MovieLens Recommendation System Project

HarvardX PH125.9x - Data Science Capstone 1

Jasmine Zhang

01 June, 2025

Contents

1 Introduction

Recommendation systems have become foundational tools in the digital economy, transforming how users interact with vast online content and services. By leveraging user behavior and historical data, these systems intelligently suggest products, movies, music, and other items tailored to individual preferences. Their applications span a wide range of industries, including entertainment, e-commerce, education, and healthcare, playing a critical role in improving user engagement and business outcomes.

One of the most influential milestones in the development of recommendation algorithms was the Netflix Prize, launched in 2006. This open competition challenged data scientists worldwide to improve the company's movie recommendation accuracy, bringing renewed focus to collaborative filtering and large-scale machine learning approaches. As a result, open datasets such as MovieLens have become central to research and practical experimentation in this field.

This project explores the construction of a movie recommendation system utilizing the MovieLens 10M dataset, which contains 10 million ratings applied by over 70,000 users to 10,000 movies. **The primary objective is to build a predictive model capable of estimating user ratings with high accuracy, specifically targeting a root mean square error (RMSE) below 0.86490 on a hold-out validation set.**

The workflow for this project consists of several stages: data preparation and cleaning, exploratory analysis, development and tuning of baseline and regularized models, and evaluation of model performance using the prescribed RMSE metric. All analysis and reporting are performed within an R Markdown environment.

2 Exploratory Data Analysis

2.1 Dataset Summary

The `edx` dataset is a data frame consisting of **9,000,055** rows. It captures movie ratings submitted by **69,878 unique users** for **10,677 unique movies**, spanning **797** distinct genres. If every user had rated every movie, the dataset would contain approximately **746 million** ratings (calculated as $69,878 \times 10,677 = 746,155,406$). However, the actual number of ratings is far lower, indicating that the dataset is highly sparse—most users have rated only a small subset of the available movies.

Table 1: MovieLens edx Dataset Summary

n_users	n_movies	n_ratings	n_genres
69878	10677	9000055	797

2.2 Most Rated and Highest Rated Movies

To further explore the dataset, we examined both the movies with the highest number of ratings and those with the highest average ratings (among movies with at least 1,000 ratings).

2.2.1 Top 10 Most Rated Movies

The table 2 below lists the ten movies that have received the highest number of ratings in the `edx` dataset. These are primarily popular, mainstream films with broad viewer engagement.

- **Pulp Fiction** received the highest number of ratings (**31,362**).
- Other frequently rated movies include **Forrest Gump** (*31,079*), **Silence of the Lambs** (*30,382*), and **Jurassic Park** (*29,360*).
- This list is dominated by well-known blockbusters and classics, reflecting their widespread popularity.

Table 2: Top 10 Most Rated Movies (by Number of Ratings)

title	count
Pulp Fiction	31362
Forrest Gump	31079
Silence of the Lambs, The	30382
Jurassic Park	29360
Shawshank Redemption, The	28015
Braveheart	26212
Fugitive, The	26020
Terminator 2: Judgment Day	25984
Star Wars: Episode IV - A New Hope (a.k.a. Star Wars)	25672
Batman	24585

2.2.2 Top 10 Highest Rated Movies (Minimum 1,000 Ratings)

We also identified the top ten movies with the highest average ratings in table 3. An important observation is that the most rated movies (well-known blockbusters) and the highest rated movies (critically acclaimed classics) **do not overlap much**. This distinction shows the difference between popularity and audience appreciation.

For this reason, in the modeling section, we will incorporate **both** the number of ratings (*rating count*) and the average rating (*mean rating*) into our methods and evaluation. This approach ensures that the recommendation system accounts for movies that are widely seen as well as those that are highly valued by viewers.

Table 3: Top 10 Highest Rated Movies (by Average Rating, min. 1000 ratings)

title	average_rating
Shawshank Redemption, The	4.455131
Godfather, The	4.415366
Usual Suspects, The	4.365854
Schindler's List	4.363493
Casablanca	4.320424
Rear Window	4.318651
Sunset Blvd. (a.k.a. Sunset Boulevard)	4.315880
Third Man, The	4.311426
Double Indemnity	4.310817
Paths of Glory	4.308721

2.3 Distribution of Movie Ratings ($rating)

The overall average rating in the `edx` dataset is **3.51**, with individual movie averages ranging from as low as 0.5 up to the maximum rating of 5. As shown in Figure 1, the distribution of average ratings across all movies is approximately bell-shaped, centered around 3 to 3.5. Most movies have an average rating between 2.5 and 4, indicating a tendency toward moderate to positive sentiment. This overall distribution serves as the baseline for the naive prediction model, which simply uses the global mean rating as the estimate for all movies.

```
mean(edx$rating)
```

```
## [1] 3.512465
```

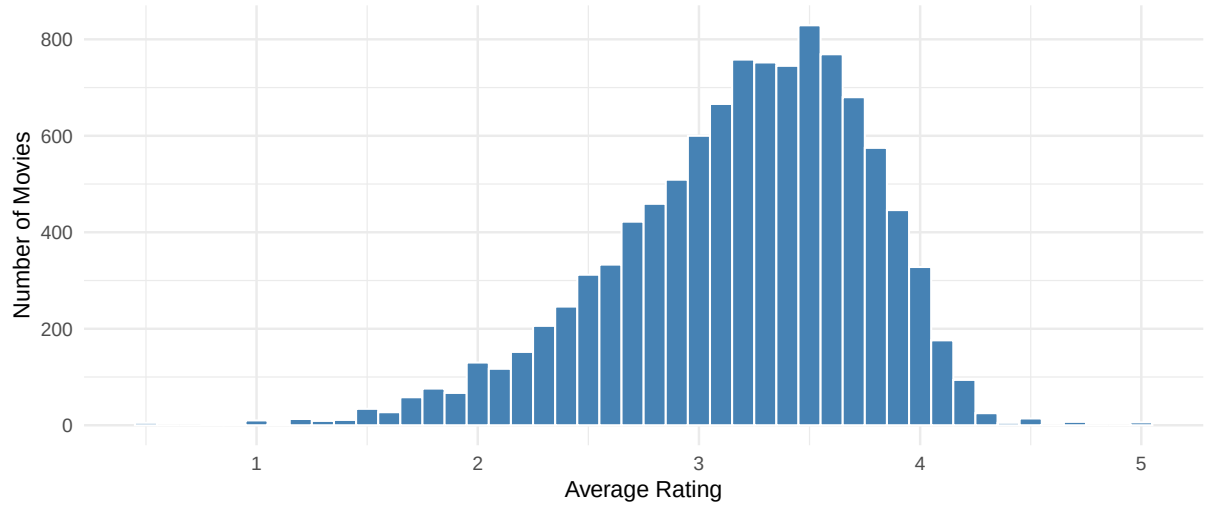


Figure 1: Distribution of Movie Ratings

2.4 Rating Density by Movie ($movieId)

Figure 2 shows the distribution of the number of ratings received by each movie, plotted on a logarithmic scale. There is considerable variability in rating density: while a small subset of popular titles have received thousands or even tens of thousands of ratings, the majority of movies in the dataset have relatively few ratings—some only a handful, or even just one. This pronounced skewness highlights the **movie effect**: certain movies attract far more user engagement than others.

Given this substantial variation, it is important to adjust for movie-specific effects in subsequent modeling to avoid biases driven by popularity alone. In particular, **movies with very few ratings may exhibit extreme average values simply due to small sample size**. To address this, we will incorporate **regularization techniques** later in the modeling process, which will help to stabilize effect estimates and prevent overfitting to movies with limited data.

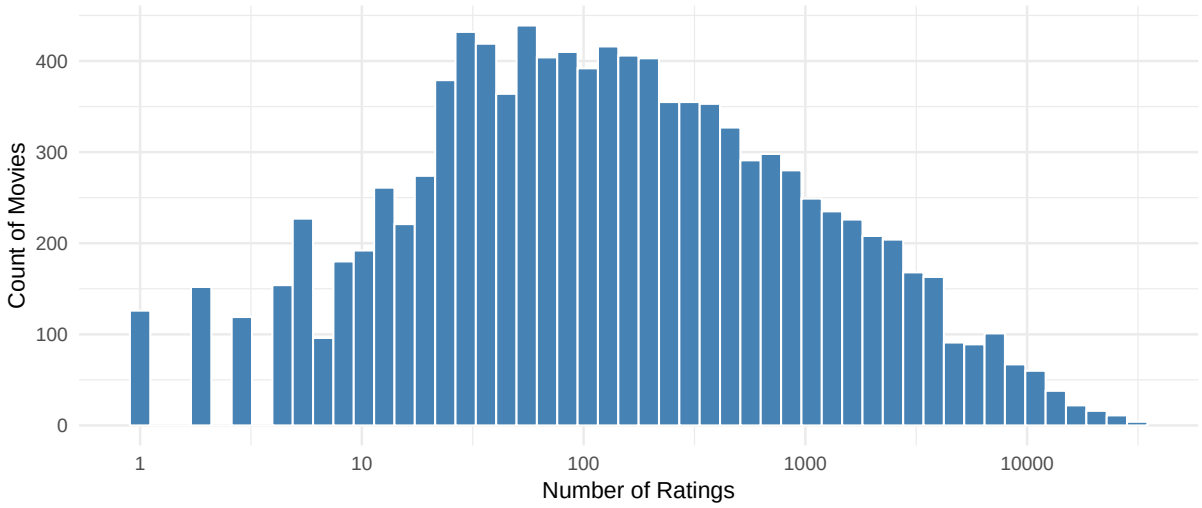


Figure 2: Rating Density by Movie

2.5 Average Rating per User ($userId)

Figure 3 shows the distribution of average ratings assigned by each user in the `edx` dataset. The histogram shows that, while the majority of users tend to give moderate to moderately high ratings (with most users averaging between 3.0 and 4.0), there is still notable variability in individual user preferences. Some users consistently rate movies much higher or lower than the overall average.

This variation demonstrates the **user effect**: each individual has a distinct baseline for how they rate movies, independent of the movie itself. For example, some users may be generally more generous or critical in their assessments. Accounting for these systematic user-specific differences is important for building an accurate recommendation system. So, **the modeling process will add user effects to adjust for these personal rating tendencies**.

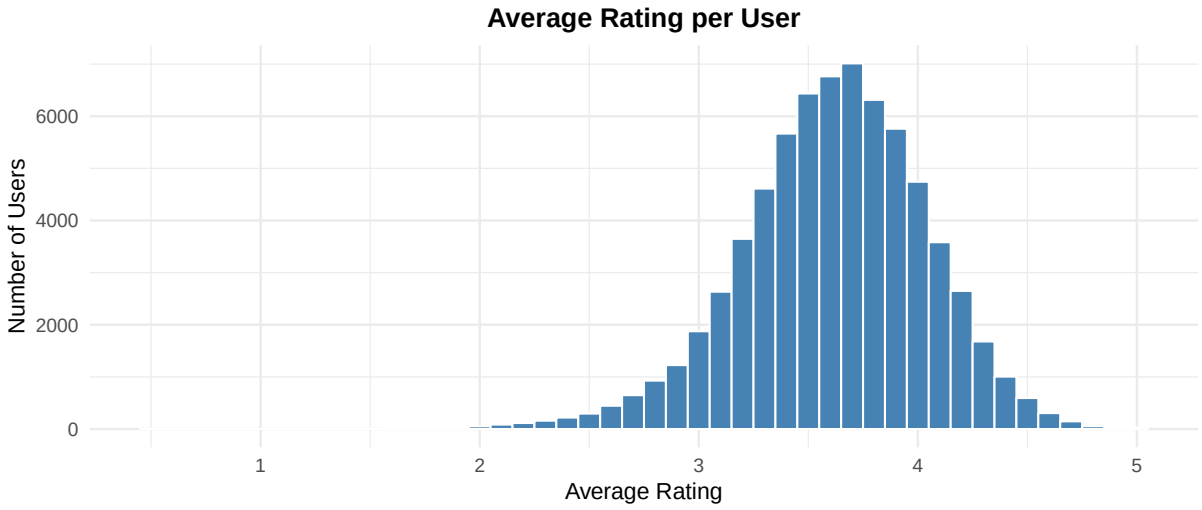


Figure 3: Average Rating per User

2.6 Rating Distribution by Movie Genre ($genres)

Figure 4 shows the average movie rating by genre combined, with 95% confidence intervals displayed for each genre group that has at least 100,000 ratings. The plot demonstrates clear variation in average ratings across genres. For example, genre combinations such as **Drama|War** and **Crime|Drama** are associated with the highest average ratings, while genres like **Comedy** and **Comedy|Romance** tend to receive lower average ratings from users.

These differences reflect some genre effects. Certain genres or genre combinations are consistently rated more favorably or critically by the user base. To improve predictive accuracy, the modeling process will incorporate a **genre effect**, adjusting for the average bias associated with genres.

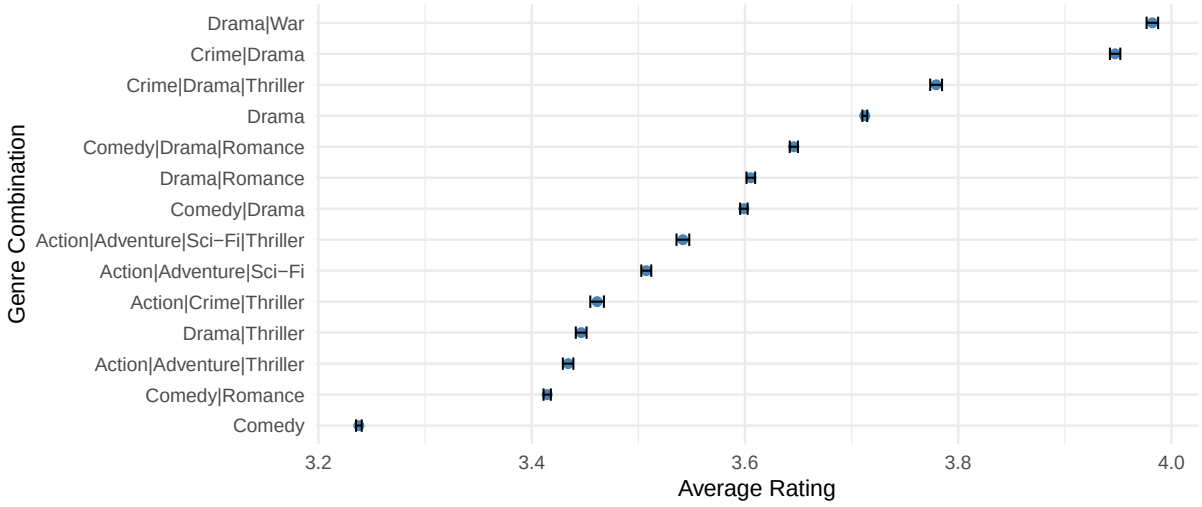


Figure 4: Average Rating by Genre Combination

2.7 Release Year (from $title)

Figure 5 displays how many ratings movies received based on their release year. The trend reveals that very few ratings are associated with movies released before the 1960s. The number of ratings then gradually increases, reaching a dramatic peak around the early 1990s. After this peak, the number of ratings per release year declines sharply.

This pattern highlights a clear release year effect: movies released in more recent decades, particularly the 1980s and 1990s, receive a disproportionately large share of user ratings compared to older or newer films. The sharp peak may reflect both the popularity of movies from this era and user engagement trends within the dataset.

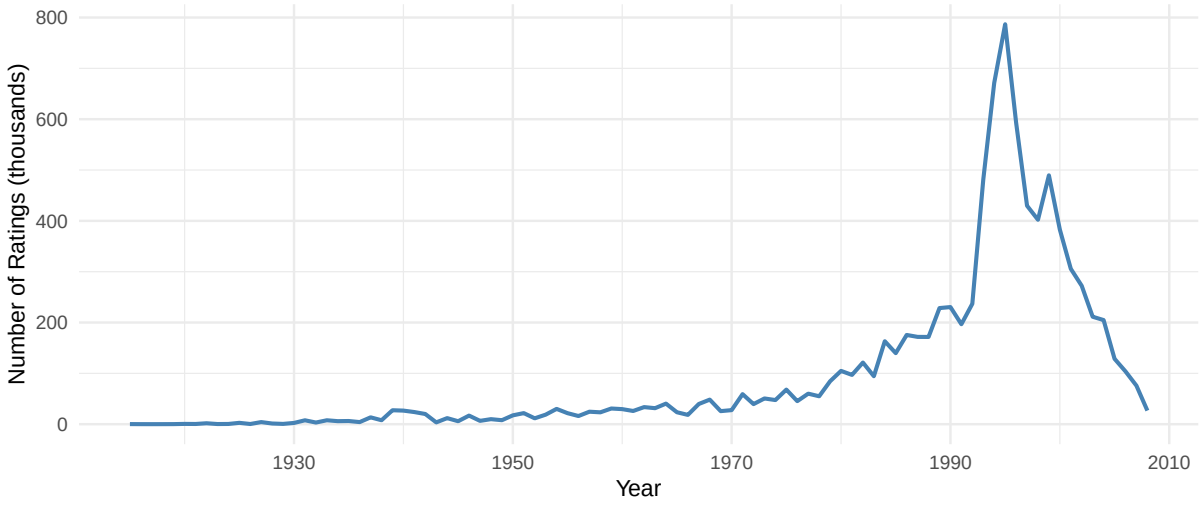


Figure 5: Number of Ratings by Release Year

Figure 6 builds on the insights from Figure 5 by examining how the average rating of movies varies with their release year. The plot shows that average movie ratings are higher for films released in earlier decades, peaking around the 1940s, and gradually declining for more recent releases. This release year effect suggests that older movies tend to be rated more favorably, possibly due to selection bias or evolving audience preferences. Recognizing and adjusting for this trend is important for building an unbiased recommendation system.

2.8 Date of review ($timestamp)

Figure 7 shows how the average movie rating varies by the date the review was submitted. The trend indicates that average ratings declined from the mid-1990s through the mid-2000s, reaching a low point around 2005, before slightly increasing in subsequent years.

This pattern suggests a date of review effect, where user sentiment and rating behavior change over time. Incorporating the date of review as an effect in the model helps to account for these temporal fluctuations and improves the accuracy of rating predictions.

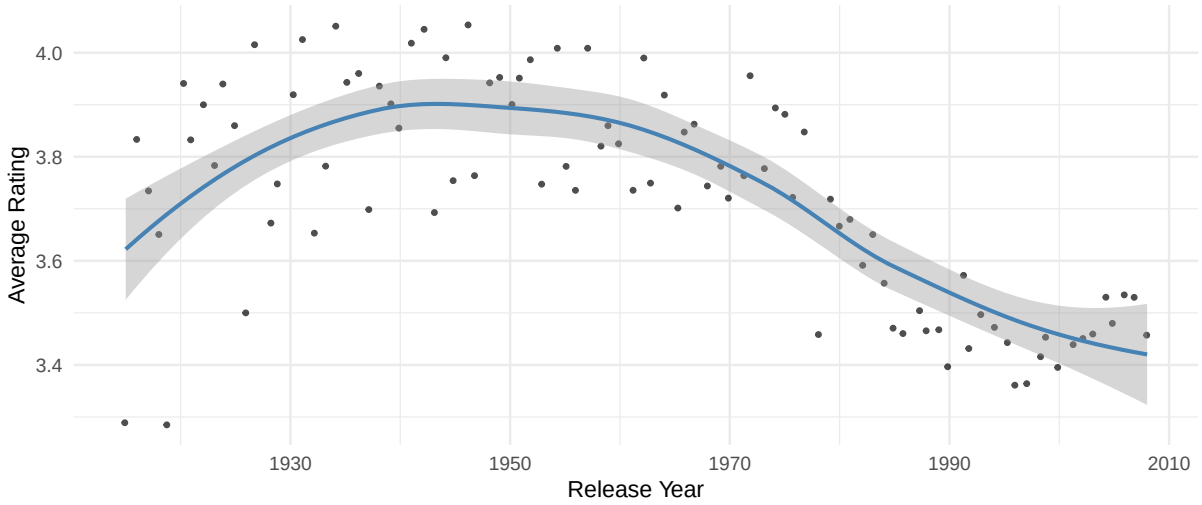


Figure 6: Average Rating by Release Year

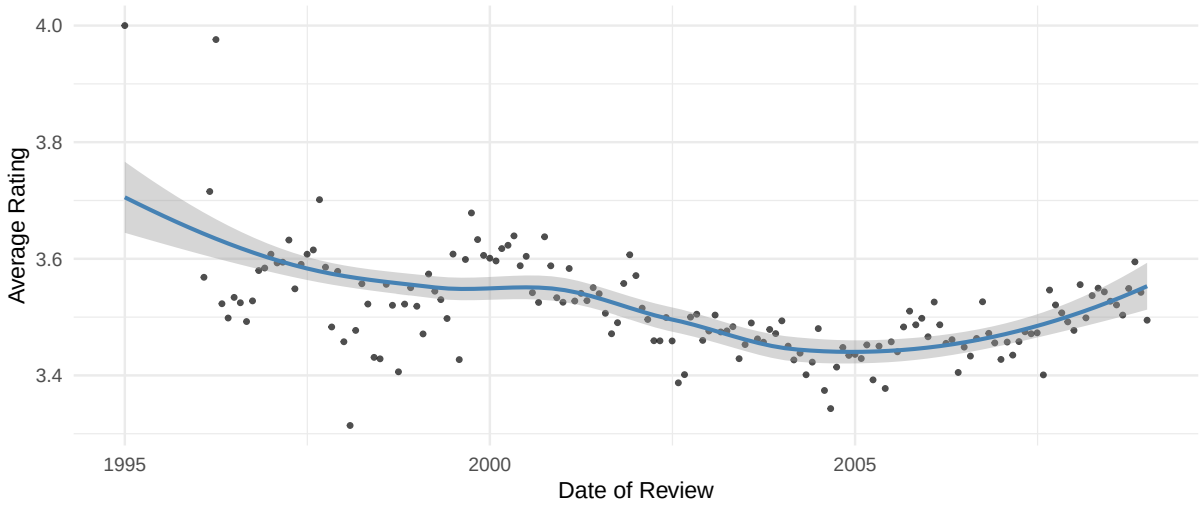


Figure 7: Average Movie Rating by Date of Review

3 Methods

3.1 Splitting the edx Dataset for Cross-Validation

To facilitate model development and parameter tuning, the `edx` dataset was partitioned into separate training and test sets. This partitioning enables the evaluation of model performance during development, while reserving the final validation set for unbiased assessment of the final model.

The `createDataPartition` function from the `caret` package was used to randomly split the `edx` data, assigning 80% of observations to the training set and the remaining 20% to the test set. To ensure the integrity of the evaluation, the test set was filtered so that it only contained **movies, users, years, genres, and review dates** present in the training set. This was achieved by applying a series of `semi_join` operations, which restrict the test set to rows with corresponding values in the training set for each key variable.

Any records in the original test split that did not meet these criteria were added back to the training set using `anti_join` and `rbind`, maximizing the data available for model training. This careful partitioning process helps prevent data leakage and ensures that the model is not tested on unseen entities, resulting in a more reliable and fair evaluation of model performance.

3.2 Model Building

To benchmark the model's performance, the project objective was set as achieving a root mean square error (RMSE) below 0.86490 on the hold-out validation set. This threshold serves as a reference point for evaluating the effectiveness of different modeling strategies.

Method	RMSE	Difference
Project objective	0.86490	-

3.2.1 Naive model

The simplest predictive approach uses the overall average rating in the training set as the predicted value for all observations. The naive model is expressed as:

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

where:

- μ = overall average rating
- $\epsilon_{u,i}$ = random error term (residuals)

3.2.2 Developing the algorithm

For brevity, we do not present the intermediate steps for each effect in this section. The modeling process for each factor (movie, user, genre, release year, review date) follows the same approach as in the final model. Therefore, we focus here on presenting the final, comprehensive model only.

The final model in this analysis predicts the rating $Y_{u,i}$ that user u assigns to movie i by adding together the overall average rating and separate effects for the movie, user, genre, release year, and review date, plus an error term. The final model is expressed as:

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

where:

- μ = overall average rating

- b_i = movie effect
- b_u = user effect
- b_g = genre effect
- b_y = release year effect
- b_r = review date effect
- $\epsilon_{u,i}$ = random error term (residuals)

Each effect is estimated from the training data by sequentially removing the influence of the effects previously calculated. This stepwise procedure ensures that each component captures the unique contribution of its corresponding factor, independent of the others. For example:

$$\hat{b}_i = \text{mean}(Y_{u,i} - \mu)$$
$$\hat{b}_u = \text{mean}(Y_{u,i} - \mu - \hat{b}_i)$$
$$\hat{b}_g = \text{mean}(Y_{u,i} - \mu - \hat{b}_i - \hat{b}_u)$$
$$\hat{b}_y = \text{mean}(Y_{u,i} - \mu - \hat{b}_i - \hat{b}_u - \hat{b}_g)$$
$$\hat{b}_r = \text{mean}(Y_{u,i} - \mu - \hat{b}_i - \hat{b}_u - \hat{b}_g - \hat{b}_y)$$

3.3 Regularization

Exploratory analysis revealed that not only are ratings influenced by factors such as movie, user, genre, release year, and review date, but the number of ratings for each group can vary dramatically. For example, some movies and genres receive far fewer ratings than others, while some users are more active in rating than others. Likewise, the number of ratings changes across different release years and review dates. As a result, effect estimates based on smaller groups are more uncertain and prone to overfitting.

To address this, regularization is employed to penalize large effect estimates that arise from small sample sizes. The penalty term, λ, is a tuning parameter determined through cross-validation on the edx dataset. This approach ensures that the estimated effects are shrunk towards zero when data is sparse, thereby improving the model's robustness and generalization.

The least squares estimate for the regularized effect, for example the movie effect b_i, is given by:

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

where n_i is the number of ratings for movie i. When n_i is large, the impact of λ is minimal and the estimate is close to the average residual. When n_i is small, the effect of λ is more pronounced, causing the estimate to shrink towards zero and thus reducing the risk of overfitting due to limited data.

In this study, the regularization model was applied to all effects considered in the analysis—movie, user, genre, release year, and review date. A range of values for the regularization parameter λ (from 3 to 6, in increments of 0.1) was tested to find the value that minimized the RMSE on the internal test set. As with previous steps, all model selection and tuning were performed using only the training and test splits of the edx dataset to prevent overfitting and ensure fair evaluation on the final validation set.

$$\frac{1}{N}\sum_{u,i}(y_{u,i}-\mu-b_i-b_u-b_g-b_y-b_r)^2+\lambda\left(\sum_i b_i^2+\sum_u b_u^2+\sum_g b_g^2+\sum_y b_y^2+\sum_r b_r^2\right)$$

3.4 Final Model Training

In the final step, the entire `edx` dataset was used to re-estimate all model parameters using the optimal value of the regularization parameter (λ) identified during cross-validation. This approach ensures that all available data is leveraged to produce the most accurate and stable estimates of the effects for movie, user, genre, release year, and review date.

To determine the best value of λ, the RMSE was computed for a range of candidate values. As illustrated in Figure 8, RMSE initially decreases as λ increases, reaches a minimum near $\lambda = 5$, and then begins to rise again. This "U-shaped" pattern demonstrates the trade-off involved in regularization: too little regularization can lead to overfitting, while too much regularization can underfit the data. The value of λ that minimizes the RMSE is selected as the optimal regularization parameter.

The final model predictions were then generated for the hold-out validation set by summing the global mean and all estimated effects for each record. The model's performance was evaluated by calculating the RMSE between the predicted ratings and the true ratings in the validation set. This procedure provides an unbiased assessment of the model's predictive accuracy on previously unseen data.

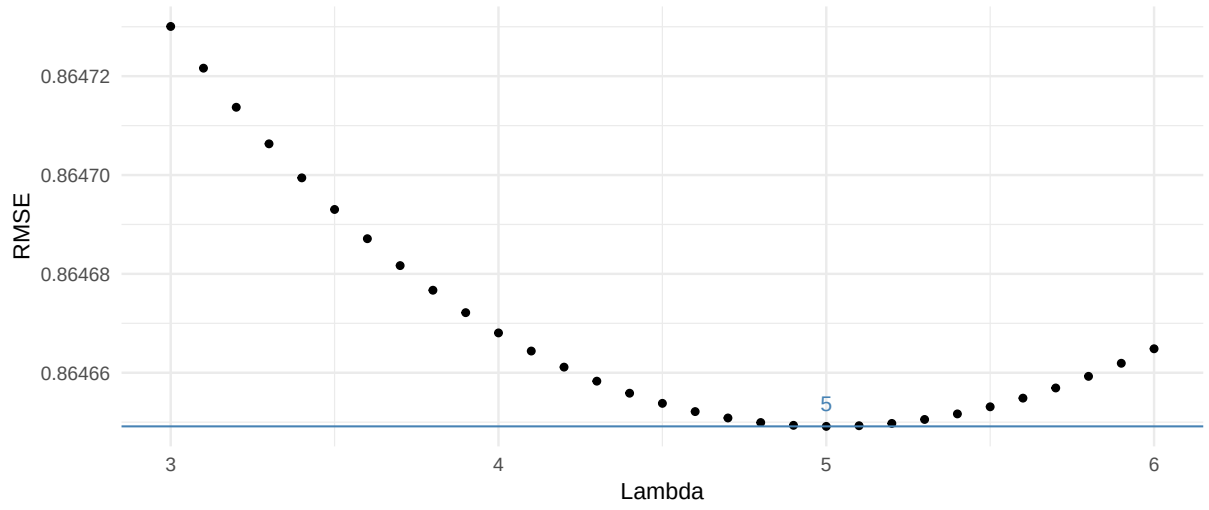


Figure 8: RMSE by Lambda

4 Summarize and Visualize Results

4.1 Naive model

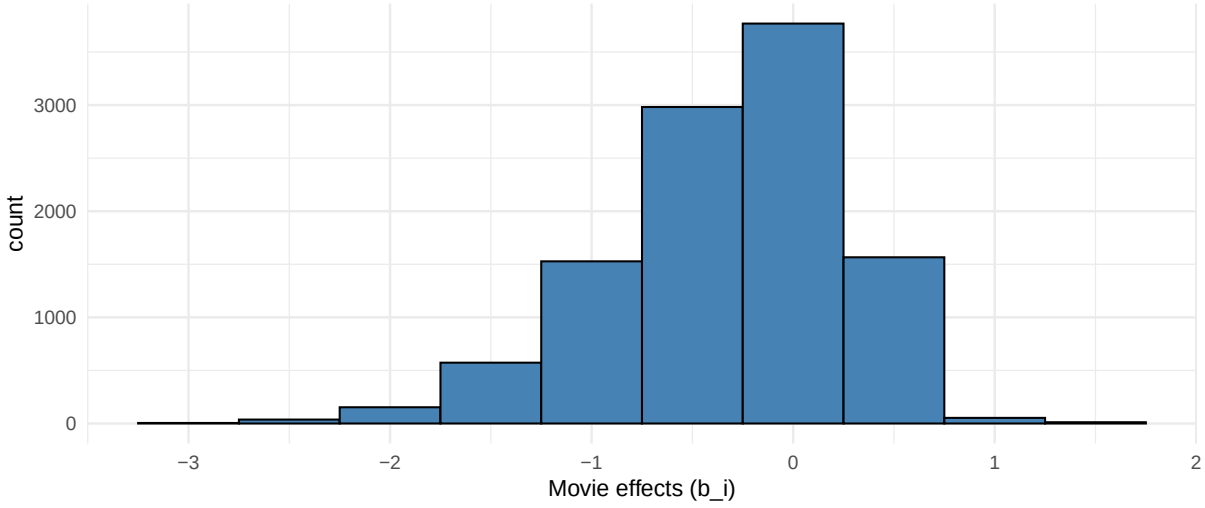
The table below summarizes RMSE for the naive model compared to the project's target objective. Predicting the average rating from the training set for every entry in the test set resulted in an RMSE of 1.06, which is substantially higher than the project's goal of 0.86490. An RMSE of this magnitude means that, on average, predicted ratings are more than one full star away from the actual rating. This level of error is unacceptably high for a movie recommendation system.

Table 4: Comparison of RMSE

Method	RMSE	Difference
Project objective	0.86490	-
Naive model	1.0599	0.195

4.2 Accounting for movie effect

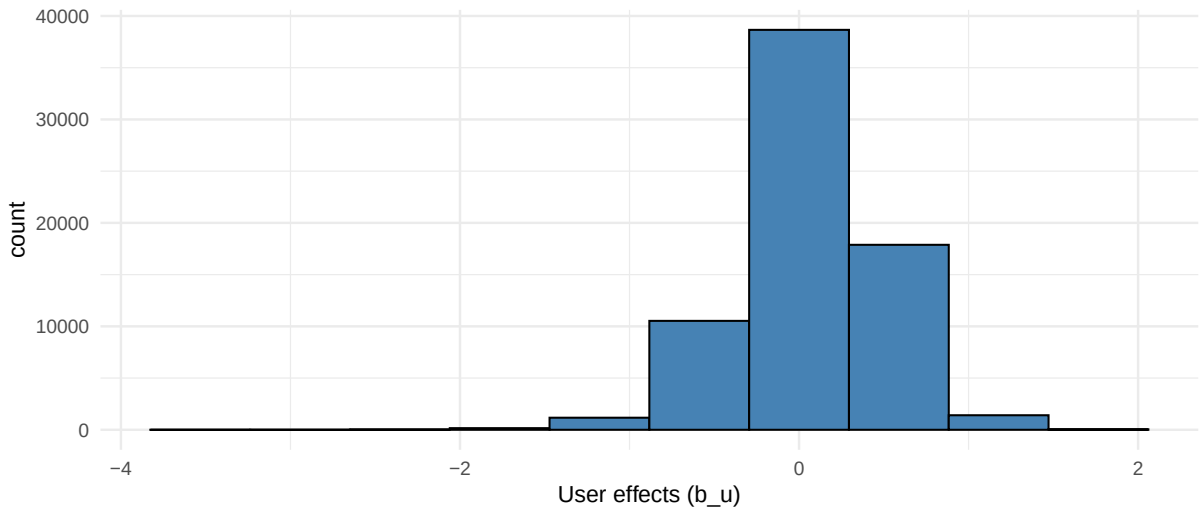
This histogram shows the distribution of estimated movie effects (b_i) after removing the overall average rating. Most movies cluster near zero, meaning they don't deviate much from the average. However, there are some with clearly positive or negative values, meaning that certain movies consistently receive higher or lower ratings than average, justifying the need to include movie-specific effects in the model.



4.3 Accounting for user effect

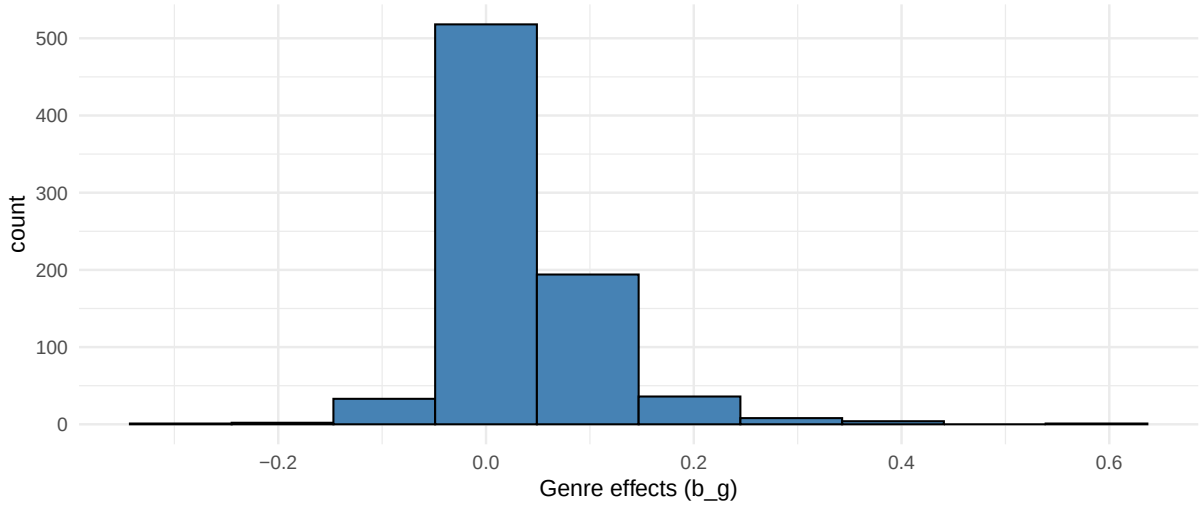
This histogram displays the distribution of estimated user effects (b_u) after accounting for the global average and movie effects. Most user effects are close to zero, indicating that the majority of users

do not consistently rate much higher or lower than average. However, there are some users with more extreme positive or negative values, which shows that a small number of users have persistent rating tendencies that differ from the norm.



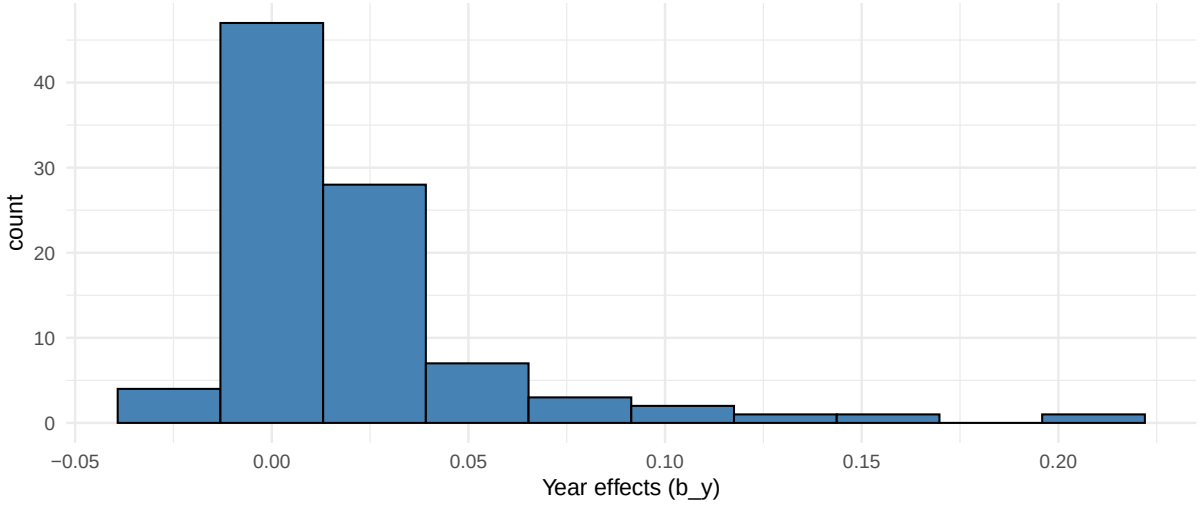
4.4 Accounting for genre effect

Figure 11 shows the distribution of estimated genre effects (b_g) after accounting for other variables in the model. Most genre effects are centered close to zero, indicating that the average rating does not differ greatly across most genres. However, there are a few genres with notably higher or lower effects, suggesting that some genres are consistently rated above or below the overall average. Including genre effects helps the model capture these systematic preferences.



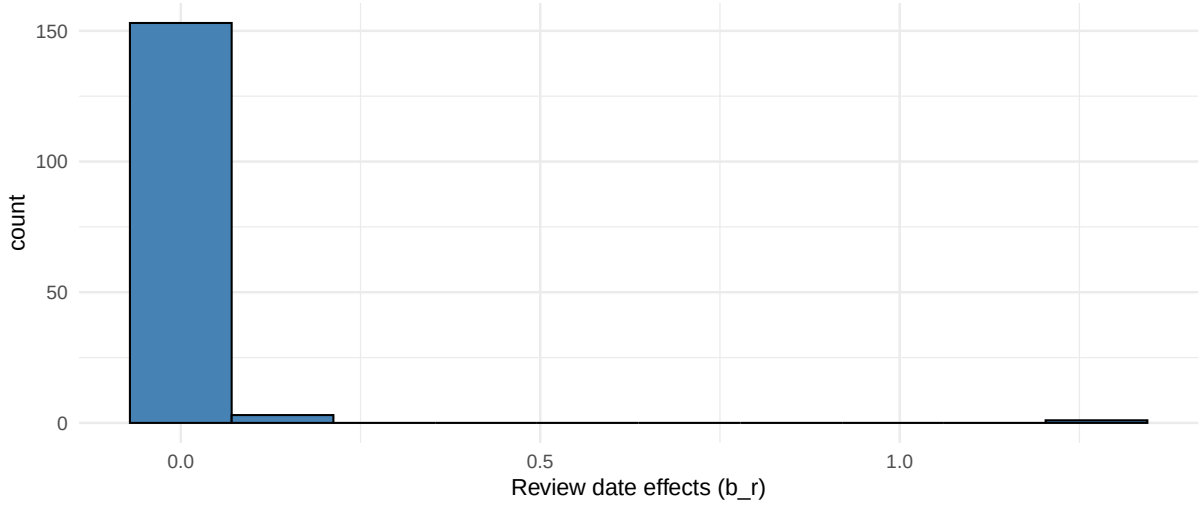
4.5 Accounting for release year effect

Figrue 12 shows the distribution of estimated release year effects (b_y) in the model. Most release year effects are clustered around zero, indicating that for most years, the average rating does not differ much from the overall mean. However, a few years have slightly higher positive effects, suggesting that movies from certain years tend to receive higher ratings on average. Including release year effects allows the model to adjust for temporal trends in movie ratings.



4.6 Accounting for review date effect

Figure 13 shows the distribution of estimated review date effects (b_r) in the model. Most review date effects are very close to zero, indicating that the timing of reviews generally does not have a large influence on the average rating. However, there are a few instances where the review date effect is noticeably higher, reflecting occasional periods where ratings deviated from the overall trend. Including review date effects helps capture any subtle temporal shifts in rating behavior.



Method	RMSE	Difference
Project objective	0.86490	-
Naive model	1.0599	0.195
Movie effects (b_i)	0.94374	0.07884
Movie + User effects (b_u)	0.86593	0.00103
Movie, User and Genre effects (b_g)	0.86559	0.00069
Movie, User, Genre and Year effects (b_y)	0.86542	0.00052
Movie, User, Genre, Year and Review Date effects (b_r)	0.86531	0.00041
Regularised Movie, User, Genre, Year and Review Date effects	0.86465	-0.00025

4.7 Regularized model

The table below shows the step-wise improvement in model performance as more effects are included. The largest reduction in RMSE occurs when user effects are added, highlighting the significant impact of individual user preferences on ratings. Additional factors such as genre, release year, and review date produce only small further improvements, suggesting these variables explain less of the remaining variation. The final regularized model, which applies a penalty to prevent large effect estimates and overfitting, achieves the best RMSE and meets the project's target objective.

To further visualize model performance across all approaches, the figure above presents RMSE values for each method as horizontal bars. The red dashed line marks the project objective (RMSE = 0.8649) for reference

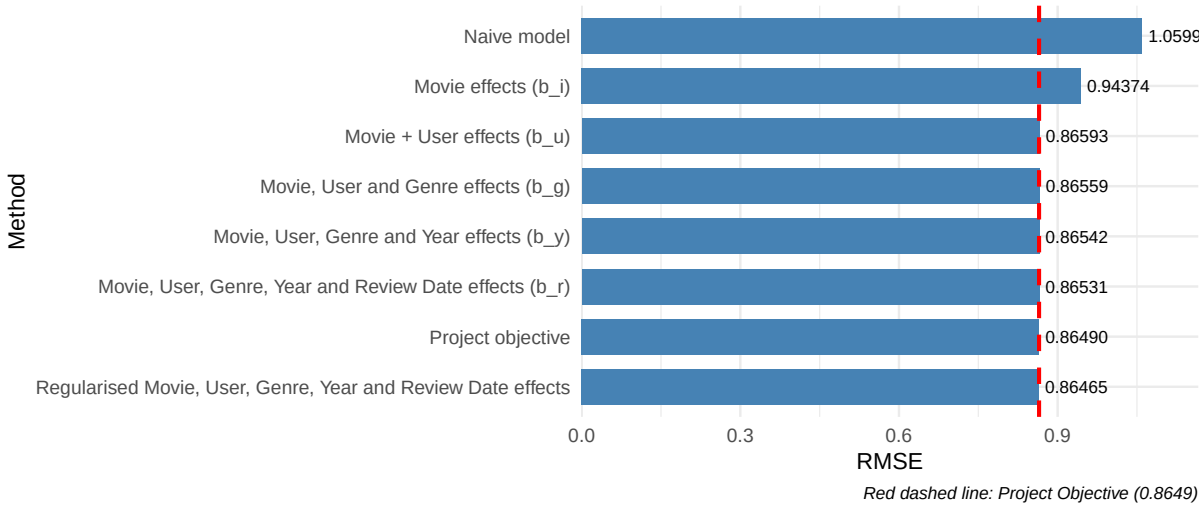


Figure 9: Model Performance: RMSE by Method

4.8 Performance on the validation set

The final model achieved an RMSE of 0.8642, when evaluated on the hold-out validation set. This value is lower than the required benchmark of 0.8649. The recommendation algorithm has met the project objective.

Table 5: Validation of Final Model

Method	RMSE	Difference
Project objective	0.86490	-
Validation of Final Model	0.86412	-0.00078

5 Conclusion

This project developed a movie recommendation system using the MovieLens 10M dataset. The process began with data preparation and exploratory analysis, then moved through stepwise model building that added movie, user, genre, release year, and review date effects. Each step resulted in improved predictive accuracy, with the greatest improvement coming from the inclusion of user-specific effects. Regularization techniques were applied to prevent overfitting and optimize model performance. The final model achieved an RMSE of 0.8642 on the hold-out validation set, meeting and slightly exceeding the project's objective.

There are some limitations to this approach. The model is based on linear additive effects and does not capture complex interactions or hidden patterns that might influence ratings. It also assumes the relationships in the data remain stable over time and may not perform well for new movies or users with limited data. Future improvements could involve using matrix factorization, neural network models, or hybrid approaches that blend collaborative filtering with content-based methods. Using more user and movie information, as well as models that account for changing patterns over time, could further improve the accuracy and reliability of recommendations.