Movie Rating Prediction Model Using the MovieLens Dataset

Charles Zimmerman

4/25/2021

I. Overview/Executive Summary

The goal of the project was to develop a movie recommendation system–specifically, predict the ratings of prospective film watchers, based on their own and others' previous ratings. The prediction model was developed using the 10M version of the MoveLens data set, as provided by the movielens data in the R dslabs package. The data set contains over 10 million ratings for 10,681 movies by 71,567 users. Each anonymous user, tracked only by a numeric unique identifier, rated a minimum of 20 movies.[1]

The model was developed by starting with the overall mean of ratings across the data set as the predicted value for every rating, and then refining it by evaluating the following effects:

- *Main effect of movie:* As movies vary in quality and popularity, some titles will receive higher ratings than others solely based on this factor.
- *Main effect of user:* Users will vary in the harshness or generosity of their ratings overall.
- *Main effect of genre:* Each movie in the dataset was associated with one or more genres, from the following list. Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western. For example, a single film might be characterized as one of these (e.g., Drama, the most frequently occurring category) or multiple (e.g., 21 films were classified as "Action|Comedy|Crime|Thriller"); overall, the data set included 797 distinct genre categories.

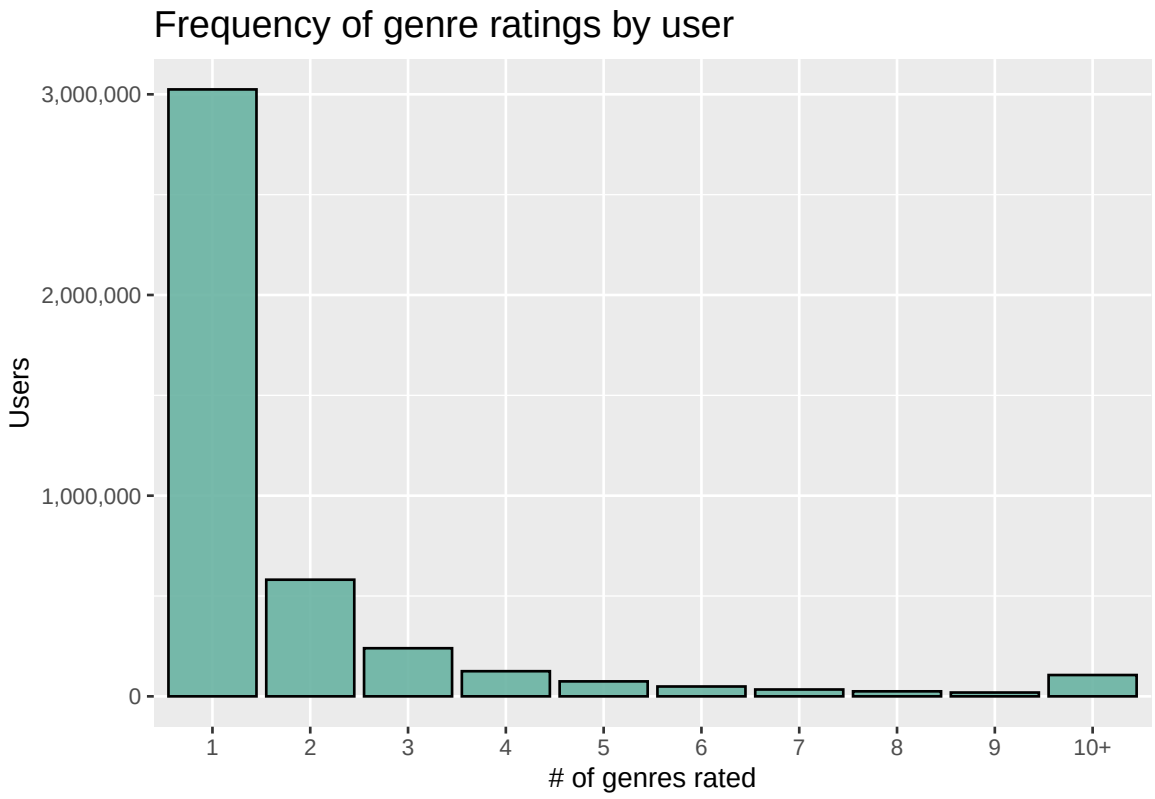
  Model development proceeded on the hypothesis that the high level of granularity in the classifying the movies might translate into predictive power. If movies in a very specifically defined category are relatively homogeneous in terms of their quality, there

[1] Department of Computer Science and Engineering at the University of Minnesota. *MoveLens 10M Dataset* https://grouplens.org/datasets/movielens/10m/

might be systematic differences in ratings across such categories. Thus, in evaluating the main-effect of genre, the model retained the classifications as provided, as opposed to slicing them into broader, more inclusive categories. For instance, if a film's genre was "Drama |Comedy", the data was kept as is, rather than assigning separate "Drama" and "Comedy" flags to the film.

- *Interaction between Genre and User:* Individual movie watchers tend to favor some types of movies over others; e.g., some people prefer comedies to thrillers, some like romances and dislike all other films. These preferences could obviously be leveraged to predict a watcher's rating—someone who loves comedies would rate them relatively higher.

  Theoretically, due to the high granularity of the genre classifications, this model component should offer strong predictive power. But the high number of genres also causes a sample size issue for the user-genre aggregation. As one might expect, individual users mostly did not provide ratings across many different genres. As the figure below illustrates, of the 4,279,145 distinct user/genre combinations in dataset, 3,024,430 only occurred once.



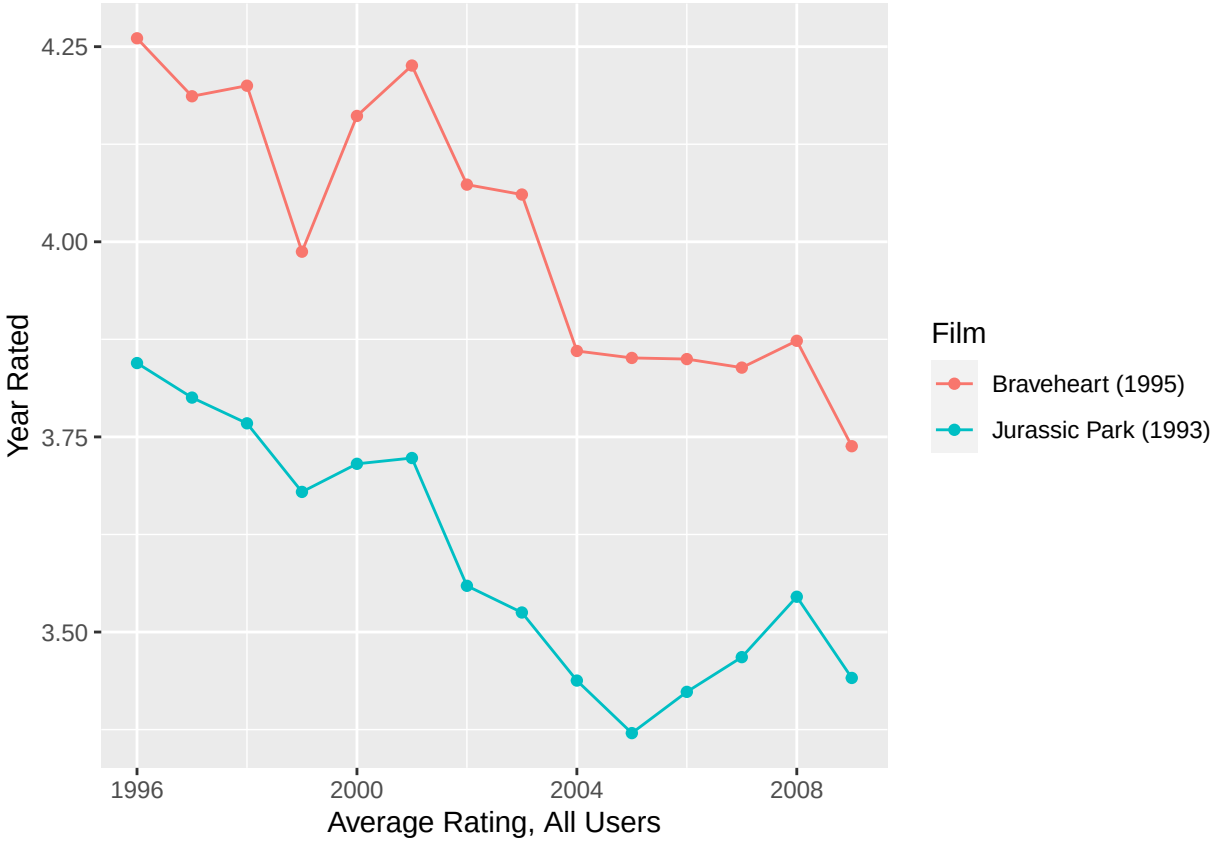
  Because small sample size is associated with overfitting in prediction models (especially with k-fold cross-validation[2], the method used here), whereby significant effects do not generalize to test set or real world data, genre was defined differently for evaluating

[2]Vabalas , Andrius; Gowen, Emma;Poliakoff, Ellen; and Casson, Alexander J.(2019) *Machine learning algorithm validation with a limited sample size.* https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0224365

this interaction, as opposed to the genre main effect. In this case, the model did use singular more inclusive categories so that, for example, a movie could be classified as a Drama and/or as Comedy, but not as a Drama|Comedy. The categories chosen were based on frequencies of occurrence in the dataset, as well as the clarity and sharpness of the distinctions. The specifics are described in the Methods/Analysis section.

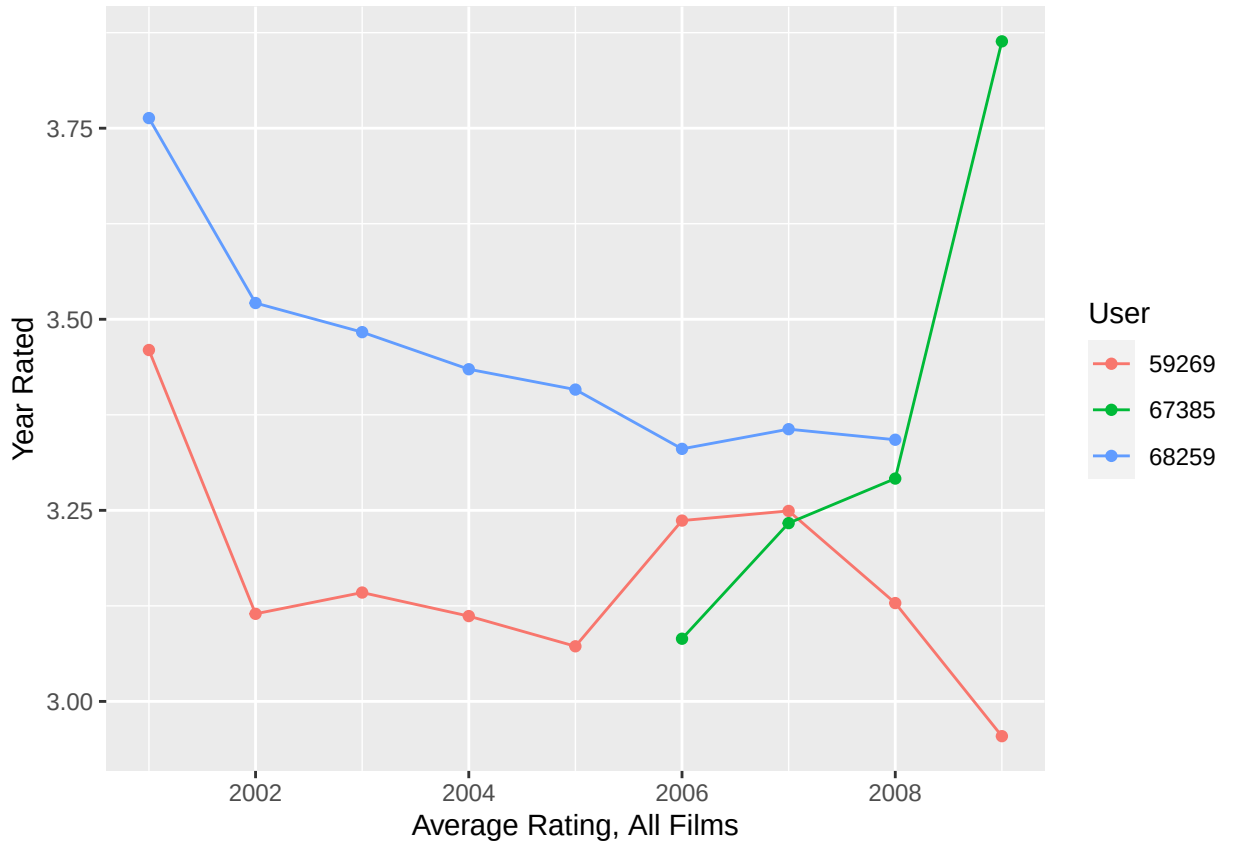
- *Interaction between title and time:* To some degree, ratings of specific movies may be a function of time. For example, a movie might receive higher average ratings around the time it wins an Academy Award, or lower ratings over time if it addresses a topical issue that become less pressing or forgotten over time.

The below graph of average ratings over time for two of the more frequently rated films illustrates that they can vary depending on when they are rated:



- *Interaction between user and time:* Specific users' rating might systematically vary to some degree over time. For example, some users might rate silly comedies lower or dramas with serious themes higher as they grow older.

  The below shows how average ratings vary over time for three users who rated a relatively high number of movies:



The dataset was partitioned into a training and a test set, with 90% of the ratings randomly assigned to the training set. The model was tested and validated solely on the training set, with evaluation on the test set occurring only after the model components and the tuning parameters were in place. The goal was to attain a residual mean square error (rmse) of 0.86490 or lower on the test set.

The tuned parameters were regularization constants, which were tested and chosen separately for each model component. Regularization involves adding a value to the denominator of a model's component's derivation logic, so as to penalize inputs that are derived from smaller sample sizes. These constants were chosen separately, because the average number of observations that were aggregated in derivations varied across model components.

The model development proceeded in what has been described as a "manual greedy manner."[3] In other words, sequentially for each model component, multiple values for the regularization parameter were tested across a 5-fold cross-validation partition, and the parameter resulting in the lowest average rmse was selected. These parameters, once chosen, were not revisited, even though theoretically, the results could have changed based on the parameter values derived later in the model's development.

[3]Koren, Yehuda (2009) *The BellKor Solution to the Netflix Grand Prize.* Koren, Yehuda. https://www.netflixprize.com/assets/GrandPrize2009_BPC_BellKor.pdf

II. Methods/Analysis

The entire project was developed in the R programming language, using version 4.0.3 of R and version 1.3.1903 of R Studio as the IDE.

A. Data download and structure

The 10M version of the MoveLens data set was downloaded programmatically from the GroupLens website in .zip format, then unzipped and parsed, resulting in an r data frame with below columns:

UserId: Unique and anonymous numeric identifier of the user rating the film.
MovieId: Unique numeric identifier for the film.
Rating: The users rating of the film. Values from 0 to 5.0 in intervals of 0.5 are valid ratings, but none of the 10 million+ ratings was 0.
Timestamp: A numeric timestamp, convertible to date/time format, indicating the date and time the rating was recorded.
Title: The movie title.
Genres: A pipe-delimited list of one or more genre categories associated with the film. Refer to "Main Effect of Genre" section in the Overview for more details.

B. Test and Train Set Partition

The data was randomly (but deterministically so that the partitions would be the same for all re-executions of the code) into a train set and a test set, with 90% of the rows assigned to the former. Following the partition, rows in the test set with userId and movieId combinations not present in the train set were removed from the former and added to the latter. The test set partition was modified to add the derived variables for Year Rated, so as to evaluate the movie-time and user-time effects, and also to add the variables Drama, Comedy, Thriller, Romance, SciFi, Children, and Action. These variables were set to 1 if the pipe-delimited string for genre contained the relevant category and 0 otherwise, per the below:

```
mutate(Drama=ifelse(str_detect(genres, 'Drama')==1, 1,0)) %>%
mutate(Comedy=ifelse(str_detect(genres, 'Comedy')==1, 1,0)) %>%
mutate(Thriller=ifelse(str_detect(genres, 'Thriller')==1, 1,0)) %>%
mutate(Romance=ifelse(str_detect(genres, 'Romance')==1, 1,0)) %>%
mutate(Children=ifelse(str_detect(genres, 'Children')==1, 1,0)) %>%
mutate(SciFi=ifelse(str_detect(genres, 'Sci-Fi')==1, 1,0)) %>%
mutate(Action=ifelse(str_detect(genres, 'Action')==1, 1,0))
```

C. Cross-validation and Sequential Evaluation

The training set was further partitioned into 5 random subsets used for cross-validation. These partitions were used to tune parameters for each model component, and because the

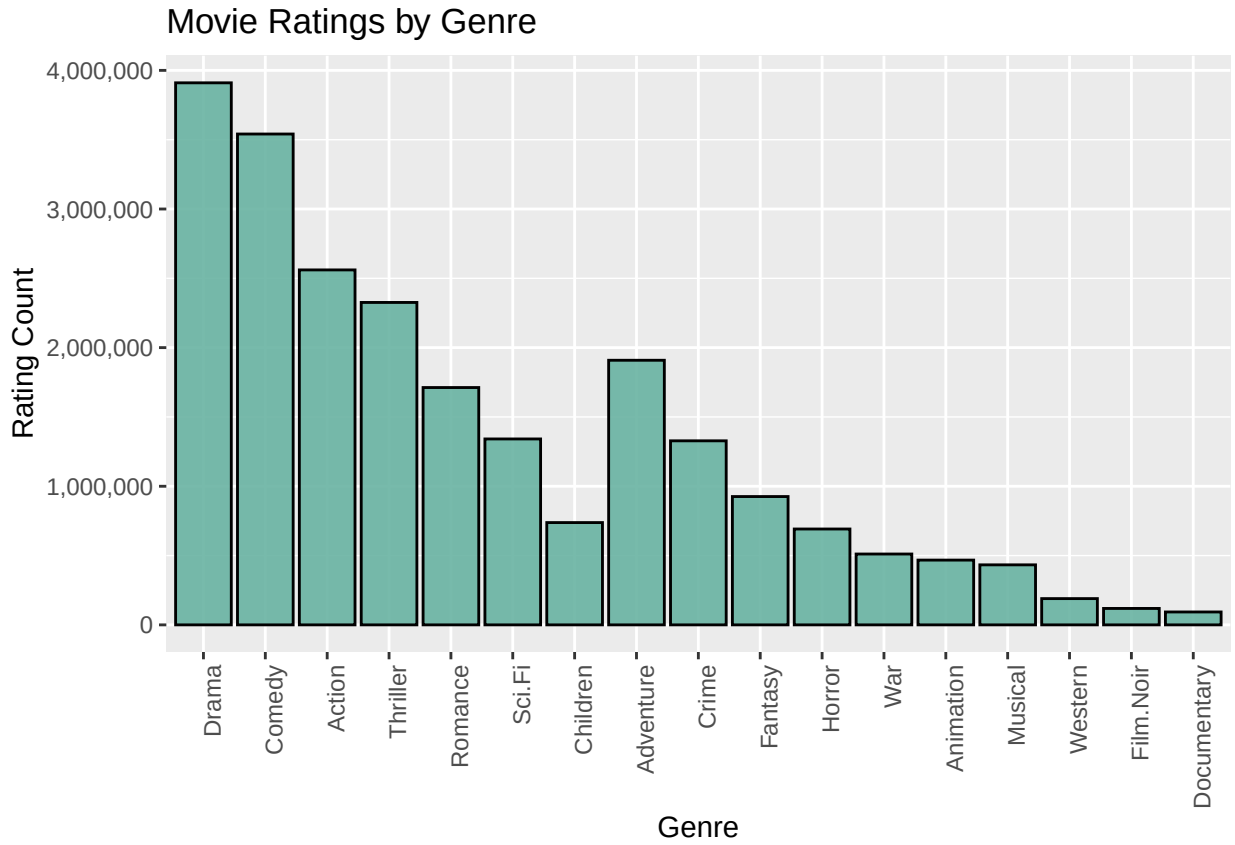
seeding variable for randomization was held constant, they contained the same data for each model run. This ensured the results could be replicated across runs.

In the course of model development, the tuning parameters were validated across a wide range of values for each component. However in the final code, in order to optimize the execution speed of the code, while at the same time demonstrating what was done, a limited subset (including the final value chosen) is tested.

The model was derived and tested sequentially, one component at a time. For each component, the optimal regularization parameter was identified and the the rmse is calculated for the model, such as it was at the point of component addition. The model's first component is the overall mean of all ratings, i.e., every film rating is predicted to be the mean. Then, development proceeds by adding components and evaluating the model at that point. The components were added in the following order.

Main effect of title
Main effect of user
Main effect of genre
User-genre interaction
Title-time interaction
User-time interaction

As noted earlier, the main effect of genre was tested by using the genres as provided. The user-genre effect was evaluated with user averages for the non-mutually exclusive single genre categories of Drama, Comedy, Thriller, Romance, Children, SciFi, and Action. The below shows the distribution of all such categories across ratings, with the ones chosen to evaluate this component on the left:



The goal in choosing categories was to balance a concern for including greater number of observations with that of including distinct categories. For example, "Adventure" was not included, even though movies in that category had more ratings than others chosen, because it is conceptually similar to "Action". And "Children" was chosen even though it had fewer ratings than some of the excluded categories, because it was viewed as distinct from the other included categories.

To test the components, the weights were applied for all components included in the model at the point of evaluation, with the result being a predicted rating for every row in the dataset. Any prediction over 5.0 or under 0.5 was smoothed to those values respectively.

Below is an illustrative example: it was run after the tuning parameter for the user-genre interaction was derived. All of the components previously evaluated are included in the prediction. The movie-time and user–time components are not present because they are evaluated at a later point in the model development code.

```
predicted_ratings <- edx %>%
 left_join(edx_movie_avgs, by='movieId') %>%
 left_join(edx_user_avgs, by='userId') %>%
 left_join(edx_genre_avgs, by='genres') %>%
 left_join(edx_user_drama_avgs, by=c('userId','Drama')) %>%
 left_join(edx_user_comedy_avgs, by=c('userId','Comedy')) %>%
 left_join(edx_user_thriller_avgs, by=c('userId','Thriller')) %>%
 left_join(edx_user_romance_avgs, by=c('userId','Romance')) %>%
```

left_join(edx_user_children_avgs, by=c('userId', 'Children')) %>%
left_join(edx_user_scifi_avgs, by=c('userId', 'SciFi')) %>%
left_join(edx_user_action_avgs, by=c('userId', 'Action')) %>%
mutate(pred=mu +b_i + b_u + b_g + b_u_d + b_u_c
+ b_u_t + b_u_r + b_u_ch + b_u_s + b_u_a) %>%
mutate(pred=ifelse(pred<.05, .05, pred)) %>%
mutate(pred=ifelse(pred>5.0, 5.0, pred)) %>%
pull(pred)

In the bolded portion, mu represents the overall rating mean, and the values prefixed by "b_" are the weights derived for each of the component in the model included up to that point.

At each step, after the predictions were derived, the rmse was calculated, using the predicted and actual train set ratings. The rmse was then compared to the one calculated in the previous sequential step. For example, the user-genre interaction was evaluated after the main effect for genre, and the resulting rmses for these two steps were compared to evaluate the degree to which error decreased.

D. Final Evaluation

The rmse calculated in the evaluation of the final component (the user-time effect) was also the overall result on the training set. At this point, the model is evaluated against the test set. The same derived variables described in section II.B were added to the test set data. Additionally, the prediction was modified so that the weights for the predictors involving an aggregate of userId/genre, userId/Year Rated, and movieId/Year Rated would be set to 0 if those combinations were not present in the test set. The final rmse served as the answer to the research question of whether the model could meet the target of an rmse of 0.86490 or lower. The performance of the model against the test set was not evaluated at any point in the model's development until this final step.

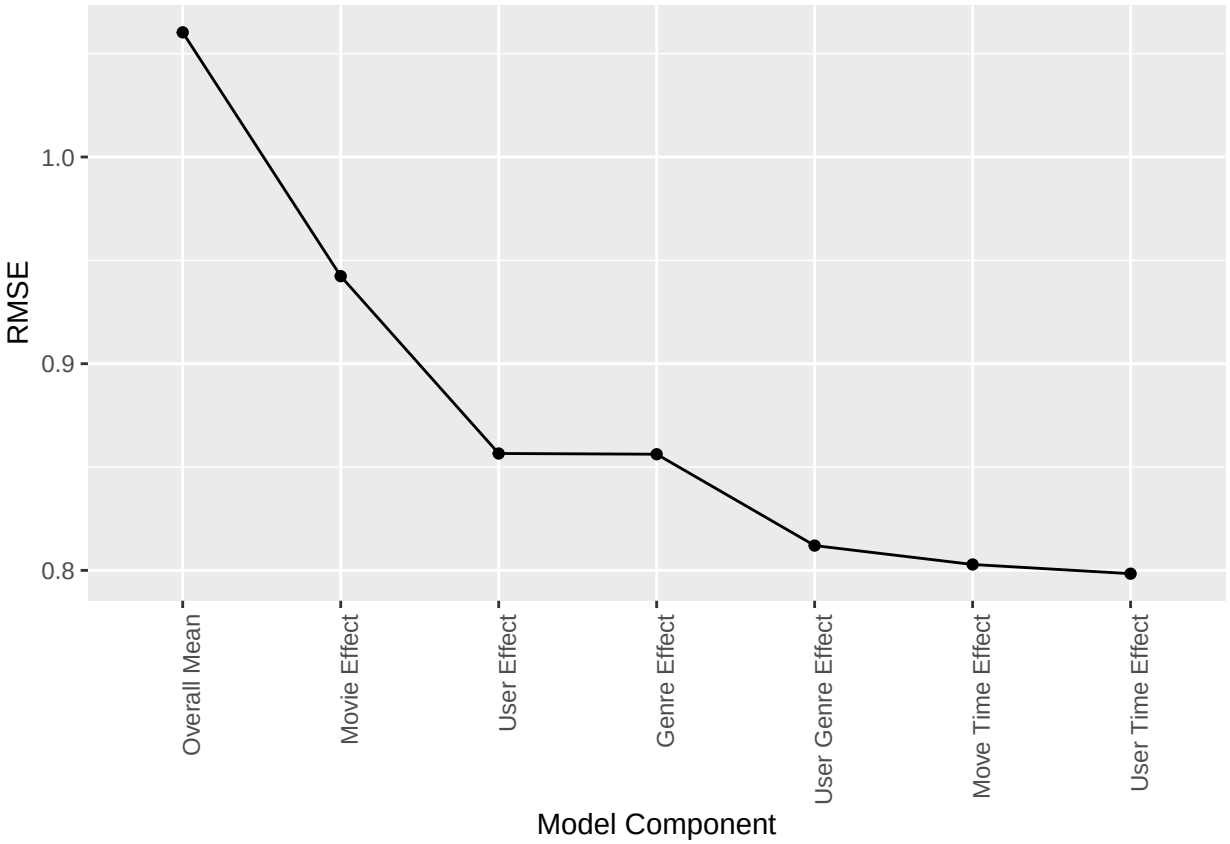
III. Results

The final rmse on the training set was 0.79840 and for the test set it was 0.85696.

The below shows all of the relevant metrics that were evaluated sequentially in the development of the model: The rmse for the train set at the point the component was introduced, the improvement over the rmse relative to the previously-introduced component, and the regularization parameter calculated and used for subsequent evaluation.

Model Component	RMSE	Incremental Improvement	Regularization parameter
Overall Mean	1.06030		
Movie Effect	0.94235	0.11798	0
User Effect	0.85655	0.085797	0
Genre Effect	0.85620	0.00035	0.01
User Genre Effect	0.81200	0.0442	0
Move Time Effect	0.80286	0.00914	0
User Time Effect	0.79840	0.0044557	0

The drop in the model's error with the introduction of each component is illustrated below:



The project's goal of obtaining a test set rmse at or below 0.86490 was achieved, suggesting that the model performs well enough to function as a predictor of individual film ratings.

IV. Discussion

The results confirm the findings of others that baseline predictors reflecting the simple main effects of movie and watcher are "at least as significant as coming up with modeling breakthroughs."[4] In the present study, these effects contributed more than any others to the model's accuracy.

The user-genre effect also appeared to be quite important, as the third most effective predictor, and notably more important than the remaining components. This finding makes sense intuitively—it basically means that people like some types of movies better than others. The finding may also have to do with the richness of genre information provided in this dataset, a potential asset that may not have been leveraged to the degree it could have been. The model development proceeded based on assumptions regarding which genres were most distinct and had the greatest potential for prediction, but these assumptions were not empirically validated. Follow-up analysis could be done on the relationships between ratings of films in different single genres and genre combinations in order to tease out the most distinct and predictive categories.

The other components added minimal or negligible incremental improvement. The user-time and movie-time predictors were both based solely on the year the film was rated, which may have been too simplistic. For example, the ratings of some films might rise around the time the Oscars are announced, or the ratings of some users might vary more systematically as a function of season rather than year. These examples, purely hypothetical, are intended to illustrate that time related interactions with films and users could have modeled in a more sophisticated manner. Further empirical analysis would be required to identify possible optimal ways of modeling these effects more precisely.

The main effect of genre was the least effective predictor, suggesting that a movie's genre in and of itself, is not predictive of its rating. The reasons might be related to those for why the user-genre was a good predictor: users prefer movies within particular genres based on their own tastes, and these preferences rather than the genre itself influence ratings. The finding could also be related to how the genre was modeled for this effect. However, some exploratory analysis using single genre categories to evaluate the genre main effect was conducted and it did not show promise.

Regularization did not improve the model's performance on the training set. The reasons for this are not known, but some authors suggest that regularization could improve a model's generalization even when it does not improve its performance on training data.[5] This phenomenon may be related to overfitting, whereby a model is tuned specifically to training data-the logic being that any biases in training a model, including those biases intended to be attenuated by regularization, reduce model errors only on the training data.

This possibility could have been evaluated by testing the regularization parameters on the test set; however, this project's design did not permit this step. The cross-validation was

[4]Koren (2009)

[5]Raschka, Sebastion. *Does regularization in logistic regression always results in better fit and better generalization?* https://sebastianraschka.com/faq/docs/regularized-logistic-regression-performance.html

intended to control for overfitting bias, but no follow-up was executed to confirm that this procedure was effective. The notable difference between the final rmses on the training vs. the test set also suggests the possibility of overfitting. Perhaps the specific way genre categories were defined led to some overfitting; as already noted, more follow-up is required in this area.

This research did not evaluate every potentially useful predictor. For instance, it did not employ matrix factorization techniques, such as principal components analysis or singular value decomposition, that identify latent user and movie related factors within the predictors. For (another purely hypothetical) example, maybe people tend to either really love or really hate movies starring Jack Nicholson; such patterns could have been used in predicting ratings. That information was not provided in the data set, but matrix factorization techniques, by identifying latent clusters of movie and user ratings that are highly correlated, could have teased predictors such as these out of the data.

In conclusion, the project goal of identifying a useful model for predicting user ratings was achieved. However, the research left a number of questions unanswered, leaving open the possibility that even better models could be developed using the same data. More research and analysis would be required to address these questions.