Movie Recommender System

Christopher Wheatley

15 September 2022

Contents

1 Executive Summary:

Blah blah

Initial Setup:

Fortunately, the Harvard X Data Science capstone course has provided the necessary code to initialize both the training and validation data sets.

2 Verification:

Let's verify the initialization was a success; analyzing the following data frames:

- edx == "training" data set.
- validation == "test" data set.

*Note. Our validation data set is not to be utilized for adjusting our model, it is reserved for generating our final predictions and assessing accuracy, as measured with the 'Root Mean Square Error' (RMSE) method.

```
## [1] 9000055       6
```

```
## Classes 'data.table' and 'data.frame':   9000055 obs. of  6 variables:
##  $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ movieId  : num  122 185 292 316 329 355 356 362 364 370 ...
##  $ rating   : num  5 5 5 5 5 5 5 5 5 5 ...
##  $ timestamp: int  838985046 838983525 838983421 838983392 838983392 838984474 838983653 838
##  $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Outbreak (1995)" "Stargate (1994)"
##  $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Action|Drama|Sci-Fi|Thriller" "
##  - attr(*, ".internal.selfref")=<externalptr>
```

The dimensions of the 'edx'/training set; details a data.table of over 9 million observations [m]; with 6 variables associated with each observation. Let's assess the completeness of this data, looking for missing values in each column/variable.

```
## [1] 0
## [1] 0
## [1] 0
## [1] 0
## [1] 0
## [1] 0
```

Nil missing values in the training set, lets evaluate our 'validation' (test) set the same way.

```
## [1] 999999      6
```

```
## Classes 'data.table' and 'data.frame':   999999 obs. of  6 variables:
##  $ userId   : int  1 1 1 2 2 2 3 3 4 4 ...
##  $ movieId  : num  231 480 586 151 858 ...
##  $ rating   : num  5 5 5 3 2 3 3.5 4.5 5 3 ...
##  $ timestamp: int  838983392 838983653 838984068 868246450 868245645 868245920 1136075494 11
##  $ title    : chr  "Dumb & Dumber (1994)" "Jurassic Park (1993)" "Home Alone (1990)" "Rob Ro
##  $ genres   : chr  "Comedy" "Action|Adventure|Sci-Fi|Thriller" "Children|Comedy" "Action|Dra
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
## [1] 0
## [1] 0
## [1] 0
## [1] 0
## [1] 0
## [1] 0
```

The dimensions of the 'validation'/test set; details a data.table of just under 1 million observations [m]; with 6 variables associated with each observation. The data.table has identified Nil missing values.

---

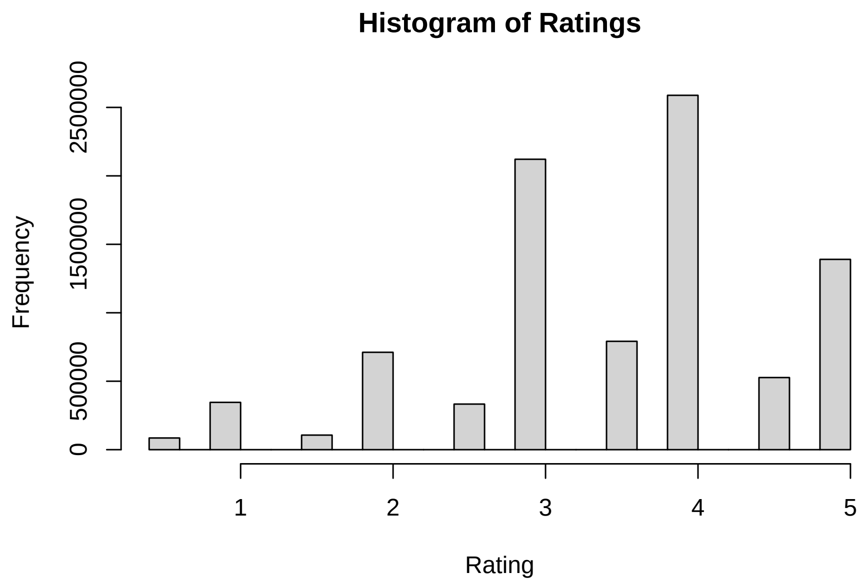
3 Strategy | Objective

Since the objective of this project is to develop a recommendation system able to predict ratings of movie [i] for user[u]. Lets examine the relationship between ratings [dependent variable] and; userId, movieId and genres [independent variables]. My reasoning for choosing only these variables; is for brevity and also based off the intuition that in a significant volume of reviews. Each user, movie and genre should detail a generalized bias relative to our prediction. As such, given new observations with the same independent variables a model can be formed to compute and add these bias terms and return a probabilistic estimate of a rating [y_hat].

4 Variable Analysis and Visualization

Variable: rating

Frequency distribution of ratings.



Numerical representation of the above histogram is as follows:

rating	count	proportion
4.0	2588430	0.29
3.0	2121240	0.24
5.0	1390114	0.15
3.5	791624	0.09
2.0	711422	0.08
4.5	526736	0.06
1.0	345679	0.04
2.5	333010	0.04
1.5	106426	0.01
0.5	85374	0.01

The most common value for ratings is 4.

Lets look at the mean rating.

```
## [1] 3.512465
```

To summarize, what we have found with the 'rating' variable. The mode is 4, mean is 3.512. Ratings between whole numbers are less frequent then their whole number equivalent. This variable can be utilized in a supervised learning model to provide real outcomes to train on. As such, it may be necessary to convert this variable into a factor or category for optimization purposes.

Variable: userId

Magnitude of unique users.

N_UserTrain	N_UserTest	Delta
69878	68534	1344

The table above depicts the number of unique users in both the training data set [69978] test data set [68534]. Also highlighting the difference [1344] between each. This apparent difference, may cause a problem later in the system design/utilization, if we constrain the algorithm to only users seen within the training set; and our model is required to take on new user data. If this is to be the case; we will replace missing values with the mean bias value for each parameter.I.e.

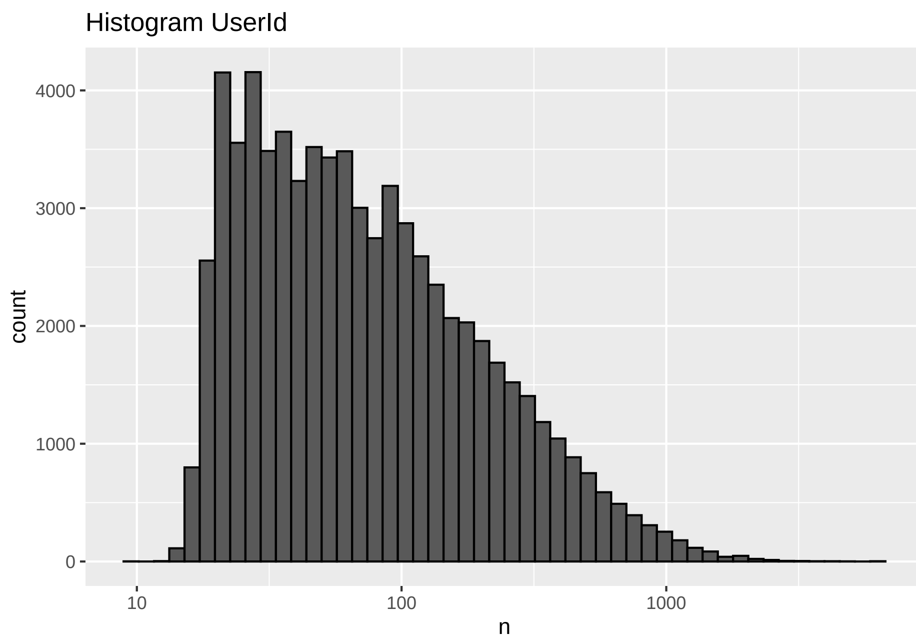
`ifelse(user bias missing / == NA, then mean(all$userBias), else leave value)`

Let's look at the number of reviews for each user.

userId	count
59269	6616
67385	6360
14463	4648
68259	4036
27468	4023
19635	3771
3817	3733
63134	3371
58357	3361
27584	3142

userId	count
57894	14
62317	14
63143	14
68161	14
68293	14
71344	14
15719	13
50608	13
22170	12
62516	10

Next we will assess the distribution of user reviews.



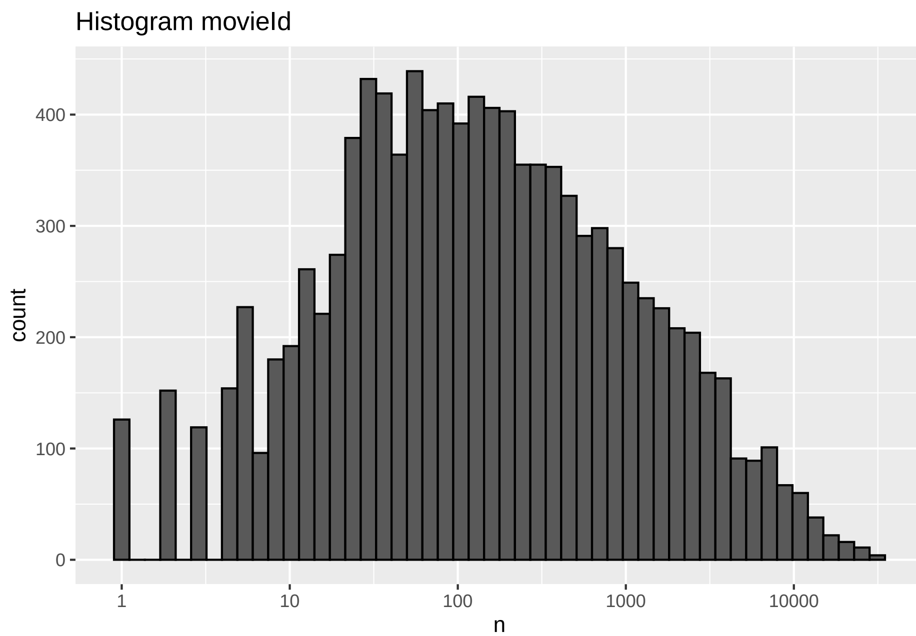
The above plot highlights certain 'outlier' users, at the higher end of total reviews. Regularization may be useful to penalize predictions with users with the largest variance.

Variable: movieId

Magnitude of unique movies.

N_MoviesTrain	N_MoviesTest	Delta
10677	9809	868

Frequency of 'movieId' within training data-set:



Frequency range of movie reviews:

title	count
Pulp Fiction (1994)	31362
Forrest Gump (1994)	31079
Silence of the Lambs, The (1991)	30382
Jurassic Park (1993)	29360
Shawshank Redemption, The (1994)	28015
Braveheart (1995)	26212
Fugitive, The (1993)	25998
Terminator 2: Judgment Day (1991)	25984
Star Wars: Episode IV - A New Hope (a.k.a. Star Wars) (1977)	25672
Apollo 13 (1995)	24284

title	count
Twice Upon a Time (1983)	1
Uncle Nino (2003)	1
Valerie and Her Week of Wonders (Valerie a tÃ½den divu) (1970)	1
Variety Lights (Luci del varietÃ) (1950)	1
Vinci (2004)	1
When Time Ran Out… (a.k.a. The Day the World Ended) (1980)	1
Where A Good Man Goes (Joi gin a long) (1999)	1
Won't Anybody Listen? (2000)	1
Young Unknowns, The (2000)	1
Zona Zamfirova (2002)	1

Similar to userId, movieId may need to be penalized based off the number of reviews observed within the training data set.

Variable: genres

genres
797

genres	count
Drama	733296
Comedy	700889
Comedy|Romance	365468
Comedy|Drama	323637
Comedy|Drama|Romance	261425
Drama|Romance	259355
Action|Adventure|Sci-Fi	219938
Action|Adventure|Thriller	149091
Drama|Thriller	145373
Crime|Drama	137387

Analyzing the 'genres' variable depicts 797 unique categories within the 'edx' data-set. People love Drama…

5 Hypothesis and Method

For simplicity sake and for brevity. I will choose a Naive Bayes approach to generating the recommender system model. This approach starts with the mean rating of all reviews and adds bias terms in an iterative fashion, assessing with each addition the accuracy of the model. Accuracy will be measured through RMSE calculations between a training set and one cross validation set. Regularization will be applied where necessary.

Model	RMSE
Mode	1.166756
Mean	1.060054

$$y_{hat} = \mu_{ratings} + bias_{term1} + bias_{term2} + ..n$$

Let's add a bias term for Genre with our first iteration.

$$y_{hat} = \mu_{ratings} + bias_{genres}$$

genres	b_g
(no genres listed)	0.1303919
Action	-0.5735849
Action|Adventure	0.1477141
Action|Adventure|Animation|Children|Comedy	0.4493543
Action|Adventure|Animation|Children|Comedy|Fantasy	-0.5185258
Action|Adventure|Animation|Children|Comedy|IMAX	-0.1761016

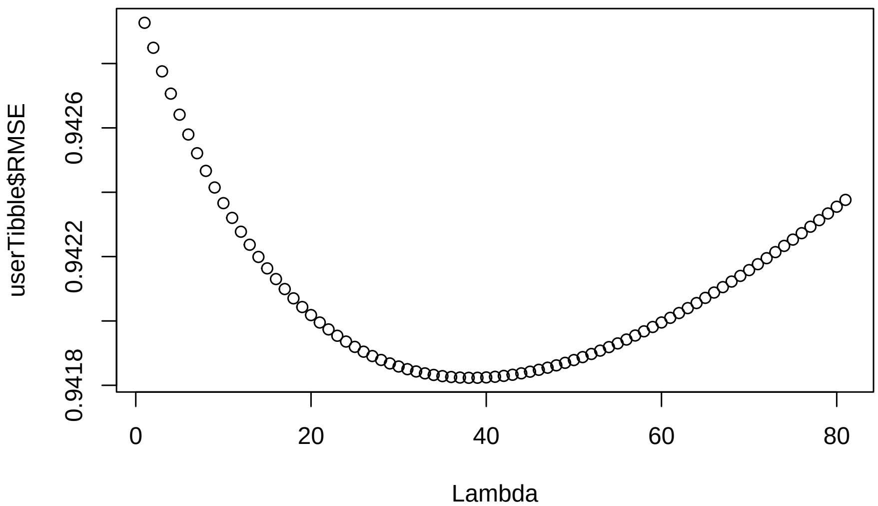
Model	RMSE
Mode	1.166756
Mean	1.060054
MeanPlusGenre	1.017501

We seem to be tracking in a positive direction, let's add another bias term - userId. As observed through our exploratory data analysis, certain users are seen to be outliers with reference to the majority; specifically in terms of their frequency of reviews. As such we will apply regularization to this term to reduce the overall variability of our bias term and hopefully increase the accuracy of predictions.

$$y_{hat} = \mu_{ratings} + bias_{genres} + \frac{bias_{userId}}{(n + \lambda)}$$

userId	b_u
1	1.4875348
2	-0.5124652
3	0.4319792
4	0.6087469
5	0.4552767
6	0.4349032

```
## [1] 1.009182
```




```
## [1] 38
```

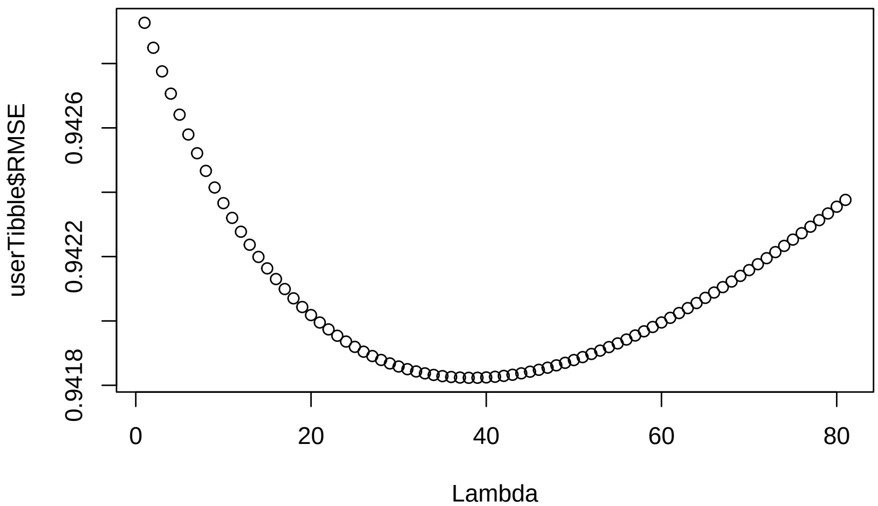
```
## [1] 0.9418233
```

Lambda set to 38, produces the highest performing model. With a RMSE score of 0.9418233

Model	RMSE
Mode	1.1667562
Mean	1.0600537
MeanPlusGenre	1.0175012
MeanPlusGenre_PlusUserRegularized	0.9418233

In finale, let's add a regularized bias term for each movie to our model.

$$y_{hat} = \mu_{ratings} + bias_{genres} + \frac{bias_{userId}}{(n + \lambda)} + \frac{bias_{movieId}}{(n + \lambda)}$$




```
## [1] 38
```

```
## [1] 0.9418233
```

6 Result

Utilizing a Naive Bayes approach and with the following variables available we have achieved a RMSE accuracy score of XXXXX

7 Conclusion

Limitations / Future work.. given enough computing power, i am eager to utilize a matri