Movie Recommender System

Christopher Wheatley

17 September 2022

Contents

1 Executive Summary

This report satisfies the first capstone project of two; within the Data Science program provided by edX and HarvardX. The primary objective being to generate a movie rating recommendation system through data; analysis, visualization, hypothesis generation, optimization and testing. As such; we have proven that given a large enough data set of movie reviews with variables for movie rating, movie genre, user identification, movie identification. One is able to build the following generalized machine learning algorithm which is able to predict a rating for movie i and user u; to a Root Mean Squared Error (RMSE) / accuracy of XXXX.

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

2 Initial Setup | Verification

Setup:

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

Verification:

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

- 'edx'; which will be our *training* data set.
- 'validation'; which will be our *test* data set.

rows	variables
9000055	6

variable	class	first_values
userId	integer	1, 1, 1
movieId	double	122, 185, 292
rating	double	5, 5, 5
timestamp	integer	838985046, 838983525, 838983421
title	character	Boomerang (1992), Net, The (1995), Outbreak (1995)
genres	character	Comedy|Romance, Action|Crime|Thriller, Action|Drama|Sci-Fi|Thriller

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

Checking for missing Data:

I like to utilize a for loop across each variable looking for missing data.

```
#example method to assess NAs with for loop.

for (i in 1:ncol(edx)) {
  na <- sum(is.na(edx[,..i]))
  print(na)
}
```

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

All 6 variables have zero missing values in the training set, lets evaluate our ‘validation’ (test) set the same way.

rows	variables
999999	6

variable	class	first_values
userId	integer	1, 1, 1
movieId	double	231, 480, 586
rating	double	5, 5, 5
timestamp	integer	838983392, 838983653, 838984068
title	character	Dumb & Dumber (1994), Jurassic Park (1993), Home Alone (1990)
genres	character	Comedy, Action|Adventure|Sci-Fi|Thriller, Children|Comedy

```
## [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 [n] associated with each observation. The data.table has identified Nil missing values.

3 Objective | Strategy

Objective:

Develop and test a movie recommendation model able to predict ratings of movie [i] for user[u]. With accuracy measured through a RMSE < 0.86490.

Strategy:

First examine the relationship between the following variables: 'ratings' [dependent variable] and; 'userId', 'movieId' and 'genres' [independent variables]. The reasoning for choosing only these variables; is for brevity and based off an intuition that in a significant volume of reviews. Each user movie and genre should detail a generalized bias relative to the mean rating. Thus, given new observations with the same independent variables a model can be formed to compute [add the bias terms to a mean] and return a probabilistic estimate of a rating [y_hat].

4 Variable Analysis | Visualization

Variable: rating

Frequency distribution of ratings.

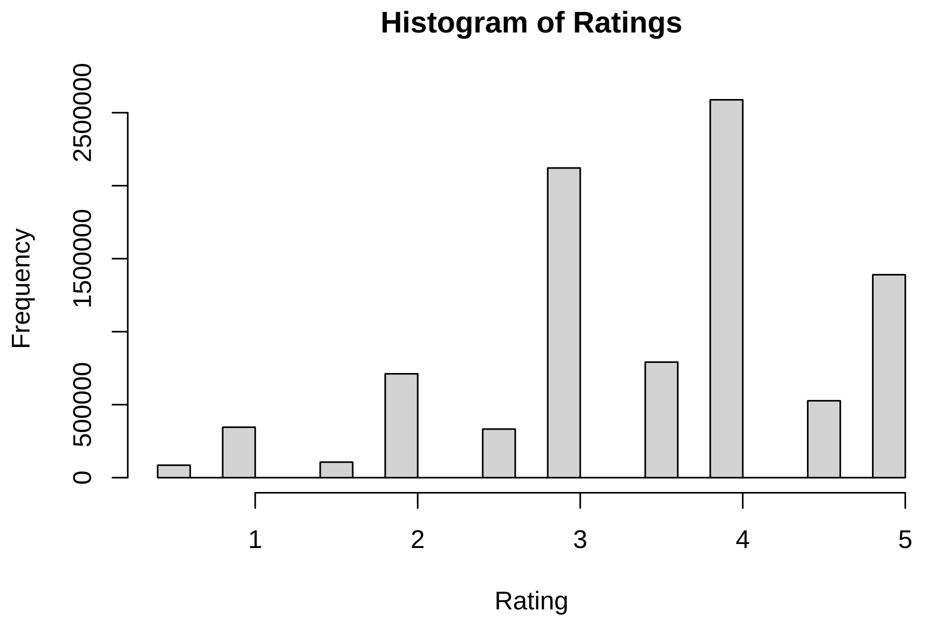


Figure 1: Histogram of Ratings

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

To summarize what we have found with the 'rating' variable. The mode is 4 and mean is 3.5124652. 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 for training a hypothesis. 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] and the 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, then replace with mean(all$userBias), else leave value)
```

Now let's plot the frequency distribution of reviews from users.

The above right skewed plot highlights 'outlier' users, at the higher and lower end of the number of reviews. Regularization may be useful to penalize predictions from users with the largest variance from the mean.

Variable: movieId

Number of unique movies between data sets and delta.

N_MoviesTrain	N_MoviesTest	Delta
10677	9809	868

Frequency of 'movieId' within training data-set:

Similar to userId, movieId contains outliers; which may need to be penalized also.

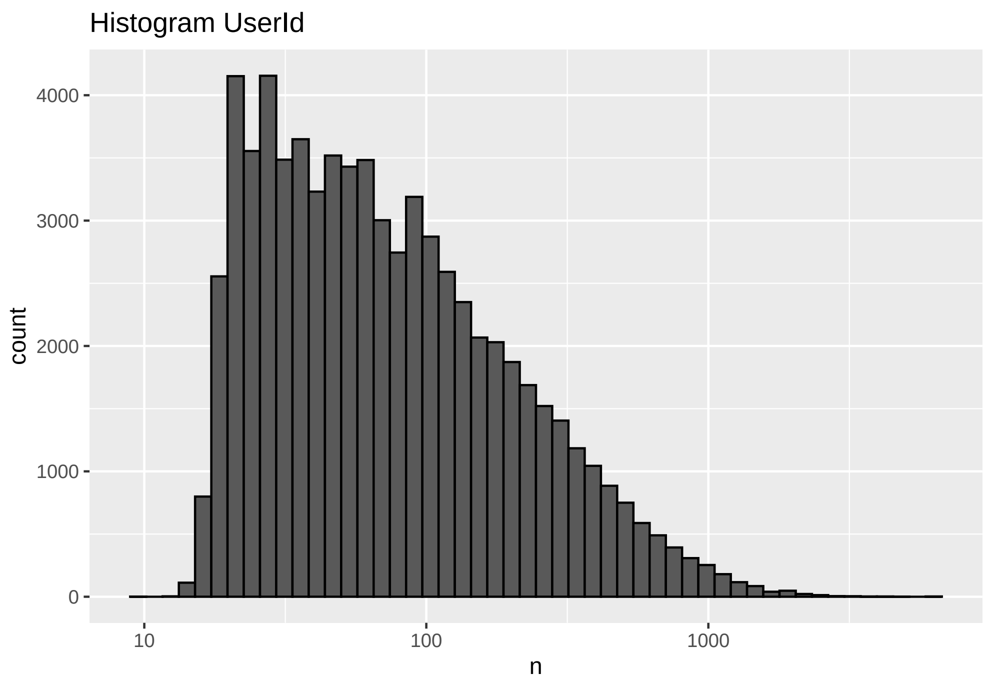


Figure 2: Histogram UserId

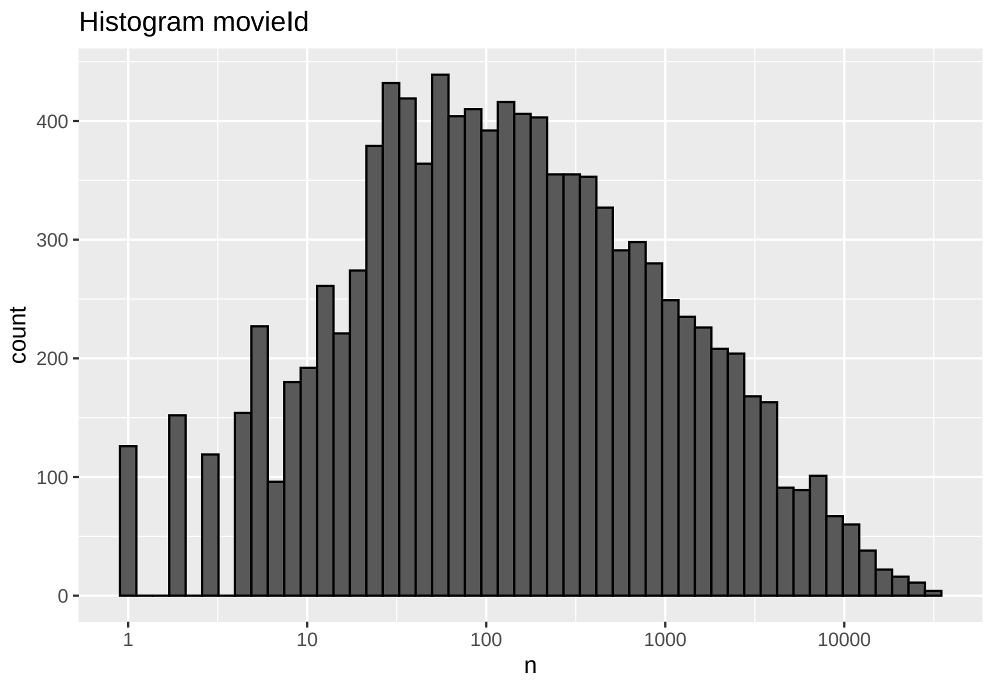


Figure 3: Histogram movieId

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 | Method

A Naive Bayes method will be utilized to form the model hypothesis. 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. As stated in the objective, 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.

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

Model	RMSE
Mode	1.166756
Mean	1.060054

Model	RMSE
MeanPlusGenre	1.017501

With the first iteration - our model has increased accuracy. Now let's add another bias term - userId.

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

```
## [1] 0.9394804
```

Model	RMSE
Mode	1.1667562
Mean	1.0600537
MeanPlusGenre	1.0175012
MeanPlusGenre_PlusUser	0.9394804

The next bias term is for movies.

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

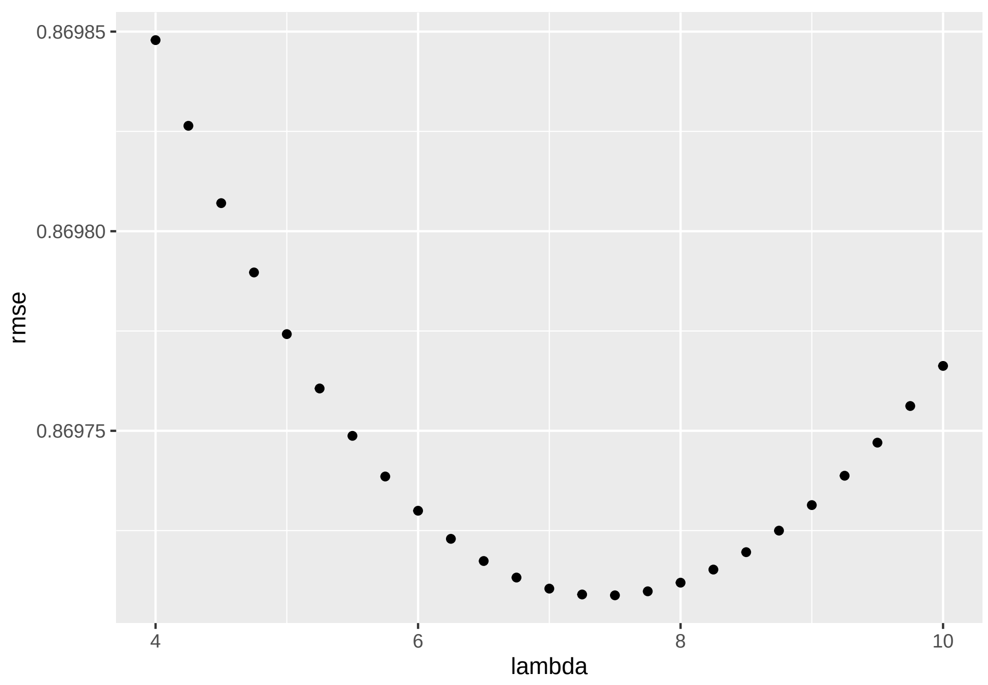
```
## [1] 0.8705983
```

Model	RMSE
Mode	1.1667562
Mean	1.0600537
MeanPlusGenre	1.0175012
MeanPlusGenre_PlusUser	0.9394804
MeanPlusGenre_PlusUser_PlusMovie	0.8705983

As highlighted through the analysis and visualization phase. Certain parameter values have significantly less frequent observations, and since we are trying to generalize an average prediction for each dimension. These outlier values could add unwanted variability with our predictions. As such we will add the penalty term 'Lambda' to our hypothesis, reducing the outlier affect from less frequent values skewing our predictions.

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

To optimize our hypothesis and select the most accurate value for λ, we will iterate values from 4 to 10; in increments of .25. Plotting our results and selecting the λ value which returns the minimum RMSE.



Model	RMSE
Mode	1.1667562
Mean	1.0600537
MeanPlusGenre	1.0175012
MeanPlusGenre_PlusUser	0.9394804
MeanPlusGenre_PlusUser_PlusMovie	0.8705983
MeanPlusGenre_PlusUser_PlusMovie_Regularized	0.8697088

Lambda set to 7.5, produces the highest performing model. With a RMSE score of 0.8697088.

We haven't quite met the accuracy objective of < 0.86490.

Delta = 0.0048088.

As the submission deadline is getting closer, I will implement this hypothesis on the validation data set and submit the project. Given more time; I would continue to add and evaluate new bias terms.

6 Result

Utilizing a Naive Bayes approach - the following model achieved a 'test' RMSE accuracy of: RMSE = 0.8704178.

7 Conclusion

Given more time, I would have liked to add additional bias terms or dimensions to the hypothesis. Given more computational power, a matrix factorization algorithm would have been interesting to work on.