Movie Ratings Predictor

Harit Bandi-2017120005 , Suyash Joshi-2012120023, Aditya Khambete-2012120031

Department of Electronics & Telecommunication - Sardar Patel Institute of Technology
Bharatiya Vidya Bhavans Sardar Patel Institute of Technology Munshi Nagar, Andheri (West), Mumbai 400 058

harit.bandi@spit.ac.in
suyash.joshi@spit.ac.in
aditya.khambete@spit.ac.in
https://www.spit.ac.in/

***Abstract*— The quality of a movie is determined by the ratings from a reliable rating site. Movie ratings are considered to be a reference point by movie lovers these days as it gives the basic idea of how good a movie is going to be. Based on various parameters provided by a dataset it is quite possible to build a model which predicts the rating of a particular movie. This project focuses on exploratory data analysis of Ratings of many movies and predicting rating of a movie based on the parameters available in the dataset.**

I. Dataset

The **MovieLens 10M Dataset** was used. The data set consists of 10 million ratings and 100,000 tag applications applied to 10,000 movies by 72,000 users. All the movies are rated between 0.5 to 5.

II. Exploratory Data Analysis

Two datasets viz. movies and ratings are loaded and left join function is used to combine them. Column names are changed as per convenience. as_tibble() and summarize() functions are helpful in deriving some basic insights which enable us to understand the dataset properly. knitr::kable() function tabulates a data for html files. Then a matrix plot gives an idea of reviews given by first 100 users to first 100 movies. The plot is as follows-:

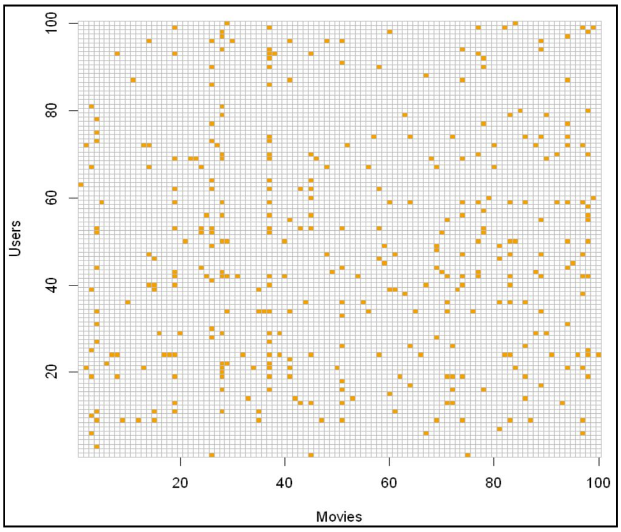


Fig. 1 reviews given by first 100 users to first 100 movies

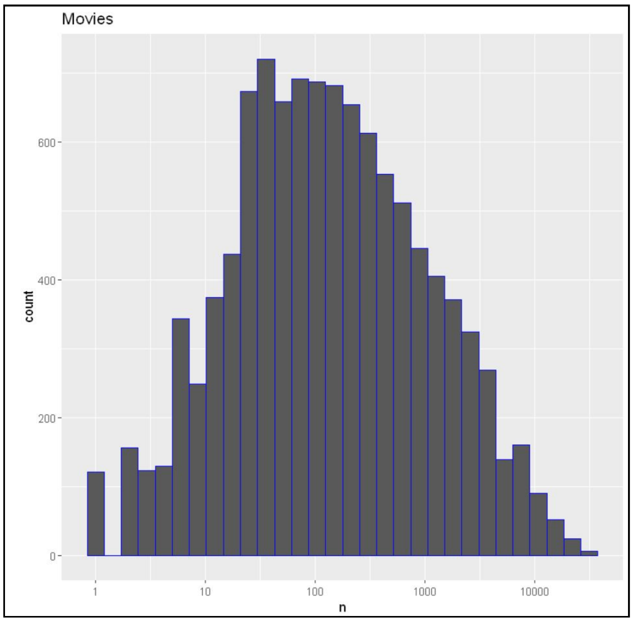


Fig. 2 Using ggplot() function the number of ratings of every movie

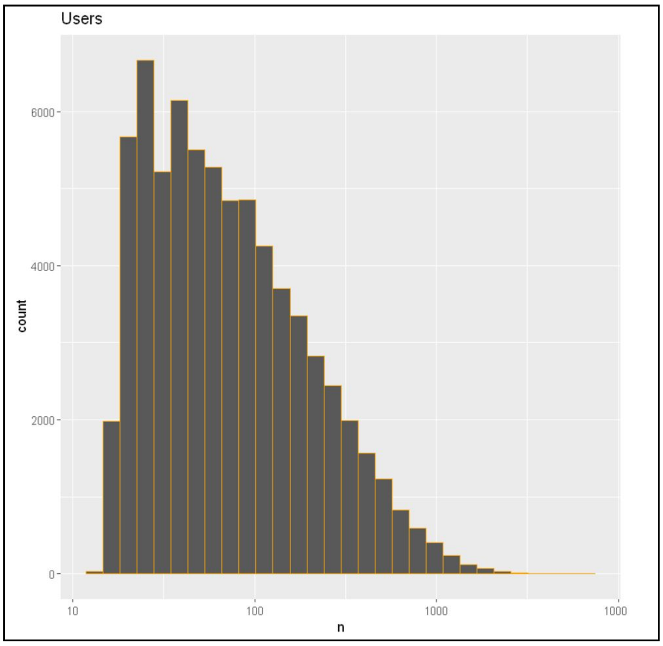


Fig. 3 the number of ratings by every user

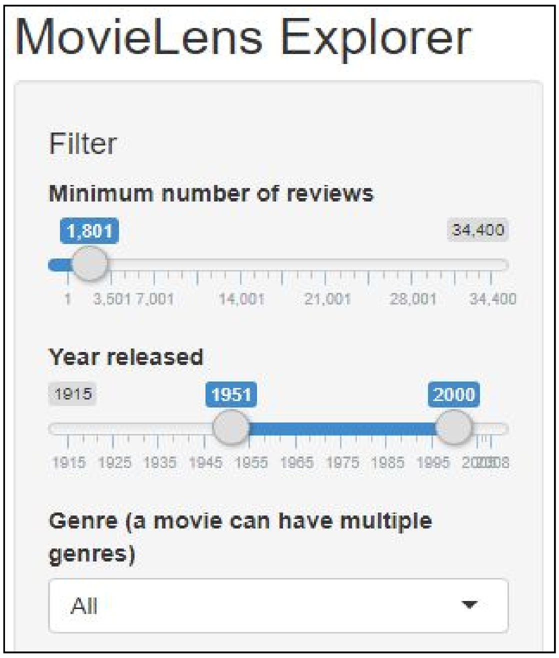
III. Rshiny Application

The Rshiny app is developed using ggvis and dplyr libraries. The innovative visualization tool in the project is the Rshiny app which provides great insights and searching tool to user to thoroughly understand the available data set. The application contains several slicers by which users can see the rating, number of reviews along with the name of a movie and year released. The slicers are as follows-:

1) Minimum number of reviews: User can display only those movies having some specific number of reviews.
2) Year released: This helps user to view movies released in some specific interval of years.
3) Genre: The movies have been categorised in different Genres. This helps to view genre specific movies at a time.

Based on the selection of specific movies selected by slicers, we get a visual which instantly provides the insights by hovering the cursor over the points on the canvas.At the bottom of the canvas the number of movies selected by the filters applied using slicers. Rshiny application link is as follows:

https://haritbandirocks.shinyapps.io/ayay/



MovieLens Explorer

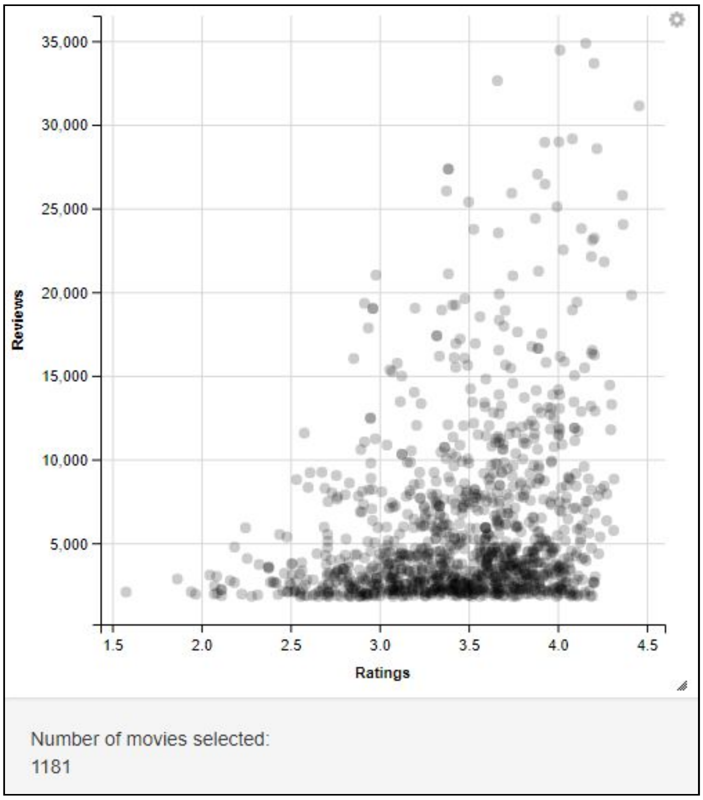
Filter

Minimum number of reviews: 1,801 (1 – 34,400)

Year released: 1951 – 2000 (1915 – 2008)

Genre (a movie can have multiple genres): All

Fig. 4 Selection Filters/Slicers



Number of movies selected:
1181

Fig. 5 Result of movies based on selected filters

IV. Model Building

The model building for this project has been done using R programming language on Jupyter notebook environment. tidyverse, caret, data.table, recosystem are the packages used for the same.

1) Mean

Mean is a basic statistical measure for predictions. If mean is used for predicting the value of model regardless of any parameter related to user, we get RMSE=1.060393

2) Normalization – Ratings with respect to Mean

The (rating - mean) value is plotted on the x-axis and the count of such rating is plotted on the y-axis. The count is maximum at mean i.e. b_i=0.

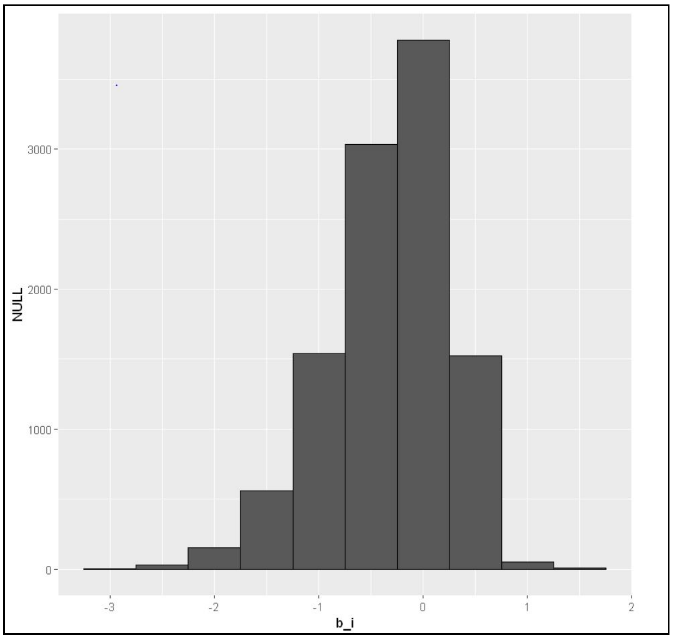


Fig. 6 distribution of movie ratings with respect to the mean

This normalization technique improves the model and gives RMSE=0.942368

3) Normalization-Mean of Ratings with respect to Users

The model is normalized with respect to users in order to further improve the model accuracy.

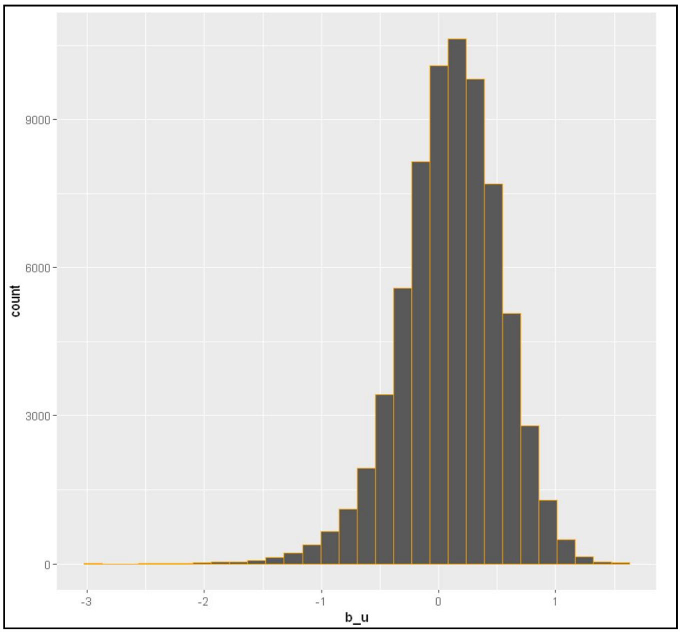


Fig. 7 distribution of movie ratings with respect to the users.

This normalization improves the model to RMSE=0.8566699

4) Regularization

Regularization permits us to penalize large estimates that are formed using small sample sizes.These are noisy estimates that we should not trust, especially when it comes to prediction. Large errors can increase our RMSE, so we would rather be conservative when unsure. We use cross-validation to find lambda for which RMSE is minimum.

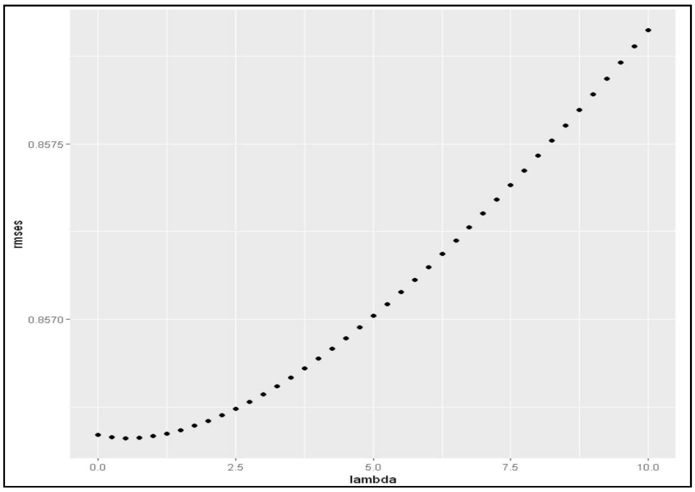


Fig 8. Cross Validation

After cross-validation we get RMSE=0.8566611. The long dataset prevents obscure movies with just a few users increase our RMSE, so the regularization does not produce significant improvements in performance using the RMSE as a metric. Thus, we focus on the movies and users' effects to calculate the residuals. These residuals are modeled using Matrix Factorization.

After Matrix Factorization, the model improves significantly to RMSE=0.7425414

V. Conclusions

The final model is used to validate on the validation set which gives RMSE=0.7948597.

References

[1] https://github.com/rstudio/shiny-examples/tree/master/051-movie-explorer

[2] https://shiny.rstudio.com/tutorial/

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

[4] https://www.inderscienceonline.com/doi/abs/10.1504/IJHPCN.2017.083199

[5] http://www.imdb.com.

[6] Bruke, R.Hybrid recommender systems: survey and experiments, User Modeling and User Adapted Interaction 12(2002) 331-370.

[7] P. Melville, R.J. Mooney, R. NagarajanContent-BoostedCollaborative Filtering Improved Recommendations, Proceedings of the 18th National Conference on Artificial Intelligence (AAAI-2002), July 2002, Edmonton, Canada

[8] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.151.7838&rep=rep1&type=pdf