Report on Movilens Project

Asham Vohra

6/16/2021

Overview

Recommendation systems have been there like forever and over time they have been incorporated by almost every business application. They are common in video steaming platforms like Netlix, Amazon Prime, restaurants' reviews and delivery applications like Yelp, Uber eats and many more. Based on application, these systems recommend a movie or a restaurant and leverage the past interaction of customer with the application or how similar users may be watching or ordering food, among others.

Taking inspiration from the recommendation systems available all around us, we have attempted to build a movie recommendation system. The recommendation system leverages the publically available MovieLens dataset.

The MovieLens dataset used here is hosted at GroupLens.org. The data set is stable ensuring results are reproducible and has **10 million** ratings applied to **10,000** movies by **72,000** users. One of the components of the data set includes information about movies and its associated genres. While the other component has details of how and movies were rated by users.

The movie recommendation system developed as part of the project, looks into leveraging patterns identified across movies, users, genres and more and uses the available data set to predict movie ratings for given observation, which includes user and movie identifiers, movie title and genres.

The goal of the project was to build a machine learning recommendation system which can predict movie ratings keeping RMSE(Root mean square error) minimal.

In order to achieve this goal, we took incremental approach of identifying predictors, which can predict the rating. Incrementally, we took a predictor analysed its relationship with the rating, incorporated in our model and tested our updated model against the test data. To ensure the behavior and model performance was not due to random choice of data set, the model was tuned, trained and tested against multiple partitions of the dataset. Only if the predictor helped improve our metric i.e. RMSE, the predictor under analysis and evaluation was added to the model and the steps were repeated with new predictor. Otherwise the predictor was ignored.

This report walks through the approach, analysis and evaluation carried out to achieve our movie recommendation system.

Analysis

Data Wrangling

A critical part for using any publically available dataset is converting it to usable form. So we carried out below steps to convert it to usable form:

- started from extracting data downloaded from source,
- the data files received had different delimiters separating various attributes of an observation. The same were handled, read and converted into a data frame.
- Some of the attributes were not of desired data type. The concerned attributes were converted from undesired data types like strings or factors to numeric data as required.

This is how snippet of the wrangled data looked like.

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	231	5	838983392	Dumb & Dumber (1994)	Comedy
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

Paritioning dataset for training and validation

Before using the wrangled data for analysis, we partitioned the data into training and validation data sets. The idea was to keep validation data set aside and use only training data for any analysis, model development and tuning. In addition to this, we ensured that our validation set only had data for same movies and users which were present in the training data set.

The data sets were named as below for furture reference:

- training data set as **edx**
- validation data set as **validation**

Below were the details about the elements in each of the partitioned datasets.

data_set	name	count
Training data set	edx	9000055
Validation data set	validation	999999

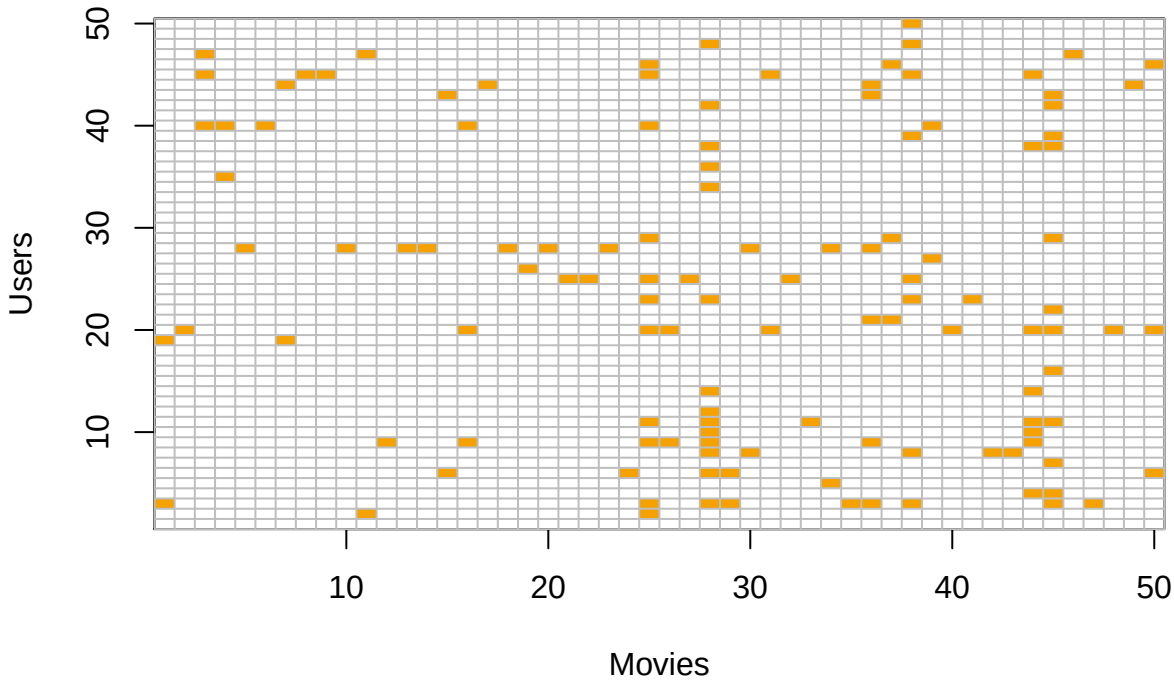
For analysis, we used only the training dataset.

Below was the summary of number of unique movies and users in the training data set.

users_count	movies_count
69878	10677

Since total_ratings in training dataset were **9000055**, the average number of movies rated per user were **128.8**. This means that average user rated only a subset of movies.

To visualise how sparse our rating dataset was, we randomly selected 50 users and 50 movies with atleast one rating made by the selected user group and plotted a graph.



Note: Here the white space represents that for a given user, rating for a specific movie is missing and needs prediction.

It it these missing ratings, our recommendation system attempted to predict once ready while maintaining minimal RMSE.

So to build a recommendation system which could predict the missing ratings, we used our intuition as guidance to identify few areas to analyse and to dive into for preparing our model. The same are shared below:

- use knowledge of average rating across all movies
- use knowledge of ratings for a movie
- use knowledge of how movies similar to movie in question are rated
- use knowlege of how movies in different genres are rated
- use knowledge of how a user rates movies,
- use knowledge of how users similar to user under question rate a movie
- use knowledge of year of movie release

To start building our algorithm, we further paritioned our training data set i.e. edx into five sets each having a training set and testing set. This ensured that we used only edx dataset while we developed our model. In addition, the multiple partitions helped to remove any randomness from our tuning parameters and to get an appopriate estimate of performance for our incremental model. This helped reduced the probability of overfitting.

Before diving into analysis and modeling carried out, it is important to understand the metric of choice i.e. RMSE

- RMSE is the root mean square error and that's what was used for this project to evaluate models and to optimize them for fitment.
- As we know, $RMSE = sqrt(1/N * sum((\hat{y} - y)^2))$
  - Here, $\hat{y}$ (to be referred as y_hat) is the prediction. In context of this project, predicted rating for movie i and user u,
  - and y is the actual outcome. In context of this project, actual rating for movie i and user u

Initial Model

To begin with we took brute force approach i.e. all movies are rated same but their rating vary randomly only by variation explained by

$y_{u,i} = true\_rating + error_{u,i}$

where $y_{u,i}$ is the rating of movie i for user u, true_rating is the true rating across users and movies, and $error_{u,i}$ is the independent error.

RMSE is minimized when we minimize least square error and this was achieved by using true_rating as average of all ratings

```
## Total mean: 3.51254074454833
```

Using this as a model, the performance results were found as below:

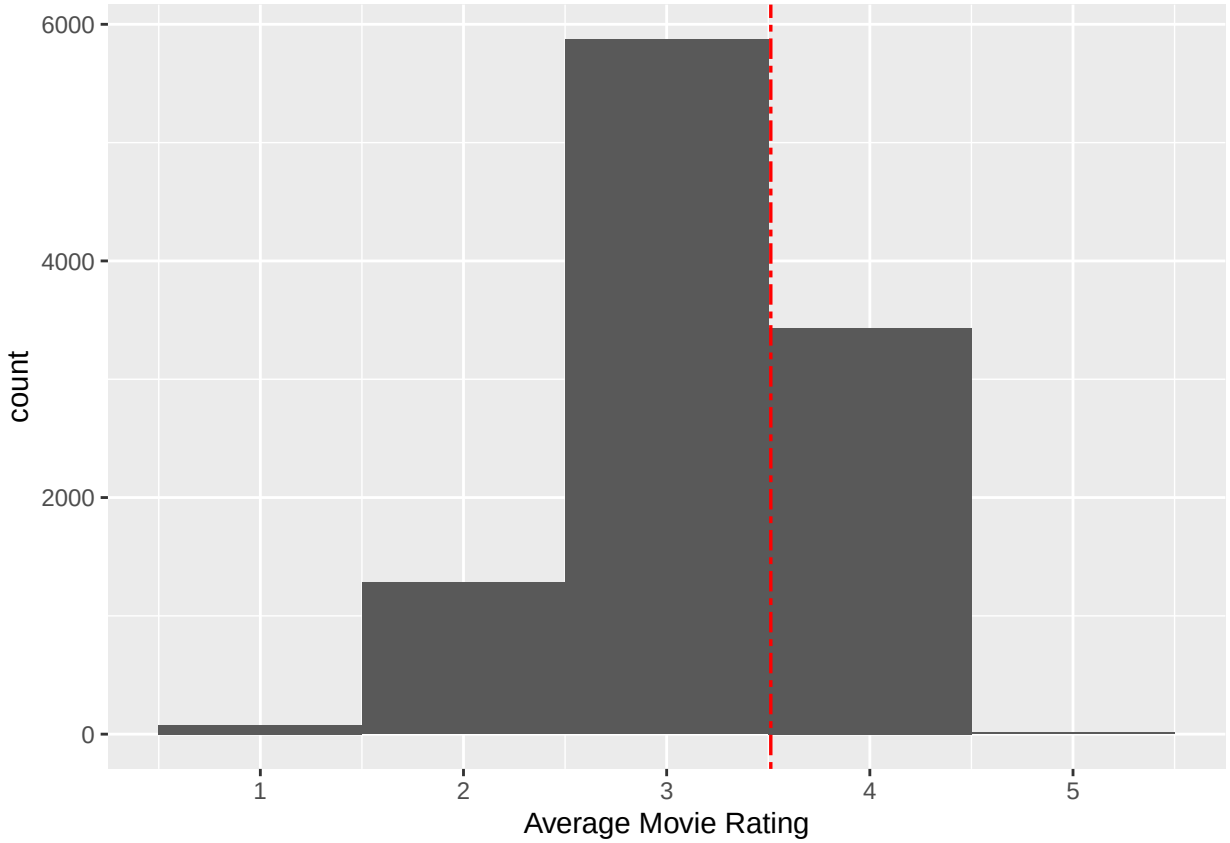
model_name	rmse
Brute force model	1.060523

This was a good starting point and the RMSE value captrued here was used to analyse effectivness of any new predictor.

Movie effects

As one may note that the brute force model depended on mean of all ratings present in the data set. However, based on the intution that the movies may be rated differently, we further explored possibility of information of movie influencing the prediction results.

Below histogram depicts how mean rating of movie across movies is distributed.



Note: The red dotted line depicts the average rating across all movies and users.

In above histogram, we noted that for lot of movies average rating was higher than the average rating across all movies and users. And for some it was lower. So our intution that different movies are rated differently was confirmed.

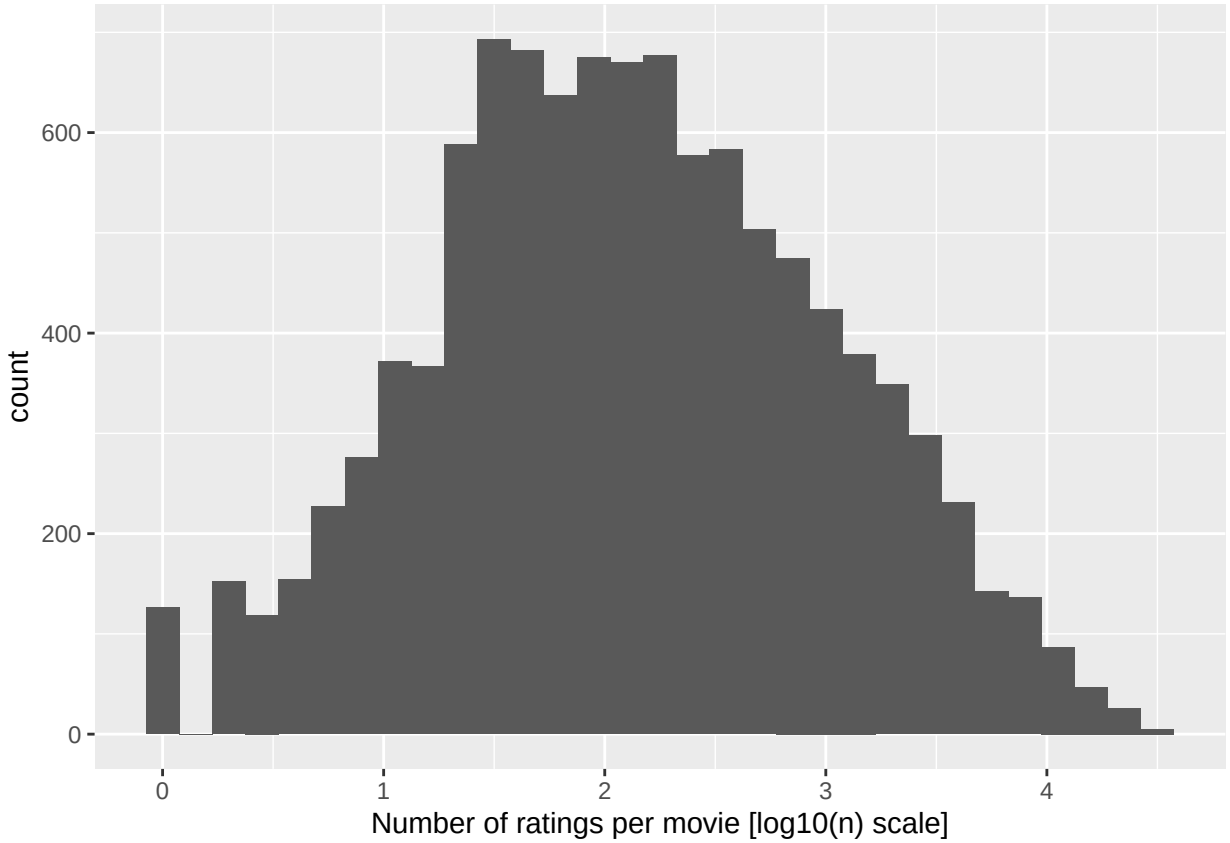
The variability across movies looked good enough to be analysed for inclusion in our model. Movie effect for analysis when incorporated in our model resulted to the below form:

$y_{u,i} = true\_rating + b_i + error_{u,i}$

where $y_{u,i}$ is the rating of movie i for user u, true_rating is the true rating across users and movies, b_i is movie effect for movie i, and $error_{u,i}$ is the independent error.

As result, estimate of b_i for movie i i.e $\hat{b_i} = mean(y_{u,i} - true\_rating)$

Before estimating movie effects for our model, we checked how the number of ratings per movie were distributed across movies and if it could be a concern.



The graph suggests that many movies have been rated very less number of times. This could be due to some movies being being obscure or newly released, among others. Regardless of the reason, with such a low set of data points, there could be errors in estimating role of movie effect in the prediction of the rating i.e basically the risk of overfitting. So any movie effect needed to be penalized for less number of ratings and we used regularization to prevent our model from overfitting.

To identify regularization factor i.e. lambda(λ) to use, we took a range of values and evaluated model to identify λ which led us to the lowest RMSE. This was done across partitions of training data to ensure, the choice of λ was not due to random nature of the dataset.

With λ our estimate of b_i for movie i could be computed as below:

$\hat{b_i} = sum(y_{u,i} - true\_rating)/(n + \lambda)$

where n is the number of ratings for the given movie.

So resulting model post regularization of the form $y_{u,i} = true\_rating + b_i + error_{u,i}$, resulted in the RMSE of 0.9436834

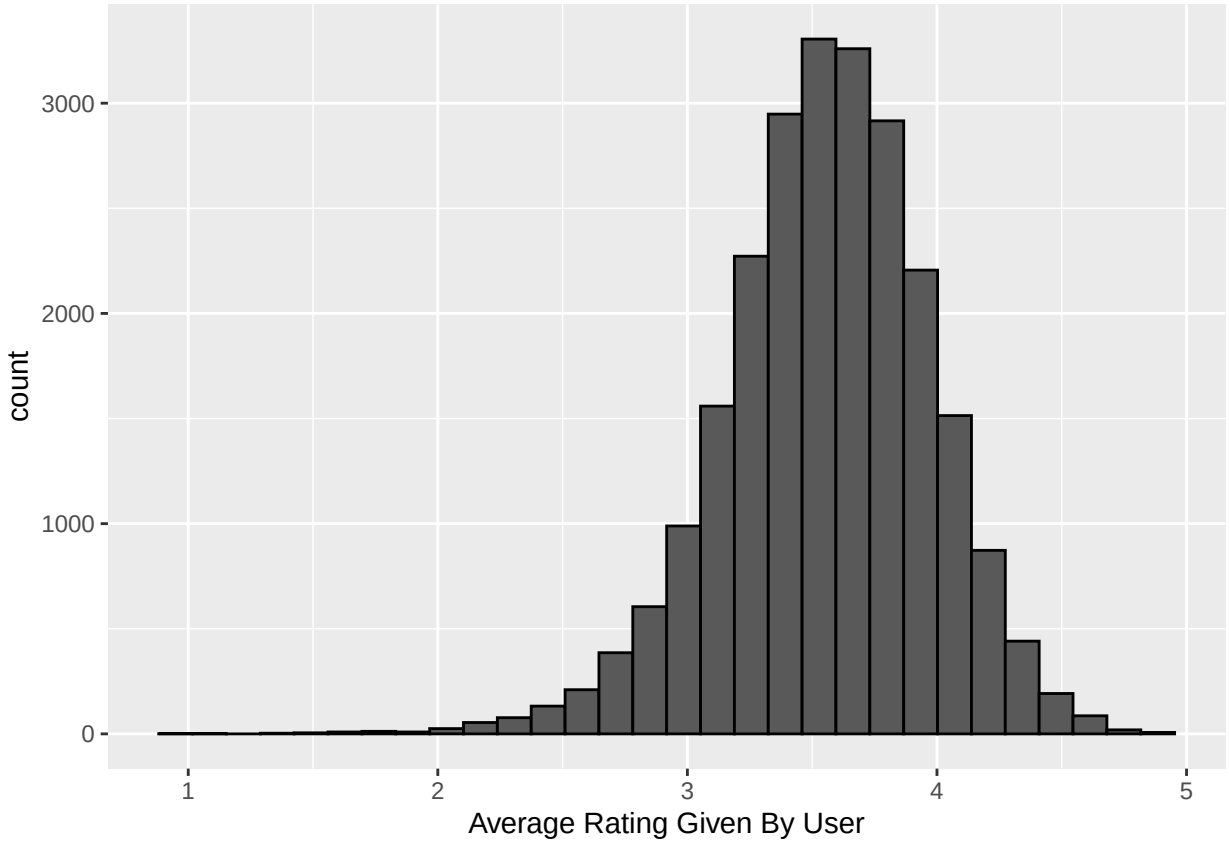
model_name	rmse
Brute force model	1.0605234
With only movie effect model	0.9436834

As we can see the evaluation metric i.e. RMSE improved considerably and therefore the movie effect was retained in our model.

User effects

Now with model taking into account movie effect, we decided to look into other intution i.e. different users rate movies differently and to evaluate its influence on the prediction results.

Below histogram depicts how average rating given by users is distributed. For this, we looked at only those users who had rated more than 100 movies.



In above histogram, we noted that average rating given by user varies across users and it is likely that a user who is critical or due to different preferences may rate even highly rated movies differently than other users. So our intution that different users rate movies differently was confirmed.

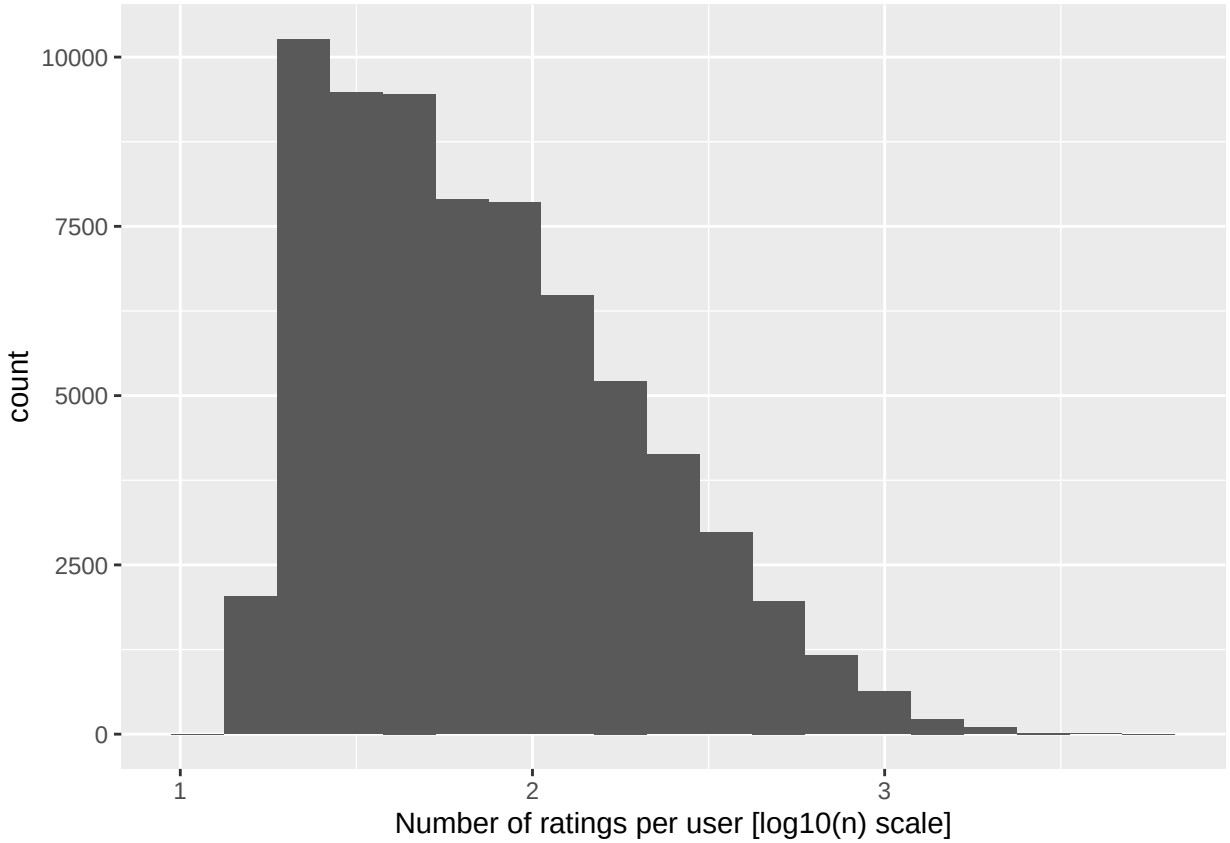
The variability across mean rating given by users deserved analaysis for inclusion in our model. User effect for analysis when incorporated in our model resulted to the below form:

$y_{u,i} = true\_rating + b_i + b_u + error_{u,i}$

where $y_{u,i}$ is the rating of movie i for user u, true_rating is the true rating across users and movies, b_i is movie effect for movie i, b_u is user effect for user u, and $error_{u,i}$ is the independent error.

As a result, estimate of b_u for user u i.e $\hat{b_u} = mean(y_{u,i} - true\_rating - \hat{b_i})$

Before estimating user effects for our model, we checked how the number of ratings per user were distributed across users and if it could be a concern.



The graph suggests that many users have rated very less number of movies. Some of the reasons for this could be that some users rate a lot, while some don't or it could be that some users were new to the movies streaming platform, among others. Regardless of the underlying reason, with such a low set of data points, there could be errors in estimating role of user effect in the prediction of the rating i.e basically the risk of overfitting. So any user effect needed to be penalized for less number of ratings made by a user and we used regularization to prevent our model from overfitting.

Similar to movie effect, to identify regularization factor i.e. lambda(λ) to use, we took a range of values and evaluated model to identify λ which led us to the lowest RMSE. This was done across partitions of training data to ensure, the choice of λ was not due to random nature of the dataset.

With λ our estimate of b_u for user u could be computed as below:

$\hat{b_u} = sum(y_{u,i} - true\_rating - \hat{b_i})/(n + \lambda)$

where n is the number of ratings for the given movie.

So resulting model post regularization of the form $y_{u,i} = true\_rating + b_i + b_u + error_{u,i}$, resulted in the RMSE of 0.8645158

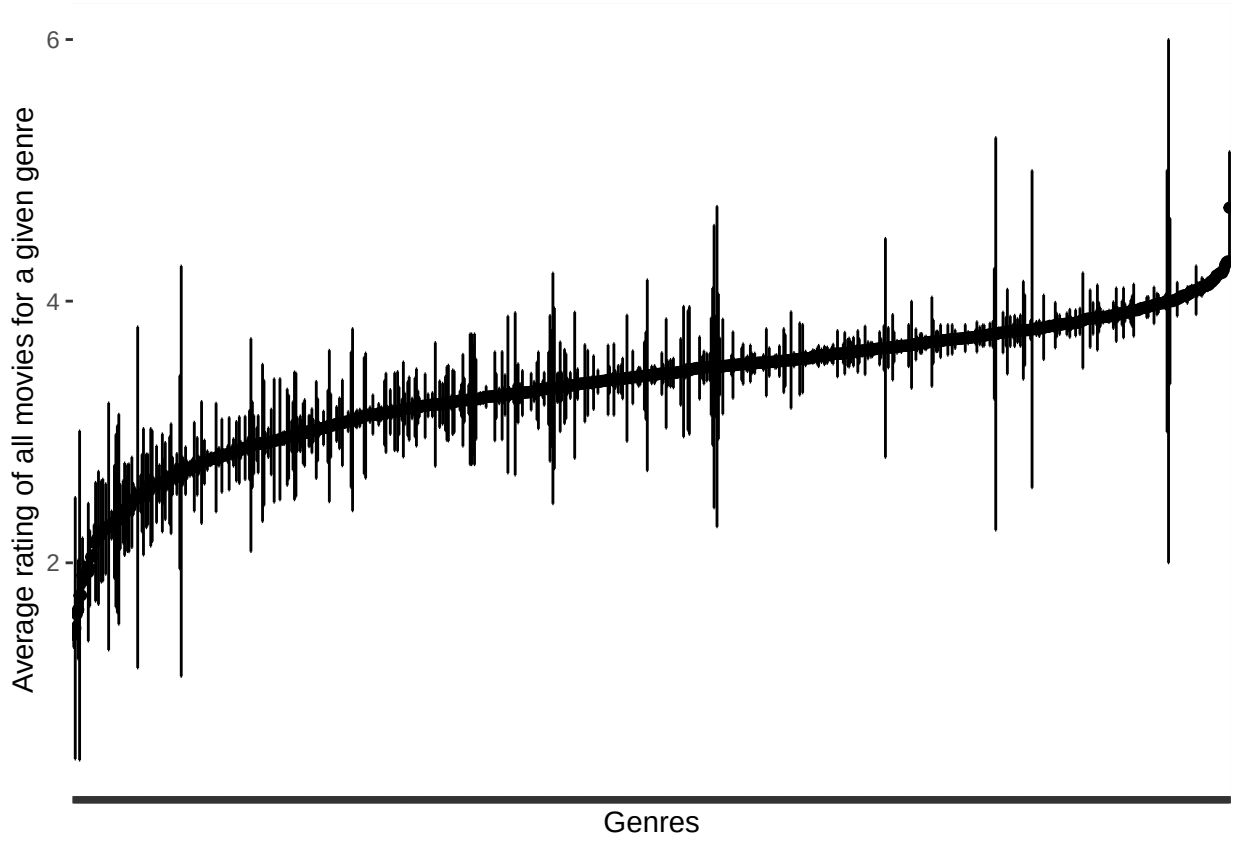
model_name	rmse
Brute force model	1.0605234
With only movie effect model	0.9436834
With movie and user effect model	0.8645158

As we can see the evaluation metric i.e. RMSE improved considerably and therefore the user effect was retained in our model.

Genres effect

Now with model already coming in shape, we decided to look into other intution i.e. movies within a genre are rated differently and evaluate its influence on the prediction results.

Below errorbar plot depicts how mean rating of a genre vary across genres while at the same time capturing the confidence interval of rating for each genre.



In above plot, we noted that geners are rated differently contributing to variability in average rating of genres. One likely reason is that users rate movies belonging to one genres differently than movies belonging to other genres due to their preference. So our intution that movies in different genres are rated differently was confirmed.

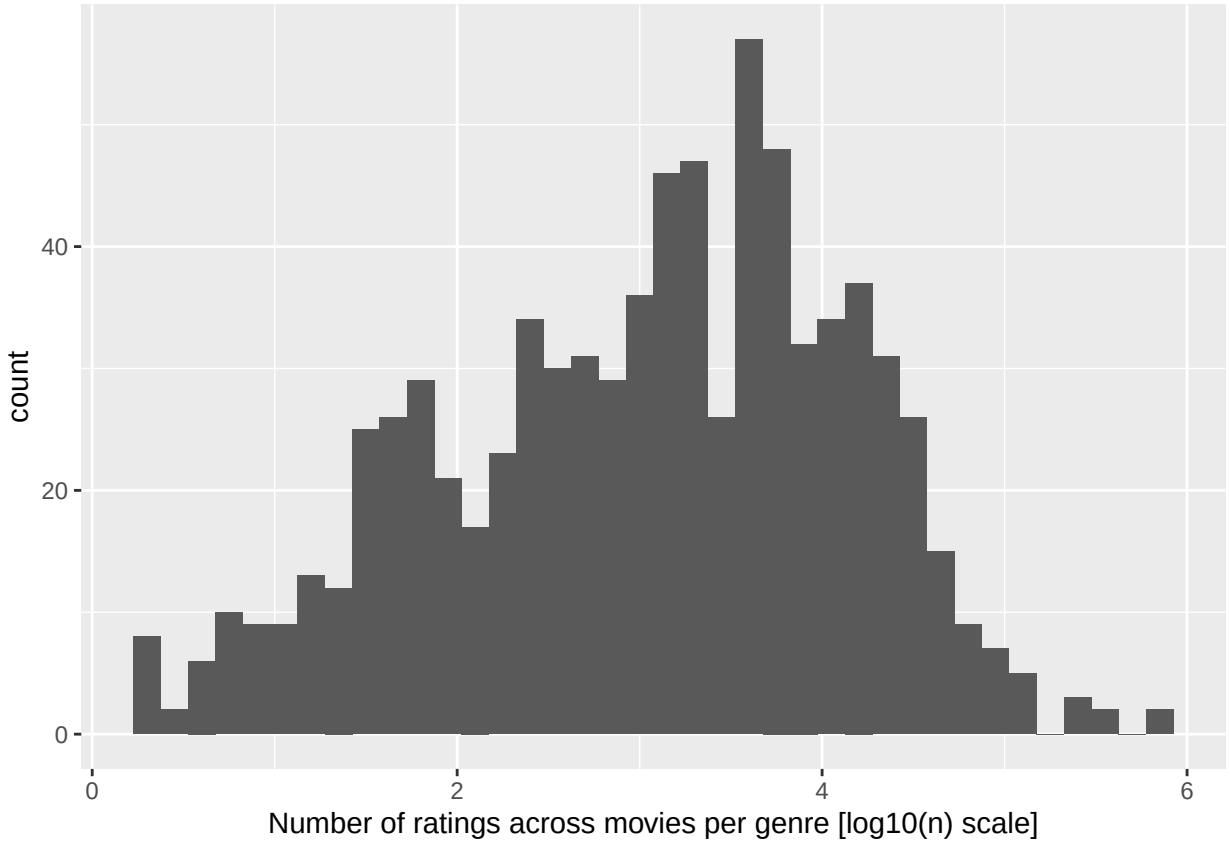
The variability of mean rating across genres could influence our final prediction and as a result the genres effect was considered for further analaysis for inclusion in the model. Genre effect for analysis when incorporated in the current model resulted to the below form:

$y_{u,i} = true\_rating + b_i + b_u + b_g + error_{u,i}$

where $y_{u,i}$ is the rating of movie i for user u, true_rating is the true rating across users and movies, b_i is movie effect for movie i, b_u is user effect for user u, b_g is the genre effect for genre g and $error_{u,i}$ is the independent error.

As a result, estimate of b_g for each genre i.e $\hat{b_g} = mean(y_{u,i} - true\_rating - \hat{b_i} - \hat{b_u})$

Before estimating genres effects for our model, we checked how the number of ratings per genres were distributed across genres and if it could be a concern.



The graph suggests that many genres have very less number of ratings as the movies belonging to the genres were rated insufficient number of times. And with such a low set of data points, there could be errors in estimating role of genres effect to the prediction of the rating i.e basically the risk of overfitting. So any genres effect needed to be penalized for less number of ratings and we used regularization to prevent our model from overfitting.

Similar to other effects discussed, to identify regularization factor i.e. lambda(λ) to use, we took a range of values and evaluated model to identify λ which led us to the lowest RMSE.

With λ our estimate of b_g for each genre could be computed as below:

$\hat{b_g} = sum(y_{u,i} - true\_rating - \hat{b_i} - \hat{b_u})/(n + \lambda)$

where n is the number of ratings across movies associated with the given genre

So resulting model post regularization of the form $y_{u,i} = true\_rating + b_i + b_u + b_g + error_{u,i}$, resulted in the RMSE of 0.8585168

model_name	rmse
Brute force model	1.0605234
With only movie effect model	0.9436834
With movie and user effect model	0.8645158
With movie, user and genre effect model	0.8585168

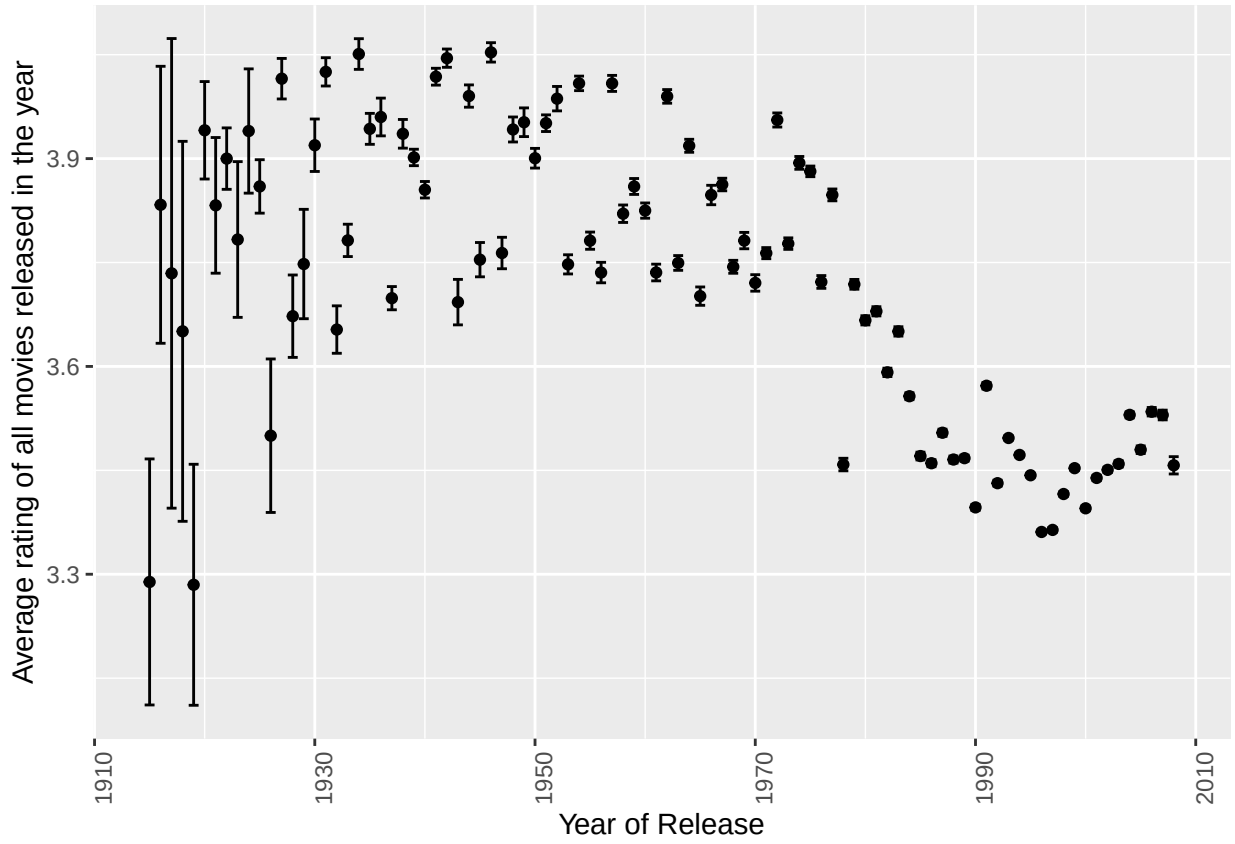
As we can see the evaluation metric i.e. RMSE improved and therefore the genre effect was retained in our model.

Year of release effect

Now with model already coming in shape, we decided to look into other intution i.e. year of release of movie influences the rating of the movie and to evaluate its influence on the prediction results.

For this, we extracted the year of release of movie from the movie title attribute available in the dataset and later we used this information to compute mean rating of all movies released in that year.

Below errorbar plot depicts how mean rating of movies released in a year varies as year of release increases. while at the same time capturing the confidence interval of rating for each year of release



In above plot, one can see that recently released movies tend to have less mean rating than earlier released movies. This confirmed our intuition that year of release of movie influenced the rating of a movie.

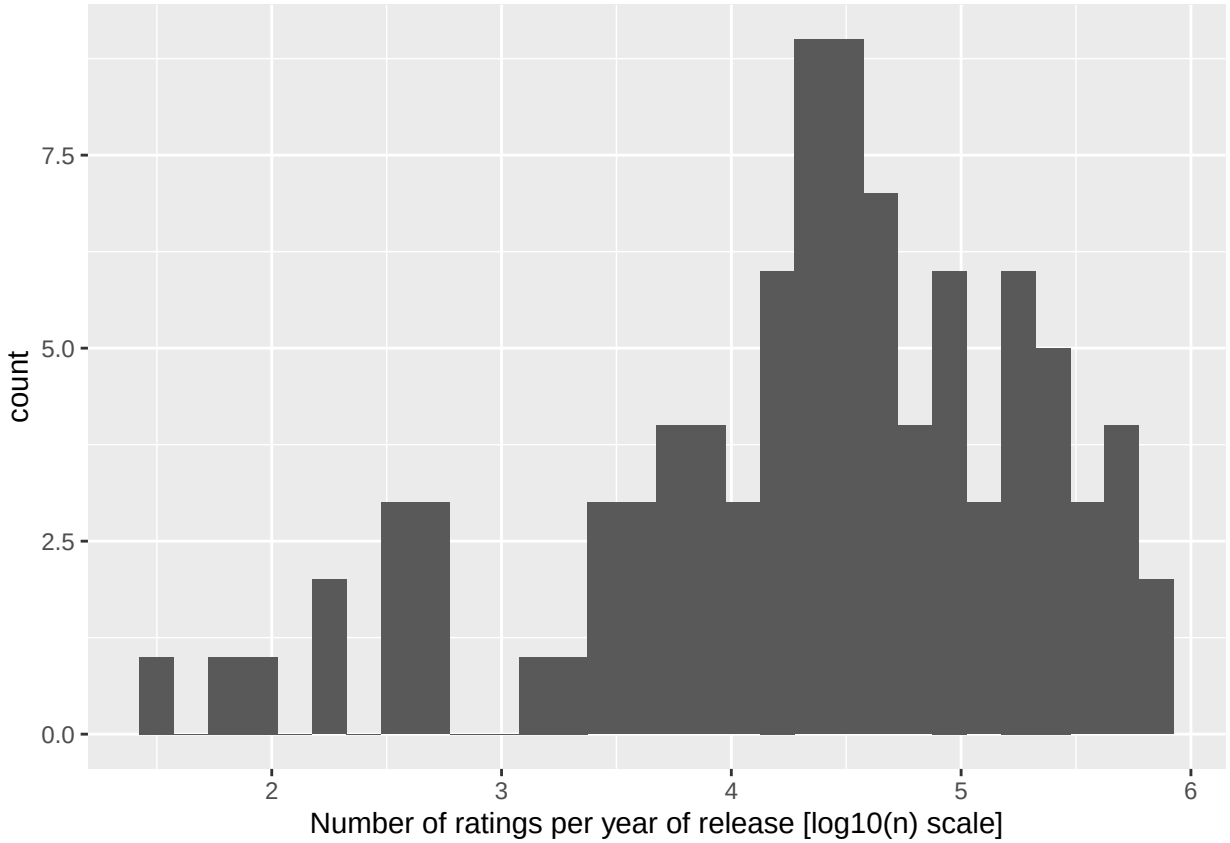
The variability of mean rating across years of release could influence our final prediction and as a result the year of release effect was considered for further analysis for inclusion in the model. Year or release effect for analysis when incorporated in the current model resulted to the below form:

$y_{u,i} = true\_rating + b_i + b_u + b_g + b_y + error_{u,i}$

where $y_{u,i}$ is the rating of movie i for user u, true_rating is the true rating across users and movies, b_i is movie effect for movie i, b_u is user effect for user u, b_g is the genre effect for genre g and b_y is the year of release effect for year y and $error_{u,i}$ is the independent error.

As a result, estimate of b_y for each year of release i.e $\hat{b_y} = mean(y_{u,i} - true\_rating - \hat{b_i} - \hat{b_u} - \hat{b_g})$

Before estimating year of release effects for our model, we checked how the number of ratings per year of release were distributed across years and if it could be a concern.



The graph suggests that many years of release have very less number of ratings as the movies were rated very less number of times. Few reasons for this could be ratings for older movies were not captured or the recently released movies may have less number of ratings or number of movies released in year were few. Regardless of the underlying reason, with such a low set of data points, there could be errors in estimating role of years of release effect to the prediction of the rating i.e basically the risk of overfitting. So any years of release effect needed to be penalized for less number of ratings and we used regularization to prevent our model from overfitting.

Similar to other effects discussed, to identify regularization factor i.e. lambda(λ) to use, we took a range of values and evaluated model to identify λ which led us to the lowest RMSE.

With λ our estimate of b_y for each year of release can be computed as below:

$\hat{b_y} = sum(y_{u,i} - true\_rating - \hat{b_i} - \hat{b_u} - \hat{b_g})/(n + \lambda)$

where n is the number of ratings across all movies for the given year of release

So resulting model post regularization of the form $y_{u,i} = true\_rating + b_i + b_u + b_g + b_y + error_{u,i}$, resulted in the RMSE of 0.8583126

model_name	rmse
Brute force model	1.0605234
With only movie effect model	0.9436834
With movie and user effect model	0.8645158
With movie, user and genre effect model	0.8585168
With movie, user, genre and release year effect model	0.8583126

As we can see the evaluation metric i.e. RMSE improved and therefore the year of release effect was retained in our model.

Results

Post evaluation with multiple predictors, the final model resulted to the below form:

$y_{u,i} = true\_rating + b_i + b_u + b_g + b_y + error_{u,i}$

where $y_{u,i}$ is the rating of movie i for user u, true_rating is the true rating across users and movies, b_i is movie effect for movie i, b_u is user effect for user u, b_g is the genre effect for genre g and b_y is the year of release effect for year y and $error_{u,i}$ is the independent error.

Till this point, we had used only training data set i.e. edx to develop the model and tune it. However, since the model was now identified and regularized, we trained the final model over the entire training data set i.e. edx and then tested it against the validation data set i.e. validation.

Below is the summary of each incremental model and our final model against the performance metric i.e. RMSE

model_name	rmse
Brute force model	1.0605234
With only movie effect model	0.9436834
With movie and user effect model	0.8645158
With movie, user and genre effect model	0.8585168
With movie, user, genre and release year effect model	0.8583126
Final model	0.8643264

For the **final model**, the performance metric i.e. RMSE = **0.8643264** captured against **validation** dataset was inline with the training data set and it was within desired range for model to be useful for predicting rating of a movie for a given user.

Conclusion

As part of this project, the goal was to build a movie recommendation system which could predict the rating for a movie and a user with minimal RMSE.

During the course of project, we evaluated various predictors backed by intution and data, and applied multiple techniques to reach to a model which performs within desired range of RMSE and can be useful for predicting rating of a movie for a given user. Due care was taken to prevent overfitting from influencing our model.

While we did take this approach given the hardware constraints, however, given more resources we could explore standardised supervised learning algorithms like k nearest neighbours or random forests among others or build incremental learning on top of the model as the recommendation systems will have to deal with new users and new movie releases and eventually learn from them and predict for those cases.

Having said that the movie recommendation system built as part of the project is performant and can be leveraged to develop further.