MovieLens Project

Introduction

The goal of this this project is to build a Predictor Engine (a recommender system) using the MovieLens data set in R Studio to try to predict what movies particular movie users would prefer, and ultimately receive a personalized movie recommendation. The version of movielens included in the dslabs package is just a small subset of a much larger dataset with millions of ratings. The dataset used will be the 10M version of the MovieLens dataset.

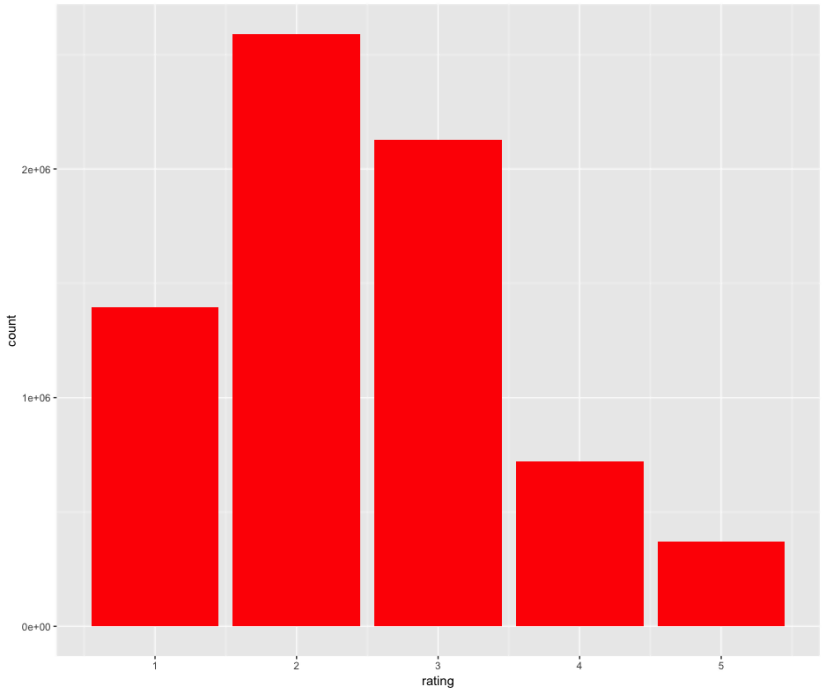
Recommendation systems use ratings that users have given on movies in order to make specific recommendations to their liking. Companies nowadays are building smart and intelligent Recommendation Systems in many different fields, applications, products, and services, by studying the past behaviour of their users. Data is used to make recommendations and choices relating to their interest in terms of “Relevant Job postings”, “Movies of Interest”, “Suggested Videos”, “Facebook friends that you may know”, “People who bought this also bought this”, so on and so forth. This project focuses on movies and was fun to work with.

The dataset used for modelling for the recommendation system includes 69,878 users and 10,677 movies.

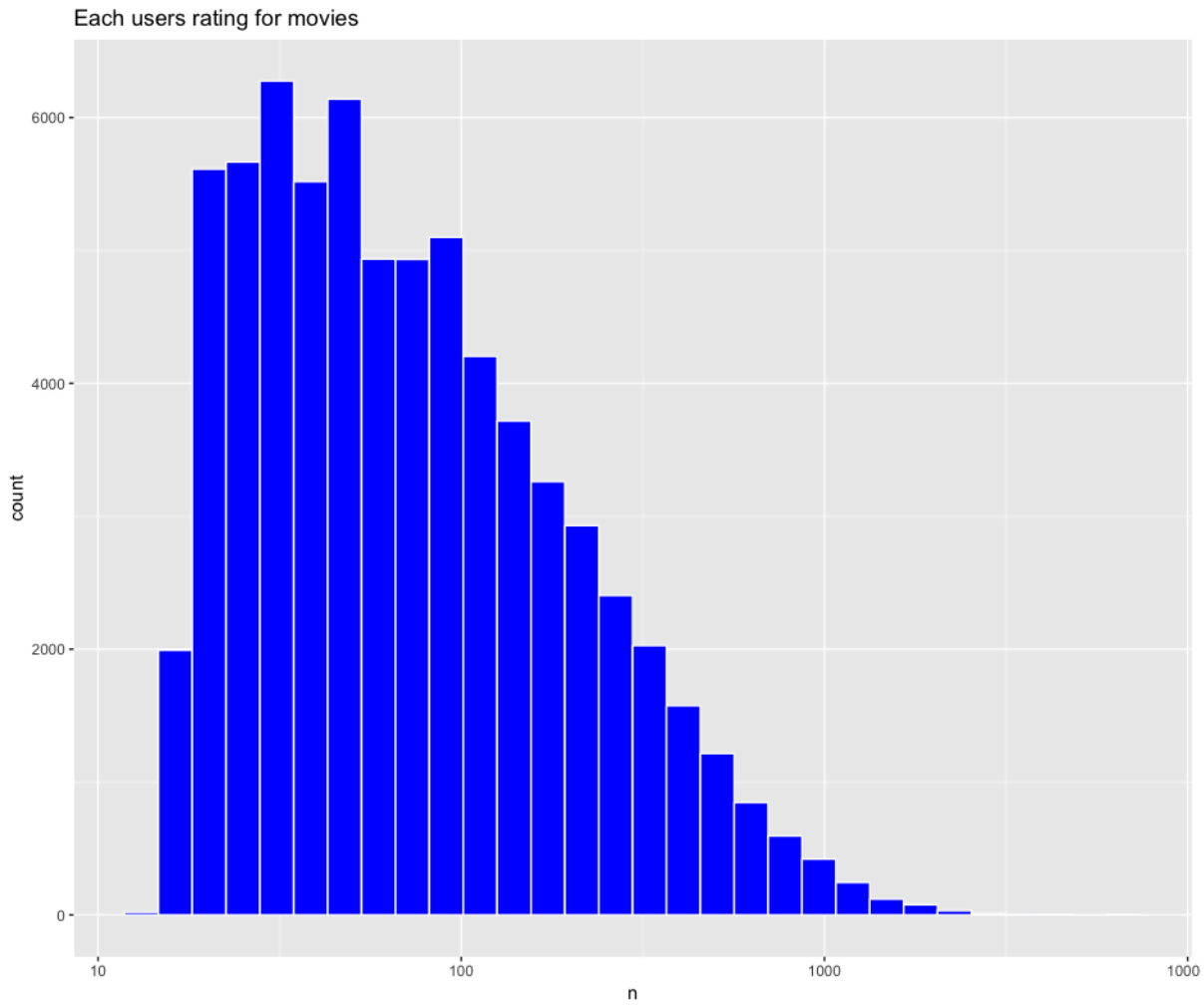
Key steps that were performed are Data Exploration, Visualization, evaluating the model’s performance using the Root Mean Square Error (RMSE) value, then using the Regularization-Based approach to improve the RMSE value to 0.8624.

Methods

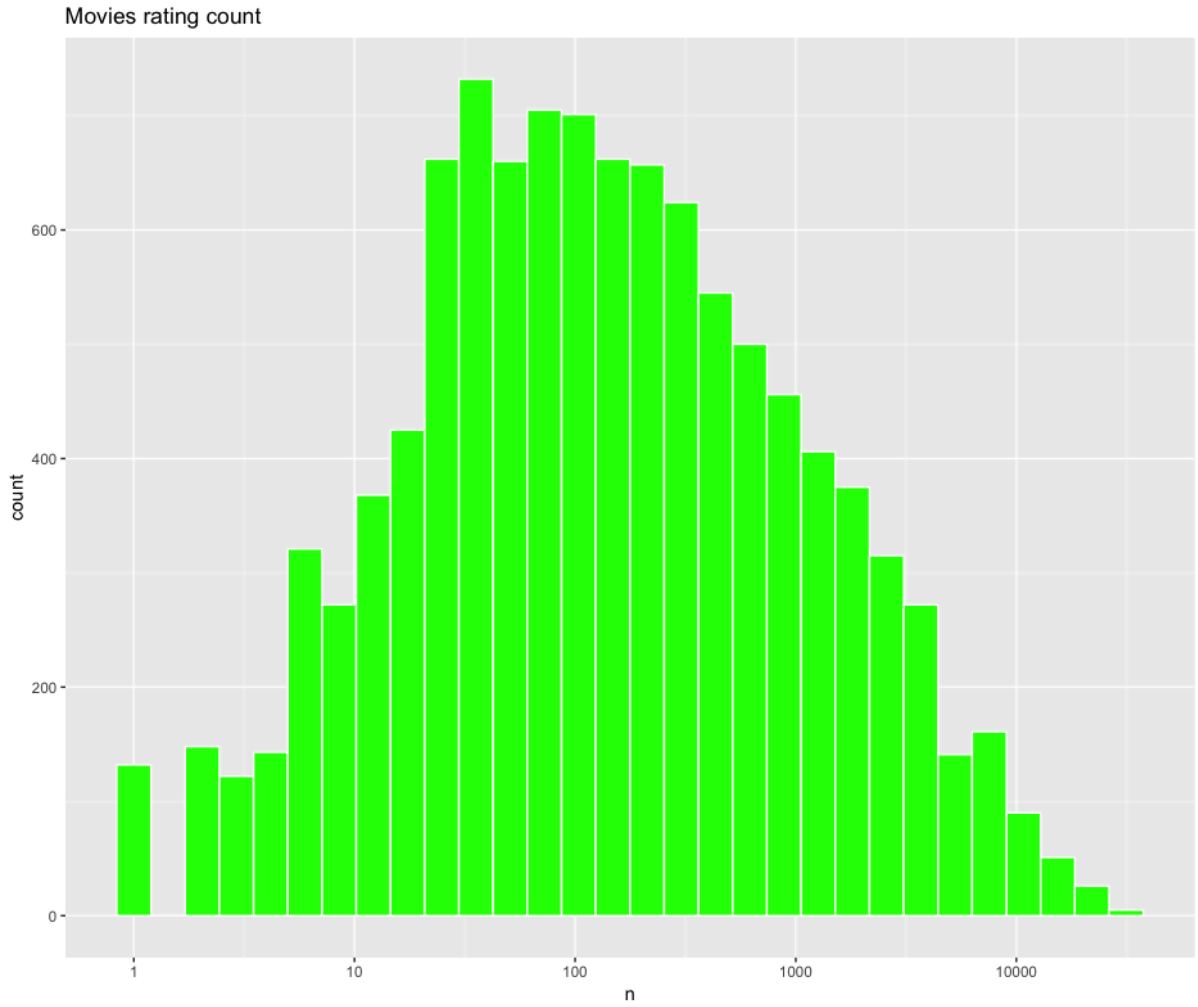
Firstly, an exploratory analysis of the data was performed to look into insights with different features affecting the rating. This was done to help in the modelling process. The technique used was data visualisation, as we can see in the form of different bar charts, which gained useful insights.



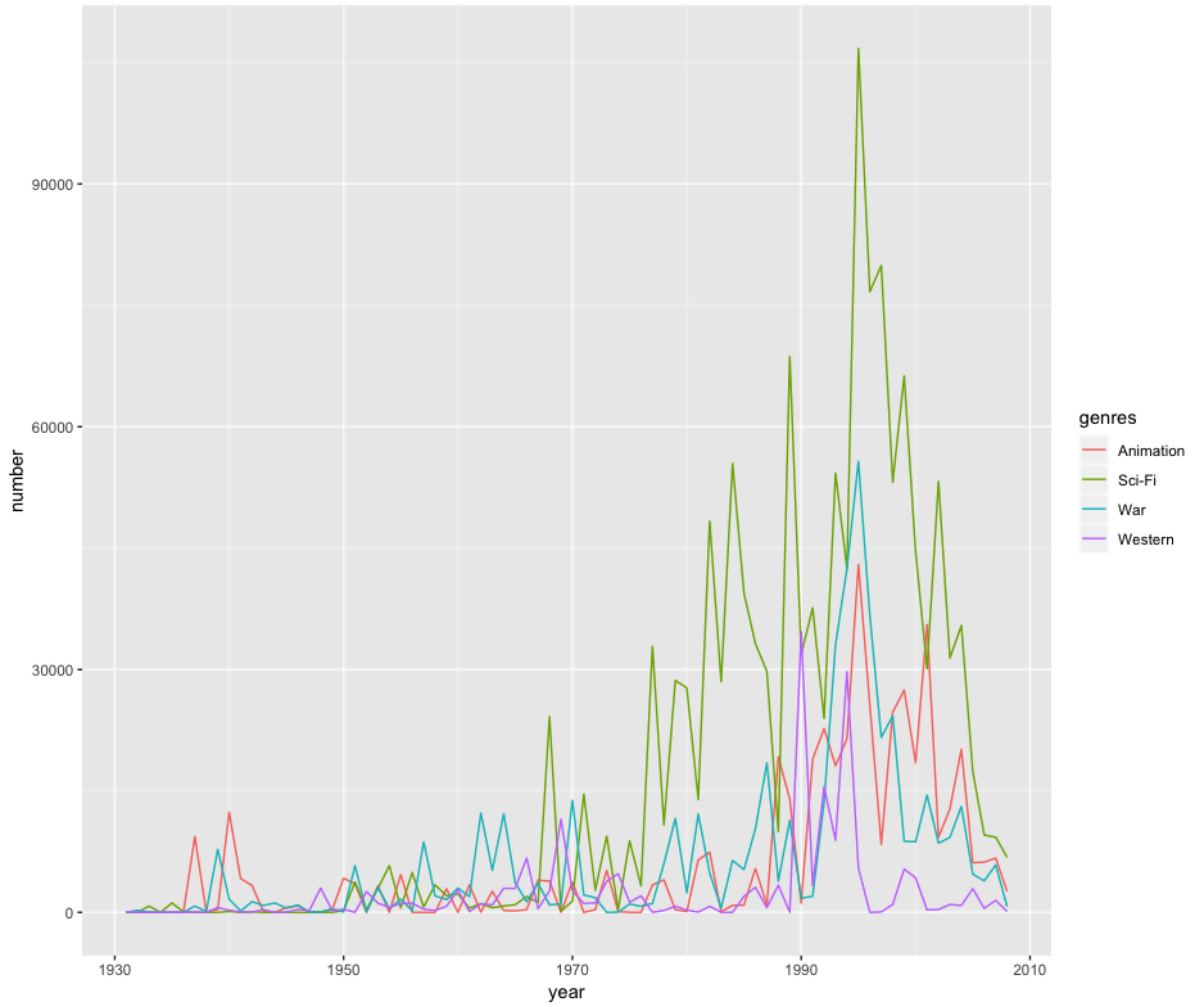
The above bar graph shows that the users mostly rate the movies between 2 and 3, conservatively. But this is just the initial analysis. For making a good model, different feature effects need to be studied as well.



The above plot shows that some users have rated very few movies and the count of users rating varies. This might affect the model results.



The above plot shows the movie rating count. It shows that some movies are rated very less and so are of least importance in the modelling process.



The plot above shows different genre's popularity varies over the years. This is interesting to visually see because global and world events impact what users are interested in and want to be engaged in watching.

Results

The model's performance will be seen by evaluating using the Root Mean Square Error (RMSE) value. Model results will help us in getting the best model.

Modelling is done by building the simplest possible recommendation system: we predict the same rating for all movies regardless of user and so mean rating.

```
# mean rating for prediction
mu_hat <- mean(edx$rating)
mu_hat
```

If we predict all unknown ratings with ^μμ^ we obtain the following RMSE is 1.06

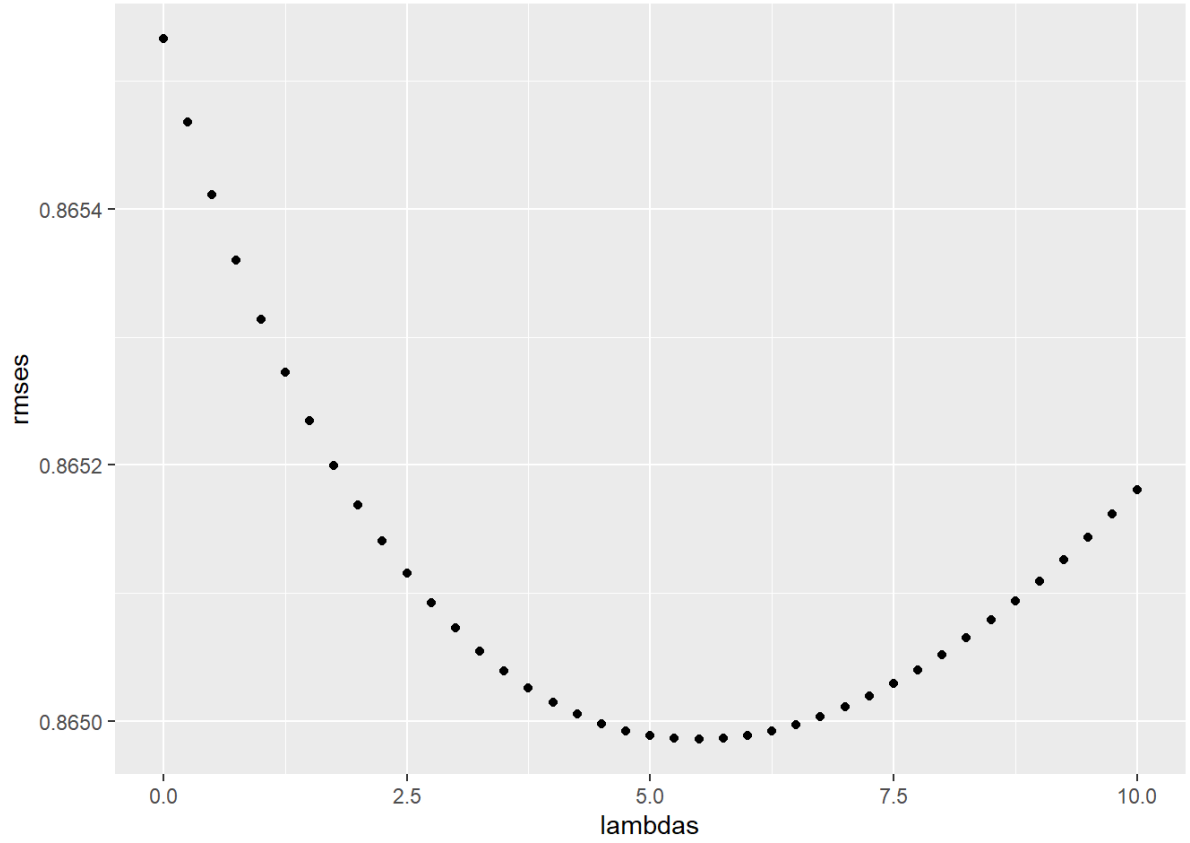
Different approaches are modelled. Now we model the movie effect as some movies are rated higher than the others. A better RMSE value is obtained by adding movie affect, as can be seen that RMSE is 0.94 in this case.

```
# movie effect model
predicted_ratings_movie_norm <- validation %>%
  left_join(movie_avgs_norm, by='movieId') %>%
  mutate(pred = mu_hat + b_i)
model_1_rmse <- RMSE(validation$rating,predicted_ratings_movie_norm$pred)
rmse_results <- bind_rows(rmse_results,
                          data_frame(method="Movie Effect Model",
                                     RMSE = model_1_rmse ))
rmse_results %>% knitr::kable()
```

A further improvement is seen in the model with both movie and user effect and the RMSE in this case is 0.86

The next method that was used is the regularization-based approach. This gave less importance to users who rated less movies, and therefore the effect is being penalized.

For the full model, the optimal λ is used against minimum RMSE and so RMSE in this case is 0.8649



Th above approach is repeated with adding variables such genre and release year effects and the RMSE is 0.8624

The RMSE values for the used models are:

Model	**RMSE**
With mean	1.06
Movie Effect	0.94
Movie and User Effect Model	0.865
Regularized Model using User and Movie Effect	0.864
Regularized Model using User, Movie, Genre, Year Effect	0.8624

Conclusion

I use AmazonPrime Video and Netflix's recommender systems to choose which movie to watch next, so this project was relatable and interesting to see what goes on behind the scenes to create recommendations. It's a simple model that recommends movies based on past preferences and ratings, and this project allows me to explore ways in which it’s done.

RMSE was used as a metric to gauge model performance which improves upon using different criteria. With only using mean the RMSE is more than 1 and next it’s tried with movie and user effect lowering the RMSE by 5% and 13% which means better value.

But there are limitations attached to the previously used models as some customers rate more movies than the others, thus affecting results so regularization is used. The regularization model was used to penalize some major data variations. So, then we finally obtained an RMSE value of 0.8624 with more improvement and so we conclude that we can predict the ratings given by users.

Future work can be done by modelling the movie recommendation system using other machine learning techniques such as random forests and decision trees as they require more computational power.