Machine Learning: Movielens Project Report

Xiaoqing Wu

6/5/2020

1. Introduction

This document describes a machine learning project using the 10-million movie rating MovieLens data hosted on grouplens, it's a part of Harvard edX Data Science course and the project goal is to develop a movie recommendation system with a loss error lower than Root Mean Squared Error (RMSE) 0.86490.

Machine Learning has two common approaches: supervised learning and unsupervised learning. Supervised learning involves data visualization, data analysis, manual training and tuning data models, whereas unsupervised learning focuses on finding predictors to group observations as clusters and the prediction would be to identify the right clusters for new data entries. This document reports the exploration and findings from a supervised study of movielens.

Supervised learning generally has the following phases and each phase will be discussed in a corresponding section:

- Visualizing and analysing data in order to identify predictors that determine rating outcomes
- Wrangling data
- Developing training model for the machine to learn behaviors from observations
- Applying the training model to test data set to obtain predictions
- Measuring the accuracy of the predictions against true results and evaluating the performance of the recommendation system

2. Methods and Analysis

2.1. Training and Validation Separation

In a supervised machine learning process, observation data is typically split into a training set and a validation set. The training set is used to learn predictor behaviors and create recommendation algorithms, the validation set is used to produce final prediction result. The movielens data was split into a 9 million ratings training set, called edx and a 1 million ratings validation set, called validation.

Table 1: edx Data Set Summary

User Count	Movie Title Count	Genres Count	Rating Count
69878	10677	797	9000055

Table 2: validation Data Set Summary

User Count	Movie Title Count	Genres Count	Rating Count
68534	9809	773	999999

2.2. Data Wrangling

When the project started a curiosity question was raised about how 'old' movies were rated and whether a movie's longevity affected rating. In order to find out the movie age's effect, two year related columns were added in the datasets and timestamp was formatted to date. Below are two examples:

Table 3: New edx data

userId	movieId	rating	title	genres	rlyear	rtdate	mvage
1	122	5	Boomerang (1992)	Comedy|Romance	1992	1996-08-04	4
1	185	5	Net, The (1995)	Action|Crime|Thriller	1995	1996-08-04	1

rlyear: Movie release year extracted from title

mvage: Movie age: the year of rating - release year

2.3. Data Exploration and Visualization

2.3.1. User Data Analysis

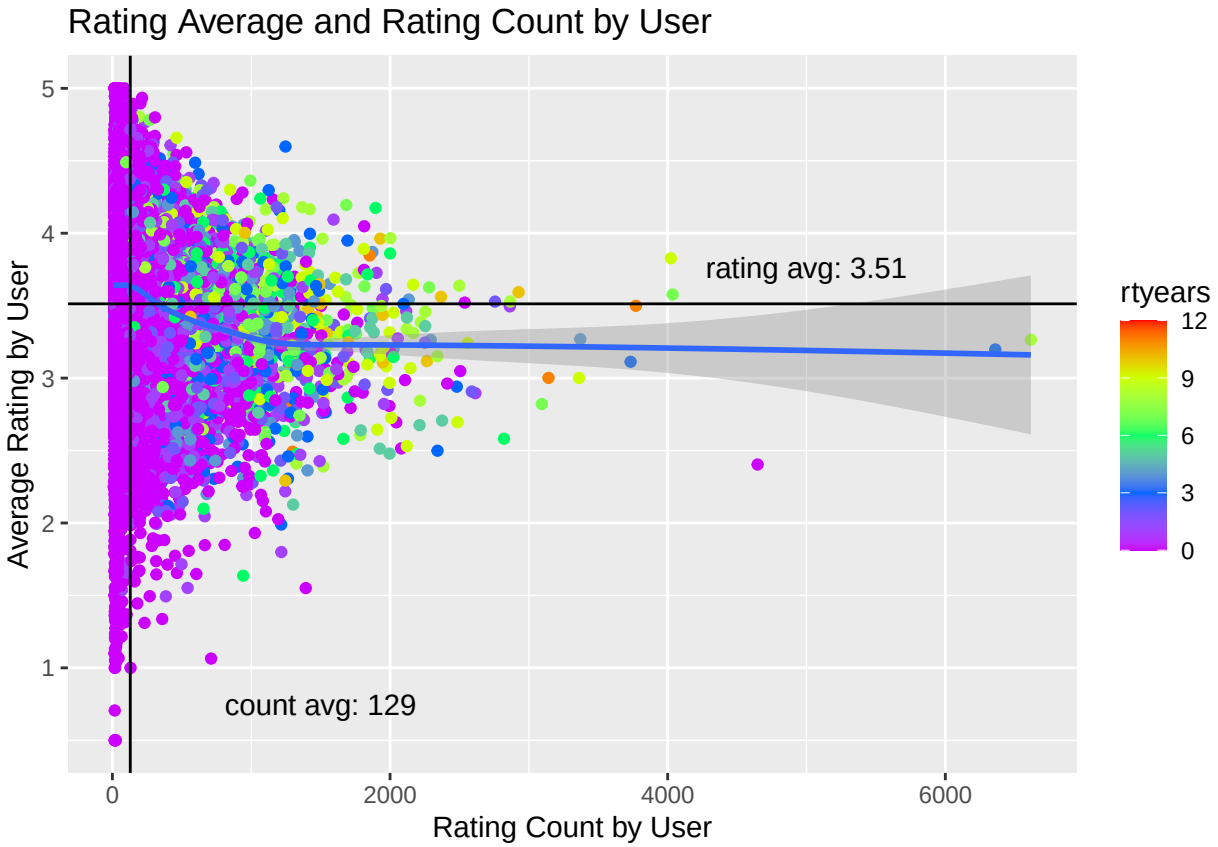


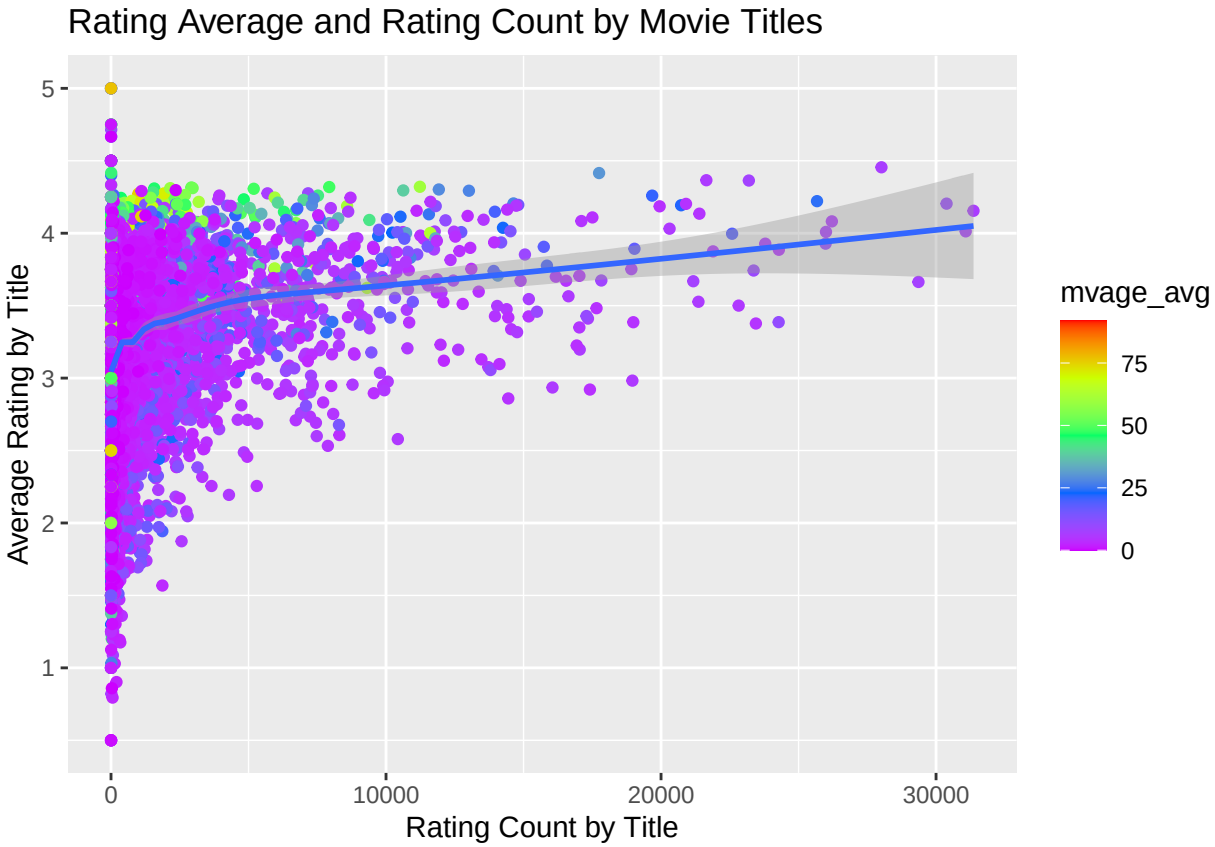
Table 4: User Statistics

Average Number of Ratings per User	Rating Average	Average Rating Experience (years)
129	3.51	0.402

The average number of ratings per user was 129, very few rated less than 10 or more than 6,600 movies. Rating years, rtyears, is the number of years an user participated in movie rating, if the user entered 1st rating in 2015 and last 2018, his/her rating experience would be 3 years (2018 - 2015), average rating experience is 0.4 year.

The plot above shows that rating variation stabilized as viewers became experienced in rating, viewers with 3-6 years experience rated movies in the range of 2 to 4.5 stars, those with fewer than 1.5 years experience spreaded rating to 0.5 to 5 stars.

2.3.2. Title Analysis



The Rating Average and Rating Count by Movie Titles plot shows: Rating average and the number of ratings have a slight up trend correlation, Compared to the 3.51 rating average, movies that had more than 20,000 ratings saw higher average ratings in the range of 3.7 to 4 stars.

2.3.3. Temporal Analysis

The second observation from The Rating Average and Rating Count by Movie Titles plot is that the movie 'age' had an effect on rating. Movie 'age', mvage, is defined as (the year of the rating) minus (movie release year), legendary movies (green dots) are the movies that had been released for 30-50 years when rated, they received 4.25 average rating which is 0.75 point higher than the 3.51 overall average.

2.3.4. Algorithm and Modeling approach

Based on insights gained from data analysis, four factors, movie item, user, genres, and movie age are selected as predictors and their effects are gathered by three linear regression models. In Netflix challenge, predictor effect is called bias hence bias and predictor effect are used interchangeably in this document.

To avoid over training edx was split to train_set and test_set; all models were run with the train_set and test_set, the validation set was only used to obtain final rmse.

2.3.4.1 Regularization

Movie rating often has noise formed by small sample sizes, to offset the noise predictor biases were calculated by Penalized Least Squares (PLS). In PLS λ is a tuning parameter, when sample size n is small, λ makes $\hat{b}_i(\lambda)$ smaller, when n is large the effect of λ diminishes.

$$\hat{b_i}(\lambda) = \frac{1}{\lambda + n_i} \sum_{u=1}^{n_i} (\hat{Y}_{u,i} - \hat{\mu})$$

2.3.4.2 Linear Regression Baseline Models

Following three baselines were created to quantify predictor behaviors:

Baseline 1 Prediction Model: Predicted Rating $\hat{Y}_{u,i} = \mu$
Baseline 2 Prediction Model: Predicted Rating $\hat{Y}_{u,i} = \mu + b_u + b_i$
Baseline 3 Prediction Model: Predicted Rating $\hat{Y}_{u,i} = \mu + b_u + b_i + b_g + b_{i,t}$

μ: average of movie ratings

b_i: movie i's rating bias, average of $y_{u,i} - \mu$

b_u: user u's rating bias, average of $y_{u,i} - \mu - b_i$

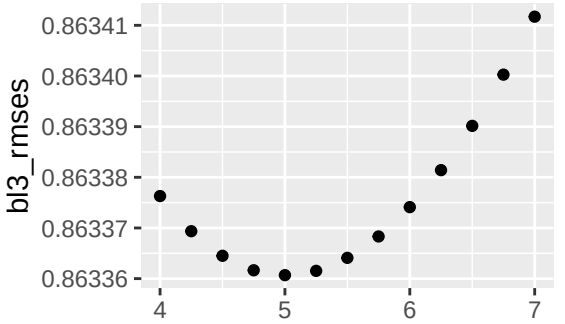
$b_{i,t}$: movie i's 'age' bias:

- t is the years between the movie release year and the year of rating
- average of $y_{u,i} - \mu - b_i - b_u$

b_g: genres g's rating bias, average of $y_{u,i} - \mu - b_i - b_u - b_t$

2.3.4.3. Tuning Parameter

An optimal λ should be the one that yields the lowest RMSE. Baseline 3's best lambda was 5 and corresponding RMSE was 0.86336, 5 was chosen after running the model iteratively with a number sequence (4, 7, 0.25)



3. Results

3.1. Prediction Accuracy

Prediction accuracy is measured by RMSE, it calculates the loss error between the predicted ratings and actual ratings in the test set.

$$RMSE = \sqrt{\frac{1}{N}\sum_{u,i}(\hat{y}_{u,i} - y_{u,i})^2}$$

N: the number of user/movie combinations

$y_{u,i}$: user u's rating of movie i

$\hat{y}_{u,i}$: user u's predicted rating of movie i

An RMSE 0.5 means the prediction is 0.5 stars off from true rating.

The RMSEs from above baselines and the final run are:

Table 5: RMSE for mu + b_i + b_u + b_t + b_g

	method	RMSE	best_lambda
bl1	Baseline 1: predicted_rating = mu	1.06005	NA
bl2	Baseline 2: predicted_rating = mu + b_i + b_u	0.86414	5
bl3	Baseline 3: predicted_rating = mu + b_i + b_u + b_t + b_g	0.86336	5
final	Final: predicted_rating = mu + b_i + b_u + b_t + b_g	0.86398	5
\			

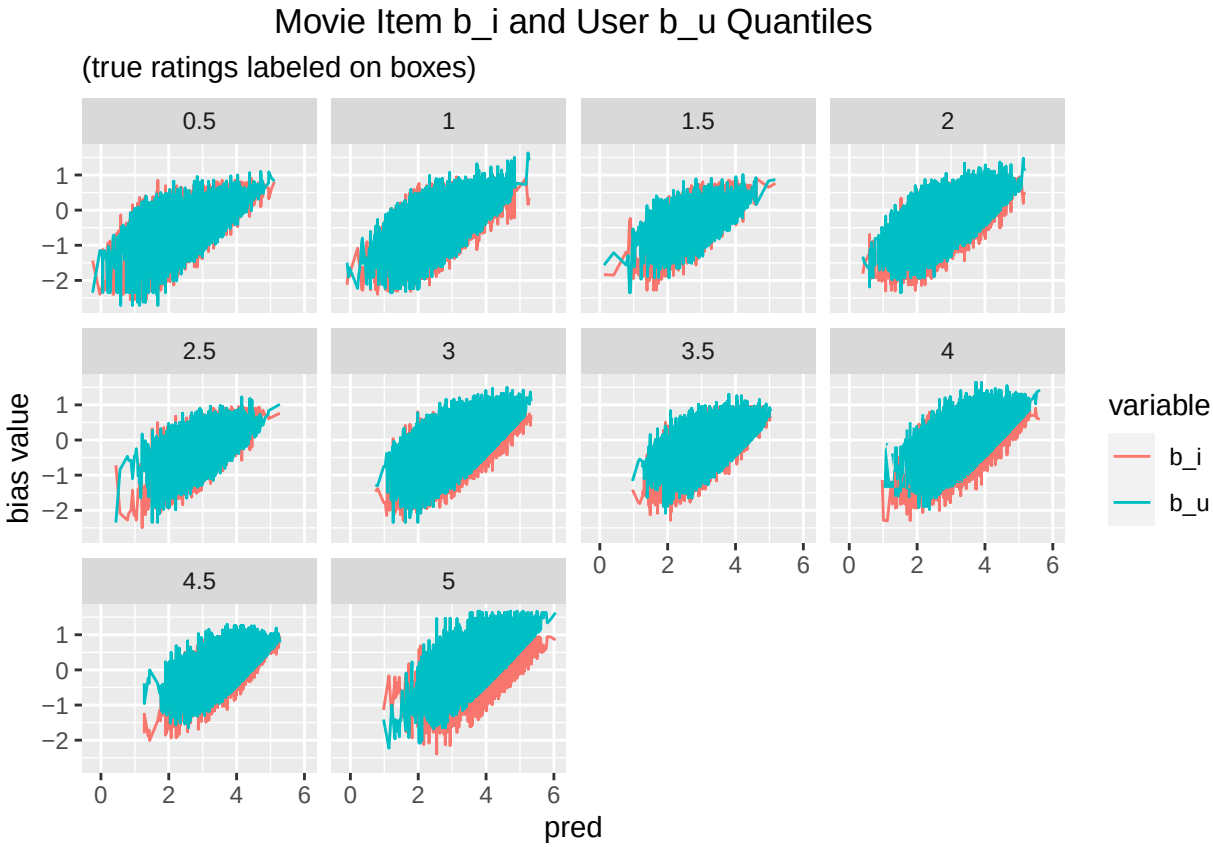
3.2. Rating Changes by Predictors

Predictors Quantiles table below shows that movie item b_i and user biase b_u have much higher absolute values than genres and movie age and hence are principal components in rating predictions.

Table 6: Predictors Quantiles

	0%	25%	50%	75%	100%
b_i	-2.5465	-0.2936	0.0782	0.3629	0.9425
b_u	-2.71213	-0.22790	0.00896	0.23670	1.66964
b_g	-3.54e-01	-1.35e-02	3.41e-05	1.92e-02	3.33e-01
b_t	-0.11584	-0.02082	-0.00121	0.02100	0.16283
\					

To understand the effects of predictor biases further, 500,000 randomly samples were selected from the validation test result and created bias plots. Movie Bias b_i and User Bias b_u Quantiles illustrate that movie title and user predictors made the most changes to high true ratings (5 & 4.5), and low true ratings (0.5-2.0).



3.3. Final Result

Prediction model: $\hat{Y}_{u,i} = \mu + b_u + b_i + b_g + b_{i,a}$

RMSE: 0.86398

lambda: 5 (from baseline 3)

3.4. Performance

The prediction model ran on a power lacking 2013 MacBook Pro with 8 GB Memory and 2.4 GHz Dual-Core Intel Core i5. Running baseline 3 took 24 seconds. The entire script took about 9 minutes including downloading the data file, wrangling data, creating edx and validation summaries, running three baselines and creating predictors value plots.

4. Conclusion

This project created a movie recommendation model by learning behaviors of four predictors from a 10-million movie rating dataset, the algorithm is simple and memory efficient.

This study was limited to linear regression because running any other models would run out of memory and crash R.

Another limitation is the two different rating number schemes used in this study. The predicted rating uses continuous numerics whereas the true rating is a half-star numbering system, comparing them with different precisions made assessing accuracy less than ideal.

This is a preliminary step in machine learning; much more can be done in future. For example developing user clusters and making the recommendation leverage user clusters can potentially help improve prediction accuracy, deeper investigation should be performed to uncover implied bias such as how user preference change over time affects movie rating.

A next major step will be to find adequate hardware to study unsupervised machine learning using similar data.

References

I. Rafael and Harvard edX. Introduction to Data Science 2020

Y. Koren. The BellKor Solution to the Netflix Grand Prize 2009