Movie Recommendation system

Hildsley Noome

17 January 2019

Introduction

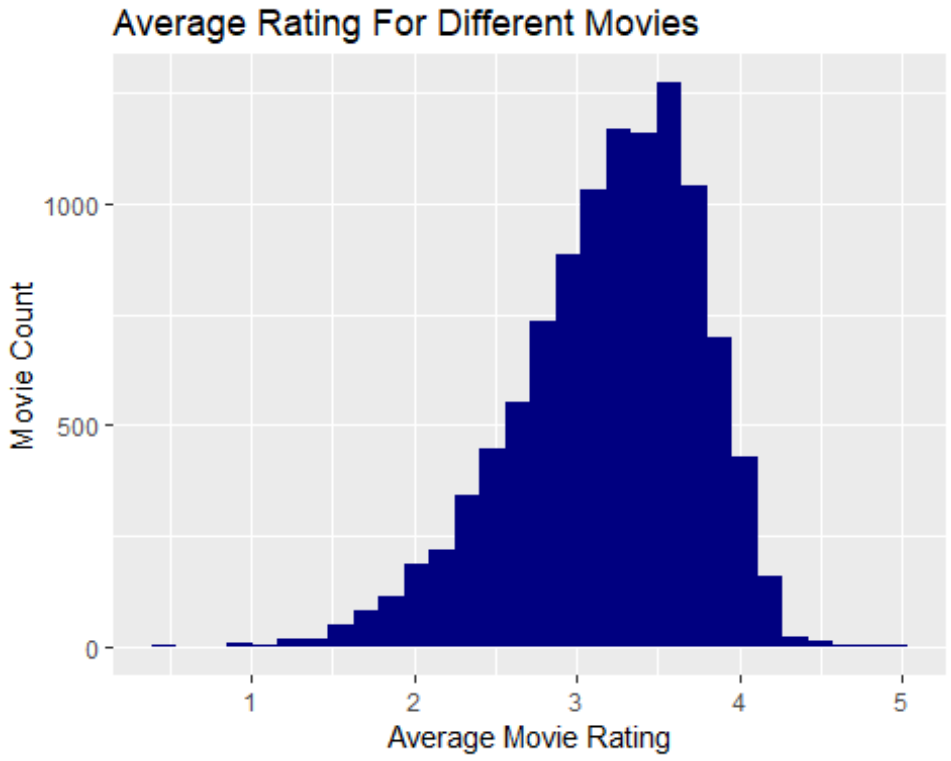
Movies are a popular form of entertainment, with a wide variety of genres. Different people have different preferences towards certain genres of which they are willing to spend time and money on. An abundance of movies are released every year and some may be excellent, while some may not fit the viewer's taste. Therefore reviews and ratings of movies could indicate whether a movie would be of good quality or one a specific viewer would enjoy watching. Grouplens is a research lab that makes data available regarding ratings of movies. Datasets containing this kind of information could be used for research and to gain insight regarding movies and how different people may enjoy different kinds of movies.

Below is an example of the data that is collected and details thereof.

```
##    userId movieId rating timestamp
## 1       1     122      5 838985046
## 2       1     185      5 838983525
## 4       1     292      5 838983421
## 5       1     316      5 838983392
## 6       1     329      5 838983392
## 7       1     355      5 838984474
## 8       1     356      5 838983653
## 9       1     362      5 838984885
## 10      1     364      5 838983707
## 11      1     370      5 838984596
##                                       title
## 1                           Boomerang (1992)
## 2                             Net, The (1995)
## 4                             Outbreak (1995)
## 5                             Stargate (1994)
## 6               Star Trek: Generations (1994)
## 7                     Flintstones, The (1994)
## 8                         Forrest Gump (1994)
## 9                     Jungle Book, The (1994)
## 10                      Lion King, The (1994)
## 11 Naked Gun 33 1/3: The Final Insult (1994)
##                                       genres
## 1                             Comedy|Romance
## 2                       Action|Crime|Thriller
## 4                  Action|Drama|Sci-Fi|Thriller
## 5                       Action|Adventure|Sci-Fi
```

```
## 6                Action|Adventure|Drama|Sci-Fi
## 7                       Children|Comedy|Fantasy
## 8                     Comedy|Drama|Romance|War
## 9                   Adventure|Children|Romance
## 10 Adventure|Animation|Children|Drama|Musical
## 11                                Action|Comedy
```

Movies are rated very differently between different people due to overall movie quality and personal taste. Below is a chart showing the average rating for the movies contained in the dataset.



Due to these differences in ratings, many attempts are made to predict the rating a viewer may give due to different reasons. Therefore this report investigates how viewers may rate movies and how closely one can predict these ratings.

Executive summary

The goal of this report is to build a model which can predict a rating that will be given by a user for a certain movie. This model will be built and trained on the dataset that was mentioned in the introduction. This model includes three variables that affect the predicted rating, the mean, the movie-specific effect and the user-specific effect. This model's accuracy is measured by the root mean squared error (RMSE) analysis technique. The final model that was trained was tested on a validation dataset of which no training was done beforehand.

Methods

Step 1:

A common measure of how close predictions are is the root mean squared error (RMSE). This calculation will be used as a measure of the overall accuracy of the predictions compared to the actual ratings. This piece of code generates a function RMSE which will be used to calculate the RMSE for the model as it is built and improved.

```
RMSE <- function(predicted_ratings, actual_ratings) {
  sqrt(mean((actual_ratings-predicted_ratings)^2))
  }
```

Step 2:

As shown in the plot in the introduction, most of the ratings lie between 3 and 4. The average rating across the dataset equals 3.5124652.

The mean across the dataset will be the first part of the prediction, hence the first variable. The model's RMSE using only the mean as the prediction equals 1.0603313.

Step 3:

The movie averages could be a powerful predictor as shown in the figure above for the average movie ratings for all movies. Although most movies are rated between 3 and 4, there are still many movies more than 4 and less than 3. To possibly improve the RMSE value, the movie-specific average ratings that deviate from the mean are calculated as b_m (the difference between mu and movie average rating). This deviation includes the movie-specific averages as a predictor.

The following code calculates b_m for each movie.

```
```r
b_m <- edx %>% group_by(movieId) %>% summarise(b_m = sum(rating - mu)/n())
head(b_m)
```


```
## # A tibble: 6 x 2
##   movieId    b_m
##     <dbl>  <dbl>
## 1       1  0.415
## 2       2 -0.307
## 3       3 -0.365
## 4       4 -0.648
## 5       5 -0.444
## 6       6  0.303
```
```

The following code will then calculate the associated predictions after b_m has been calculated.

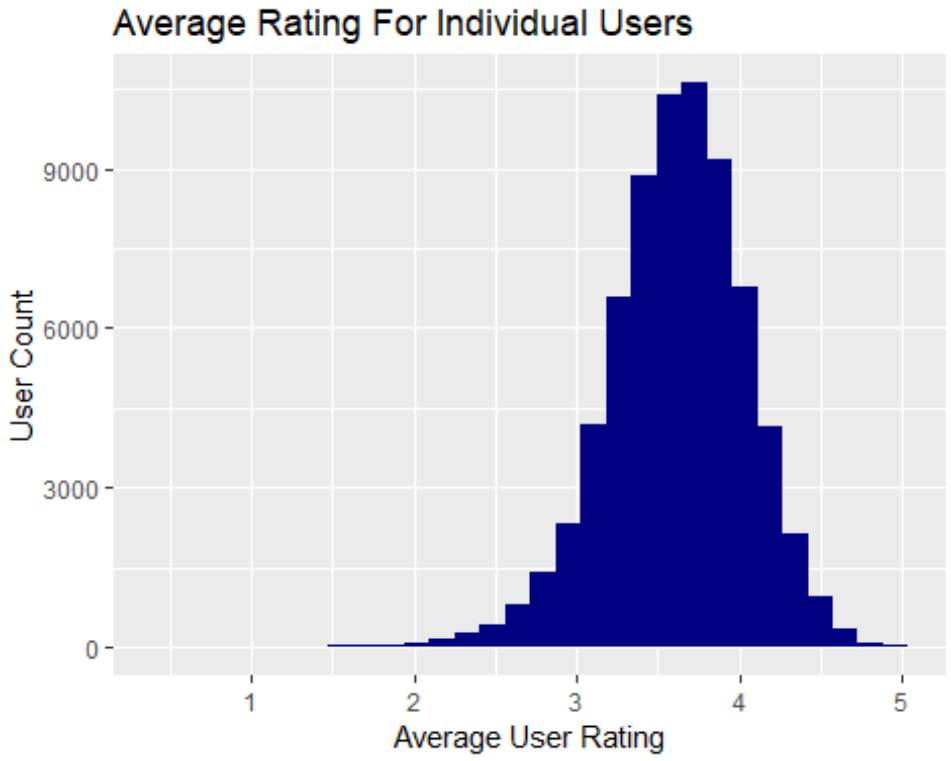
```
  predicted_ratings_b_m <- edx %>% left_join(b_m , by = "movieId") %>%
mutate(pred = mu + b_m) %>% .$pred
```

The model including the movie effects' RMSE equals 0.9423475.

This model improved the RMSE value compared to using only the mean as a predictor from 1.0603313 to 0.9423475

Step 4:

The users' personal taste and the likelihood of giving ratings could also be a powerful predictor. The following plot shows the frequency of users' average rating.



This plot shows a similar trend when compared to the average movie rating plot in the introduction. Although most user's average rating is approximately 4, there are quite a few that are a lot less or more than 4. Each user's specific effect will be determined by calculating the difference between the mean and the overall rating for each user's average rating given across al of their ratings for different movies. This value will be denoted as b_u.

The following code will calculate the user's specific effect.

```
b_u_seperate <- edx %>% group_by(userId) %>% summarise(b_u = mean(rating -
mu))
```

The following code will then calculate the associated predictions after b_u has been calculated.

```
predicted_ratings_b_u <- edx %>% left_join(b_u_seperate, by = "userId") %>%
mutate(pred = mu + b_u) %>% .$pred
```

The model including the mean and user's effect RMSE equals 0.9700086.

Following the three different steps that lead to predictions we have the following RMSE's:
Mean Predictions = 1.0603313

Movie effects = 0.9423475

User effects = 0.9700086

Step 5:

For the following predictions, the movie effect and user effect was combined.

The following code will calculate both the movie effect and the user effect, and then determine the associated predictions.

```
b_m <- edx %>% group_by(movieId) %>% summarise(b_m = sum(rating - mu)/n())
b_u <- edx %>% left_join(b_m , by = "movieId") %>% group_by(userId) %>%
summarise(b_u = mean(rating - mu - b_m))
predicted_ratings_both <- edx %>% left_join(b_m, by = "movieId") %>%
left_join(b_u , by = "userId") %>% mutate(pred = mu + b_m + b_u) %>% .$pred
```

Following the combined effects the RMSE value decreased to 0.8567039.

Step 6

After careful evaluation of the predicted ratings, the maximum predicted value was 6.1765904 while the minimum predicted value was -1.8551863. These values do not make sense because the highest rating a user is allowed is 5 and the lowest 0. Therefore these values should be changed accordingly.

The following code changes predictions above 5 to be exactly 5 and those below 0 to be exactly 0.

```
for (x in 1:length(predicted_ratings_both)) {
  if (predicted_ratings_both[x] > 5) predicted_ratings_both[x] = 5   #
changes ratings to a 5 for those above 5 predicted
  else if (predicted_ratings_both[x] < 0) predicted_ratings_both[x] = 0 #
changes ratings to 0 for predicted less than 0
}
```

The RMSE value now slightly decreased to 0.856551.

Step 7

After careful inspection of the b_m and b_u values, it was found that movies with a low number of ratings had the biggest values of b_m and b_u. Therefore predictions of these movies would yield more inaccurate results.

```
edx %>% group_by(title) %>% summarise(b_m = sum(rating -
mu)/n(),number_ratings = n()) %>% top_n(b_m,n = 10) %>% knitr::kable()
```

title	b_m	number_ratings
Blue Light, The (Das Blaue Licht) (1932)	1.487535	1
Constantine's Sword (2007)	1.237535	2
Fighting Elegy (Kenka erejii) (1966)	1.487535	1
Hellhounds on My Trail (1999)	1.487535	1
Human Condition II, The (Ningen no joken II) (1959)	1.237535	4
Human Condition III, The (Ningen no joken III) (1961)	1.237535	4
Satan's Tango (SÃ¡tÃ¡ntangÃ³) (1994)	1.487535	2
Shadows of Forgotten Ancestors (1964)	1.487535	1
Sun Alley (Sonnenallee) (1999)	1.487535	1
Who's Singin' Over There? (a.k.a. Who Sings Over There) (Ko to tamo peva) (1980)	1.237535	4

```
edx %>% group_by(title) %>% summarise(b_m = sum(rating -
mu)/n(),number_ratings = n()) %>% top_n(b_m,n = -10) %>% knitr::kable()
```

title	b_m	number_ratings
Accused (Anklaget) (2005)	-3.012465	1
Besotted (2001)	-3.012465	2
Confessions of a Superhero (2007)	-3.012465	1
Criminals (1996)	-2.512465	2
Disaster Movie (2008)	-2.653090	32
Dischord (2001)	-2.512465	1
Dog Run (1996)	-2.512465	1
From Justin to Kelly (2003)	-2.610455	199

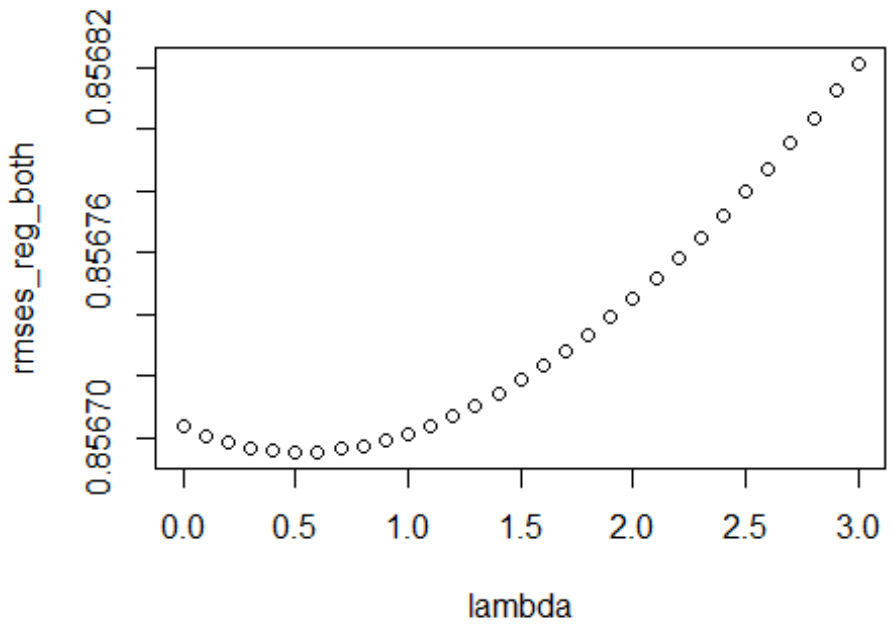
Hi-Line, The (1999)	-3.012465	1
Hip Hop Witch, Da (2000)	-2.691037	14
Monkey's Tale, A (Les ChÃ¢teau des singes) (1999)	-2.512465	1
Mountain Eagle, The (1926)	-2.512465	2
Relative Strangers (2006)	-2.512465	1
Stacy's Knights (1982)	-2.512465	1
SuperBabies: Baby Geniuses 2 (2004)	-2.717822	56
War of the Worlds 2: The Next Wave (2008)	-3.012465	2
When Time Ran Out... (a.k.a. The Day the World Ended) (1980)	-2.512465	1

To improve the model the b_m and b_u values will be regularised, taking into account the number of ratings. The following piece of code calculates the optimal penalty value of lambda that will shrink the b_m and b_u values towards zero. Higher values of lambda will shrink the values more.

```
lambda <- seq(0,3,0.1) # different values of lambda to test

rmses_reg_both <- sapply(lambda, function(l){
  mu <- mean(edx$rating)
  b_m <- edx %>% group_by(movieId) %>% summarise(b_m = sum(rating - mu)/
(n()+l)) # regularizes movies first
   b_u <- edx %>% left_join(b_m, by = "movieId") %>% group_by(userId) %>%
summarise(b_u = sum(rating - mu - b_m)/(n()+l)) # regularizes users second
predicted_ratings_reg_both <- edx %>% left_join(b_m, by = "movieId") %>%
left_join(b_u , by = "userId") %>% mutate(pred = mu + b_m + b_u) %>% .$pred
#prediction values
RMSE(predicted_ratings_reg_both,edx$rating)
})
```

The following plot shows the RMSE for the different values of lambda.



The value of lambda that minimizes the RMSE equals 0.5. This value will be used for the prediction model.

The following code calculates the regularized b_m and b_u with the associated predictions.

```
lambda_pred = lambda[which.min(rmses_reg_both)]

  b_m_reg <- edx %>% group_by(movieId) %>% summarise( b_m = sum(rating - mu)/
(n()+lambda_pred) ) # regularizes movies' effect
b_u_reg <- edx %>% left_join(b_m, by = "movieId") %>% group_by(userId) %>%
summarise(b_u = sum(rating - mu - b_m)/(n()+lambda_pred)) # regularizes
users' effect
predicted_ratings_reg_both <- edx %>% left_join(b_m, by = "movieId") %>%
left_join(b_u , by = "userId") %>% mutate(pred = mu + b_m + b_u) %>% .$pred #
prediction values

for (x in 1:length(predicted_ratings_reg_both)) {
  if (predicted_ratings_reg_both[x] > 5) predicted_ratings_reg_both[x] = 5
# changes ratings to a 5 for those above 5 predicted
  else if (predicted_ratings_reg_both[x] < 0) predicted_ratings_reg_both[x] =
0 # changes ratings to 0 for predicted less than 0
}
```

The RMSE following regularization equals 0.856551

The model's predicted rating depends on three variables, the mean, the movie-specific effect and the user-specific effect. Each of these three variables will , therefore, contribute to the final prediction of the model.

Results

The prediction model was built given a subset of an entire dataset containing 90% of the data. This model was evaluated on the validation subset of the data which no training or testing was done.

The following code will test the RMSE of the model given the validation dataset.

```
predict_val <- validation %>% left_join(b_m_reg , by = "movieId") %>%
left_join(b_u_reg , by = "userId") %>% mutate(pred = mu + b_m + b_u) %>%
.$pred

for (x in 1:length(predict_val)) {
  if (predict_val[x] > 5) predict_val[x] = 5
  else if (predict_val[x] < 0) predict_val[x] = 0
}

rmse_result_val <- RMSE(predict_val,validation$rating)
```

The RMSE of the model on the validation dataset equals 0.8650633.

This value compares excellently with the model that was built on the training dataset.

Conclusion

The prediction model's RMSE value is relatively good. This model only includes three variables and could possibly be improved in the future by adding more variables or by using other techniques either in conjunction or from another point entirely. Models predicting ratings will aid in research and help companies in the movies industry to make more informed choices regarding which types of movies are more likely to get better ratings, or even to expand on this to improve the ratio between budget spent and overall ratings versus their gross profit. Models like these could be very useful if their results are understood and if the information gained is used intelligently.