Movie Recommendation System Project

Yigit Kalyoncu

12/15/2020

OVERVIEW:

The goal of this project is to develop a movie recommendation system model using the EDX data set provided by the course.

Upon completing the task, the RMSE would be calculated using the hold-out validation set, also created with the code provided by the course.

Since the EDX data set itself, and the corresponding training set created is quite large with approximately 7 million instances that involves tens of thousands of unique users, movies, and even hundreds of genres, exploring via straight-forward visualizations would not be so easy.

```
##      userId         movieId          rating        timestamp
##  Min.   :    1   Min.   :    1   Min.   :0.500   Min.   :7.897e+08
##  1st Qu.:18127   1st Qu.:  648   1st Qu.:3.000   1st Qu.:9.468e+08
##  Median :35750   Median : 1834   Median :4.000   Median :1.035e+09
##  Mean   :35872   Mean   : 4121   Mean   :3.513   Mean   :1.033e+09
##  3rd Qu.:53607   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:7200043     Length:7200043
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
```

```
##   movies users genres
## 1  10641 69878    794
```

So instead, it makes sense to take a more generalized and examplified approach to explore the data and come to conclusions.

Upon exploration of the data, average overall ratings, regularized movie effect, user effect, genre effect and year effect are used to predict the ratings for each user & movie pair in the data set. Penalty terms are calculated to minimize RMSE. Lambda penalty terms are tuned via 10 fold cross validation.

Lastly, the final model is applied to the hold-out validation set, RMSE is calculated & results, performance, limitations and possible improvements are discussed.

METHOD:

PREPARATION AND EXPLORATION:

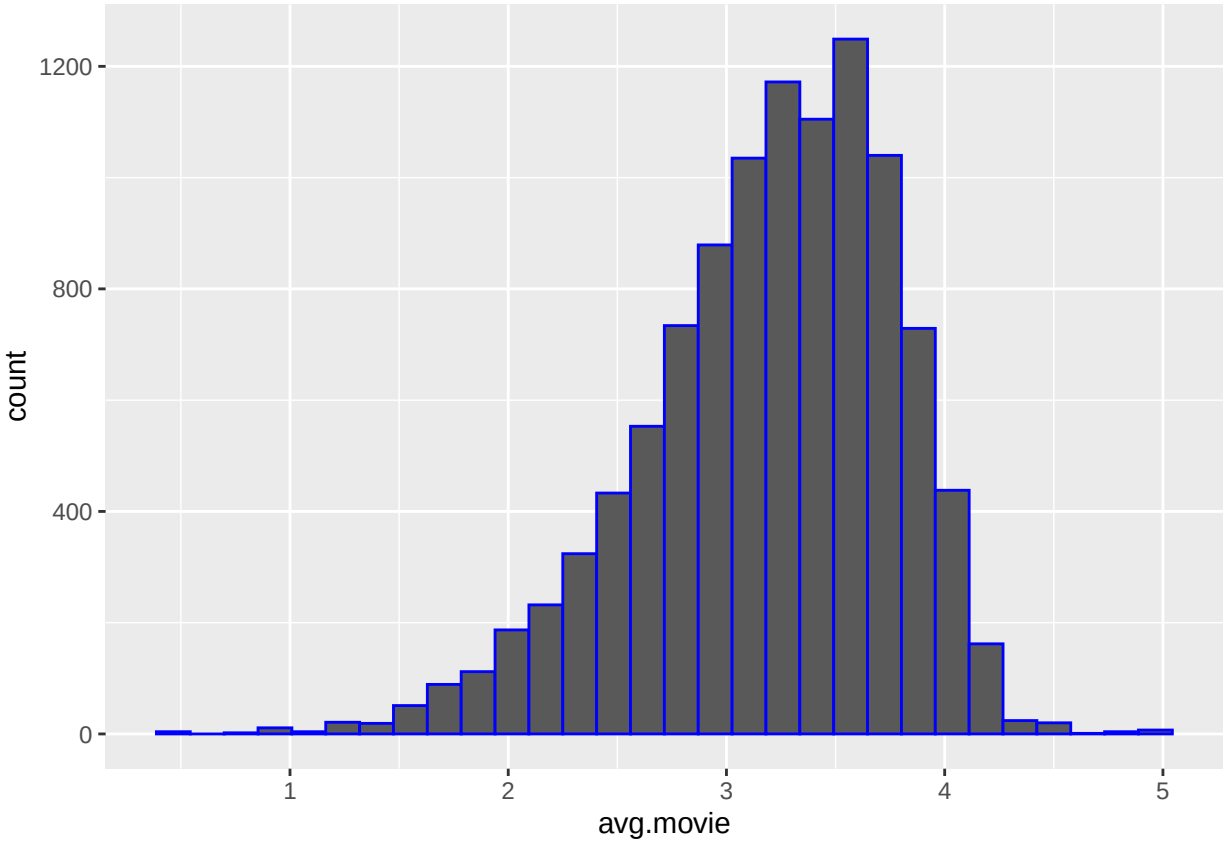
The EDX data set consists of approximately 9 million instances with 6 variables. Among the 6 variables, 3 appear to be usable when creating a model. These are userId, movieId & genre.

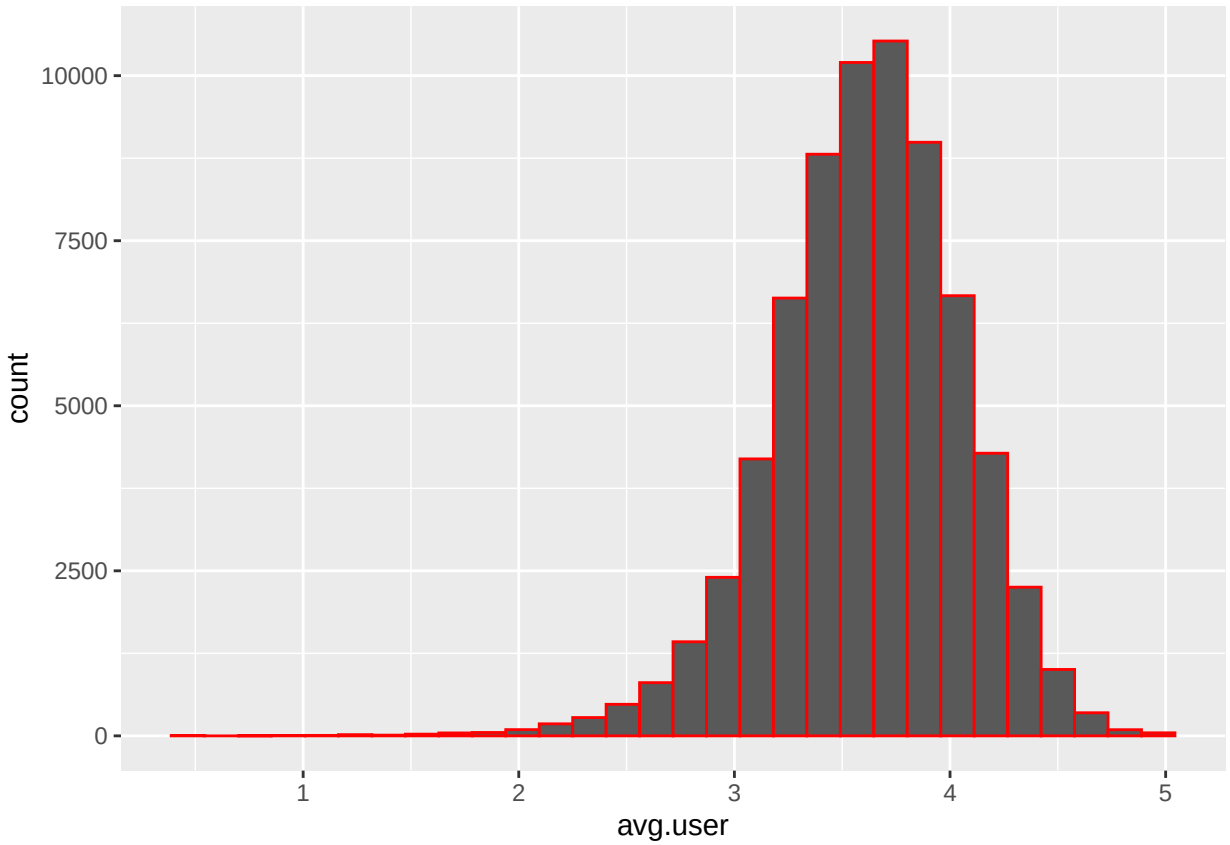
In addition, there is 1 more hidden variable that is embedded within the title varible, which is the year of the movie. This can also be used to create a model for predictions, once it is extracted from the title.

After exanining the data set broadly, training and test sets are created. 20% of the EDX data is used for the test set, and 80% is used for the training set. Then users and movies that do not appear in the training set are removed from the test set to avoid NA's.

The overall mean rating for the training set is calculated in order to be used as a benchmark for further exploration.

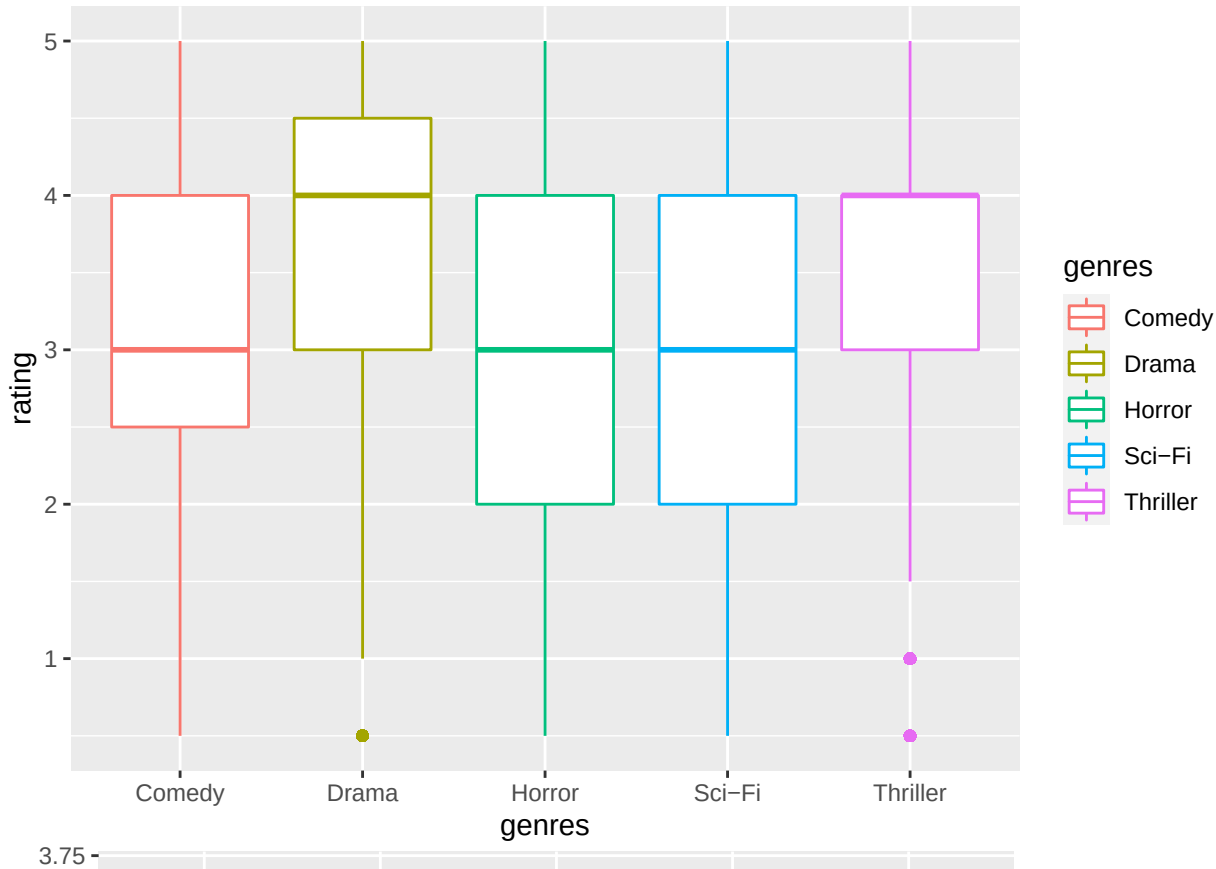
Mean rating for movies with more than 100 ratings, and users with more than 80 ratings is calculated and plotted. The predictors are filtered as such since the result we get on more prominent users and movies would be reflective enough to make a generalization.

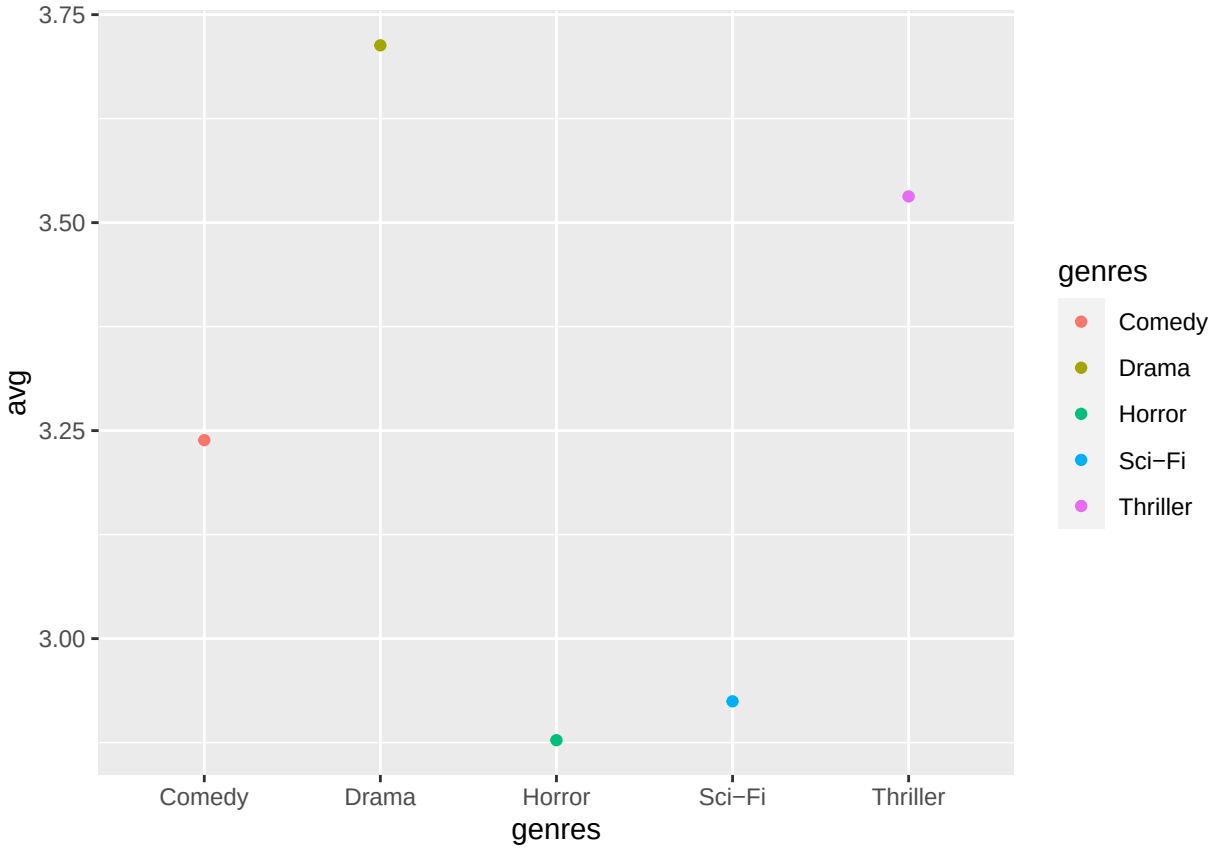




These plots indicate that there is indeed a movie and user effect for ratings, meaning certain movies appear to be consistently rated with a certain way, and users also appear to rate movies with biases of their own.

To find out more about the genre effect, select genres are filtered and plotted as a boxplot. The mean rating for selected genres are also calculated. The resulting boxplot and mean ratings vary enough to conclude that there is also a genre effect. The movies that belong to different genres tend to get different ratings overall.


rating
5
4
3
2
1
Comedy
Drama
Horror
Sci−Fi
Thriller
genres
genres
Comedy
Drama
Horror
Sci−Fi
Thriller


avg
3.75
3.50
3.25
3.00
Comedy
Drama
Horror
Sci−Fi
Thriller
genres
genres
Comedy
Drama
Horror
Sci−Fi
Thriller

genres	avg.rating
Comedy	3.238538
Drama	3.713107
Horror	2.877901
Sci-Fi	2.924619
Thriller	3.531481

To examine the year effect, the years in which the movies were made are extracted from the title variable.

The code used to extract and modify the year predictor is shown below:

```
year.train <- training.set %>% mutate(year = str_extract(title, pattern =  "\\(\\d{4}\\)")) %>%
  mutate(year = str_remove(year, '\\(')) %>%
  mutate(year = str_remove(year, '\\)')) %>%
  mutate(year = as.numeric(year))%>%
  select(movieId,year) %>% distinct()
```

After the extraction, year effect is explored via first calculating average rating per year, and the examining the standard deviation, minimum, median and maximum ratings among years.

There seems to be note worthy difference between ratings among different years. Both standard deviation and variance are quite large.

mean.avg.years	sd.avg.years	min.avg.years	median.avg.years	max.avg.years
3.724424	0.2064097	3.33913	3.758257	4.054017

Anova test performed also supports the rating difference between years is significant since it yields a very low P-Value and high F-Value.

```
##                   Df  Sum Sq Mean Sq F value Pr(>F)
## year              93  208618  2243.2    2037 <2e-16 ***
## Residuals   8999961 9910166     1.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

MODEL AND APPROACH:

The model proposed to predict ratings consist of overall mean rating as a base. Then movie effect, user effect, genre effect and year effect are calculated and added in order.

All effects are calculated using a penalization term to minimize the undesired effect of fringe data points.

Further, penalty terms for each effect are tuned in order achieve the lowest RMSE possible. The initial tuning was done using broader ranges and fewer points(This step is not included in the final code). Using the result of this initial tuning, final tuning was done in a more precise manner.

To tune the penalty terms while avoiding overtraining, 10 fold cross validation is used. The folds are created using the training set.

The coding approach used is similar for all effects:

- The overall average is deducted from the rating
- Observations are grouped by the effect being measured

- The penalty term is optimized based on the RMSE of the validation sets, and average of the penalty terms that result in lowest RMSE's are used
- The effects are calculated using the tuned penalty term
- The model is applied to the test set and rmse is obtained
- The next effect is calculated using the same steps, but also accounting for the effects that were calculated previously

Code used for the movie effect calculations is show below:

```
movie.effect.tuner <- seq(1, 3, 0.1)

movie.tuner.rmses <- lapply(indexes, function(fold){
  rmses <- sapply(movie.effect.tuner, function(lambda){
    mu <- mean(training.set[-fold,]$rating)
    summer <- training.set[-fold,] %>%
      group_by(movieId) %>%
      summarize(sum = sum(rating - mu), rated = n())
    predicted <- training.set[fold,] %>%
      semi_join(training.set[-fold,], by = 'movieId') %>%
      left_join(summer, by = 'movieId') %>%
      mutate(me = sum/(rated + lambda)) %>%
      mutate(pred = mu + me) %>%
      summarize(rmse = sqrt(mean((rating - pred)^2)))
  })
  return(movie.effect.tuner[which.min(rmses)])
})

#determining the best penalty term
best.movie.tuner <- mean(as.numeric(movie.tuner.rmses))

#calculating movie effect for the whole training set & test set

tr.movie.effect <- training.set %>%
  group_by(movieId) %>%
  summarize(me = sum(rating - mu.train)/(n() + best.movie.tuner))

edx.movie.effect <- edx %>%
  group_by(movieId) %>%
  summarize(me = sum(rating - mu.train)/(n() + best.movie.tuner))

#calculating rmse on the test set

movie.model.rmse <- test.set %>%
  left_join(tr.movie.effect, by = 'movieId') %>%
  mutate(predicted = mu.train + me) %>%
  summarize(rmse.movie = sqrt(mean((rating - predicted)^2)))
```

RESULT:

As became evident when the models were applied to the test set, the effect by movie and user seem to be the strongest, since they lower the RMSE most significantly.

Genre and year effects can be concluded to be weaker since the resulting change in RMSE is lower when they are added to the model.

	RMSE
Overall Mean Model	1.06070449352326
Reg. Movie Effect Model	0.943680219141195
Reg. Movie + User Effect Model	0.865550241538415
Reg. Movie + User + Genre Effect Model	0.865257560516438
Reg. Movie + User + Genre + Year Effect Model	0.865110531833507

The tuned penalty terms for each effect are listed below:

Effect Type	Lambda
Movie Effect	2.31
User Effect	5.1
Genre Effect	4.48
Year Effect	5.23

On the validation set, due to a reason that was unable to be found, there were 8 rows films where the movie effect turned out to be NA. These rows were removed to calculate the RMSE for the final model.

For the final validation, the model is created using the whole EDX data set to increase accuracy. Since the final validation set still remains untouched, usage of the whole EDX data is believed to be appripriate and could not be considered overtraining. (If the validation set did not exist, then using a model based on the total data set would most likely yield overtrained results.)

The penalty terms thats were tuned with training validation sets are applied to corresponging predictors on EDX set, and the effect of the predictors are calculated.

Finally, the effects are added to the validation data set by predictor, and the final RMSE is obtained.

```
##   rmse.validation
## 1       0.8643249
```

The RMSE result of the validation test is lower than what was seen on the test set. This is due to the fact that only training set was used for model that was tested on the test set.

However, since the final model is produced using the whole data set, the model became more accurate and yielded a notably improved result.

The general performance of the model is limited, although when the simple approach and the relatively short computing time is taken into account, it can still be considered a quite good and useful model.

CONCLUSION AND FURTHER ANALYSIS:

As mentioned above, the model used is certainly one of the simplest approaches that can be taken, but still yields a satisfactory result.

The number of useful predictors in the given data is relatively small. This approach could have been extended to include further predictors (for example Age Group of the User, Country of the User, Budget of the Movie etc.), which could in return improve the predictions made. Given that this is a movie recommendation system, the company that requires the system would most likely have access to the data regarding the possible predictors given as exmaple, so it would not be a unrealistic idea to include them in the model.

In addition, more complex and time consuming approaches & models such as clustering, matrix factorization, k-nearest neighbors, or regression could almost certainly improve the results.

An ensemble of such models, used by averaging the predictions gained from each model would probably be the approach that produces the most refined result, and it could further be improved by getting a weighted average based on how much they improve the RMSE individually.