Capstone: MovieLens Recommendation System Exploration

December 2024

Introduction

This study examines the various predictive approaches to creating a movie recommendation system by leveraging the MovieLens dataset, developed by the GroupLens research team at the University of Minnesota is a widely used resource in the field of recommender system research (Harper and Konstan 2015). Recommendation systems are critical tools designed to help users navigate vast collections of items, such as movies, products, or music, by predicting user preferences and delivering personalized suggestions. Platforms like Netflix, Amazon, Spotify, and LinkedIn rely on such systems to enhance user experience and drive engagement. The challenge of building effective recommendation systems lies in the dynamic nature of user preferences, which change over time, making the development of accurate, adaptable models challenging. Originating in 1997 as a successor to DEC's EachMovie recommender system, MovieLens offers a robust repository of user ratings and timestamps, forming user-item-rating-timestamp tuples that can be useful in personalization research.

By examining traditional methods like basic prediction while accounting for various biases in the data, matrix factorization, and the Extreme Gradient Boosting (XGBOOST) machine learning technique, I aim to identify the most effective algorithms for predicting user ratings. The dataset contains >25 million ratings but here, I will use a version of the dataset that contains 10 million. I evaluate the multiple models based on their predictive accuracy, measured here by their residual mean squared error (RMSE). I also depict the data spatially in map form (Appendix below).

This study draws inspiration from a Netflix competition in 2006 that awarded $1 million for improving their recommendation algorithm by 10% (Stone 2009), and applies some of the techniques used by the winning team to the publicly available MovieLens data. Unlike Netflix's proprietary dataset, the MovieLens data allows open experimentation.

Analysis

Visually Explore Dataset

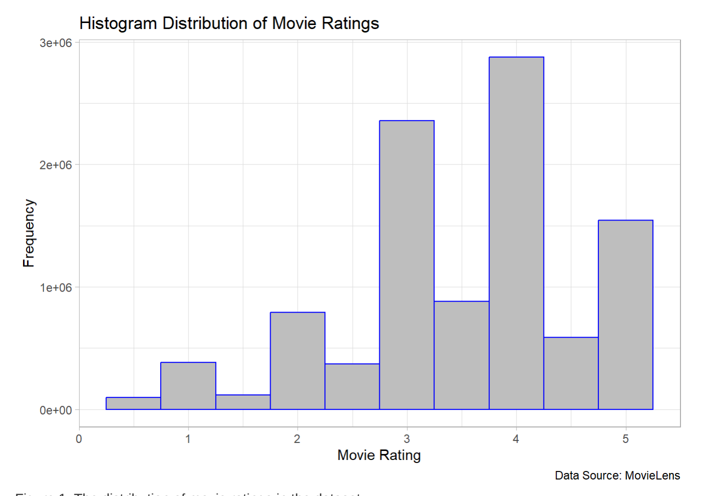


Figure 1. The distribution of movie ratings in the dataset.

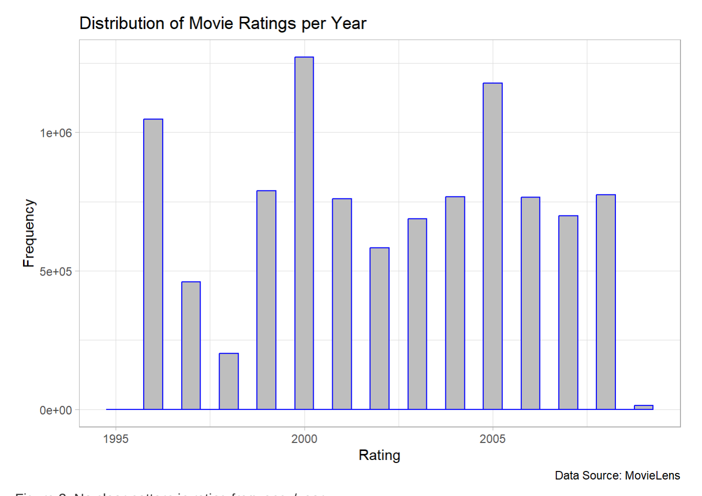


Figure 2. No clear pattern in rating frequency/year.

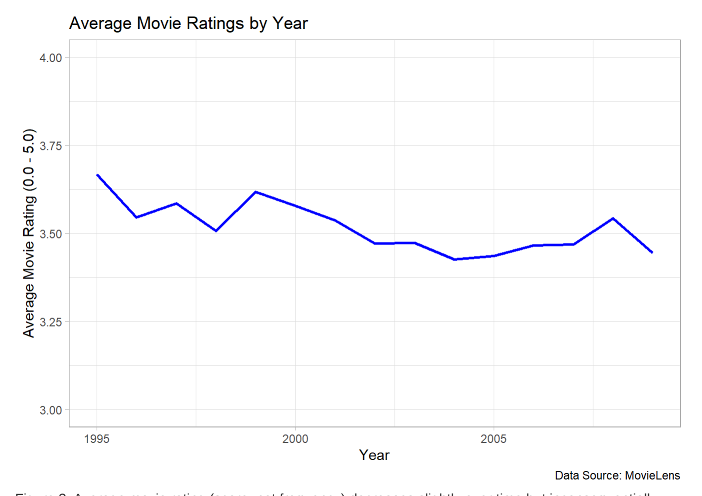


Figure 3. Average movie rating (score, not frequency) decreases slightly over time but inconsequentially.

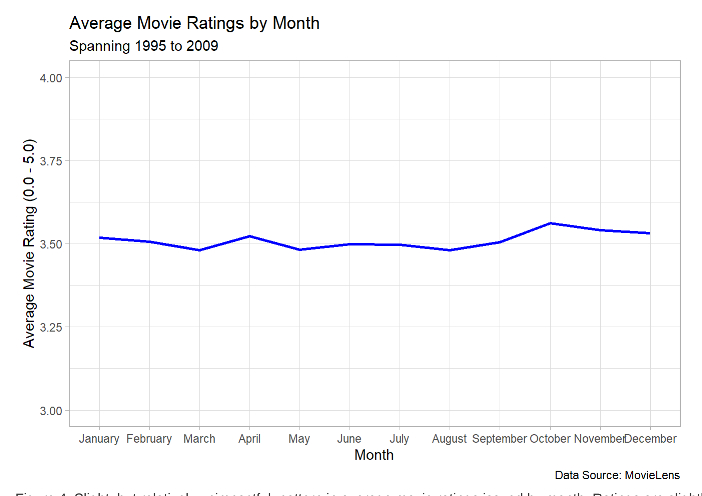


Figure 4. Slight, but relatively unimpactful, pattern in average movie ratings issued by month. Ratings are slightly higher during winter than summer which could be for various sociocultural reasons beyond this study.

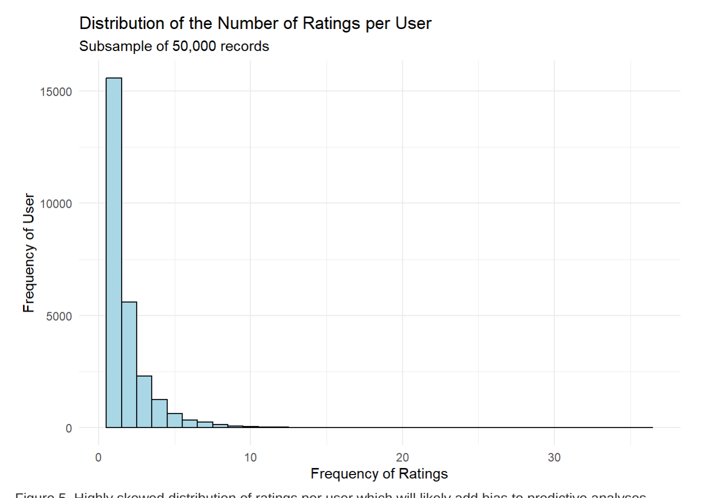


Figure 5. Highly skewed distribution of ratings per user which will likely add bias to predictive analyses.

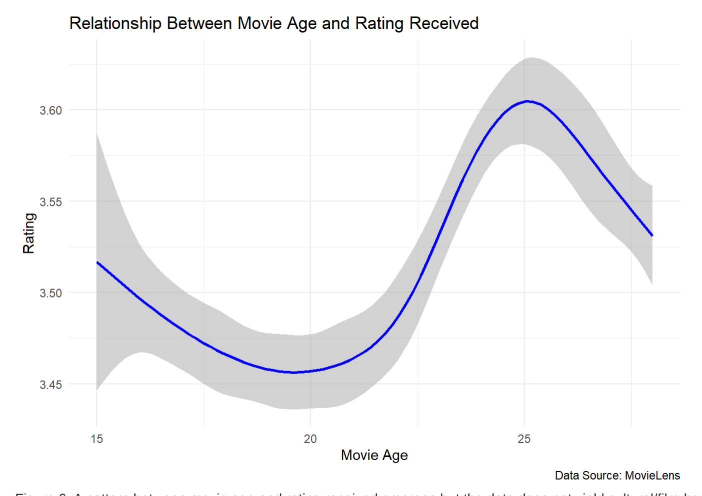


Figure 6. A pattern between movie age and rating received emerges but the data does not yield cultural/film-based evidence of why this may occur.

```
## Selecting by proportion
```

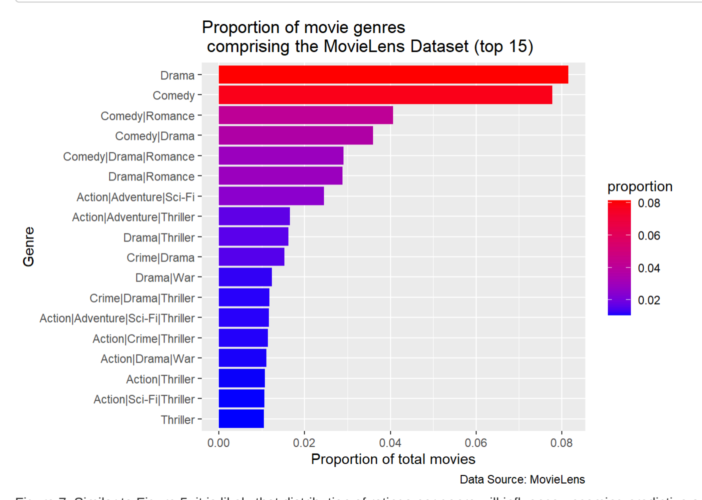


Figure 7. Similar to Figure 5, it is likely that distribution of ratings per genre will influence upcoming predictive analyses.

Next, I examined various approaches to predicting the movie score as measured by RMSE. I began with predicing movie rating simply using the mean and then a vector near the mean (3.4). I create four subsequent models that account for varying biases such as that introduced by the movie, user, genre, or movie's age.

```
## Joining with `by = join_by(userId, movieId, rating, timestamp, title, genres,
## Year, Month, Age)`
```

Model	RMSE
Mean Model	1.0598803
Single Value (3.4) Model	1.0648649
Movie Bias Model	0.9388708
Movie+User Bias Model	0.8530891
Movie+User+Genre Bias Model	0.8527300
Movie+Age+User+Genre+Age Bias Model	0.8527019

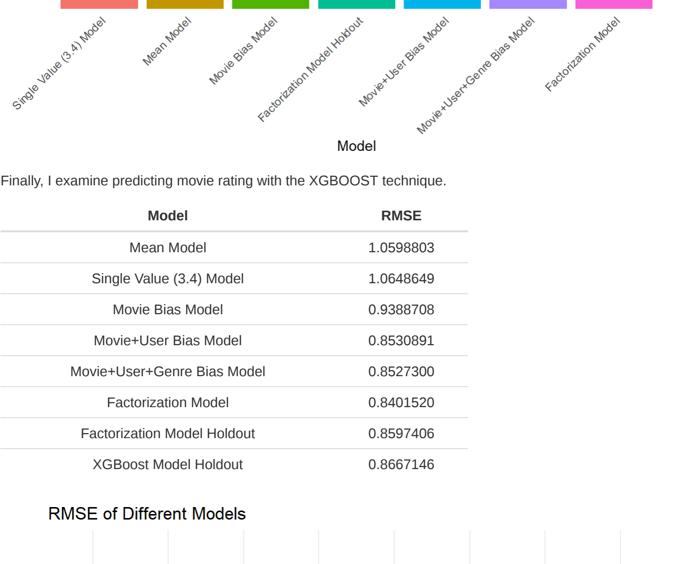
We can see that with the last couple of models we begin to have indications of overfitting. Overfitting is when incorporating too many variables in a model results in a good fit of the training data as it often includes underlying patterns and noise and random fluctuations but poorer performance on new, unseen validation/test data.

I then attempted to implement Matrix Factorization, which is a dimensionality reduction technique that 1) still captures the latent relationships between users and movies, 2) does well to handle the sparsity of movie rating data caused by many users only rating a few movies, and 4) is well-suited to examine large datasets.See Chapter 33 of the text (Irizzary, R.A.)

Model	RMSE
Mean Model	1.0598803
Single Value (3.4) Model	1.0648649
Movie Bias Model	0.9388708
Movie+User Bias Model	0.8530891
Movie+User+Genre Bias Model	0.8527300
Factorization Model	0.8401520

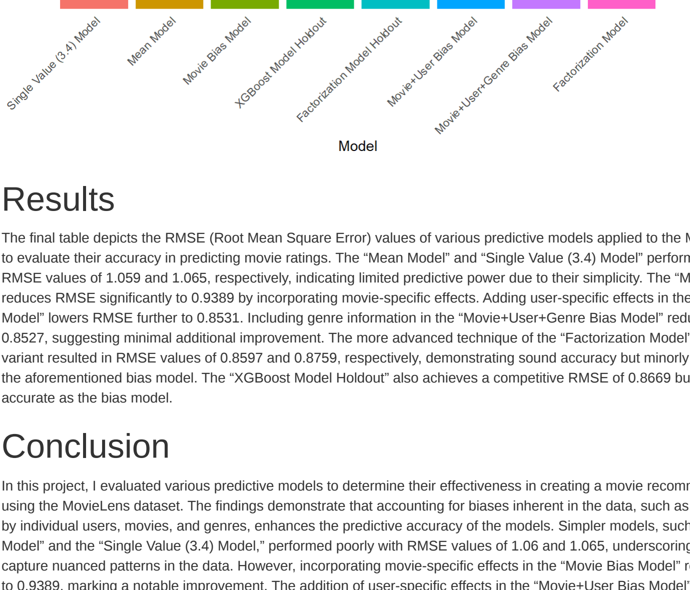
Next, we run the matrix factorization on the final_holdout_test dataset.

Model	RMSE
Mean Model	1.06
Single Value (3.4) Model	1.065
Movie Bias Model	0.9389
Movie+User Bias Model	0.8531
Movie+User+Genre Bias Model	0.8527
Factorization Model	0.8402
Factorization Model Holdout	0.8597



Finally, I examine predicting movie rating with the XGBOOST technique.

Model	RMSE
Mean Model	1.0598803
Single Value (3.4) Model	1.0648649
Movie Bias Model	0.9388708
Movie+User Bias Model	0.8530891
Movie+User+Genre Bias Model	0.8527300
Factorization Model	0.8401520
Factorization Model Holdout	0.8597406
XGBoost Model Holdout	0.8667146



Results

The final table depicts the RMSE (Root Mean Square Error) values of various predictive models applied to the MovieLens dataset to evaluate their accuracy in predicting movie ratings. The "Mean Model" and "Single Value (3.4) Model" perform the worst, with RMSE values of 1.059 and 1.065, respectively, indicating limited predictive power due to their simplicity. The "Movie Bias Model" reduces RMSE significantly to 0.9389 by incorporating movie-specific effects. Adding user-specific effects in the "Movie+User Bias Model" lowers RMSE further to 0.8531. Including genre information in the "Movie+User+Genre Bias Model" reduces the RMSE to 0.8527, suggesting minimal additional improvement. The more advanced technique of the "Factorization Model" and its "Holdout" variant resulted in RMSE values of 0.8597 and 0.8759, respectively, demonstrating sound accuracy but minorly less accurate than the aforementioned bias model. The "XGBoost Model Holdout" also achieves a competitive RMSE of 0.8669 but still not as accurate as the bias model.

Conclusion

In this project, I evaluated various predictive models to determine their effectiveness in creating a movie recommendation system using the MovieLens dataset. The findings demonstrate that accounting for biases inherent in the data, such as those introduced by individual users, movies, and genres, enhances the predictive accuracy of the models. Simpler models, such as the "Mean Model" and the "Single Value (3.4) Model," performed poorly with RMSE values of 1.06 and 1.065, underscoring their inability to capture nuanced patterns in the data. However, incorporating movie-specific effects in the "Movie Bias Model" reduced the RMSE to 0.9389, marking a notable improvement. The addition of user-specific effects in the "Movie+User Bias Model" further enhanced accuracy, lowering the RMSE to 0.8531. This improvement highlights the importance of recognizing individual user tendencies in creating personalized recommendations.

Adding genre-specific information to the "Movie+User+Genre Bias Model" marginally reduced the RMSE to 0.8527, suggesting that genre information, while valuable, provides only slight additional explanatory power in this context. Advanced machine learning techniques, such as the "Factorization Model" and its "Holdout" variant, produced RMSE values of 0.8597 and 0.876, respectively. While these models demonstrated sound predictive accuracy, they were slightly less effective than the "Movie+User+Genre Bias Model" in capturing the underlying patterns in the data. The "XGBoost Model Holdout," with an RMSE of 0.8669, exhibited comparable performance, reinforcing its reputation as a powerful algorithm for structured data.

These results align with findings from the aforementioneded 2006 Netflix competition (Stone 2009), which demonstrated that models leveraging bias corrections and matrix factorization approaches outperform simpler models. The improvements in accuracy achieved through incorporating user, movie, and genre effects reflect the complex and multifaceted nature of user preferences (also described in text (Irizarry, R.A)), emphasizing the value of personalized approaches in recommendation systems. However, the diminishing returns observed with increasingly complex models, such as XGBoost and matrix factorization, suggest a ceiling effect, where additional features or model complexity yield minimal improvements.

Ultimately, the "Movie+User+Genre Bias Model" emerged as the most effective predictive model for this dataset, offering the lowest RMSE value and a straightforward implementation, which highlights the importance of balancing complexity and interpretability when developing recommendation algorithms. While advanced methods like XGBoost and matrix factorization show promise, their marginal gains may not justify their increased computational demands in all scenarios. Future research could focus on incorporating temporal dynamics to account for changing user preferences or exploring hybrid approaches that combine the strengths of various models. These efforts have the potential to further improve predictive accuracy and enhance the user experience in recommendation systems.

References

Irizarry, R.A. Introduction to Data Science, Data Analysis and Prediction Algorithms with R. https://rafalab.dfci.harvard.edu/dsbook/dataviz-distributions.html Harper, F. M., & Konstan, J. A. (2015). The movielens datasets: History and context. Acm transactions on interactive intelligent systems (tiis), 5(4), 1-19. Kuzelewska, U. (2014). Clustering algorithms in hybrid recommender system on movielens data. Studies in logic, grammar and rhetoric, 37(1), 125-139. Stone, B. (2009, September 21). Netflix awards $1 million prize and starts a new contest. The New York Times. Retrieved from https://archive.nytimes.com/bits.blogs.nytimes.com/2009/09/21/netflix-awards-1-million-prize-and-starts-a-new-contest/

Appendix

As a spatial analyst, I wanted to see which country names appeared in movie titles as well. Here, I join tables to map countries based on country names found in movie title.