Data Science: Capstone Project - Movie Lens Report

Man Chun Hui

Abstract

In this **project**, we created a movie recommendation system using a large subset of the MovieLens dataset. Exploration of the *edx* dataset showed strong evidence of a genre effect and if used to augment the ***Regularized Movie + User Effect Model***[1] could yield improvements in the accuracy of the model. This hypothesis was validated when the final model, with the addition of an regularized predicted genre effect "*bias*" term, yielded a residual mean squared error (**RMSE**) of **0.8575** on the *validation* dataset.

1 Introduction

The goal of this project was to create a movie recommendation system using a subset of the MovieLens dataset generated by the the GroupLens research lab[2]. It is worth noting that the version of the movielens dataset included in the dslabs package (which was used for some of the exercises in PH125.8x: Data Science: Machine Learning) is just a small subset of the much larger dataset which has millions of ratings, whereas in this project we be will using a larger proportion of the MovieLens dataset, that has 10M ratings, to help create our recommendation system.

The code provided in the project brief partitions the Movielens data into two seperate subsets one named **edx** and the other **validation**. The **edx** subset was used to train a machine learning algorithm and the **validation** subset was used to predict movie ratings and provide the final **RMSE** value.

Table 1: EDX dataset exploration

userId	movieId	rating	timestamp	title	genres
1	122	5	838985046	Boomerang (1992)	Comedy|Romance
1	185	5	838983525	Net, The (1995)	Action|Crime|Thriller
1	292	5	838983421	Outbreak (1995)	Action|Drama|Sci-Fi|Thriller
1	316	5	838983392	Stargate (1994)	Action|Adventure|Sci-Fi
1	329	5	838983392	Star Trek: Generations (1994)	Action|Adventure|Drama|Sci-Fi
1	355	5	838984474	Flintstones, The (1994)	Children|Comedy|Fantasy

Exploration of the **edx** subset, in Table 1 above and Table 2 below, showed that the data is in tidy format with each row representing a rating given by one user to one movie and in total there are approximately 9M ratings from 69,878 unique users that provided ratings for 10,677 unique movies.

Table 2: EDX dataset exploration

No. Of Rows	No. Of Columns	No. of Movies	No. Of Users
9000055	6	10677	69878

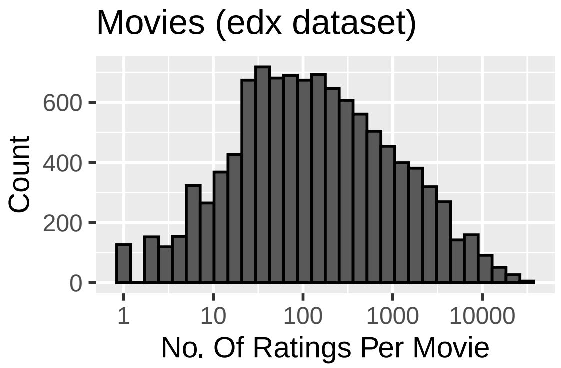
[1] https://rafalab.github.io/dsbook/large-datasets.html#recommendation-systems

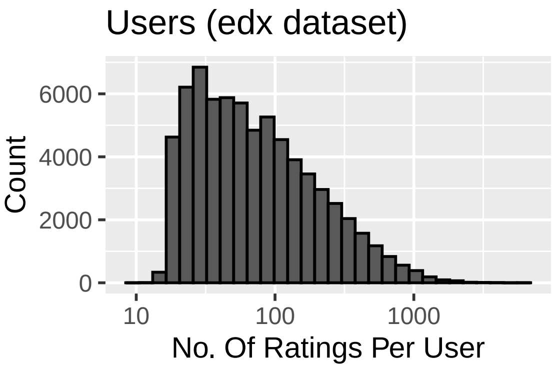
[2] https://grouplens.org/

2 Methods / Analysis

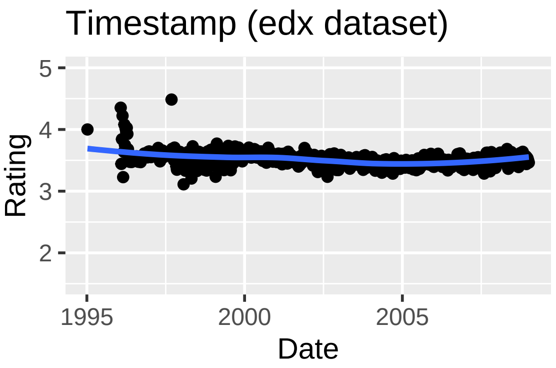
2.1 Movielens edx data

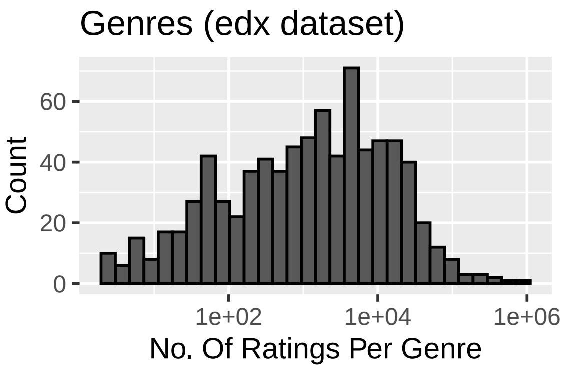
Exploration of the edx data confirmed that the observations[3] noted in ***Chapter 33.7.1 - Movielens data*** of the smaller movielens data set still holds true for the larger dataset, this therefore confirms the continued usefulness that movie effects and users effects will have on predicting moving ratings.


Movies (edx dataset)


Users (edx dataset)

To improve accuracy of the ML algorithm / model it was necessary to explore if either of the remaining predictors, timestamp or genres, in the edx dataset could prove useful. It quickly became evident, from the two figures below, that there is a genre effect where certain genres are rated more than others while there is little evidence of a useful time effect as the average rating is fairly consistant overtime.


Timestamp (edx dataset)


Genres (edx dataset)

2.2 Modelling genre effects

Adapting the model detailed in ***Chapter 33.7.6 - User Effects***[4] to add the genre effects term to predict the rating of a movie ***i*** by user ***u*** with different biases for different genre's ***g***:

$$Y_{g,u,i} = \mu + b_i + b_u + b_g + \epsilon_{g,u,i}$$

Where $\hat{b_g}$ could be estimated by taking the average of:

$$\hat{b_g} = \hat{y_{g,u,i}} - \hat{\mu} - \hat{b_i} - \hat{b_u}$$

However to ensure that the **RMSE** is not negatively affected by movies rated by very few users, regularization[5] is required.

[3] https://rafalab.github.io/dsbook/large-datasets.html#movielens-data, **Observation 1** - Not every user rated every movie, **Observation 2** - Some movies get rated more than others, **Observation 3** - Some users are more active than others at rating movies.

[4] https://rafalab.github.io/dsbook/large-datasets.html#user-effects

[5] ***Chapter 33.9 - Regularization*** - https://rafalab.github.io/dsbook/large-datasets.html#regularization

2.3 Regularization

Regularization of each of the effects $\hat{b_i}$, $\hat{b_u}$ and $\hat{b_g}$ was required, to ensure the **RMSE** is not negatively affected by movies rated by very few users, and is covered below. It is worth highlighting the the equations for $\hat{b_i(\lambda)}$ and $\hat{b_u(\lambda)}$ is shown in ***Chapter 33.9 - Regularization***[6], so therefore only the equation form is shown in the interests of leaning out this report.

The equation for movie effect estimate $\hat{b_i(\lambda)}$ is as below:

$$\hat{b_i(\lambda)} = \frac{1}{\lambda + n_i} \sum_{n=1}^{n_i} (Y_{g,u,i} - \hat{\mu})$$

The equation for user effect estimate $\hat{b_u(\lambda)}$ is as below:

$$\hat{b_u(\lambda)} = \frac{1}{\lambda + n_i} \sum_{n=1}^{n_i} (Y_{g,u,i} - \hat{\mu} - \hat{b_i})$$

The equation for genre effect estimate $\hat{b_g(\lambda)}$ is as below:

$$\hat{b_g(\lambda)} = \frac{1}{\lambda + n_i} \sum_{n=1}^{n_i} (Y_{g,u,i} - \hat{\mu} - \hat{b_i} - \hat{b_u})$$

And when coded estimate $\hat{b_g(\lambda)}$ is as below:

```
b_g <- train_set %>%
  left_join(b_i, by="movieId") %>%
  left_join(b_u, by="userId") %>%
  group_by(genres, userId) %>%
  summarize(b_g = sum(rating - mu - b_i - b_u)/(n()+l))
```

2.4 Loss function

Before moving forward we will briefly go over the the **RMSE** function detailed in ***Chapter 33.7.3 - Loss function***[7] as it was re-used to measure the accuracy of the ML model, refer to the equation and code used below:

$$RMSE = \sqrt{\frac{1}{N} \sum_{u,i} (\hat{y_{u,i}} - y_{u,i})^2}$$

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

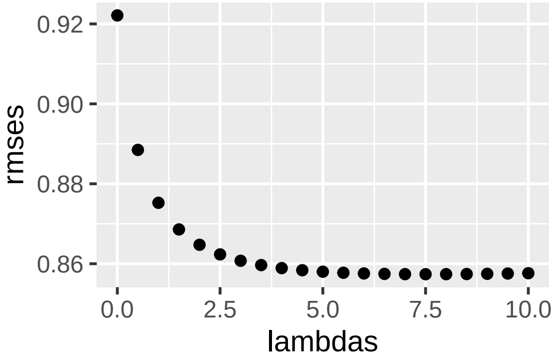
[6] https://rafalab.github.io/dsbook/large-datasets.html#regularization

[7] https://rafalab.github.io/dsbook/large-datasets.html#netflix-loss-function

2.5 Choosing the penalty term

Cross validation is used select the best λ[8], the following code was used achieve this:

```r
lambdas <- seq(0, 10, 0.5)

rmses <- sapply(lambdas, function(l){

  mu <- mean(train_set$rating)

  b_i <- train_set %>%
    group_by(movieId) %>%
    summarize(b_i = sum(rating - mu)/(n()+l))

  b_u <- train_set %>%
    left_join(b_i, by="movieId") %>%
    group_by(userId) %>%
    summarize(b_u = sum(rating - mu - b_i)/(n()+l))

  b_g <- train_set %>%
    left_join(b_i, by="movieId") %>%
    left_join(b_u, by="userId") %>%
    group_by(genres, userId) %>%
    summarize(b_g = sum(rating - mu - b_i - b_u)/(n()+l))

  predicted_ratings <-
    test_set %>%
    left_join(b_i, by = "movieId") %>%
    left_join(b_u, by = "userId") %>%
    left_join(b_g, by= c('userId','genres'))
  predicted_ratings$b_g[is.na(predicted_ratings$b_g)] = 0

  predicted_ratings <- predicted_ratings %>%
    mutate(pred = mu + b_i + b_u + b_g) %>%
    .$pred

  return(RMSE(predicted_ratings, test_set$rating))
})
qplot(lambdas, rmses)
```



[8] https://rafalab.github.io/dsbook/large-datasets.html#choosing-the-penalty-terms

3 Results

3.1 Optimal lambda

For the full ***Regularized Movie + User + Genre Effect Model***, the optimal λ is:

```
lambda <- lambdas[which.min(rmses)]
```

```
## [1] 7.5
```

3.2 Predicted ratings and Final RMSE

Using the optimal λ all the effects estimates $\hat{b_i(\lambda)}$, $\hat{b_u(\lambda)}$ and $\hat{b_g(\lambda)}$ was re-calculated and used to create the final predicted ratings $\hat{Y_{g,u,i}}$ using the follow code:

```
mu <- mean(train_set$rating)

b_i <- train_set %>%
  group_by(movieId) %>%
  summarize(b_i = sum(rating - mu)/(n()+lambda))

b_u <- train_set %>%
  left_join(b_i, by="movieId") %>%
  group_by(userId) %>%
  summarize(b_u = sum(rating - mu - b_i)/(n()+lambda))

b_g <- train_set %>%
  left_join(b_i, by="movieId") %>%
  left_join(b_u, by="userId") %>%
  group_by(genres, userId) %>%
  summarize(b_g = sum(rating - mu - b_i - b_u)/(n()+lambda))

predicted_ratings <- validation %>%
  left_join(b_i, by = "movieId") %>%
  left_join(b_u, by = "userId") %>%
  left_join(b_g, by= c('userId','genres'))
predicted_ratings$b_g[is.na(predicted_ratings$b_g)] = 0

predicted_ratings <- predicted_ratings %>%
  mutate(pred = mu + b_i + b_u + b_g) %>%
  .$pred
```

And the final RMSE value is:

```
RMSE_Final <- RMSE(predicted_ratings, validation$rating)
RMSE_Final
```

```
## [1] 0.857548
```

This final residual mean squared error (**RMSE**) of **0.8575** shows that additional usage of the genre effect significantly improves the accuracy of the model.

4 Conclusion

In this **project**, we created a movie recommendation system using a large subset of the MovieLens dataset. Exploration of the *edx* dataset showed strong evidence of a genre effect and if used to augment the ***Regularized Movie + User Effect Model***[9] could yield improvements in the accuracy of the model. This hypothesis was validated when the final model, with the addition of an regularized predicted genre effect "*bias*" term, yielded a residual mean squared error (**RMSE**) of **0.8575** on the *validation* dataset.

Finally a potential limitation is the use of a common penalty λ for each of the effects, therefore further improvement in future work to improve the prediction accuracy would be to look at incorporating individual penalty terms, λ_i, λ_u and λ_g for each of the effects.

[9] https://rafalab.github.io/dsbook/large-datasets.html#recommendation-systems