edX MovieLens Submittal

Eduardo Caballero

2/26/2022

Executive Summary

The objective of this project was to develop a model to predict the rating a user would give a movie. The data set used was the movielens dataset with 10 million observations. Code was provided to download the files and split the data into a validation set and a training set. The dimensions and structure of the data set was as given below:

```
## Classes 'data.table' and 'data.frame':   9000055 obs. of  14 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 8389848
##  $ title        : chr  "Boomerang (1992)" "Net, The (1995)" "Outbreak (1995)" "Stargate (1994)" ...
##  $ genres       : chr  "Comedy|Romance" "Action|Crime|Thriller" "Action|Drama|Sci-Fi|Thriller" "Acti
##  $ rate_half    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ user_pct_half: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ userAvg      : num  5 5 5 5 5 5 5 5 5 5 ...
##  $ userStd      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ userReviews  : int  19 19 19 19 19 19 19 19 19 19 ...
##  $ MovieAvg     : num  2.86 3.13 3.42 3.35 3.34 ...
##  $ MovieStd     : num  0.932 0.955 0.868 0.941 0.942 ...
##  $ MovieReviews : int  2178 13469 14447 17030 14550 4831 31079 3612 18921 7331 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
##    userId movieId rating timestamp                        title
## 1:      1     122      5 838985046              Boomerang (1992)
## 2:      1     185      5 838983525               Net, The (1995)
## 3:      1     292      5 838983421               Outbreak (1995)
## 4:      1     316      5 838983392               Stargate (1994)
## 5:      1     329      5 838983392 Star Trek: Generations (1994)
##                          genres rate_half user_pct_half userAvg userStd
## 1:               Comedy|Romance         0             0       5       0
## 2:         Action|Crime|Thriller         0             0       5       0
## 3:  Action|Drama|Sci-Fi|Thriller         0             0       5       0
## 4:       Action|Adventure|Sci-Fi         0             0       5       0
## 5: Action|Adventure|Drama|Sci-Fi         0             0       5       0
##    userReviews MovieAvg  MovieStd MovieReviews
## 1:          19 2.858586 0.9324804         2178
## 2:          19 3.129334 0.9548463        13469
## 3:          19 3.418011 0.8678273        14447
## 4:          19 3.349677 0.9411659        17030
## 5:          19 3.337457 0.9424496        14550
```

The training dataset was subdivided into a training and testing set. The training set consisted of 80% of the data.The data was then reviewed and a modeling approach was developed. The breakdown of unique users and movies is given below.

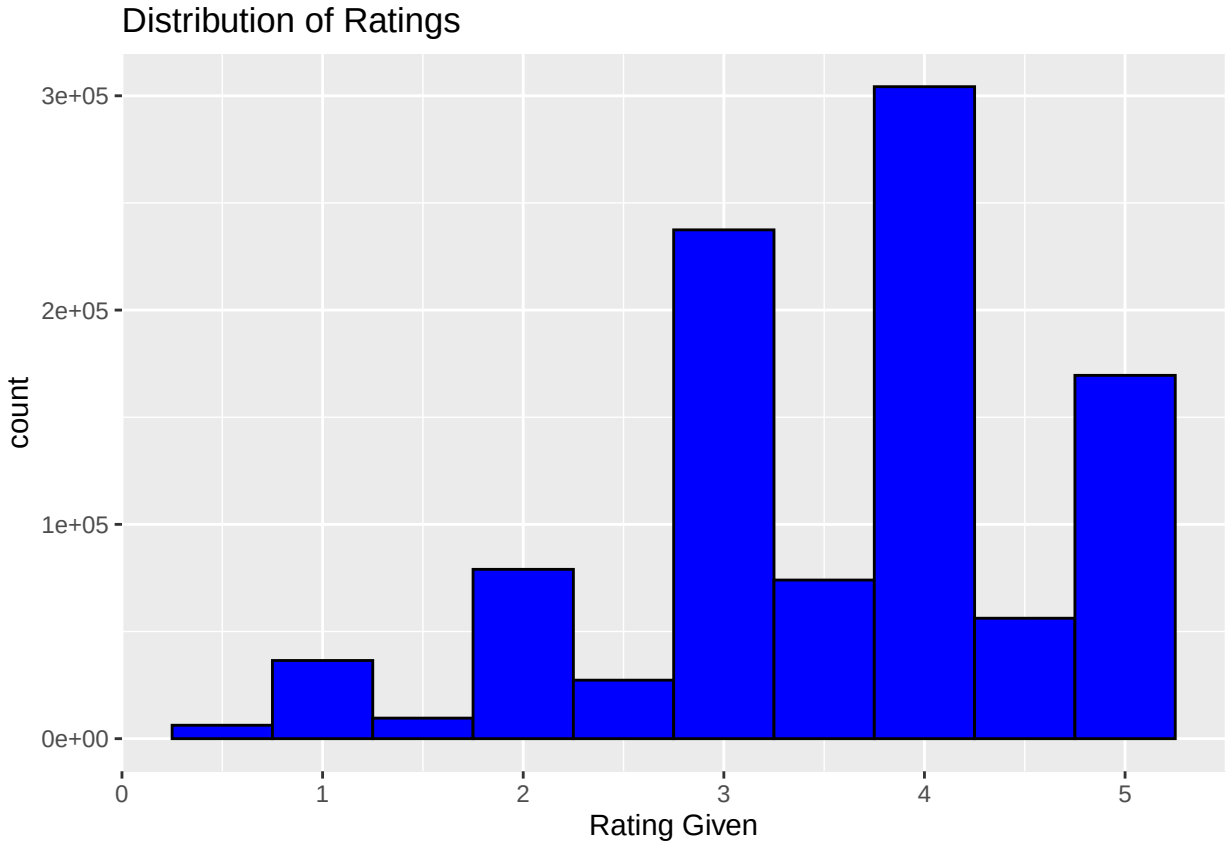
```
##   unique_users unique_movies
## 1        69878         10677
```

Various models were tested using the caret library. Models tested included knn, glm, random forest, rpart and ranger. The root mean squared error (RMSE) was utilized to determine the best model. The glm model gave the best results of the models tested.

Analysis

section that explains the process and techniques used, including data cleaning, data exploration and visualization, insights gained, and your modeling approach

As part of the analysis the data was reviewed to determine what the best approach would be. A sample of the dataset was taken to determine the distribution of the ratings.



As can be seen in the graph, ratings were mostly 3's and 4's with much fewer 3.5's or 4.5's. In order to determine how an individual might rate a movie, additional columns based on past ratings by the user were developed. Columns for the mean rating a user gave, standard deviation of the user's rating and the percentage of times that a user would give a rating with a half point such as 0.5, 1.5, 2.5, 3.5 or 4.5.This percent was calculated as the rounded rating to zero decimal points minus the rating. If a whole number rating was given this would evaluate to a 0 and if a rating with a half was given it would evaluate to a -.05 or 0.5. This value was then doubled and the absolute value taken so that the final value was a 0 if a

whole number rating was given or a 1 if a rating with a half was given. The average value over all ratings was then taken to determine the percent of time the user would rate on a half. This variable was named user_pct_half

Other columns were added to get the average rating a specific movie ,MovieAvg, had and the standard deviations ,MovieStd, of that average and also the number of reviews ,MovieReviews, the movie had. This was also done for each user creating columns named userAvg ,userStd and userReviews.

The skimr package was then used to get a quick summary of the data as shown below.The data was not missing fields except for the Movie standard deviations which was for movies that only had one rating. Because of this preprocessing of the data was unnecessary.

Table 1: Data summary

Name	edx
Number of rows	9000055
Number of columns	14
Key	NULL
Column type frequency:	
character	2
numeric	12
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty
title	0	1	8	161	0
genres	0	1	3	60	0

Variable type: numeric

skim_variable	n_missing	complete_rate
userId	0	1
movieId	0	1
rating	0	1
timestamp	0	1
rate_half	0	1
user_pct_half	0	1
userAvg	0	1
userStd	0	1
userReviews	0	1
MovieAvg	0	1
MovieStd	126	1
MovieReviews	0	1

The data file created was very large for the processing power of the computer used. So in order to run the models a random sample of the data was taken to run the models on a smaller data set. Data sets of 10,000, 100,000 and 1 million were created and the models were run progressively until the computer would freeze.

Results

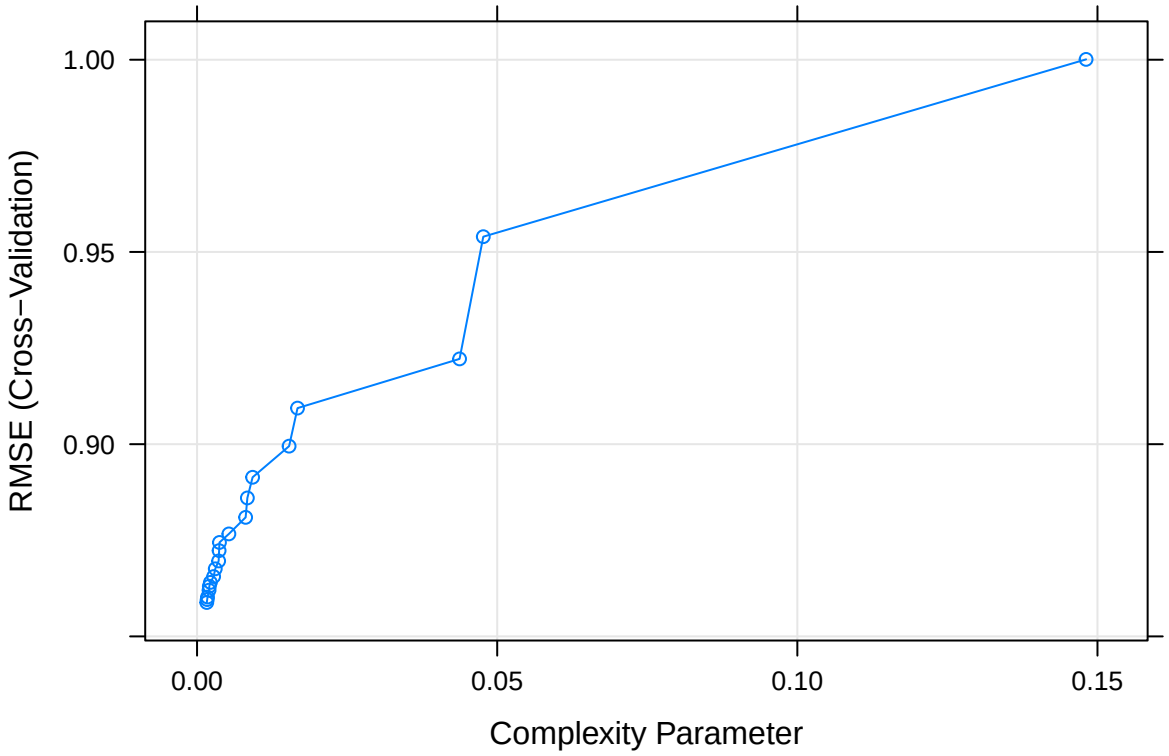
Various machine learning models were tested to determine which one gave the best predicted results. Models tested were knn, glm, random forest, rpart and ranger. The models were tested on increasingly sized data sets. The predictors used to train the model were user_pct_half + userAvg + MovieAvg

The knn model was run with the default cross validation. The model was run with 10,000 and 100,000 records to train the model. Once a trained model was established the model was tested on the test data set. The RMSE results for the 10,000 model was 0.8832128 and when ran with the 100,000 data set RMSE was 0.9454087

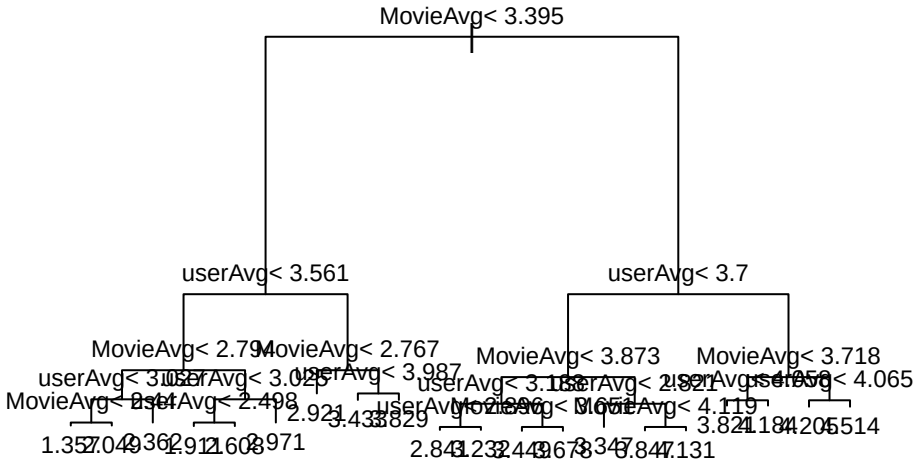
The Random Forest model was run with cross validation. The model could only be run with 10,000 data set before the laptop would crash. The RMSE results for the 10,000 model was 0.9597575

The rpart model was run with cross validation. The model was run with data sets 10,000, 100,000 and 1,000,000. The RMSE results for the 10,000 model was 0.8945104, when ran with the 100,000 data set RMSE was 0.8916313 and with the 1,000,000 data set RMSE was 0.8918748. The model information is shown below.

```
## CART
##
## 1e+06 samples
## 3e+00 predictors
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 899999, 899999, 900001, 900000, 900000, 899999, ...
## Resampling results across tuning parameters:
##
##   cp           RMSE       Rsquared   MAE
##   0.001618524  0.8588189  0.3287589  0.6674065
##   0.001684960  0.8595831  0.3275646  0.6676292
##   0.001744223  0.8602337  0.3265462  0.6677925
##   0.002004210  0.8620327  0.3237242  0.6686793
##   0.002040633  0.8630535  0.3221238  0.6683842
##   0.002226391  0.8640455  0.3205641  0.6689795
##   0.002760787  0.8655706  0.3181625  0.6703684
##   0.003034554  0.8675799  0.3149918  0.6705059
##   0.003598731  0.8696236  0.3117631  0.6698828
##   0.003672676  0.8723144  0.3074965  0.6738923
##   0.003734245  0.8744414  0.3041157  0.6779764
##   0.005303210  0.8766595  0.3005776  0.6818307
##   0.008092184  0.8809654  0.2936906  0.6883893
##   0.008379892  0.8860155  0.2855730  0.6944314
##   0.009254077  0.8913894  0.2768768  0.7016228
##   0.015334957  0.8994827  0.2636695  0.7129153
##   0.016738739  0.9093846  0.2473696  0.7270965
##   0.043732496  0.9221813  0.2259534  0.7328240
##   0.047682574  0.9539715  0.1716243  0.7363606
##   0.148101015  1.0000903  0.1472572  0.7875294
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was cp = 0.001618524.
```




```
##           cp
## 1 0.001618524
```



The Ranger model could only be run with 10,000 data set before the computer would crash. The RMSE results for the 10,000 model was 0.9072315

The glm model was run with data sets 10,000, 100,000 and 1,000,000. The RMSE results for the 10,000 model was 0.8714957, when ran with the 100,000 data set RMSE was 0.8716486 and with the 1,000,000 data set RMSE was 0.8716314. The model information is shown below.

```
## Generalized Linear Model
##
## 1e+06 samples
## 3e+00 predictors
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 1000000, 1000000, 1000000, 1000000, 1000000, 1000000, ...
## Resampling results:
##
##   RMSE       Rsquared  MAE
##   0.8517405  0.339782  0.6607571
```

Since this model provided the best RMSE on the test data. The model was used on the validation data set to determine how well it predicted unknown ratings The RMSE for the Validation set is shown below.

```
## [1] 0.8787814
```

Conclusion

Based on the models run the glm model performed the best. With more processing power on a computer the other models could have been tested with larger data sets which may have given better results. Also looking for ways to make the code more efficient may have helped with the processing limitations that were encountered. Additional fine tuning of all the model parameters could be attempted in the future as well as testing additional models to find a better performing model.