Movie Recommendation Model

Overview

The purpose of this machine learning project was to analyze a collection of move ratings provided by the MovieLens package (https://movielens.org/ (https://movielens.org/)) and design a recommender model to accurately predict a user's rating of a given movie. A 10 million observation MovieLens dataset was selected and split into a train and test dataset. Final model was then executed against a provided validation dataset to determine a root mean square error rating (RMSE). Generally, if the RMSE error is greater than 1, then the model prediction will be off by more than one star. The rating is on a 5 star scale.

Analysis

The final model incorporated the following effects or biases as attributes in the overall model:

Simple average

The model begins with a simple average of all movies for all users in the dataset and adjusts from there.

Movie Effect

Some movies are rated differently based on its content, genre, etc. This effect recognizes the individuality of the movie and calculates a least square estimate using the individual movie's (moveid from the movielens dataset) average.

User Effect

Similarly, individual users have tendencies, maybe an attitude towards one movie or a whole movie type. The model adds a least square estimate using each userid's average rating. The idea here is to categorize users based on their judgements. Do they tend to be critical with lower star ratings or lenient with higher star ratings across the board?

Regularized Move + User Effect

Realizing the model is not accounting for variability due to the number of occurrences of both movie and user effects, we use a penalized least squares Regularization adjustment. Essentially, the model introduces a 'penalty' which tunes out movie ratings with very small populations of ratings. (i.e. a movie with one user rating should not indicate the same reliability of a movie with 1000 ratings) As the dataset occurrences of user ratings for a given movie increases, the penalty reduces. To optimize such a regularization of data, a tuning parameter, lamda, is calculated. Cross-validation is used to select the point where lamda provides the minimum RMSE.

Plot of Lambda vs RMSE

```
## Loading required package: tidyverse
```

```
## -- Attaching packages ------------------------------------------------------------
---- tidyverse 1.2.1 --
```

```
## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0
```

```
## -- Conflicts -------------------------------------------------------------------- t
idyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

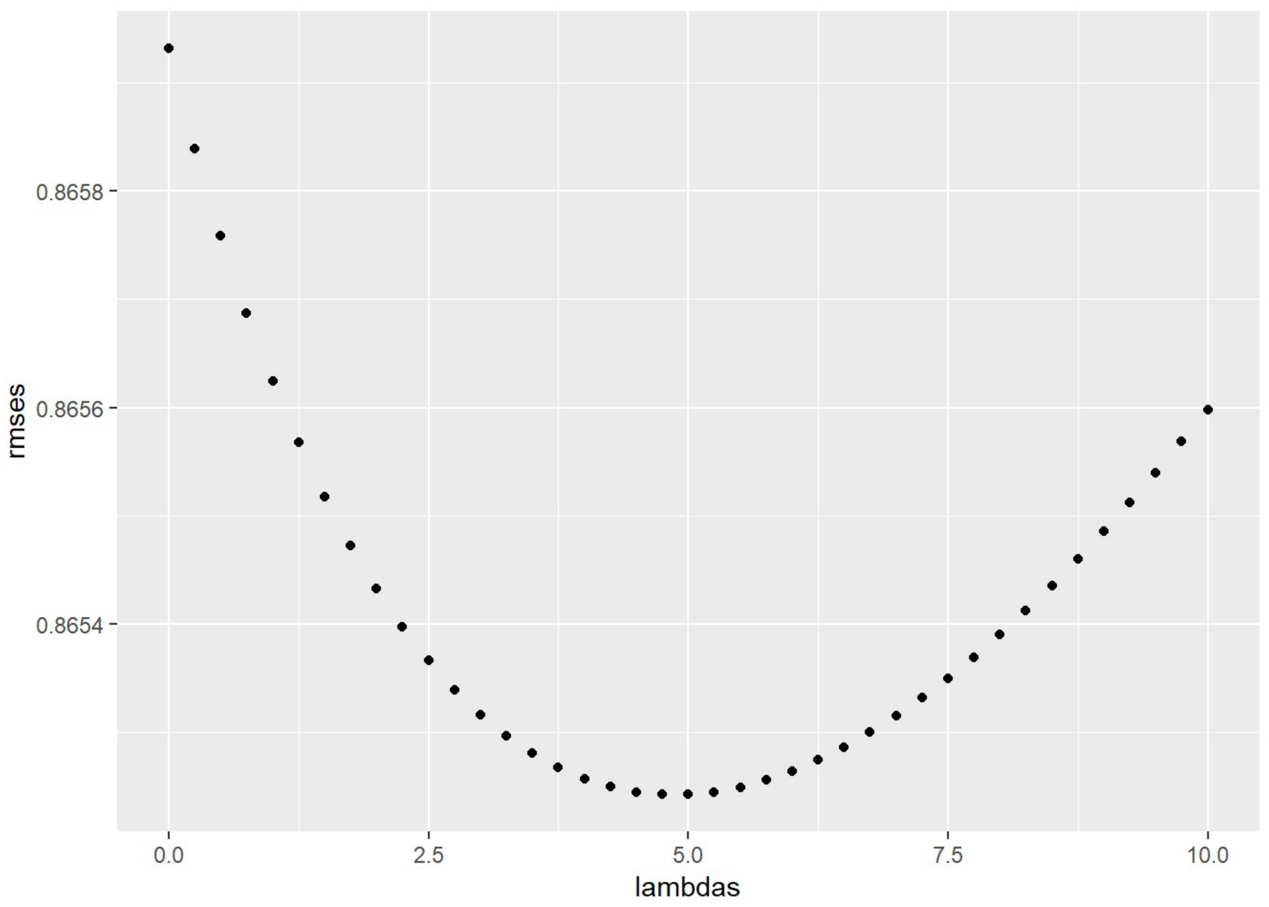
```
## Loading required package: caret
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
## Joining, by = c("userId", "movieId", "rating", "timestamp", "title", "genres")
```



```
## [1] 4.75
```

Validation

method	RMSE
Just the average	1.0599043
Movie Effect Model	0.9437429
Movie + User Effects Model	0.8659320
Regularized Movie Effect Model	0.9436762
Regularized Movie + User Effect Model	0.8652421
Regularized Movie + User Effect Model Validation Run	0.8656928

Conclusion

The RMSE score for the validation run is:

```
## [1] 0.8656928
```

The model’s best result was using regularization with a movie and user effect model. Regularization provided a small decrease in RMSE. The largest decrease in RMSE occurred when taking into account the user effect in combination with the movie effect. In observing remaining data columns to look for other data attributes to consider in the model, we know no more on a user. We do know more on the movies: their genre. Any deeper analysis to further improve/lower RMSE should look to include a genre effect.