HarvardX Data Science Program:

PH125.9x - Capstone - MovieLens Project

Travis Horesh

June to November 2022

Introduction

The goal of this project was to build a recommendation system for movie ratings, using the knowledge and techniques developed over the course of the HarvardX Data Science program on edX. Primarily, this focused on the machine learning topics discussed during the preceding course (HarvardX PH125.8x).

The project utilized the MovieLens 10M dataset, which is a collection of 10,000,000 movie ratings maintained by GroupLens Research. Other dataset notes from GroupLens Research are: *Stable benchmark dataset. 10 million ratings and 100,000 tag applications applied to 10,000 movies by 72,000 users. Released 1/2009.* The dataset also includes genres and year of release for each movie and a timestamp for when the rating was given.

The project is inspired by the Netflix Prize challenge of October 2006. At that time, "Netflix released a dataset containing 100 million anonymous movie ratings and challenged [the broader data science] communities to develop systems that could beat the accuracy of its recommendation system, Cinematch." Ultimately, "the accuracy of [Cinematch] is determined by computing the root mean squared error (RMSE) of the system's prediction against the actual rating that a subscriber provides." And, According to this program's textbook (Introduction to Data Science by Rafael A. Irizarry, pg. 641), "To win the grand prize of $1,000,000, a participating team had to get an RMSE of about 0.857."

In summary, the project consisted of ingesting and transforming the raw dataset, creating the train, test, and validation subsets, exploring the data, and developing and refining an algorithm to predict movie ratings.

Analysis

Being new to data science, I generally stuck to the approach outlined by the textbook and extrapolated where possible. My analysis explored a number of individual and combined effects using root mean squared error (RMSE).

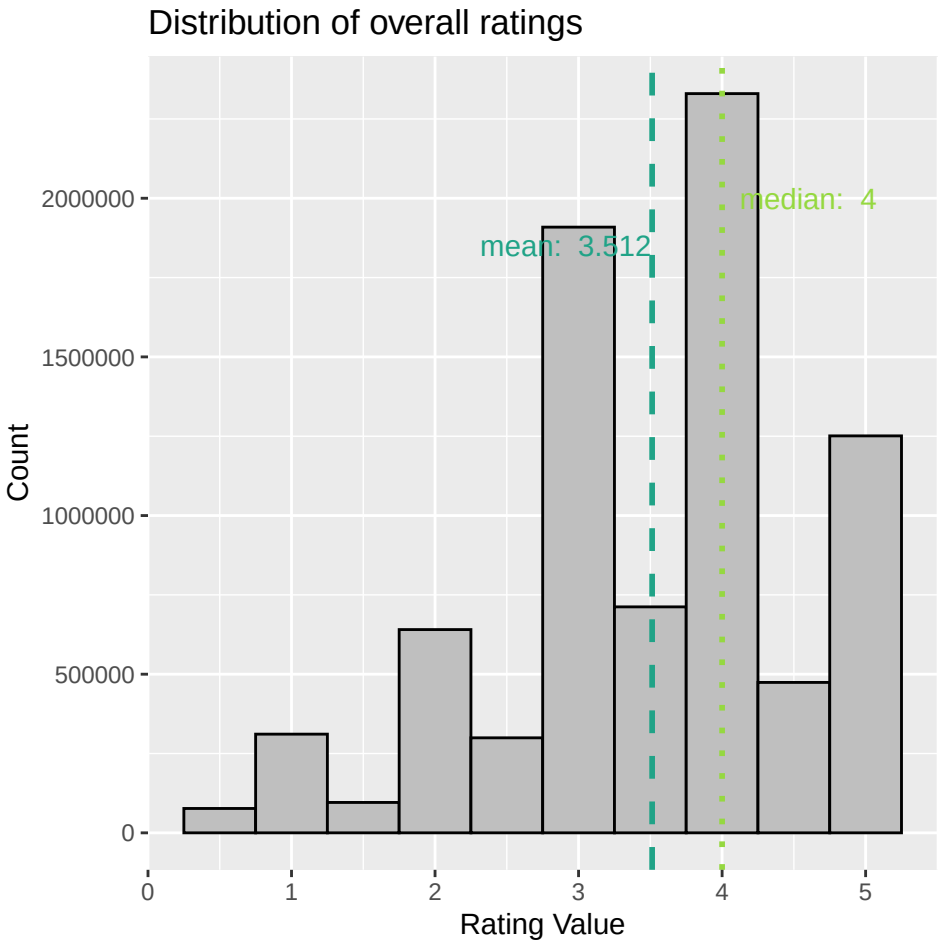
Data preparation was minimal. The only major changes I made to the dataset were: - Extracting the "primary" genre (i.e., the first) listed for each movie - Converting the serial datetime "timestamp" column to a date - Extracting the year from the timestamp column as rating year - Extracting the year of release from the movie title - Calculating the age of the rating (the time between the release of the movie and the rating) - Calculating the age of the movie (since its release)

The overall edx dataset was split 90/10, with 90% retained for use in model development and the remaining 10% set aside as a hold-out or validation dataset. The 90% retained for model development was further split into training and test sets using a test index and the createDataPartition() function, with 90% of the observations allocated to the training set and the remaining 10% to the test set. One key step in data preparation had to occur at this point - ensuring all three datasets (train, test, and validation) contained all movies and users. This was necessary for eliminating NA values introduced when later joining datasets.

The process of eliminating NAs consisted of using the semi_join() function between the training and test sets by movieId and userId, the anti_join() function to identify the rows removed from the test set, and add those rows back back to the train set using rbind(). Additionally, I set the na_matches argument to "never" for all uses of the left_join() function to ensure it operates more like the SQL join functions I am used to.

Initially, prior to implementing this data preparation step, NAs were identified using is.na() each time predictions were calculated. Any NAs were coerced to the overall rating average using if.else(). The NA correction code is still in place but is no longer needed.

Replacing NAs with the overall average was informed using a plot of the distribution of ratings, along with the mean and median ratings values. This was a good opportunity to include some ggplot2 practice. So, I embellished slightly with the labels, lines, colors, etc.



I chose to analyze a number of different effects individually, in combination, and with regularization to penalize large estimates that were formed using small sample sizes. The effects analyzed are detailed below in the bulleted lists:

Individual Effects

- Movie
- User
- Primary Genre
- Genre Combination
- Movie Age
- Rating Age

Combined Effects

- Movie and User
- Movie and Primary Genre
- Movie, User, and Primary Genre
- Movie, User, and Genre Combination
- Movie, User, and Movie Age

Regularization

- Movie
- Movie and User
- Movie, User, and Genre Combination

Based on the findings from the individual effects (i.e., those with the lowest RMSE), I proceeded to combine the individual effects in different ways. Ultimately, a combination of movie, user, and genre effects proved most accurate. Finally, some of the best performing models were repeated with regularization.

Results

Movie, User, and Genre Combination effects were found to be most informative features, and ultimately, this combination using regularization proved to be the best performing model. When applied to the validation set, the model performed marginally worse than when applied to the test set. Though, the performance trend followed that of the test set. Final best scores are listed immediately below and the overall results are contained in the Results Table, below.

Final Test Set Score = 0.86381

Final Validation Set Score = 0.86485

Final Results:

Model	RMSE_Test	RMSE_Validation
Naive	1.0600537	NA
Movie Only	0.9429615	0.9439729
User Only	0.9777090	NA
Primary Genre Only	1.0482023	NA
Genre Combo Only	1.0175012	NA
Movie Age Only	1.0492828	NA
Rating Age Only	1.0583887	NA
Movie and User	0.8646843	NA
Movie and Primary Genre	0.9429615	NA
Movie, User, Primary Genre	0.8646843	NA
Movie, User, Genre Combo	0.8643241	0.8654490
Movie, User, Movie Age	0.8643301	NA
Movie Regularized	0.9429370	NA
Movie-User Regularized	0.8641362	NA
Movie, User, Genre Combo Regularized	0.8638141	0.8648545

Conclusion

I succeeded in developing a model that performs better than simply guessing the average and is inline with some of the Netflix Prize attempts, but there is significant room for improvement still. Several features or combinations of features were not fully explored, as well as other more advanced techniques.

So, RMSE as a method of recommending movie ratings is viable but may not necessarily be the preferred method.

Overall, the project provided an excellent opportunity to test and solidify knowledge gained in the latter portions of the HarvardX Data Science program. I look forward to feedback from my peers and further improving my skills and knowledge in this area!