Movielens Report

Ain Sanchez

2023-12-02

Introduction

The goal of the Capstone Project of the HarvardX Data Science Program is to train an algorithm that would estimate movie recommendations based on a dataset of the 2009 Movielens movie ratings, which would provide an RMSE (root-mean-squared-error) of less or equal to 0.86490. In order to describe the process to train the algorithm, this document incorporates three main sections related to: 1) Methods and Analysis, 2) Results, and 3) Conclusion. In the first section, the document will describe the process and techniques used including data-cleaning, data-exploring, visualization, insights gained, and the modeling approach. In the second section, the document will show the results of the methods employed and analysis performed. Finally, the document will provide a conclusion stating the limitations of the project and will recommend further considerations for future work in this area.

Section 1: Methods and Analysis

Once downloaded the Movielens zip file, the decompressed folder shows three main datasets: 1) movies.dat, 2) ratings.dat, and 3) tags.dat. There are three other files, two of which are scripts that contain code in Unix shell and Perl respectively, to split the ratings dataset with a five-fold cross-validation. The last file is a readme file, which summarizes the contents of each dataset. For instance, the movies.data contains 10,681 observations, while the ratings.data contains 10,000,054 observations. It is worth noting that there are also "… 95,580 tags applied to 10,681 movies by 71,567 users of the online movie recommender service MovieLens."

From the description of the content and use of each file, the readme file states that Movielens users were selected randomly. However, the users were selected separatedly from the ratings and tags datasets, which implies that the users that appear in one dataset might not be the same in the other. For the purpose of this project, only the movies and ratings datasets will be used, so the tags dataset will be disregarded accordingly. In the case of ratings, each observation contains four variables: 1) userID, 2) movieID, 3) rating, 4) and timestamp. Each rating is made on a 0 to 5 scale, with increments of 0.5. Finally, the movies dataset include three variables: 1) movieID, 2) title, and 3) genres.

Upon merging the ratings and movies datasets into the movielens dataset, the resulting dataset contains six variables. This combined dataset was split once again to create the edx dataset (train dataset), and the final_holdout_test (test dataset). Taking into consideration that movies' and users' ids are included in the edx dataset, the final_holdout_test contains 10% of the observations of the movielens dataset, while the edx dataset includes the rest. For the Methods and Analysis section, only the edx dataset will be used to describe the process and techniques of data-cleaning, data-exploring, visualization, insights gained, and the modeling approach.

Data-Cleaning

To start the data-cleaning process, the six main variables of the edx dataset will be analyzed using three metrics based on completeness, consistency and clarity. In relation to completeness, the "table" function is used to obtain frequencies of the discrete values of ratings. For the other variables, a foor-loop will be used to determine the number of missing values that should be excluded of the training model.

```
#Completeness

#This line is only run to match the original edx dataset
edx <- edx[,-c(7:10)]

#Generate a table to determine the completeness of the rating variable
table(edx[,3], useNA = "always")
```

```
## 
##     0.5       1     1.5       2     2.5       3     3.5       4     4.5       5
##   85374  345679  106426  711422  333010 2121240  791624 2588430  526736 1390114
##    <NA>
##       0
```

```
#Generate a for-loop to identify the completeness of the other variables
for (i in c(1:2,4:6)){
  print(sum(is.na(edx[,i])|edx[,i]==""))
}
```

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

Based on the completeness results, none of the variables in the edx dataset contains missing values, so that it is not necessary to delete any observation.

In terms of consistency, all variables were analyzed to determine if their values are adequately represented on the dataset. In that sense, it was necessary to change the variable type of the userId and movieId from numeric to character, to assure that ids are treated properly. Similarly, timestamp was transformed from a numeric variable into a a datetime format. Other variables such as rating, title and genres have variable types which are directly related to their variable names and the documentation, so no further process was necessary to apply in terms of consistency.

In order to acknowledge clarity, the current structure of variables was changed in order to improve its conciseness and interpretability. For instance, genres is a categorical variable but contains different combinations of independent categories separated by a vertical bar ("|"). To improve the variable's clarity, 18 new dummy variables were generated based on the unique genre categories described in the Movielens documentation.

Data-Exploring

Since id values cannot be analyzed per se, the main variables to be explored in this section are rating, genres and timestamp. First we start obtaining summary statistics of rating.

```
##   meanRating stdvRating medianRating
## 1   3.512465   1.060331            4
```

```
##
##    0.5      1    1.5      2    2.5      3    3.5      4    4.5      5
##  0.95%  3.84%  1.18%   7.9%   3.7% 23.57%   8.8% 28.76%  5.85% 15.45%
```

As shown in the tables above, rating has a mean of 3.51, a median of 4 and a standard deviation of 1.06, which implies that users tend to rate movies higher than the mean of the rating variable's scale, which is 2.5. In fact, 28.76% of movie ratings have a score of 4. The next variable analyzed is genres.

```
##   meanNroGenres stdvNroGenres medianNroGenres
## 1        2.5139      1.103816               2
```

```
##
##      0      1      2      3      4      5      6      8
##  0.01% 20.78% 29.92% 30.59% 15.02%  3.16%  0.51%     0%
```

```
##
##   1995   1996   1997   1998   1999   2000   2001   2002   2003   2004   2005
##     0%   3.3% 10.71%  1.73%  1.94% 12.64% 11.02%  7.02%  6.44%  7.23% 11.17%
##   2006   2007   2008   2009
##  8.16%  7.65%  6.41%  4.57%
```

```
##
##      1      2      3      4      5      6      7      8      9     10     11
##  8.36%  6.87%  8.27%  7.46%  7.39%  8.32%  8.92%  8.05%   6.3%  9.22% 10.84%
##     12
## 10.01%
```

While ratings have been given between 1995 and 2009, 12.71% of all ratings have been recorded in the year 2000. It is also important to highlight that users do not tend to rate movies on a specific month.

Visualization

Based on the 18 new dummy variables generated in the data-cleaning section, a barplot is used to represent the distribution of movie genres across all movies in the dataset, including the year of the rating. In this section, 10,000 observations are randomly selected to display the behavior of the dataset, so that computing power is optimized when graphs are generated.

The barplot on Figure 1 shows that users tend to rate a similar number of movies every year across the different genres. In general, drama movies are more frequently rated, followed by comedy and action movies. In order to identify how users rated different movie genres, a boxplot is used.

The bocplot on Figure 2 demonstrates that some movie genres have a higher median rating than others, yet all movies have median ratings between 3 and 4. Compared to other movie genres, horror movies tend to have lower ratings, while filmNoir movies have higher ratings. To identify if ratings are affected by the year of the rating, another boxplot is used.

The boxplot on Figure 3 displays a interesting pattern where movies rated before 2003 have a different scoring-scale compared to those rated in or after 2003. In fact, the first group of movies rated before 2003, have median scores of 3 or 4, and outlier scores of 1. However, movies rated in or after 2003, have median scores of 3.5 and outlier scores of 1 and 0.5. In short, it can be deduced from this pattern that scores with increments of 0.5 points were introduced in 2003, since scores only had increments of 1 before that year.

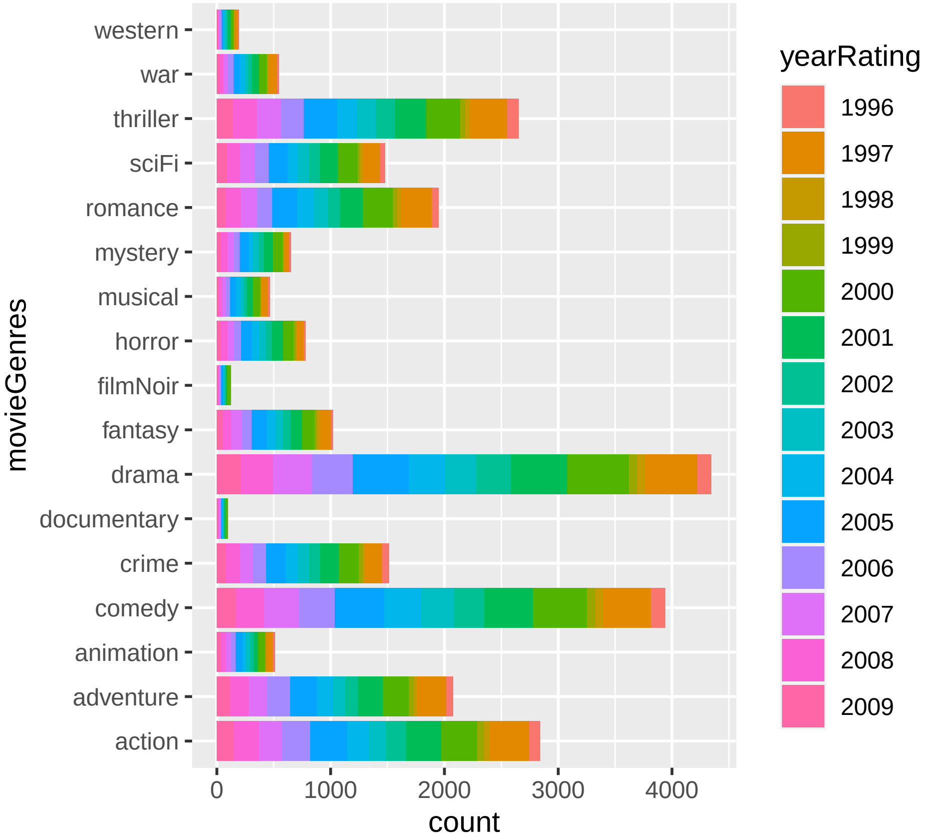

Figure 1: Distribution of movie ratings of Movielens' users

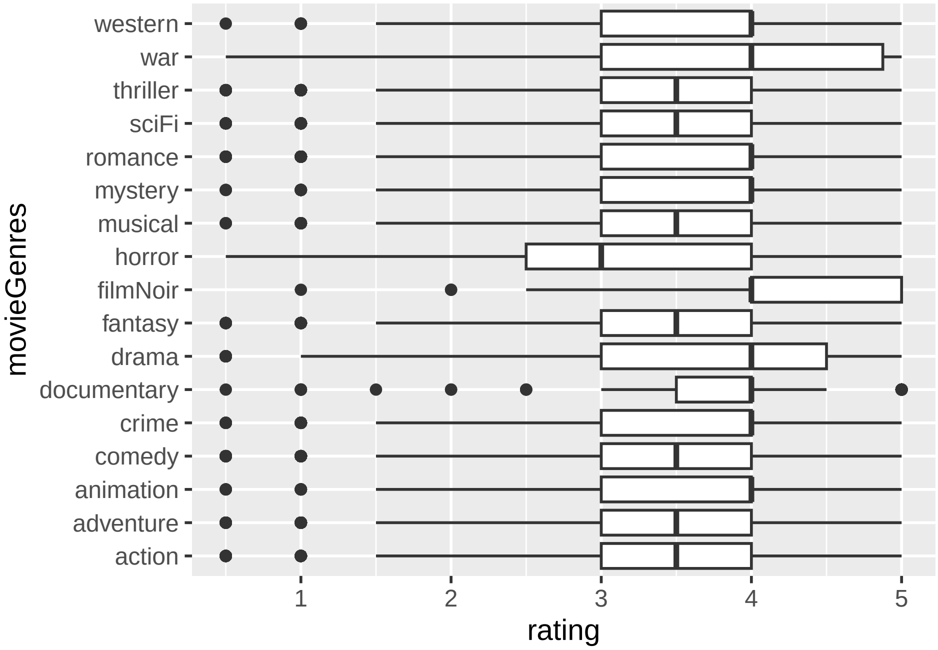

Figure 2: Distribution of movie ratings of Movielens' users

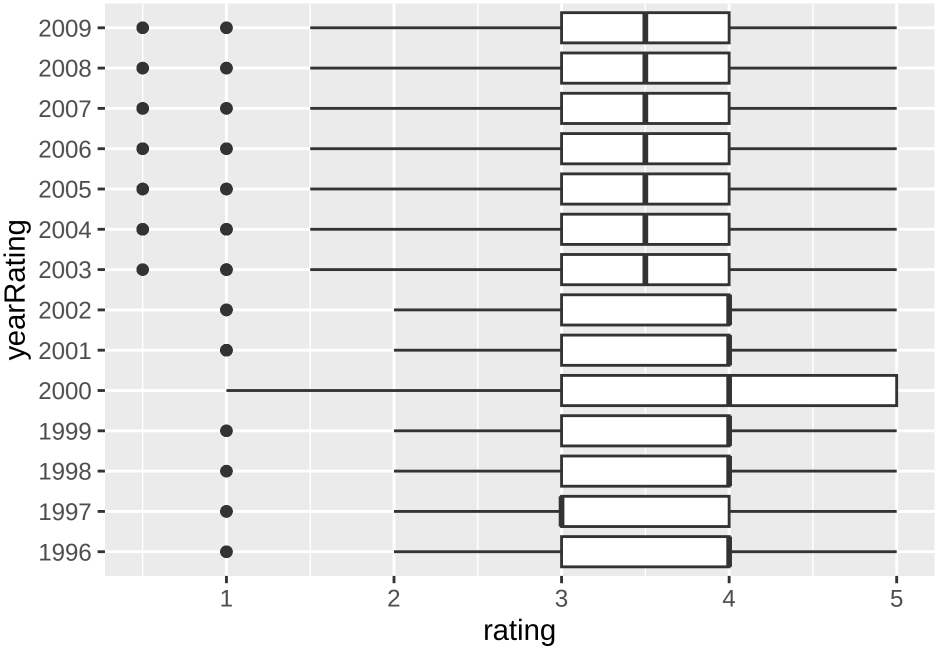

Figure 3: Distribution of movie ratings of Movielens' users

Insights

From the previous sections, three main insights can be useful to develop and algorithm that will estimate the movie ratings in the final_holdout_test dataset. 1) the 18 dummy variables could be used as individual predictors, in contrast of the genres variable which has many combinations that are difficult to interpret. 2) Year is a useful predictor, but this predictor seems to be more powerful when it discriminates between movies rated before 2003 and in or after 2003. 3) One should not discount the userId and movieId variables, which could account for all the unobservable variance of movie preferences.

Modeling Approach

For the modeling approach, rating will be the predicted variable. The predictors include the 18 dummy variables generated by unique genre names, which will be chosen in the model using the "nearZeroVar" function. Other predictor is the variable before2003, which is a dummy variable that identifies movies rated before the year 2003 and movies rated in or after 2003.Finally, userId and movieId will be added to incorporate movie and user mean ratings and movie and user number of ratings to estimate rating. In short, the last 4 predictors are: 1) number of movies each user rated, 2) number of movie ratings given to each movie, 3) average rating for each user, 4) average rating for each movie.

Based on the predicted variable which is numeric, the model to predict rating will be a linear regression model. In the training process, the model will incorporate a control with resampling through cross-validation, and will be run on a random sample of observations in order to optimize computing power.

Section 2: Results

To obtain the final algorithm, which has an RMSE lower than 0.86490 and achieves the goal of the Capstone Project, it is important to mention that several combinations of predictors were used in the process. For instance, the “nearZeroVar” function determined that only 13 out of the 18 dummny genere variables had predictive power. In fact, the genres that did not add up predictive power to the model are the ones with the lowest number of ratings on Figure 1, which include western, war, filmNoir, documentary and children’s. Model 1 includes the filtered, non-zero variance, genre variables along with other predictive variables listed in the Modelling Approach section.

In model 2, genre variables are not included purposedly in order to observe the model performance by minimizing the number of precictive variables. However, it is also necessary to highlight that Model 1 and Model 2 include an additional variable named YearDiffm which was calculated based on the difference between the movie rating represented on the timestamp variable, and the year of the movie release included in the title variable. This variable was introduced in order to further explore the effect of time on movie ratings, which is also partially captured by the before2003 variable in both models.

It is very insightful that Model 3 did not include predictors related to the movie genres and the time when the movie rated. Nonetheless, this final model using the 4 variables related to movie and user mean rating and movie and user number of ratings listed in the Modelling Approach Section, performs just as well as the other models with 15 additional predictors. It is important to mention that all models used linear regression with resampling, through a 5-fold cross-validation and a probability of 0.9. To train the model, 1,000 randomly selected observations were chosen, so that computing power would not be exhausted during the algorithm training process.

In short, the final model was obtained by training the linear regression algorithm with 4 predictors 1) number of movies each user rated, 2) number of movie ratings given to each movie, 3) average rating for each user, and 4) average rating for each movie. Also, the predicted variable used in the model was rating. To measure the predictive power of the model compared to the other two, the RMSE is calculated using with the trained models applied on the final_holdout_test dataset. The results obtained are the following:

method	RMSE
Model 1: movieGenres + before2003 + yearDiff + user & movie mean ratings + user & movie number of ratings	0.8454124
Model 2: before2003 + yearDiff + user & movie mean ratings + user & movie number of ratings	0.8454206
Model 3: user & movie mean ratings + user & movie number of ratings	0.8455515

As shown above, model 3 performs just as well as the other two models while using signifficantly fewer predictors, which allows the model to be more interpretable. As a result, model 3 is the preferred model to predict movie ratings in the Movielens dataset with a RMSE performance metric of 0.8455515.

Section 3: Conclusion

The Goal of the Data Science Capstone Project was to train an algorithm that improves the performance metric of the model to a value lower than 0.86490. Although the Movielens dataset did not have many variables, the Methods and Analysis section helped understand the data through data-cleaning, data-exploring and visualization. While elaborating Section 1, it was very insightful to acknowledge the behavior of several variables and the potential contribution that they could bring to the modelling approach. In the end, the model construction included a simplified version of a linear regression model to train on a dataset, which expanded the original dataset of 6 variables to one with more than 20 variables derived from those 6 variables.

The final model to predict movie ratings entirely used predictors that reflected movie and user mean ratings and movie and user number of ratings, by grouping the movieId and the userId, and calculating the corresponding sum of observations and average rating within each Id. Through this process, 4 predictors were obtained, which improved the model performance entirely when running the model on the test dataset. The main idea behind this exercise lies on the fact that movie ratings are very sensitive to user preferences and the quality of the movie itself. For instance, the movie-genres dummy variables, and the variable before2003 were not able to improve the model performance signifficantly because their within-group variance remained very high across movies and users.

The current limitations of this model are reflected by the performance metric obtained of 0.86490. In fact, the metric suggests that most predicted ratings can have a difference of 0.86 points compared to the true rating value. Considering that the 0-5 point rating scale after 2003 holds increments of 0.5, the current modeling error implies that most values might actually deviate from the true value. To improve the model performance even more, future work should explore high values within userId and movieId through regularization, to penalize observations that might influence predictors to deviate. Also, matrix factorization can also be effective to determine the main factors that influence ratings to vary across the dataset. Morevoer, user's demographic variables could be very powerful to explain variations within userIds. Finally, other powerful regression algorithms could be used to better capture the variance among the dataset and improve the overall prediction.