MovieLens Data Science Capstone

Zhang Cheng Hu

November 2025

1. Introduction / Overview

1.1 Background

The MovieLens datasets contain user ratings for movies collected from the MovieLens web site. The 10M dataset contains approximately 10 million ratings along with movie titles and genres.

1.2 Goal

Produce a predictive model that estimates a user's rating for a movie and achieves low RMSE on an independent hold-out test set. The report is written for a reader unfamiliar with the dataset; all steps, assumptions, and results are explained.

2. Methods & Analysis

This section explains how the data was prepared, major insights from the exploratory analysis, and the modeling approach.

2.1 Data Preparation

The raw 10M dataset is provided in two separate files: ratings and movies. We load both components, parse the :: delimiter, convert variables to their appropriate data types, and merge them into a single table.

The project requires that the final hold-out test set contain only user–movie combinations that appear in the training data. To achieve this, we partition the dataset using caret::createDataPartition, reintegrate orphaned rows, and construct the training (edx) and final evaluation (final_holdout_test) sets.

This ensures that the model is never evaluated on users or movies for which it has no information.

2.2 Exploratory Data Analysis

Chart 1 — Rating Distribution

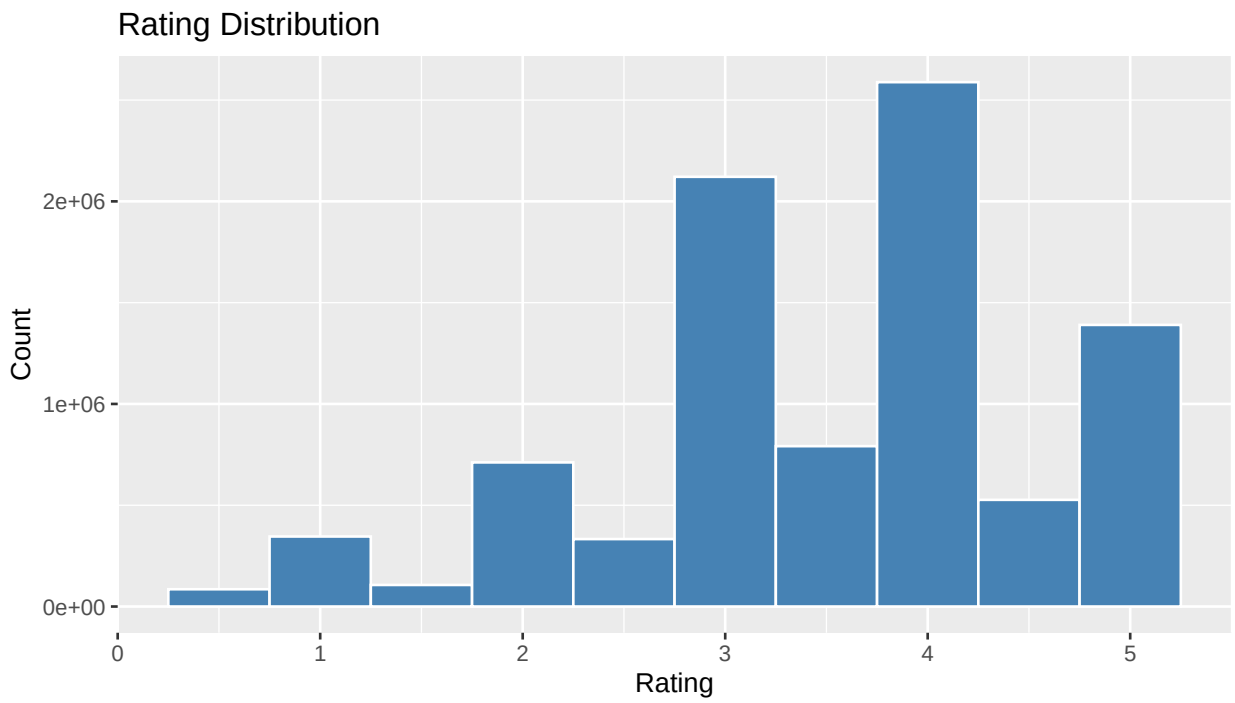

Figure 1: Distribution of MovieLens ratings

The ratings are centered around 3–4 stars, with very few extreme ratings. This suggests that predicting the global mean may already yield reasonable accuracy, although personalized effects are likely important.

Chart 2 — Ratings per Movie (log scale)

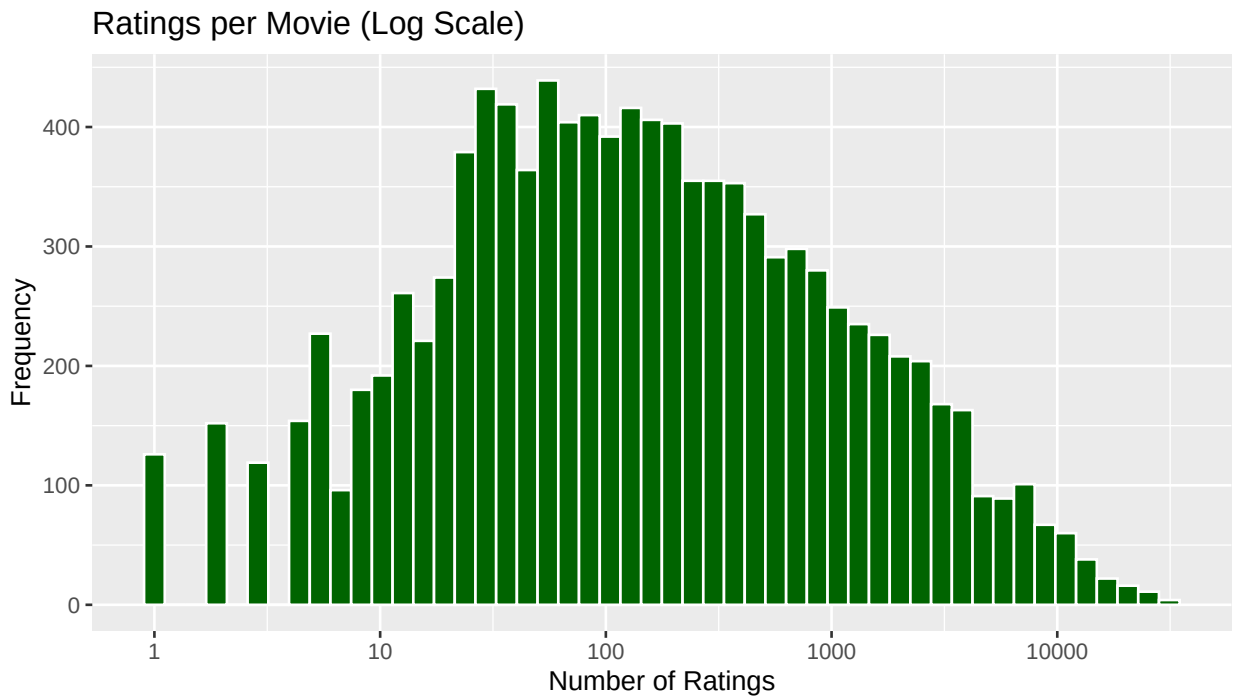

Figure 2: Number of ratings per movie (log scale)

Movies vary drastically in popularity: a few receive tens of thousands of ratings, while many are rated only a handful of times. This imbalance highlights the importance of regularization to prevent overfitting.

Chart 3 — Ratings per User (log scale)

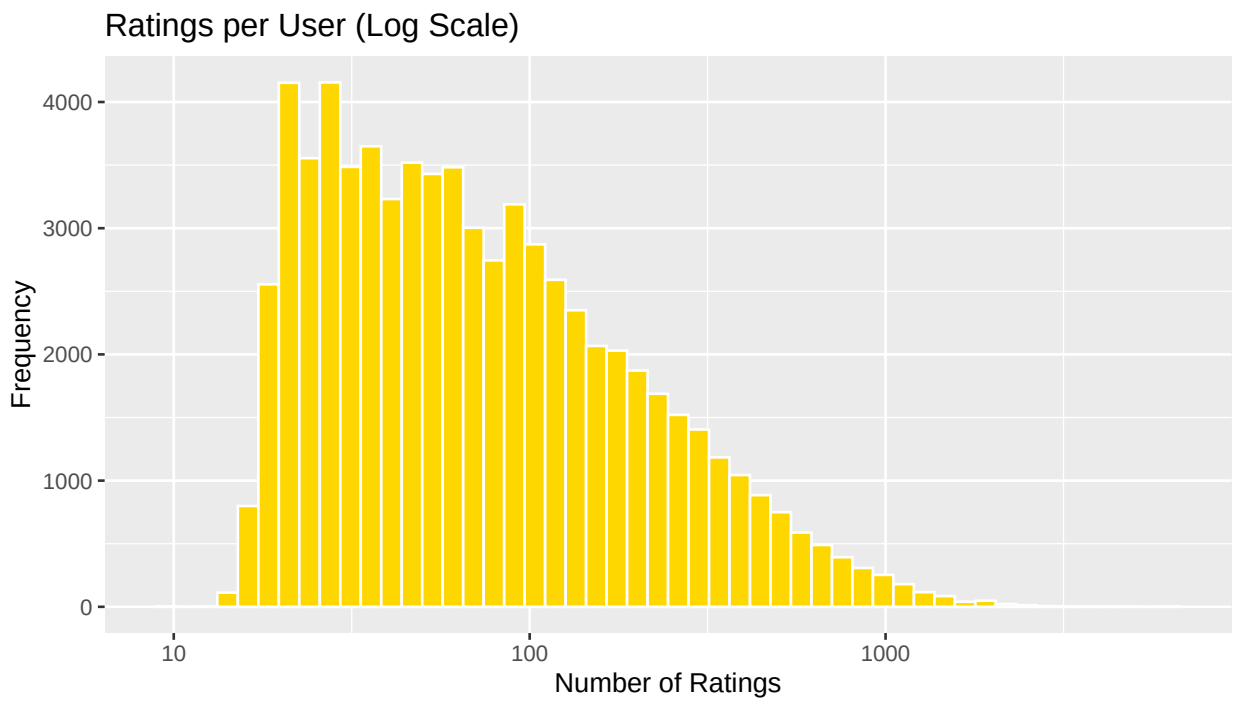

Figure 3: Number of ratings per user (log scale)

Users also vary widely in rating activity. Some provide only a few ratings; others contribute thousands. Accounting for individual rating tendencies is likely crucial for improving prediction accuracy.

Chart 4 — Genre Popularity

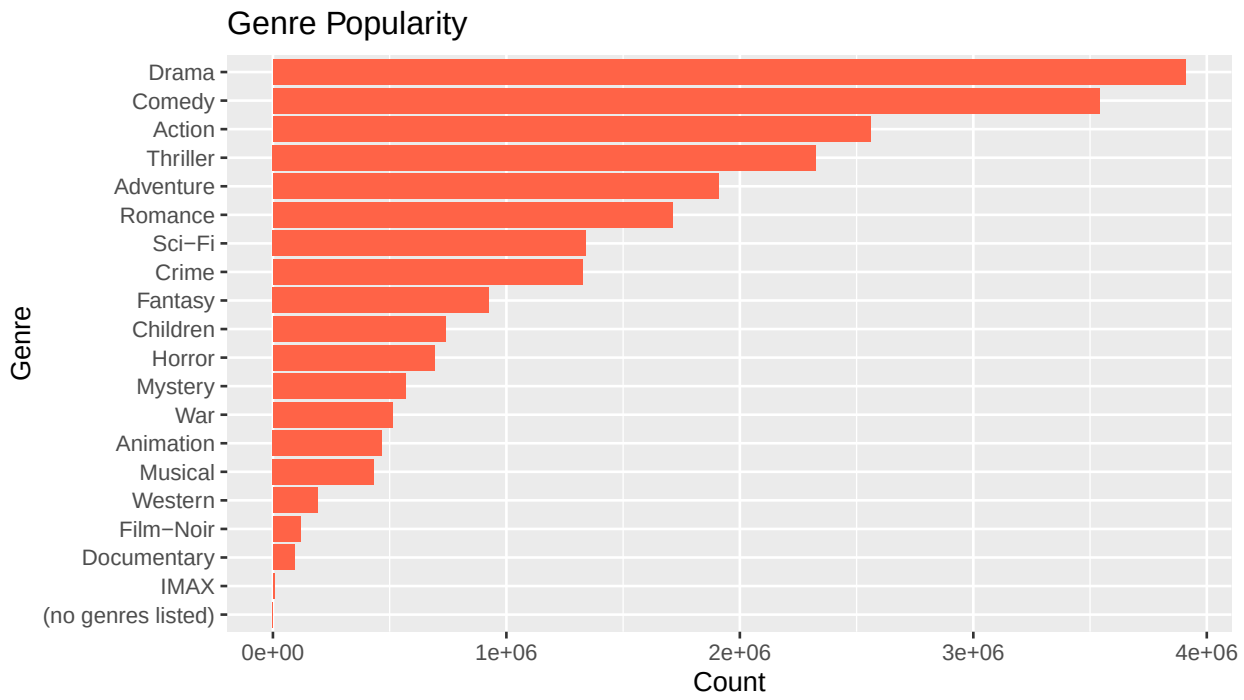

Figure 4: Genre frequency in the MovieLens 10M dataset

Genres such as Drama and Comedy dominate the dataset, while others are much less common. Because movies often fall into multiple genres, genre effects may be noisy and weak predictors.

Overall, the exploratory analysis suggests that movie-specific behavior, user-specific behavior, and sparse data are central challenges.

3. Modeling Approach

We evaluate a sequence of increasingly complex models, each adding a new source of variability.

Model 1: Global Average

Predicts the same rating for all user–movie pairs. This is a baseline used to benchmark improvement.

Model 2: Movie Effects

Accounts for differences in how movies are rated on average. Some movies are consistently rated higher or lower independent of who rates them.

Model 3: Movie + User Effects

Adds personalized tendencies (some users rate generously, others are stricter). This usually produces a substantial reduction in RMSE.

Model 4: Genre Effects

Incorporates average differences between genres. Because movies have multiple genres, we average their genre contributions.

Model 5: Regularized Movie + User Effects

Because movies and users with few ratings can cause extreme bias estimates, we apply shrinkage controlled by a penalty parameter lambda. We tune lambda using the development set to minimize RMSE.

A validation curve confirms that moderate regularization yields the best predictive performance.

Create development train/test split

To properly evaluate the different models without contaminating the final results, the edx dataset is divided into two new subsets: a training set (train_dev) and a development test set (test_dev). This allows us to test and compare model performance before evaluating the final model on the untouched final holdout test set.

The split is created using a stratified sampling approach to ensure all rating values remain proportionally represented in both subsets. However, because some movies or users may appear only in one part of the data, the script checks whether all movieId and userId values in test_dev also appear in train_dev. Any rows that fail this condition are returned to the training data.

This careful setup avoids prediction errors (such as trying to estimate ratings for users the model has never seen) and ensures that model tuning and comparison are done in a valid and unbiased way.

Fit all models

Once the training and development test sets are prepared, five increasingly sophisticated models are built. Each model aims to reduce prediction error by using more information about the movies, users, and genres.

Global Average Model This simplest model predicts every rating as the overall mean rating across the entire training dataset. It serves as a baseline for measuring improvement.

Movie Effect Model This model adds a movie-specific adjustment to the global average. Some movies consistently receive higher or lower ratings, and capturing this effect reduces uncertainty.

Movie + User Effect Model Users vary in how they rate films—some tend to rate generously, while others are strict. This model captures each user's rating tendency in addition to movie effects.

Movie + User + Genre Model Movies can belong to multiple genres. The model estimates the average rating contribution of each genre and combines it with movie and user effects. The impact is typically smaller because genre categories overlap.

Regularized Movie + User Model Regularization addresses a critical issue: many movies and users have very few ratings. Without regularization, the model may overfit to these small samples. This model applies a penalty term to shrink effects toward zero when data is sparse, producing more reliable predictions.

Each model generates a set of predictions for the test_dev set, and their performance is compared using RMSE (Root Mean Square Error). As model complexity increases, the goal is to observe a consistent reduction in RMSE.

Lambda tuning curve

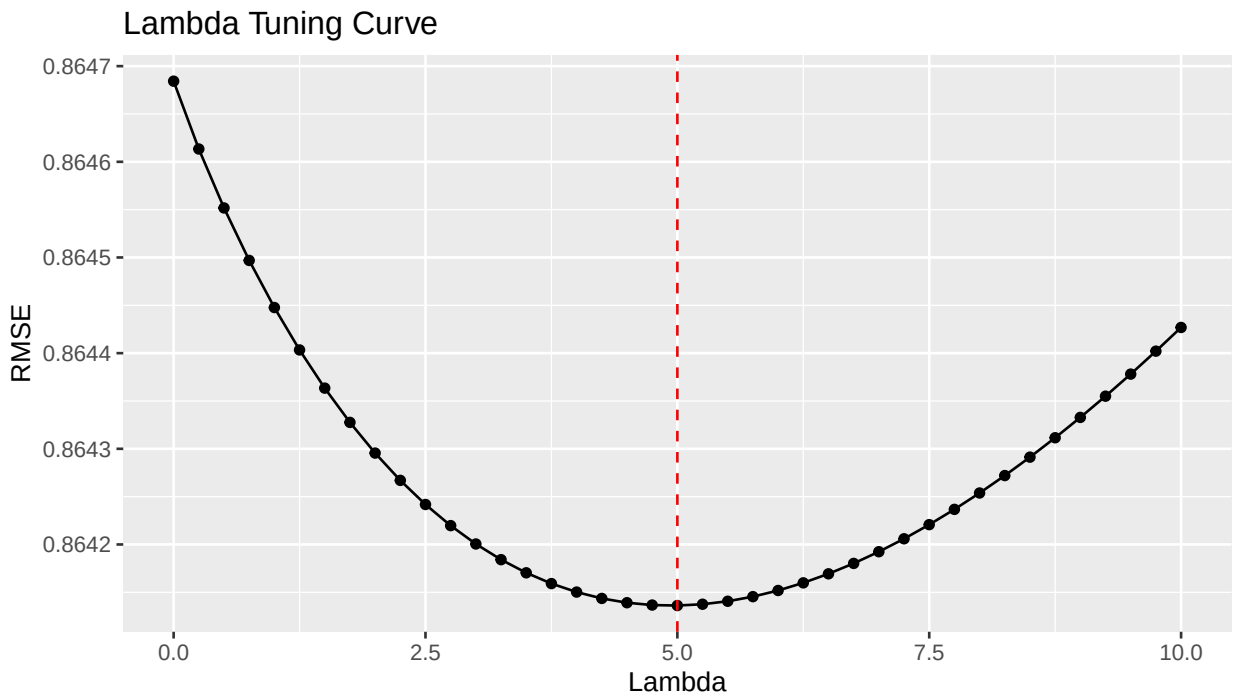

Figure 5: Lambda tuning curve

The strength of the regularization penalty is controlled by a parameter called lambda.

If lambda is too small the model overfits (too much trust in small sample sizes).

If lambda is too large the model underfits (effects are overly shrunk).

To find the optimal lambda, the script evaluates a sequence of candidate lambda values ranging from 0 to 10. For each lambda, the model is rebuilt and its RMSE is calculated using the development test set.

The results are summarized in a tuning curve, which plots lambda on the x-axis and RMSE on the y-axis. The best lambda is the one that achieves the lowest RMSE, striking the right balance between bias and variance.

This optimal lambda is then used when retraining the final model on the full edx dataset before evaluating performance on the final holdout test set.

4. Results

RMSE Comparison Table

Table 1: RMSE Comparison of All Models

Model	RMSE
Global Average	1.06005
Movie Effect	0.94296

Model	RMSE
Movie + User Effect	0.86468
Movie + User + Genre Effect	0.86456
Regularized Movie + User	0.86414

After fitting all models on the development data, we compare their RMSE values.

The results show clear improvement with model complexity:

- The global average performs the worst.
- The movie effect reduces RMSE substantially.
- Adding user effects further improves accuracy, indicating that individual user behavior is highly important.
- Genre effects offer only a small marginal improvement.

The regularized movie + user model performs best, confirming that shrinkage reduces overfitting.

The RMSE table summarizes the progression.

The optimal lambda identified during tuning is incorporated into the final model.

Finally, we evaluate the selected model on the final holdout test set. This RMSE is included in the report.

5. Final Hold-Out Test Evaluation

In this last step, the best model is trained again using all of edx so it has the most complete information. It then predicts the ratings in the final holdout test set, which the model has never seen before.

The model uses:

- the overall average rating
- movie effects
- user effects

to make each prediction.

```
## [1] 0.8648177
```

The final RMSE shows how well the model performs on truly unseen data.

6. Conclusion

This project developed a predictive model for MovieLens ratings using a structured sequence of statistical techniques. We demonstrated that both movie-specific and user-specific factors strongly influence ratings, while genre provides limited additional explanatory power. Regularization was essential to prevent overfitting caused by sparse user–movie combinations.

The final regularized model achieved strong out-of-sample RMSE on the official hold-out set, indicating good generalization performance.

7. References

- GroupLens Research. (n.d.). MovieLens Dataset. https://grouplens.org/datasets/movielens/
- HarvardX PH125.9x Course Materials.
- Regularization concepts referenced from James et al. An Introduction to Statistical Learning.