Movie Recommendation Project

Sujata Verma

9/1/2020

Table of Contents

Introduction

The ** objective** of this project is build an algorithm to predict the rating a viewer(called user) will give to a movie in the range of 0.5 (worst rating) to 5(best rating). The Movielens dataset consisting of viewer ratings of movies of variuos genres, comedy, romance etc. for the years 1995 to 2009 will be used for analysis.

I will identify the best machine learning model for predicting the user rating by fitting a few regression models to the data and calculating the Root Mean Square Error. All models with RMSE of less than 0.80 will be acceptable and the model with the lowest RMSE will be deemed best.

The key steps will be to first explore the data, data wrangling (e.g. converting timestamp into years, dividing the data set into train, test and validation sets), fitting a baseline genralized linear regression model. Lastly,the baseline model will be enhanced by fitting two additional models and to check if we get a lower RMSE.

DATA WRANGLING

In this section, the data is imported and observations with missing values are removed. The data is prepared for analysis.

The first step is to load the libraries and options.The second step is to load the Movielens data set consisting of more than 10 million ratings of 10,000 movies.

The data set, Movielens, specifically, consists of the following variables, movie titles,genre,timestamp,userId,movieId and rating.Not all users rated all the movies, so there are lot of NA's in the dataset.Entire data matrix can be used as predictors of each cell, making it hard to predict how a user will rate a movie.

The Movielens dataset has a 'timestamp' variable about when the movie was released. It will be useful to convert it into 'year' variable for better interpretation.

userId	movieId	rating	title	genres	year
1	122	5	Boomerang (1992)	Comedy|Romance	1996
1	185	5	Net, The (1995)	Action|Crime|Thriller	1996
1	231	5	Dumb & Dumber (1994)	Comedy	1996
1	292	5	Outbreak (1995)	Action|Drama|Sci-Fi|Thriller	1996
1	316	5	Stargate (1994)	Action|Adventure|Sci-Fi	1996
1	329	5	Star Trek: Generations (1994)	Action|Adventure|Drama|Sci-Fi	1996

Next we create a subset for data exploration and data analysis. We want to separate out 10% of data for validation and use the rest of the data for predicting. In the end, we will apply our best predictive model to the validation set to validate our results.

The dataset movielens is split into validation set(10%) and rest 90% of the data(edx) will be further split equally between train set and test set. The edx dta set has around 9 million movie ratings, and has six variables: MovieID, userID, Movie Title, Genre, Year and Rating. Each movie can have mulitple ratings.

Now we have two sets of data-edx and validation. In the next section, data exploration will be performed on the edx data set.

DATA EXPLORATION

In this section, summary statistics will performed on the edx data set to get familiar with it.

We begin by looking at how many unique movies and users are in the edx dataset.

```
##   n_users n_movies
## 1   69878    10677
```

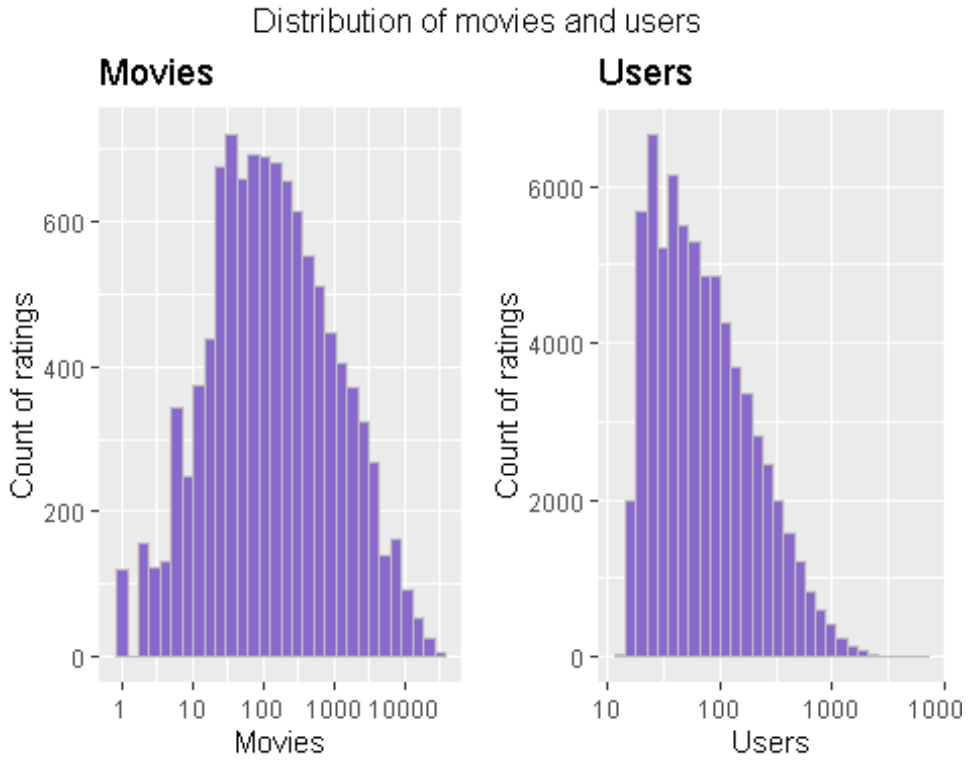
Next we find out the most given ratings in order from most to least.

```
## # A tibble: 10 x 2
##    rating   count
##     <dbl>   <int>
##  1    4   2588021
##  2    3   2121638
##  3    5   1390541
##  4    3.5  792037
##  5    2    710998
##  6    4.5  526309
##  7    1    345935
##  8    2.5  332783
##  9    1.5  106379
## 10    0.5   85420
```

Next we explore all the variables by looking at the summary statistics.

```
##      userId         movieId          rating         title
##  Min.   :    1   Min.   :    1   Min.   :0.500   Length:9000061
##  1st Qu.:18122   1st Qu.:  648   1st Qu.:3.000   Class :character
##  Median :35743   Median : 1834   Median :4.000   Mode  :character
##  Mean   :35869   Mean   : 4120   Mean   :3.512
##  3rd Qu.:53602   3rd Qu.: 3624   3rd Qu.:4.000
##  Max.   :71567   Max.   :65133   Max.   :5.000
##     genres               year
##  Length:9000061     Min.   :1995
##  Class :character   1st Qu.:2000
##  Mode  :character   Median :2002
##                     Mean   :2002
##                     3rd Qu.:2005
##                     Max.   :2009
```

Next, the distribution of movies and users will be visualised to deepen our understanding of these variables.



The above chart shows that some movies get rated more than others and also some users rate movies more actively than others.

METHOD AND ANALYSIS

In the following section, various machine learning models are built to predict the user ratings. Each model is built by using the caret package, and applying it to our edx data set. The data set is first split evenly between the training set and testing set. To make sure we don't include movies and users in the test set that don't appear in the training set, we use the semi_join function.

In the following section,six models will be fitted. The different models are:

1. Baseline Model: Mean-only Regression Model
2. Movie-effects Regression Model
3. Movie and User-effects Regression Model
4. Movie and User-effects Regression Model with regularization
5. Time-effects Model
6. User Score Model

For each model, the following steps are taken:

1. Training the model using the training set
2. Making predictions about rating using the test set
3. Determining the RMSE of the predictions
4. Storing the results of the model in a results table

Model 1. Mean only Regression Model

The first model predicts that the user will rate the movie equal to the average of all ratings in the data set.

Define Yu,i as the rating for movie i, by user,u. μ is the mean rating.

εu,i are the independent errors.

Thus the equation we are trying to fit is Yu,i = μ + εu,i

```
## [1] 3.512
```

Model	RMSE
Mean only Model	1.06

Model 1 doesn't give us the target RMSE of less than 0.80.

Model 2: Movie-effects Model

Adding movie-effects: Is their varibility in ratings due to differences in movies? We add the term bi to represent average rating for movie i. Now out model becomes Yu,i = μ + bi + εu,i

Model	RMSE
Mean only Model	1.0602
Movie Effects Model	0.9439

Movie effects model lowered the RMSE. We can also take into account user effects to look into the question- do different users differ in terms of how they rate movies?

Model 3: Movie and user-effects Model

We add the term bu to represent average rating for user, u So, the model we are trying to estimate becomes Yu,i = μ + bi + bu + εu,i

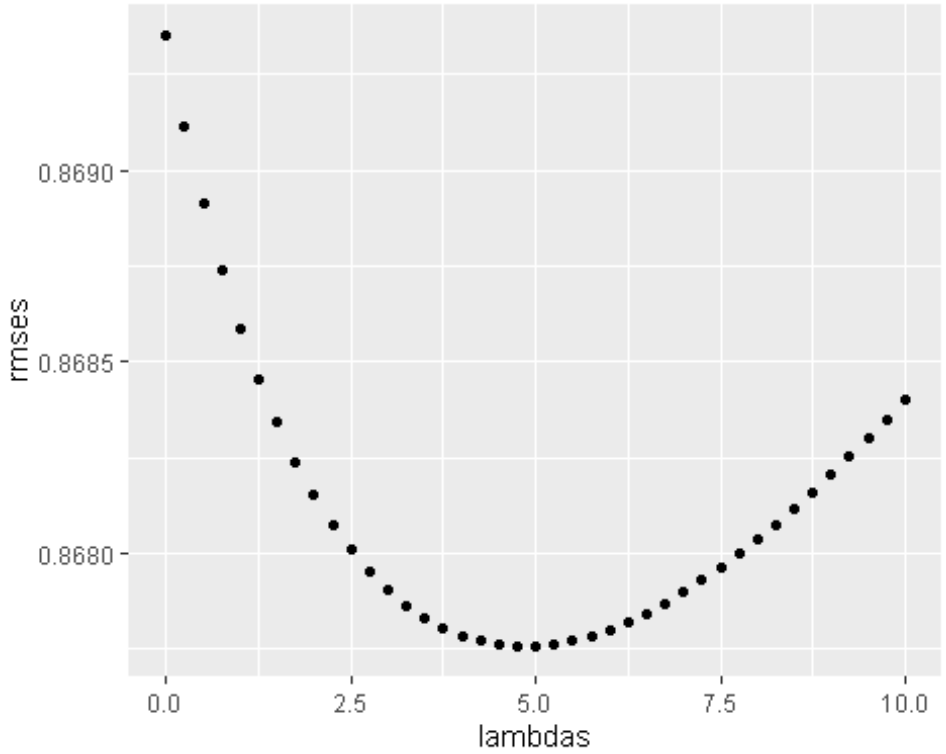
Model	RMSE
Mean only Model	1.0602
Movie Effects Model	0.9439
Movie and User Effects Model	0.8694

In Model 3, the RMSE decreases significantly when we add the user effects to movie effects and mean ratings.In the next model, we use the process of 'regularization' to remove the

disproportionate effects of movies with few ratings and users with few ratings by adding a penalty variable, lambda.

Models 4 : Movie and user effects model with regularization

We regularize the previous model, with optimized lambda to prevent overfitting.




```
## [1] 4.75
```

Next, we repaet Model 4 with the optimized value of lambda found in the previous section.

Model	RMSE
Mean only Model	1.0602
Movie Effects Model	0.9439
Movie and User Effects Model	0.8694
Regularized Movie and User Effects Model	0.8678

Model 4 did reduce the RMSE a little but did not significantly reduce it. So, instead of regularization, we will experiment with adding the time effects in the next model to explore if adding a time variable explains the variations in ratings. We add bt to capture the effect of the year on explaining the varibility in ratingst, so our equation now becomes $Yu,i = \mu + bi + bu + bt + \varepsilon u,i,t$

Model 5: Time-effects Model

Add time effects to the movie and user-effects model (no regularization)

Model	RMSE
Mean only Model	1.0602
Movie Effects Model	0.9439
Movie and User Effects Model	0.8694
Regularized Movie and User Effects Model	0.8678
Time Effects Model	0.8693

Adding the time effects to the movie and user effects model(without regularization) and then estimating the RMSE resulted in RMSE of 0.8697. It didn't perform better than the movie and user effects model with regularization, thus I didn't pursue it further.

Model 6: User Score Model

I tried another model and called it User Score Model, a combination of movie-user effects. It included the following steps:

A.Since the training set was very large, I subsetted it to include 500 movies with most ratings and 500 most prolific users to fit the models.Similarly, the test set was subsetted to include the same movieId and userId as the training subset. Note that the two subsets have different number of observations though the variables are the same, userId, movieId and rating.

```
## [1] 61104     3

## [1] 60369     3

## [1] FALSE
```

B. I calculated the average rating for all the movies in the train-subset.First, I converted the train_subset into a matrix,train_mat then found out the column means,col_means. These are the average ratings for all of the movies in the data base. Next step is to subtract the average movie rating from each of the user ratings to give us the divergence matrix. There are a number of NA's since not each user rated each movie. Each value in the divergence matrix gives us how a user rating diverged from the average rating of the movie.

```
##            1       2       3  4     5      6
## [1,]      NA      NA      NA NA    NA     NA
## [2,]      NA -0.9186      NA NA    NA     NA
## [3,] -0.3778      NA      NA NA    NA -1.474
## [4,]  0.1222      NA -0.1757 NA -0.75     NA
## [5,]      NA      NA  1.3243 NA    NA     NA
## [6,]      NA -1.9186  1.3243 NA  0.75     NA
```

C. For each user,u, I calculated the divergence of their rating from the average movie rating (ignoring NA's) of a movie,i, as shown above. The next step was to calculate the mean of this divergence among all movies each user had rated. I called it the score. Next, I combined the userId with their score. Below is the userId in the first column and their user score in the second column.

1860	-0.1713
1918	-0.1818
3810	-0.6976
3817	-0.0231
5134	0.2586
5678	0.1225

D. Now we can predict the rating a user will give any movie. We can simply add their user score to the average rating for that movie. More specifically, the prediction of user u for movie i is computed by adding the user-score, given by variable 'score' to the average of movie,i.

E.Let us take a specific example. Let us see how the model would predict all the missing ratings within the data set for userId 1860, our first user in the train subset. We add back the movie titles, given in the first column below. The column means gives us the average rating for that movie.since we know that this user diverges from average movie score by -0.1713 (from row 1,colmn 1 of the user-score matrix),we can predict rating for each movie by subtracting it from movie averages for all movies. In column 3 gives the rating user 1860 would have given each movie.

```
##                                        col_means user_1860_rating
## 1 "Toy Story (1995)"                   "3.878"   "3.7067"
## 2 "Jumanji (1995)"                     "2.919"   "2.7477"
## 3 "Grumpier Old Men (1995)"            "2.676"   "2.5047"
## 4 "Waiting to Exhale (1995)"           "2.438"   "2.2667"
## 5 "Father of the Bride Part II (1995)" "2.75"    "2.5787"
## 6 "Heat (1995)"                        "3.974"   "3.8027"
```

F.Next step is to find out the RMSE for the test-subset.It is important to note that the user-scores and prediction was done separately within the testing model and RMSE was calculated. The reason is that training and testing subsets differ in the movie-user combination as we saw earlier they are not identical in dimensions.So next, we apply the User Score Model to the test set and calculate the RMSE

Model	RMSE
Mean only Model	1.060
Movie Effects Model	0.944
Movie and User Effects Model	0.869
Regularized Movie and User Effects Model	0.868

Time Effects Model	0.869
User Score Model	0.778

From the above table, we can see that this model gives us the lowest RMSE and it below our target of 0.80. Thus we will apply this model to the validation set.

G.Lastly, we apply the score-model to the validation set and calculated the RMSE. The important caveat being that training subset and validation set had different movies and users, so the prediction had to be created within the validation set for all the movies and users.

Model	RMSE
Mean only Model	1.060
Movie Effects Model	0.944
Movie and User Effects Model	0.869
Regularized Movie and User Effects Model	0.868
Time Effects Model	0.869
User Score Model	0.778
Validation User Score Model	0.825

RESULTS

Summary of Results:

Model	RMSE
Mean only Model	1.060
Movie Effects Model	0.944
Movie and User Effects Model	0.869
Regularized Movie and User Effects Model	0.868
Time Effects Model	0.869
User Score Model	0.778
Validation User Score Model	0.825

From the results table, we note that there was a large decrease in RMSE betwen movie-effects only(Model 2) and user-movie effects(Model 3), showing that characteristics of users were the most important factor in predicting ratings.

The best results were obtained by the User score Model (Model 6) since it had the lowest RMSE. This was validated using the validation set.

CONCLUSIONS,LIMITATIONS AND RECOMMENDATIONS

I will discuss the two extensions I added to enhance the baseline model.

The Time-effects Model asked if the variation in year given in the dataset had any significant effect on how users rated the movies.No significant effect was found as the RMSE remained exactly the same as the movie and user effects. This is probably due to the short timespan 1995-2009. A further enhancement can be to add the regularization model and check if we get a lower RMSE.

I formulated the User Score algorithm, but it was hard to test in the conventional way. The idea behind the user score-method is intutive.

How would a particular user rate a particular movie? Each movie has an average rating based of millions of viewers and can be assigned a score. We can see how a particular users rating pattern is by calculating the divergence of their rating from the average movie rating from prior rating history and derive a user-score. If the user is very discerning, when everyone gives a 4 and they give a 3, their score is (-1)then for any movie we can predict they will rate it one star below the average movie rating.

Limitation of the model was that the predictions had to be created within train_subset, test_subset and validation dataset, leading to overfitting.

As an extension,I would get the average movie rating from another data source, not movielens, and then calculate the user-score and do the predictions within this model.

Other extensions could be finding similarity within users about their rating history and modifying this prediction to include these clusters.