PH125.9X Capstone Project: Movielens Recommendation System

Hendrik Adriaan Nieuwenhuizen

1. Introduction

The dataset that we will be using for the project is the Movielens 10M dataset that can be downloaded from http://files.grouplens.org/datasets/movielens/ml-10m.zip

The dataset contains movie ratings for multiple movies from unique users. The data contains 10 million ratings and 100,000 tag applications applied to 10,000 movies by 72,000 users.

The goal of the project is to create a movie recommendation system. The task of the recommendation system is to fill in any "N/A's" because not every movie is rated by every user. In a perfect world every movie would have been rated by every user in an unbiased manner but this is simply not the case and we will have to try and compensate for this fact.

First step is to do some exploratory analysis and visually look at the data provided. The second step is to run models on the edx and test dataset to train the algorithm to find the lowest RMSE. Last step is to run the final model against the validation dataset (the final hold-out test set).

2. Method/analysis

2.1 Data provided and visualization

The code provided breaks up the 10 million movie ratings into an Edx dataset (which will be used to train the algorithm) and a Validation dataset (which will be the final validation set that we will run the algorithm against) in a 90/10 proportion.

The code provided has also taken care of most data cleaning.

Edx dataset provided creates a table with 6 headings with 9 000 061 lines.

```
edx
```

```
##           userId movieId rating  timestamp                          title
##       1:      1     122    5.0  838985046              Boomerang (1992)
##       2:      1     185    5.0  838983525               Net, The (1995)
##       3:      1     231    5.0  838983392          Dumb & Dumber (1994)
##       4:      1     292    5.0  838983421               Outbreak (1995)
##       5:      1     316    5.0  838983392               Stargate (1994)
##      ---
## 9000057:  59269   59680    3.0 1229014701     One Hour with You (1932)
## 9000058:  59269   64325    3.0 1229014646       Long Night, The (1947)
## 9000059:  59342   61768    0.5 1230070861    Accused (Anklaget) (2005)
## 9000060:  60713    4820    2.0 1119156754 Won't Anybody Listen? (2000)
## 9000061:  68986   61950    3.5 1223376391              Boot Camp (2007)
##                                      genres
##       1:                     Comedy|Romance
##       2:             Action|Crime|Thriller
##       3:                            Comedy
```

```
##       4:           Action|Drama|Sci-Fi|Thriller
##       5:                Action|Adventure|Sci-Fi
##      ---
## 9000057:                 Comedy|Musical|Romance
## 9000058: Crime|Drama|Film-Noir|Romance|Thriller
## 9000059:                                 Drama
## 9000060:                           Documentary
## 9000061:                              Thriller
```

Validation dataset has the same 6 headings as the Edx dataset with 999 993 lines.

```
validation
```

```
##         userId movieId rating  timestamp
##      1:      1     588    5.0  838983339
##      2:      2    1210    4.0  868245644
##      3:      2    1544    3.0  868245920
##      4:      3     151    4.5 1133571026
##      5:      3    1288    3.0 1133571035
##     ---
## 999989:  71567    1080    4.0  912580440
## 999990:  71567    1527    5.0  912580647
## 999991:  71567    1598    2.0  912649143
## 999992:  71567    1982    1.0  912580553
## 999993:  71567    1983    1.0  912580553
##                                                        title
##      1:                                          Aladdin (1992)
##      2:       Star Wars: Episode VI - Return of the Jedi (1983)
##      3: Lost World: Jurassic Park, The (Jurassic Park 2) (1997)
##      4:                                          Rob Roy (1995)
##      5:                              This Is Spinal Tap (1984)
##     ---
## 999989:                     Monty Python's Life of Brian (1979)
## 999990:                               Fifth Element, The (1997)
## 999991:                               Desperate Measures (1998)
## 999992:                                        Halloween (1978)
## 999993:                                     Halloween II (1981)
##                                          genres
##      1: Adventure|Animation|Children|Comedy|Musical
##      2:                   Action|Adventure|Sci-Fi
##      3:     Action|Adventure|Horror|Sci-Fi|Thriller
##      4:                   Action|Drama|Romance|War
##      5:                             Comedy|Musical
##     ---
## 999989:                            Adventure|Comedy
## 999990:                     Action|Adventure|Sci-Fi
## 999991:                        Crime|Drama|Thriller
## 999992:                                      Horror
## 999993:                                      Horror
```

```
#summary of edx dataset
head(edx)   #6 columns
```

```
##    userId movieId rating timestamp                          title
```

```
## 1:      1     122      5 838985046              Boomerang (1992)
## 2:      1     185      5 838983525               Net, The (1995)
## 3:      1     231      5 838983392          Dumb & Dumber (1994)
## 4:      1     292      5 838983421               Outbreak (1995)
## 5:      1     316      5 838983392               Stargate (1994)
## 6:      1     329      5 838983392 Star Trek: Generations (1994)
##                          genres
## 1:               Comedy|Romance
## 2:         Action|Crime|Thriller
## 3:                        Comedy
## 4:  Action|Drama|Sci-Fi|Thriller
## 5:       Action|Adventure|Sci-Fi
## 6: Action|Adventure|Drama|Sci-Fi
```

```
dim(edx)    #9 000 061 lines with 6 columns
```

```
## [1] 9000061       6
```

```
str(edx)    #structure of EDX dataset
```

```
## Classes 'data.table' and 'data.frame':   9000061 obs. of  6 variables:
##  $ userId   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ movieId  : num  122 185 231 292 316 329 355 356 362 364 ...
##  $ rating   : num  5 5 5 5 5 5 5 5 5 5 ...
##  $ timestamp: int  838985046 838983525 838983392 838983421 838983392 838983392 838984474 838983653 83
##  $ title    : chr  "Boomerang (1992)" "Net, The (1995)" "Dumb & Dumber (1994)" "Outbreak (1995)" ...
##  $ genres   : chr  "Comedy|Romance" "Action|Crime|Thriller" "Comedy" "Action|Drama|Sci-Fi|Thriller"
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
summary(edx)  #basic summary
```

```
##      userId         movieId          rating        timestamp
##  Min.   :    1   Min.   :    1   Min.   :0.500   Min.   :7.897e+08
##  1st Qu.:18122   1st Qu.:  648   1st Qu.:3.000   1st Qu.:9.468e+08
##  Median :35743   Median : 1834   Median :4.000   Median :1.035e+09
##  Mean   :35869   Mean   : 4120   Mean   :3.512   Mean   :1.033e+09
##  3rd Qu.:53602   3rd Qu.: 3624   3rd Qu.:4.000   3rd Qu.:1.127e+09
##  Max.   :71567   Max.   :65133   Max.   :5.000   Max.   :1.231e+09
##     title              genres
##  Length:9000061     Length:9000061
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

```
n_distinct(edx$movieId)  #how many movies in  Edx dataset
```

```
## [1] 10677
```

```
n_distinct(edx$userId)    #how many users in Edx dataset
```

```
## [1] 69878
```

```
sapply(edx, function(x) sum(is.na(x)))    #check to see N/A's in Edx dataset
```

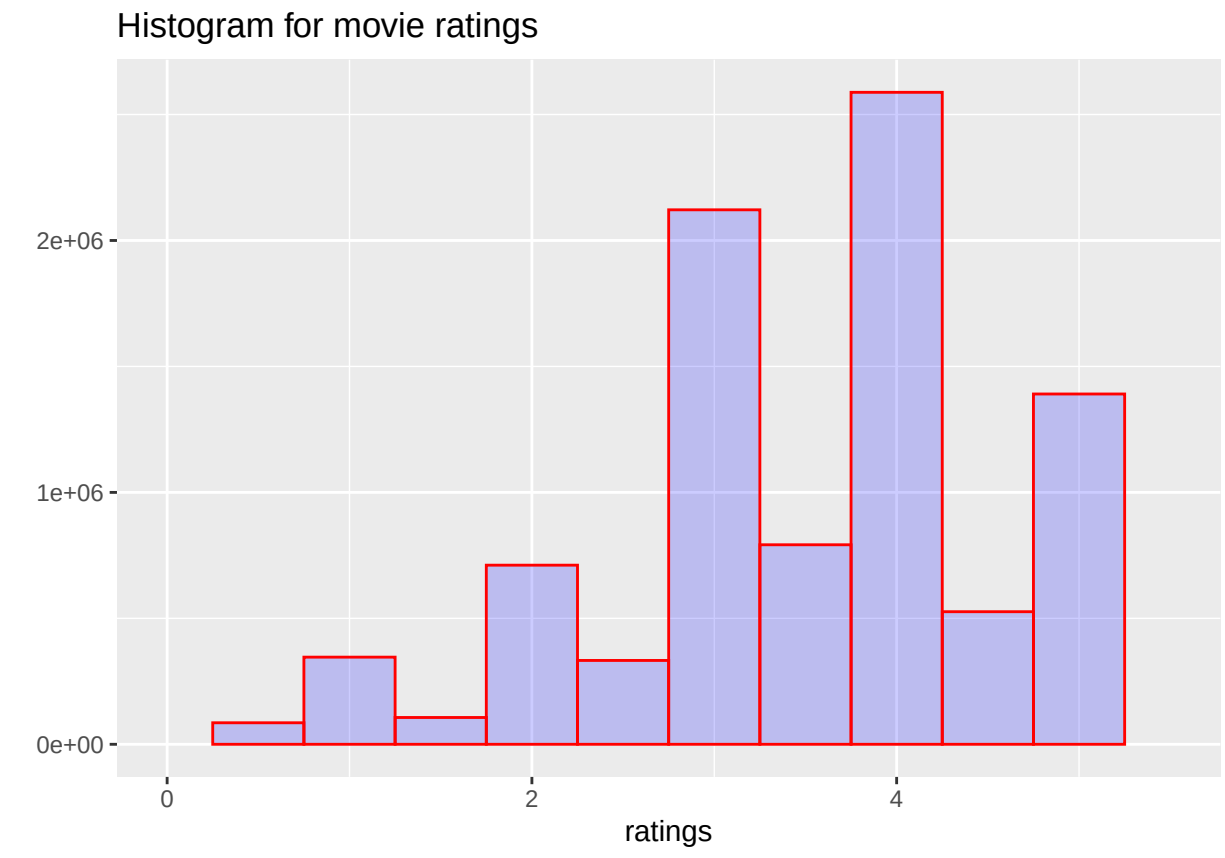
```
##    userId   movieId    rating timestamp     title    genres
##         0         0         0         0         0         0
```

```
sum(edx$rating > 5 | edx$rating <= 0)    #check to see how many ratings are not between zero and five
```

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

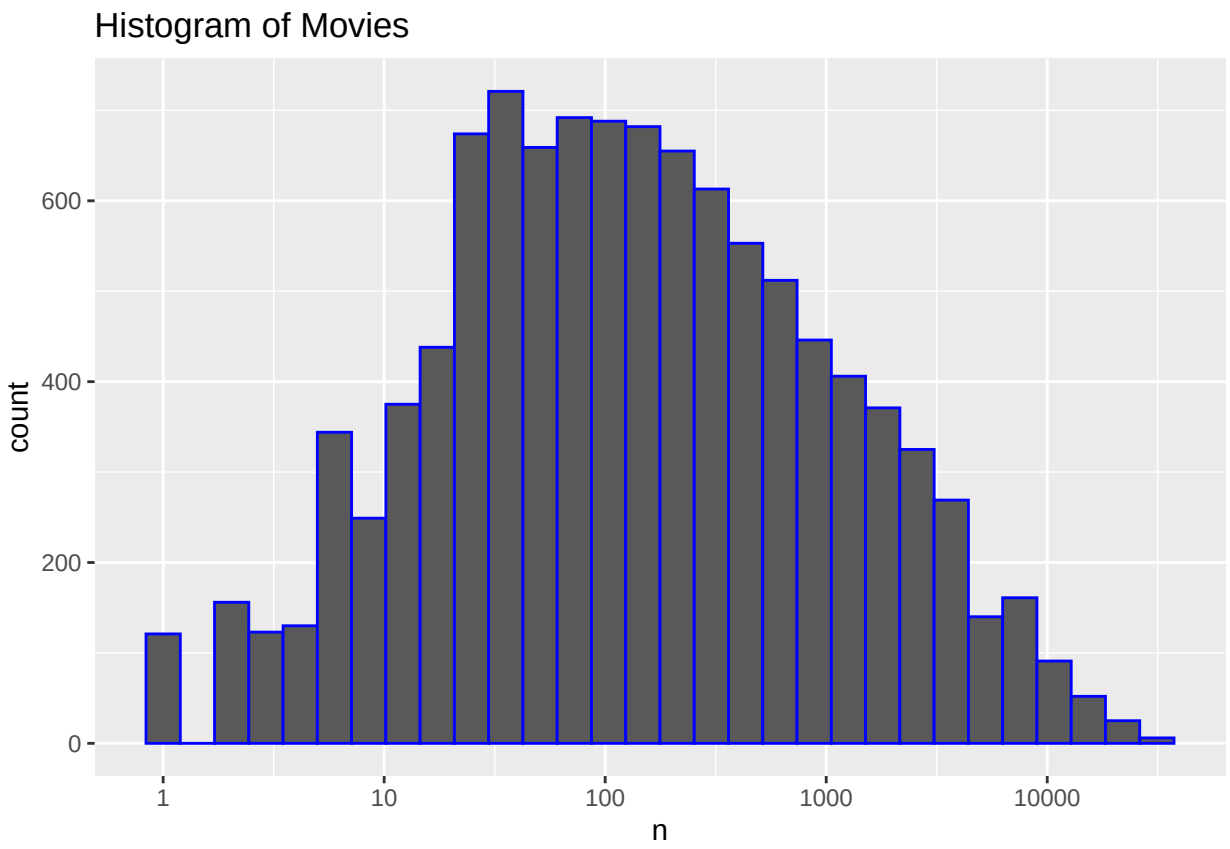
The below graph shows the distribution of movie ratings. We can see that most users prefer to give a full rating instead of a half rating.

```
qplot(edx$rating,          #distribution of movie ratings
      geom="histogram",
      binwidth = 0.5,
      main = "Histogram for movie ratings",
      xlab = "ratings",
      fill=I("blue"),
      col=I("red"),
      alpha=I(.2),
      xlim=c(0.0, 5.5))
```



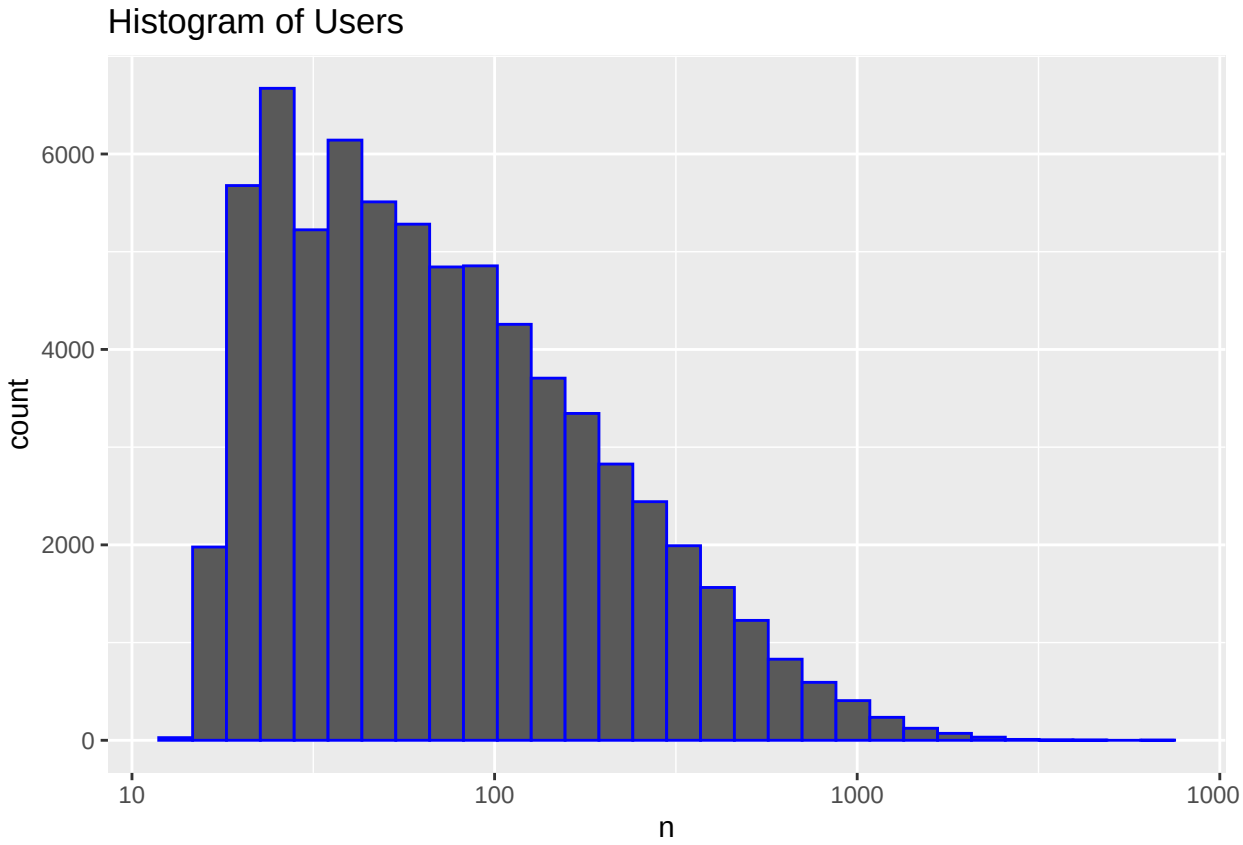
The below graph shows that some movies are rated more than others.

```
edx %>%
  dplyr::count(movieId) %>%
  ggplot(aes(n)) +
  geom_histogram(bins = 30, color = "blue") +
  scale_x_log10() +
  ggtitle("Histogram of Movies")     ##some movies are rated more than others
```



Some users have rated more than 1000 movies, this also shows than some users rate more than others.

```
edx %>%
  dplyr::count(userId) %>%
  ggplot(aes(n)) +
  geom_histogram(bins = 30, color = "blue") +
  scale_x_log10() +
  ggtitle("Histogram of Users")    #some users have rated over 1000 movies
```



2.2 Datasets to train algorithm

Next we split the EDX dataset into a train and test set in a 50/50 proportion. The training set and test set has approximately 4.5 million lines of data each. These 2 datasets will be used to train the algorithm.

```
edx_train_set
```

```
##          userId movieId rating  timestamp                         title
##       1:      1     122    5.0  838985046              Boomerang (1992)
##       2:      1     231    5.0  838983392           Dumb & Dumber (1994)
##       3:      1     316    5.0  838983392                Stargate (1994)
##       4:      1     355    5.0  838984474       Flintstones, The (1994)
##       5:      1     356    5.0  838983653            Forrest Gump (1994)
##      ---
## 4500301:  23140   39439    3.5 1159527126 God's Sandbox (Tahara) (2002)
## 4500302:  27946   63327    3.0 1226730298  Säg att du älskar mig (2006)
## 4500303:  40976   61913    3.0 1227767528            Africa addio (1966)
## 4500304:  59269   64325    3.0 1229014646        Long Night, The (1947)
## 4500305:  60713    4820    2.0 1119156754  Won't Anybody Listen? (2000)
##                                      genres
##       1:                     Comedy|Romance
##       2:                             Comedy
##       3:               Action|Adventure|Sci-Fi
##       4:               Children|Comedy|Fantasy
##       5:              Comedy|Drama|Romance|War
##      ---
```

```
## 4500301:                                  Drama
## 4500302:                                  Drama
## 4500303:                            Documentary
## 4500304: Crime|Drama|Film-Noir|Romance|Thriller
## 4500305:                            Documentary
```

```
edx_test_set
```

```
##          userId movieId rating timestamp
##       1:      1     185      5 838983525
##       2:      1     292      5 838983421
##       3:      1     329      5 838983392
##       4:      1     362      5 838984885
##       5:      1     370      5 838984596
##      ---
## 4499752:  71567    1917      4 912580787
## 4499753:  71567    1920      4 912578247
## 4499754:  71567    1984      1 912580553
## 4499755:  71567    2028      5 912580344
## 4499756:  71567    2384      2 912578173
##                                          title
##       1:                          Net, The (1995)
##       2:                          Outbreak (1995)
##       3:             Star Trek: Generations (1994)
##       4:                   Jungle Book, The (1994)
##       5: Naked Gun 33 1/3: The Final Insult (1994)
##      ---
## 4499752:                       Armageddon (1998)
## 4499753:                     Small Soldiers (1998)
## 4499754: Halloween III: Season of the Witch (1982)
## 4499755:                Saving Private Ryan (1998)
## 4499756:             Babe: Pig in the City (1998)
##                                 genres
##       1:          Action|Crime|Thriller
##       2:   Action|Drama|Sci-Fi|Thriller
##       3:  Action|Adventure|Drama|Sci-Fi
##       4:     Adventure|Children|Romance
##       5:                  Action|Comedy
##      ---
## 4499752: Action|Romance|Sci-Fi|Thriller
## 4499753: Animation|Children|Fantasy|War
## 4499754:                         Horror
## 4499755:               Action|Drama|War
## 4499756:                Children|Comedy
```

2.3 Measuring success

The loss function that we will use to measure the accuracy of our model to predict movie ratings will be the residual mean squared error (RMSE). The lower the number the better.

2.4 Modeling approach

A. We start with the assumption that all movies and all users have the same rating. We calculate the average and RMSE based on the average vs the test set. "u" represents the true rating for all movies. "e" is the independent errors sampled.

B. We then add "bi" that represents the average ranking for movie i. This movie effect comes from the observed idea that some movies are generally rated differently , also known as bias.

C. We then add "bu" to represent the user effect. This is users that give good movies a bad rating for reasons unknown.

D. Now we regularize the movie and user effect. This penalizes large estimates from small sample sizes and improves the model further. We also used optimization to get the lowest lambda for the regularization of the movie and user effect.

E. Finally we run the same models specified above on the train set against final validation set.

3. Results

A. We start with the assumption that all movies and all users have the same rating. "u" represents the true rating for all movies. We then add "bi" that represents the average ranking for movie i. "e" is the independent errors sampled. Average movie rating is 3.51 on the train set. The average movie RMSE is 1.06 and adding the movie effect gives a RMSE of 0.94.

```
#average
mu <- mean(edx_train_set$rating)
mu             #average rating on training data, mu minimizes the RMSE. We will predict the same rating
```

```
## [1] 3.512307
```

```
naive_rmse <- RMSE(edx_test_set$rating, mu)
naive_rmse         #baseline model
```

```
## [1] 1.060165
```

```
rmse_results <- data_frame(method = "The average", RMSE = naive_rmse)  #we create a table with our stor
```

```
## Warning: `data_frame()` is deprecated as of tibble 1.1.0.
## Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
rmse_results %>% knitr::kable()      #RMSE has dropped to 0.94
```

method	RMSE
The average	1.060165
Movie Effect Model on test set	0.943921

B. We then add bu to represent the user effect. This is users that give good movies a bad rating for reasons unknown. This improves the model and gives a lower RMSE of 0.87

```
rmse_results %>% knitr::kable()    #RMSE has dropped to 0.87
```

method	RMSE
The average	1.0601647
Movie Effect Model on test set	0.9439210
Movie Effect + User Effect Model on test set	0.8693527

C. Now we regularize the movie and user effect. This penalizes large estimates from small sample sizes and improves the model further. We also used optimization to get the lambda that produces the lowest RMSE possible for the movie and movie + user effect.

```
rmse_results %>% knitr::kable()
```

method	RMSE
The average	1.0601647
Movie Effect Model on test set	0.9439210
Movie Effect + User Effect Model on test set	0.8693527
Regularized Movie Effect Model on test set	0.9437904

```
rmse_results %>% knitr::kable()
```

method	RMSE
The average	1.0601647
Movie Effect Model on test set	0.9439210
Movie Effect + User Effect Model on test set	0.8693527
Regularized Movie Effect Model on test set	0.9437904
Regularized Movie Effect + User Effect Model on test set	0.8677564

D. Finally we run the same model specified above on the train set against the final validation set. The final RMSE regularized for movie and user effect was 0.8649. I'm very happy with this result.

```
rmse_results %>% knitr::kable()
```

method	RMSE
The average	1.0601647
Movie Effect Model on test set	0.9439210
Movie Effect + User Effect Model on test set	0.8693527
Regularized Movie Effect Model on test set	0.9437904
Regularized Movie Effect + User Effect Model on test set	0.8677564
Regularized Movie Effect Model on validation set	0.9436515

```
rmse_results %>% knitr::kable()
```

method	RMSE
The average	1.0601647
Movie Effect Model on test set	0.9439210

method	RMSE
Movie Effect + User Effect Model on test set	0.8693527
Regularized Movie Effect Model on test set	0.9437904
Regularized Movie Effect + User Effect Model on test set	0.8677564
Regularized Movie Effect Model on validation set	0.9436515
Regularized Movie + User Effect Model on validation set	0.8649857

4. Conclusion

The purpose of the exercise is to see if I can train an algorithm to create a movie recommendation system to fill in the N/A's because not every movie is rated by every user. This was done by starting with an average and then adding the movie and user effect. Further improvement was done by regularization of the movie and user effect which reduced the RMSE further to confirm which model is the most appropriate to use.

Some limitations of the above project are that it only focused on the movie and user effects.

Future work for me will be to implement the knowledge I've gained in this course in my analytics role within the financial services industry. Focusing spesifically on using R to create financial models and give insights on statistics.