Movie Ratings Predictor

Samuel Matsiko

12/13/2019

Introduction

A dataset is downloaded, tweaked and then split into edx and validation sets in proportions of 90% and 10% respectively. The resulting dataset comprises of 6 columns, 9000055 rows for the edx set and 999999 rows for the validation set.

While the validation set has been set aside for the final testing purposes, i have further split the edx set into train and test sets in proportions of 50% to 50% for intermediate training, testing and cross-validation purposes. With these sets,i have modelled a recommendation sytem that predicts movie ratings using two predictors(movieId and UserId). The train set has been trained using rating approximations to predict movie ratings of the test set and then compared against the actual ratings of the test set. The performance of the model has further been improved by regularization of the rating approximations. To obtain the final results, the same training approach has been adopted to train the whole edx set(not split this time round) before predicting the ratings of the validation set, that are then compared with the actual ratings of the validation set using a RMSE function.

Dimensions of the edx set

```
## [1] 9000055       6
```

Dimensions of the validation set

```
## [1] 999999      6
```

Columns of the dataset

```
## [1] "userId"    "movieId"   "rating"    "timestamp" "title"     "genres"
```

Methods

From the data overview below, it is observed that the dataset is tidy, that is to say; every variable is represented in its own column, every observation is represented in its own row and each value is stored in one cell. We also observe that there are no null values as observed in all the distinct values of rating.

Data overview

```
##   userId movieId rating  timestamp                                    title
## 1  54055    2542    5.0 1127126289 Lock, Stock & Two Smoking Barrels (1998)
## 2  44588     784    3.0  853942485                     Cable Guy, The (1996)
## 3  28957    2918    4.0 1004318839          Ferris Bueller's Day Off (1986)
## 4  67088     569    4.0  945403606                 Little Big League (1994)
## 5  48798    3371    4.0  976339222                   Bound for Glory (1976)
## 6  53123    3284    3.5 1076981907              They Might Be Giants (1971)
## 7  47019      74    5.0  842114515                     Bed of Roses (1996)
## 8  63967    7439    3.5 1111474870                    Punisher, The (2004)
## 9  57784    5464    4.0 1027126620                Road to Perdition (2002)
```

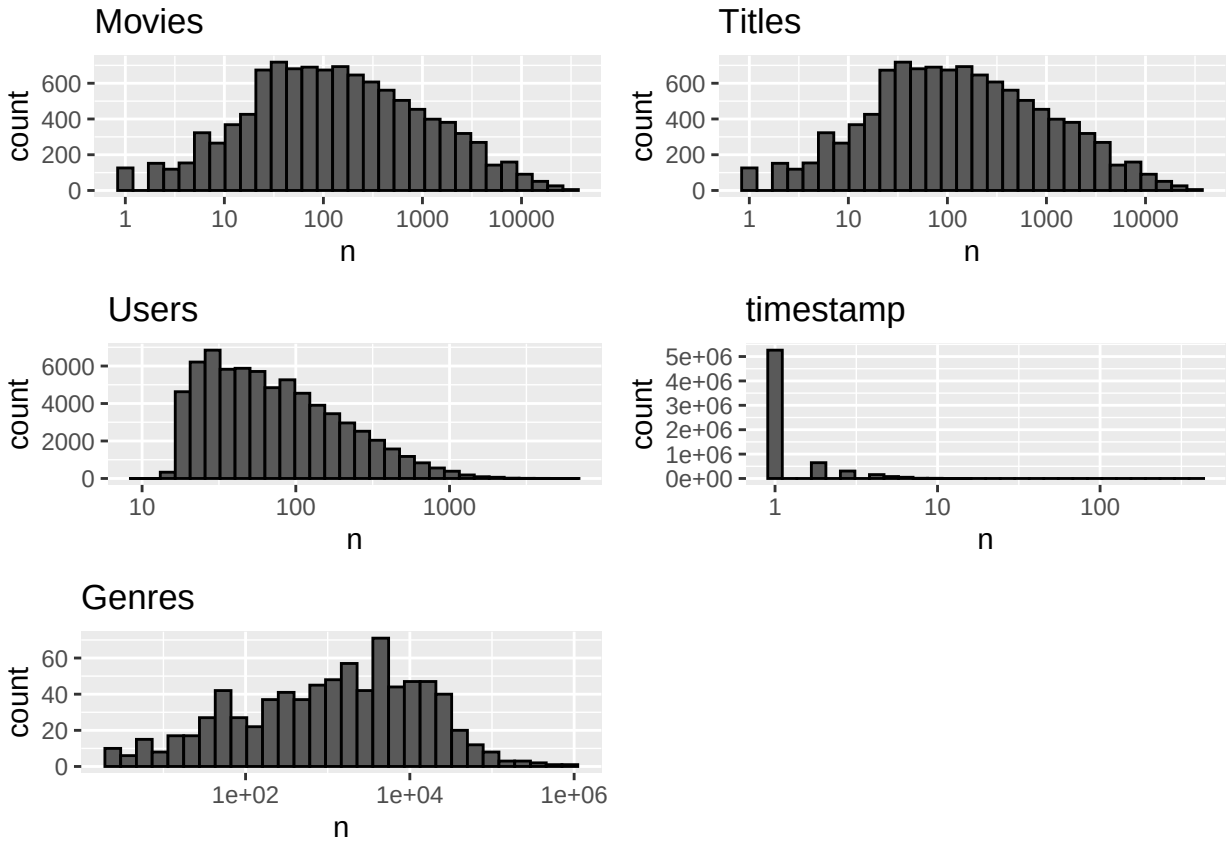
```
##                   genres
## 1 Comedy|Crime|Thriller
## 2       Comedy|Thriller
## 3                Comedy
## 4          Comedy|Drama
## 5                 Drama
## 6        Comedy|Romance
## 7         Drama|Romance
## 8 Action|Crime|Thriller
## 9           Crime|Drama
```

Distinct values of rating variable

```
##  [1] 5.0 3.0 2.0 4.0 4.5 3.5 1.0 1.5 2.5 0.5
```

The 5 variables(minus the target varibale, rating) of the dataset have been explored using the visualizations below to understand the biases that might be caused by each variable hence determining predictors.From the visualizations, we observe that graphs for movies and titles are indentical, hence highly correlated. I have therefore selected the movieId variable to cater for biases of both variables. As the graph for movies variable indicates that different movies are rated differently, i have taken into account the "movie effect" while modelling. Also important to note is the effect of users. We note that some users rate/watch more movies than their counterparts. The userId has therefore also been considered a predictor. While we also observe biases due to timestamp and genres variables, they aren't as significant as those for movieId and userId variables. Therfore, i have taken only movieId and userId as predictors.

```
## Loading required package: grid
```



I started off with a model that assumes same rating across all movies and all users, and any variations to be random. This approximation has been found by calculating the average rating across all movies and all users. Then the model has been improved by incoporating the "movie effect" and further improvements achieved by taking into account the "user effect" as well. The effects of both the movies and users(b_i and b_u respectively) have been determined by obtaining the mean of the residuals between the rating of the train set and the average rating across all movies and users for their respective movieIds and userIds. The resulting models have been used to predict test set ratings and compared with the actual ratings thereafter.

However, post this modelling, the model still made large estimates when a movie is rated by very few users. As seen in the modelling results below, it is observed that movies that have been rated very few times(indicated by n column) are estimated highly as indicated in both the top and bottom ten movies. To remedy the situation, I deployed regularization techniques to penalize the large estimates that come from very few users. An optimal penalty term(lambda) to balance out inaccurances due to very few ratings has been obtained by cross-validation. As seen in the modelling results post regulation(also below),this has not only awarded a better estimate accordingly, but also improved the performance of the model.

Top ten movies before regularization

```
## Joining, by = "movieId"
```

title	b_i	n
Heartbreak Hospital (2002)	1.487524	1
Act of Violence (1948)	1.487524	1
Satan's Tango (SÃ¡tÃ¡ntangÃ³) (1994)	1.487524	2
Fighting Elegy (Kenka erejii) (1966)	1.487524	1
Sun Alley (Sonnenallee) (1999)	1.487524	1
Class, The (Entre les Murs) (2008)	1.487524	1
More (1998)	1.362524	4
Bloody Child, The (1996)	1.287524	5
Who's Singin' Over There? (a.k.a. Who Sings Over There) (Ko to tamo peva) (1980)	1.237524	4
Star! (1968)	1.237524	2

Bottom 10 movies before regularization

```
## Joining, by = "movieId"
```

title	b_i	n
Way We Laughed, The (CosÃ¬ Ridevano) (1998)	-3.012476	1
State Property (2002)	-3.012476	1
Besotted (2001)	-3.012476	2
Manslaughter (1922)	-3.012476	1
Grief (1993)	-3.012476	1
Confessions of a Superhero (2007)	-3.012476	1
Illegal Tender (2007)	-2.762476	2
Disaster Movie (2008)	-2.726762	14
SuperBabies: Baby Geniuses 2 (2004)	-2.712476	20
From Justin to Kelly (2003)	-2.590146	103

Top ten movies after regularization

```
## Joining, by = "movieId"
```

title	b_i	n
Shawshank Redemption, The (1994)	0.9407106	13927
Godfather, The (1972)	0.9074301	8856
Usual Suspects, The (1995)	0.8543178	10807
Schindler's List (1993)	0.8492121	11637
Sunset Blvd. (a.k.a. Sunset Boulevard) (1950)	0.8383233	1461
Double Indemnity (1944)	0.8160513	1084
Third Man, The (1949)	0.8105582	1539
Rear Window (1954)	0.8042915	3923
Casablanca (1942)	0.7960299	5657
Seven Samurai (Shichinin no samurai) (1954)	0.7901492	2608

Bottom ten movies after regularization

```
## Joining, by = "movieId"
```

title	b_i	n
From Justin to Kelly (2003)	-2.475963	103
Gigli (2003)	-2.260817	144
PokÃ©mon Heroes (2003)	-2.245338	68
Barney's Great Adventure (1998)	-2.224534	94
SuperBabies: Baby Geniuses 2 (2004)	-2.191900	20
Pokemon 4 Ever (a.k.a. PokÃ©mon 4: The Movie) (2002)	-2.184900	108
Glitter (2001)	-2.176214	172
Turbo: A Power Rangers Movie (1997)	-2.129706	208
Ernest Goes to Africa (1997)	-2.074193	51
Carnosaur 3: Primal Species (1996)	-2.045407	39

Results

The performance metric that has been used to guage the performance of the model is the Root Mean Square Estimate(RMSE). The performance of the first model(average rating) returned an RMSE of 1.0600456. The second model (incoporating the movie effect) returned an RMSE of 0.9439980, the third model (incoporating both movie and user efects) returned an RMSE of 0.8695042 and the forth model(after regularization) returned an RMSE 0.8679124. The final model with the validation set returned an RMSE of 0.8648295.

Conclusion

With the provided dataset, i have been able to build a recommendation system that predicts a user's movie rating using userId and movieId predictors. Some of the limitations rotate around predictions with new users or new movies since no data would be available to learn from. Another limitation is that the model did not take into account the effects by timestamp and genres variables.

My future work is to incoporate timestamp and genres variables to improve the model's preformance further.