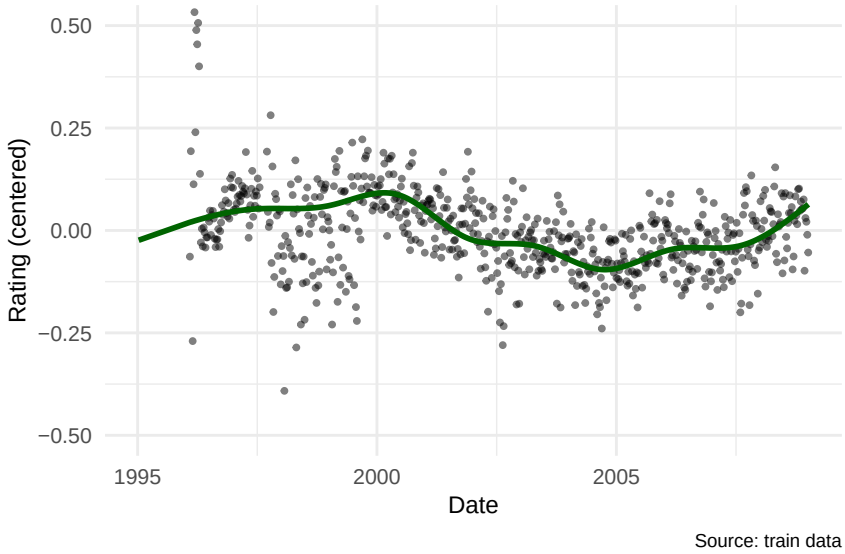

Figure 10: Fitted GAM curve with weekly averaged data points

After that, it was possible to remove the "time of rating" effect from the other effects (movie, user, release year, genre). The corrected effect deviations can be seen in Figure 11 and are designated as "time adjusted".

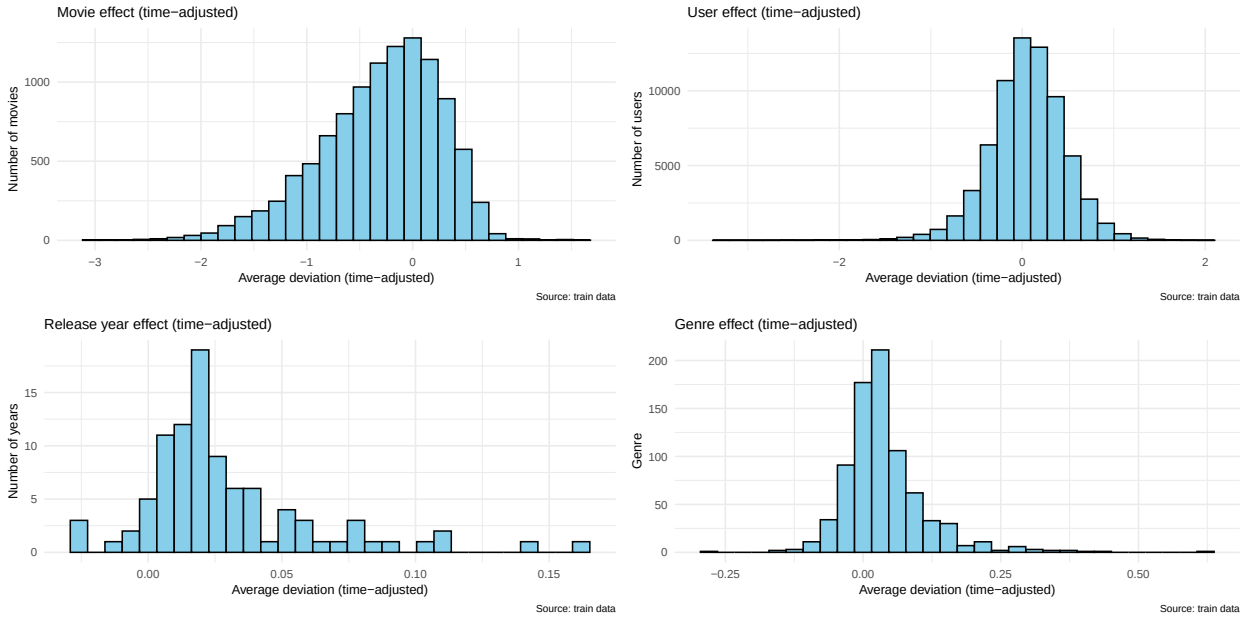

Figure 11: All effects time adjusted

During RMSE test of the time-adjusted models, it could be found out that the time-adjustment leads to better RMSE when applied to the movie, user, and release year effect. If also applied to genre effect, it is not beneficial for RMSE. So, for the following RMSE calculation and further calculations, the genre effect is not time-adjusted. The RMSE reduction for the "time-of-rating" effect is very small but nevertheless stays in the model in the rest of this project.

The RMSE change can be seen in the table below.

Remark: At the beginning of the calculations, a seed value of 1 is defined to get repeatable results. Changing this value would result in different train and test sets and could possibly lead to an RMSE increase when adding the time adjustment.

Method	RMSE
Just the average	1.0600537
Movie effect model	0.9429615
Movie + User effect model	0.8646844
Movie + User effect Model + Release year effect	0.8643302
Movie + User + Release year + Genre effect	0.8640802
Time-adjusted (Movie, User, Release year) + Genre effect	0.8640028

In Figure 12, the individual time influence on the different effects can be seen. For movie, user, and release year effects, there are clear deviations in when the rating was done. Also, the genre has some effect on it, but it is not uniform over the genres. This could explain why there is an RMSE improvement only for movie, user, and release year effects and not for genre effect because there is no clear direction for all genres. Decreasing RMSE by time adjustment for genre effect could possibly be realized by time adjustment individually per genre. This was not done in this project.

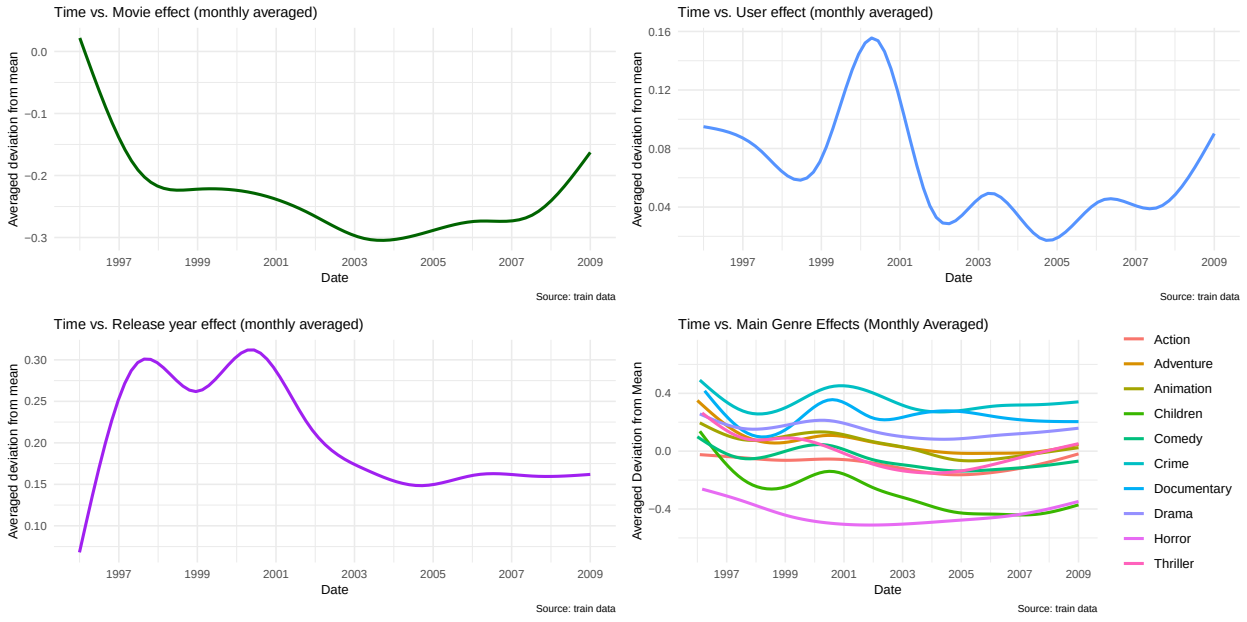

Figure 12: Time vs. all effects (monthly averaged)

4.3 Range correction

The next topic to take a further look at is range correction. Figure 13 shows the distribution of the predicted ratings. It can be seen that a certain number of values lie out of range (lower than 0.5 = 107, higher than 5 = 2152).

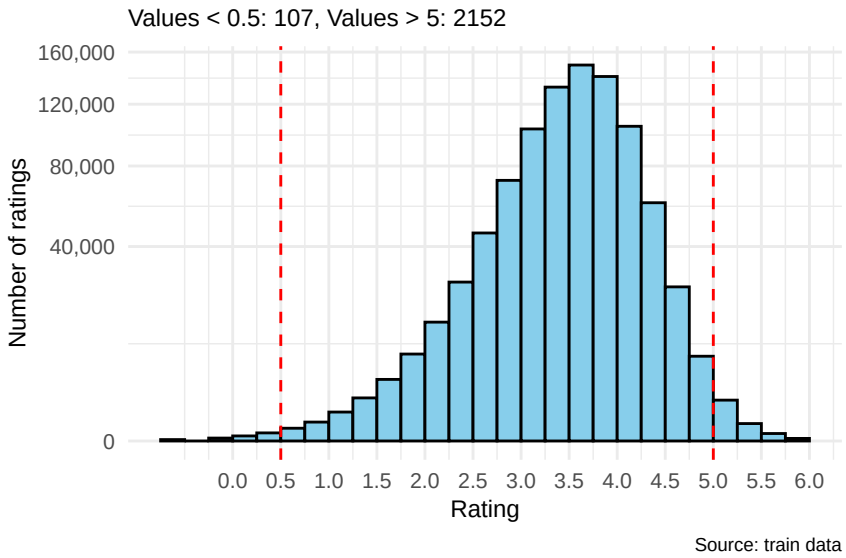

Figure 13: Distribution of the predicted ratings

Due to that, in the edx dataset, it can be seen that values below 0.5 and higher than 5 are not valid; for further decrease of RMSE, the prediction out of range can be corrected either to 0.5 or 5. The outcome (RMSE decrease) can be seen in the following table ("corr").

Method	RMSE
Just the average	1.0600537
Movie effect model	0.9429615
Movie + User effect model	0.8646844
Movie + User effect Model + Release year effect	0.8643302
Movie + User + Release year + Genre effect	0.8640802
Time-adjusted (Movie, User, Release year) + Genre effect	0.8640028
Time-adjusted (Movie, User, Release year) + Genre effect + corr	0.8637955

4.4 Regularization

In the following, regularization for penalizing large effect values is applied to all effects to prevent overfitting.

1. without the time-adjusted model
2. with the time-adjusted model

The lambda value is a variable, and a range from 4.2 to 5.0 is selected and applied in steps of 0.1 in a loop. With this procedure, the best value of lambda can be found, and it can also be seen if, after regularization, the time-adjustment still leads to a better RMSE.

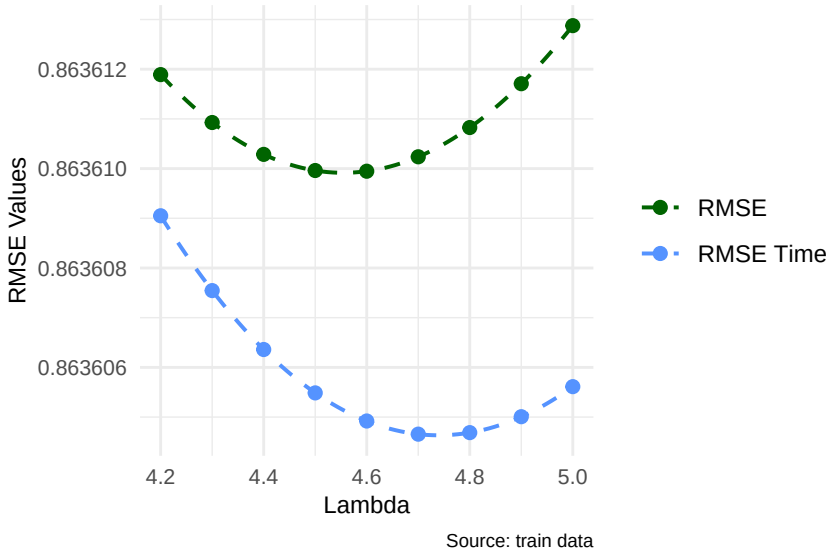

Figure 14: RMSE in regularization loops

It was found out that the time-adjusted model with regularization at a lambda value of 4.7 performs best. Again, the differences are very small. Nevertheless, this will be used, and the values out of range will again be corrected. Then the RMSE table looks the following:

Method	RMSE
Just the average	1.0600537
Movie effect model	0.9429615
Movie + User effect model	0.8646844
Movie + User effect Model + Release year effect	0.8643302
Movie + User + Release year + Genre effect	0.8640802
Time-adjusted (Movie, User, Release year) + Genre effect	0.8640028
Time-adjusted (Movie, User, Release year) + Genre effect + corr	0.8637955
Regularized, Time-adjusted (Movie, User, Release year) + Genre + corr	0.8634787

4.5 Final holdout test

Now, the model is tested in the final holdout test dataset. The best model is the "Regularized, Time-adjusted (Movie, User, Release year) + Genre" model. The final holdout dataset was tested with that model with and without correction. Therefore, the release year and the week numbers need to be mutated in the dataset. Then, all the effects were added. The results can be seen in the following table:

Method	RMSE
Regularized, Time-adjusted (Movie, User, Release year) + Genre (final test)	0.8646216
Regularized, Time-adjusted (Movie, User, Release year) + Genre + corr (final test)	0.8645075

The model does not perform as well as in the test set out of the edx dataset, but the RMSE is still in an acceptable range. Figure 15 shows the outcome of the final holdout test from the predicted "Regularized, Time-adjusted (Movie, User, Release year) + Genre + corr" model as the absolute deviations of the prediction from the real ratings in the data. With that, the average deviation of the outcome is visualized for a better evaluation of the result. It can be seen that 71% of the predictions lie between ± 1 RMSE with its final value of 0.8645075.

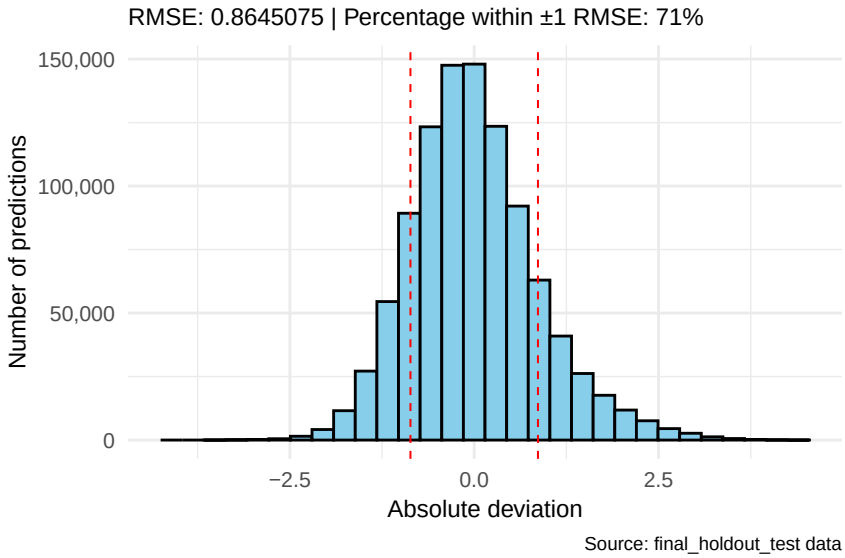

Figure 15: Distribution of prediction deviations

Both RMSE results (with and without correction) are below 0.86490, so the given target was reached.

5 Conclusion

This project report described the development of a movie recommendation system based on the MovieLens 10M dataset. The main goal was to predict the user ratings based on various effects, like move-specific, user-specific, release year, genre, and time-dependent influences, as accurately as possible. The best model with the lowest RMSE used time adjustments, movie, user, release year, genre effects, and regularization and achieved an RMSE of 0.8634787 in the test data and an RMSE of 0.86451 in the final holdout test set. Thus, it met the predefined target of below 0.86490.

However, some limitations were found during the analysis. The results are highly sensitive to both the site of train and test samples and the seed chosen for being reproducible. Changing these parameters could lead to different outcomes and could potentially have an impact on the descisions regarding which effects are to be chosen to be included in the model. Additionally, a time-of-rating effect was observed, but the impact on the rating predictions was very small.

Ratings are limited to increments of 0.5 within a range from 0.5 to 5. Therefore, alternative modeling techniques specifically for discrete ratings could potentially improve the accuracy. Matrix factorization methods, which are widely used in recommendation systems, could be more effective due to their ability to capture user-item interactions and thus, possibly provide better predictions (Koren, Bell, and Volinsky 2009). Also, applying cross-validation could help to better assess the model robustness by partitioning the dataset into multiple subsets and checking the performance of the model across these subsets. Although cross-validation was not used in this project because of computational constraints, it is recommended to be implemented in future analyses.

Usage of Artificial Intelligence

Perplexity AI – Latex and R Markdown syntax assistance

- **Website:** https://www.perplexity.ai
- **Used in:** Latex header creation for R Markdown document, code chunk headers, mathematical formula syntax in Methods section
- **Purpose:** Assistance in generating Latex headers for the R Markdown document structure, creating code chunk headers, and creating accurate mathematical syntax in the Methods section.

Perplexity AI – R code assistance

- **Website:** https://www.perplexity.ai
- **Used in:** Methods and Results sections (ggplot syntax correction, GAM fit curve implementation for the time-corrected model)
- **Purpose:** Syntax correction and debugging support for ggplot visualizations, assistance in implementing Generalized Additive Model (GAM) fit curves for time-adjusted models.

Perplexity AI – Content crosscheck and writing tips

- **Website:** https://www.perplexity.ai
- **Used in:** Introduction and Conclusion sections
- **Purpose:** Crosschecking content accuracy and logical structure; providing suggestions for improvements regarding clarity, precision, and scientific writing style.

Grammarly – Grammar and spell check

- **Website:** https://www.grammarly.com
- **Used in:** Entire report
- **Purpose:** Grammar and spelling correction.

Grok from X - R code commenting

- **Website:** https://x.com
- **Used in:** R Code
- **Purpose:** Assisting in generating code comments.

References

Clark, Michael. n.d. *Introduction | Generalized Additive Models*. Accessed March 11, 2025. https://m-clark.github.io/generalized-additive-models/introduction.html.

GroupLens. 2009. "MovieLens 10M Dataset." https://grouplens.org/datasets/movielens/10m/.

Irizarry, Rafael A. 2019. *Introduction to Data Science*.

Koren, Yehuda, Robert Bell, and Chris Volinsky. 2009. "Matrix Factorization Techniques for Recommender Systems." *Computer* 42 (8): 30–37. https://doi.org/10.1109/MC.2009.263.

"Regularization in Machine Learning." 2025. *GeeksforGeeks*. https://www.geeksforgeeks.org/regularization-in-machine-learning/.