Movielens Project Report

Magdalena Borisova

6/10/2019

Abstract

This report presents a machine learning algorithm for movie ratings. The algorithm provides a model that improves movie rating predictions in a large dataset. The model's accuracy was measured using residual mean squared error (RMSE). The lowest RMSE achieved with the minimal number of predictors was 0.86482. The final model was selected due to its simplicity and low RMSE.

Introduction

Recommendation systems are used in a wide variety of fields as a tool that makes predictions based on a user's preferences. Companies such as Netflix, Hulu and Amazon, and institutions such as the GroupLens Research Group, use movie recommendation systems to give personalized movie recommendations to their users.

Netflix uses a machine learning algorithm that makes predictions and gives movie recommendations based on many criteria - one of which is their user's movie star ratings [3, 5]. Unlike the Netflix database, which is not readily available online, the Movielens database, created by the GroupLens Research Group, is a database that resembles the Netflix database and can be used to build and optimize a movie recommendation algorithm [2, 5, 7].

The Movielens 10M dataset is a dataset with over ten million movies ratings, which was released on the GroupLens website in 2009 [7]. The data was collected by the GroupLens Social Computing Research Group and made available to the public for educational purposes.

The goal for this project was to create a movie recommendation system by building a machine learning algorithm on the Movielens 10M dataset. The dataset was split into two sets, a train set and a validation set. A machine learning algorithm that predicts movie ratings was developed and used to train a model on the train set and was then tested on the validation set. Regularization techniques were applied to the final model to prevent overfitting.

Methods

Sample

The GroupLens Research Group at the University of Minnesota created the Movielens website as a movie recommendation engine that personalizes a user's movie watching experience, by taking the user's movie preferences as a reference guide and recommending new movies based on a custom-built user movie profile [8]. Different Movielens datasets were created using data collected from users of the Movieles website.

The most current Movielens dataset contains 27,000,000 ratings and was released in 2018 on the GroupLens website. The Grouplens Group has created various sizes of the Movielens dataset that contain different number of observations collected during several time periods [2, 8].

Part of the Movielens 10M dataset was used for data analysis and will be discussed in this report. The Movielens data was taken from the Ratings and Movies files. The analyzed data contained 10000054 observations and 6 variables: "movieId", "userId", "rating", "timestamp", "title" and "genres". There were 69878 users, 10677 movies, 10000054 ratings with 10 rating categories, 7096905 timestamp entries, 10676 movie titles and 797 movie genres included in the analysis. This Movielens dataset did not contain demographic information [7].

Table 1: Sample characteristics

movielens variables	distinct	missing
users (userId)	69878	0
movies (movieId)	10677	0
rating	10000054	0
timestamp	7096905	0
title	10676	0
genres	797	0

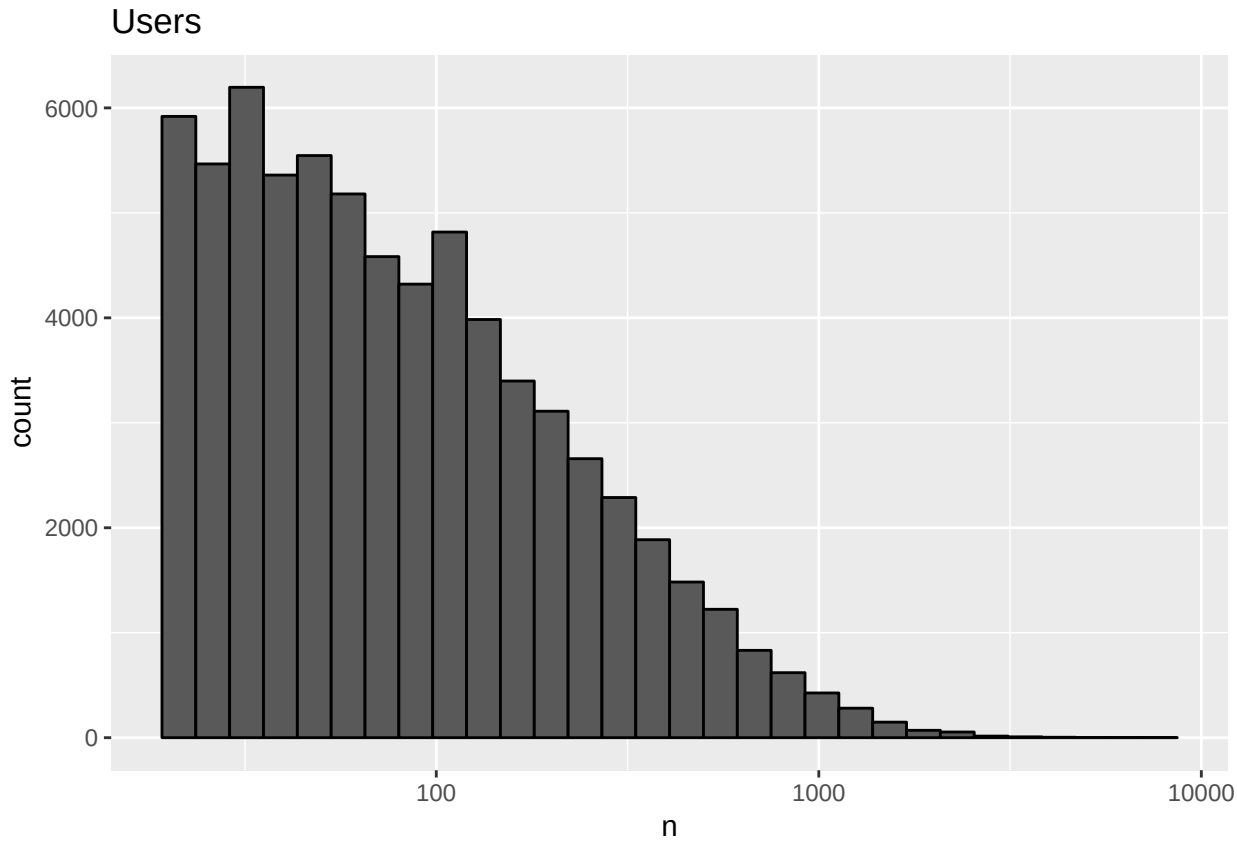
According to the Movielens 10M dataset summary, users were randomly selected and have provided a minimum of 20 movie ratings [2, 7]. Each user was given a "userId" and each movie was given a "movieId". Movie ratings represent a numeric half-star incremental rating scale with 0.5 stars being the lowest possible rating a user could give to a movie and 5 stars being the highest possible rating. The "timestamp" variable was computed by taking "seconds since midnight Coordinated Universal Time (UTC) of January 1, 1970" [7]. Movie titles were taken from the original IMDB movie titles [2, 4]. Movie genres are diverse with some genre categories representing a combination of more than one genre, such as "Forrest Gump", which was coded as a "Comedy/Drama/Romance/War".

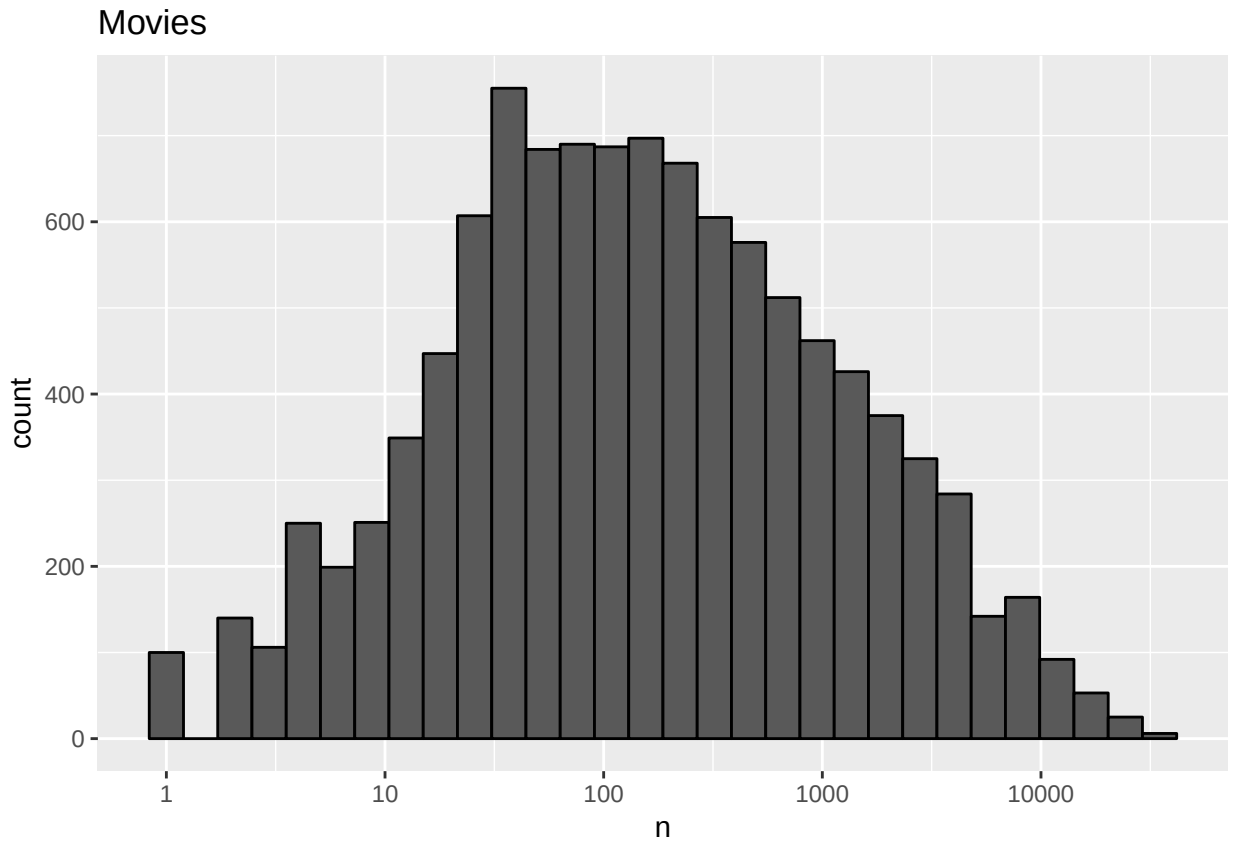
Data Cleaning, Exploration and Visualization:

The Movielens 10M dataset files were downloaded from the Grouplens website [7]. The Ratings file contained 4 variables: "userId", "movieId", "rating" and "timestamp". The Movies file contained 3 variables: "movieId", "title" and "genres".

Data cleaning consisted of splitting the string variables in the Movies file, formatting the datatype of the variables in the file, turning it into a data frame and joining it with the Ratings file into a "movielens" dataset. The dataset consisted of 6 variables: "userId", "movieId", "rating", "timestamp", "title" and "genres".

Data exploration and visualization included confirming the structure and dimensions of the movielens dataset, inspecting the first few observations in the dataset, checking for missing and distinct values for each variable in the set, calculating the observed versus expected number of movies in the set based on distinct values for users and movies, and graphing users and movies. There were no missing values for any of the 6 variables in the set. The distinct/unique number for each one of the 6 variables is listed in the Sample section above. There were 69878 distinct users and 10677 distinct movies, which should have led to a combined product of 746087406 expected ratings given the number of users and movies in the dataset, however the observed number of ratings was only 10000054. Log scale was used for the users and movies graphs to help with skewness by producing better visual representation of the data. The appearance of the distribution in the graphs of users and movies suggested that some users rated more movies that other users and some movies had more ratings that other movies.


Users
6000
4000
2000
0
count
100
1000
10000
n


Movies
600
400
200
0
count
1
10
100
1000
10000
n

Insights

The visual insight gained from the graphs was consistent with the information accumulated from the data exploration of the observed versus expected number of ratings in the movielens dataset. Some possible explanations for the trends seen through the data exploration and visualization for users could be that user rating activity is related to individual characteristics, timing of movie watching, devices used for movie watching, and length of time spent on movie watching [3]. For movies, the trend might be related to movie popularity and genre [5]. Most of these variables were not present in the dataset to allow for further analysis.

Training and test set

The movielens dataset was split into a train set named "edx" and a test set named "validation". The validation set was 10% of the movielens dataset. MovieId's and userId's in the validation set were also present in the edx set. Additional movieId's and userId's present in the validation set but not present in the edx set were removed from the validation set and added back into the edx set.

Modeling approach

A machine learning algorithm was built to predict movie ratings. The accuracy of the trained model was measured by computing RMSE. The model was optimized by minimizing the mean squared error (RMSE) in the validation set. Different models were tested. The final model was selected based on the lowest RMSE given the least number of predictors.

Model 1 "Just the Average": Y = mu_hat + e

Model 1 took a naïve approach to predicting movie ratings and was based on the assumption that all differences from the average rating were due to random variation in the distribution.

Model 2 "Movie Effects Model": Y_m = mu_hat + b_hat_m + e_m

From the movie graph in the Data visualization portion of the report, it could be seen that not all movies were rated the same. Model 2 explored the difference in ratings due to the effects of different movies.

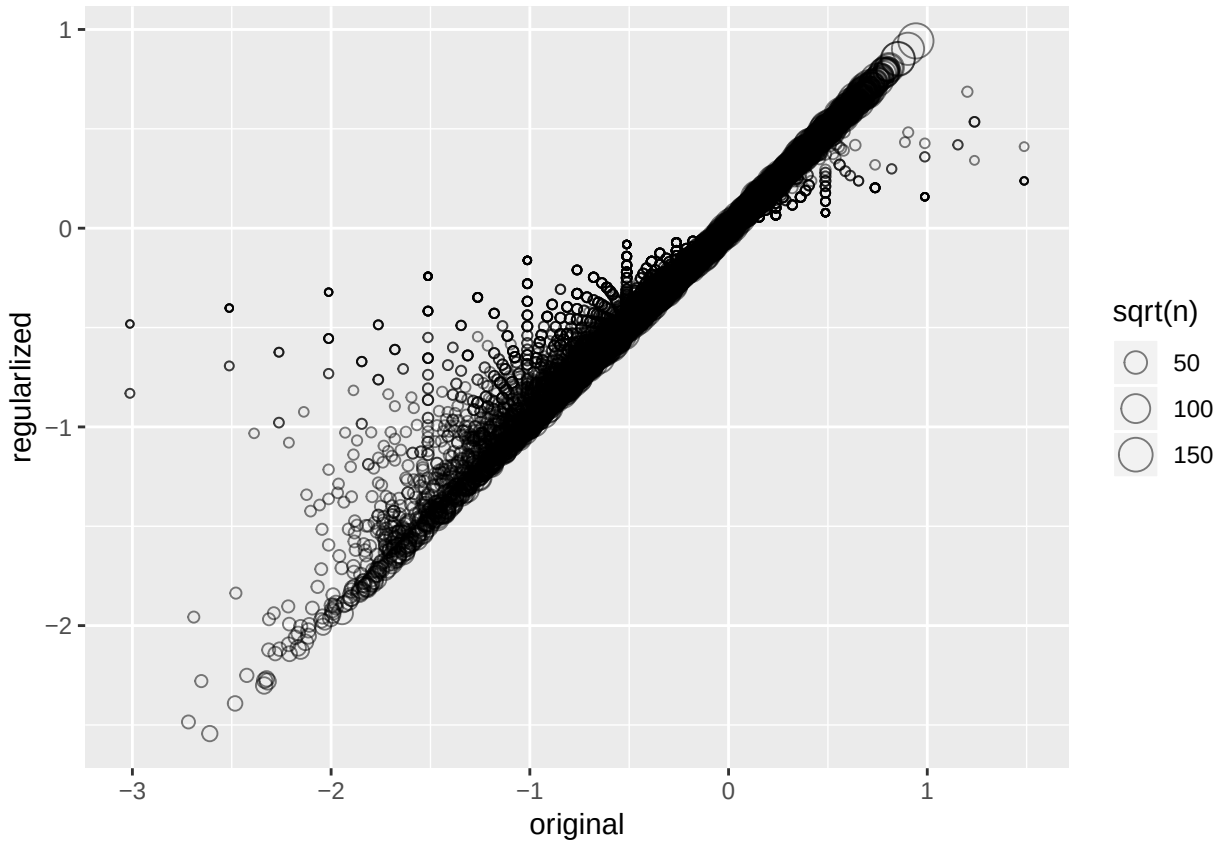
Model 3 "Movie and User Effects Model": Y_mu,u = mu_hat + b_hat_m + b_hat_u + e_m,u

From the user graph in the Data visualization portion of the report, it could be seen that not all users rated movies the same. Model 3 explored the difference in rating due the effects of both the different movies and users.

Other models were trained including the addition of an interaction term and the inclusion of the genre and title variables. However, these models did not make the cut because the RMSE achieved for most of them was not lower than the RMSE achieved by Model 3.

Model 4 "Regularized Movie and User Effects Model"

Because some model variation in Model 3 was suspected to be due to very high or vary low ratings given by only a small number of users, regularization was applied to the model. After regularization, the model RMSE decreased suggesting some model overfitting was present. Cross validation was used to pick an optimal lambda parameter for the regularized model.



Results

Different models were trained to predict movie ratings in the validation set. The goal was to train a model and test it on the validation set to produce a RMSE lower than or equal to 0.87750. Model 4, "Regularized Movie and User Effects Model", produced a RMSE of 0.86482 with two predictors – movieId and userId. The beauty of this model is that it is simple, thus it can be easily taken and reproduced for another dataset because it is not too specific to this particular dataset.

More predictors were added but the RMSE did not improve substantially. These additional models greatly contributed to the choice of the final model. Although model 10 "Movie, User and Title Effects Model", not described above, produced the lowest RMSE, it incorporated an additional predictor and it did not lower the RMSE by more than 1% to justify the addition of an extra predictor to Model 3 ("Movie and User Effects Model"). After examining the more models and the RMSE they produced, Model 3 was selected for further analysis and was regularized to prevent overfitting. It produced the final model – Model 4 ("Regularized Movie and User Effects Model").

Table 2: Models

Model	RMSE
Just the Average Mode	1.0612018
Movie Effect Model	0.9439087
Movie + User Effects Model	0.8653488
Regularized Movie + User Effects Model	0.8648170
Movie + User + Title Effects Model***	0.8640972

*** Additonal model discussed above but not selected as the final model

Conclusion

Key Findings

MovieId and userID were the two predictors included in the final model trained to predict movie ratings. Although the final model did not produce the lowest RMSE, it was selected using an additional criterion – simplicity.

Limitations

Model 4 did not address the "timestamp" variable in the dataset. This temporal variable could have provided valuable information about timing of movie watching and whether has an effect on movie ratings [3]. This report also did not address item-to-item collaborative filtering, matrix factorization or ensemble methods as possible training model techniques [1, 6]. The dataset used for this analysis did not provide demographic information or information related to user preferences on movie watching, such as devices used for movie watching.

Future areas of research

Adding a temporal component to the prediction model is a potential next step that could highlight new and unexpected relationships between movie ratings and movie popularity, user activity, and time of movie watching.

References

[1] Chen, E. Winning the Netflix Prize: A Summary [Blog post]. Retrieved from (http://blog.echen.me/2011/10/24/winning-the-netflix-prize-a-summary/)

[2] Harper F. M. and Konstan, J. A. (2015). The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems (TiiS) 5(4), 19.* doi: (http://dx.doi.org/10.1145/2827872)

[3] How Netflix's Recommendation System Works. Retrieved from (https://help.netflix.com/en/node/100639)

[4] IDBM Website. Retrieved from (https://www.imdb.com)

[5] Irizarry, R. (2019). Introduction to Data Science [Book]. Retrieved from (https://rafalab.github.io/dsbook/large-datasets.html#recommendation-systems)

[6] Koren, Y. (2009). The BellKor Solution to the Netflix Grand Prize [PDF file]. Retrieved from (https://www.netflixprize.com/assets/GrandPrize2009_BPC_BellKor.pdf)

[7] Movielens Database. Retrieved from (https://grouplens.org/datasets/movielens/)

[8] Movielens Website. Retrieved from (https://movielens.org)