BUAN 6357 Advanced Business Analytics Using R

Project Report
Movie Recommender System

- By Animesh Kansal (axk169531)

Executive Summary:

I created a movie recommender app. You select the top 7 movies of your choice and the app will recommend you similar movies, movies that similar users have seen and popular movies. It will also create a word cloud displaying your favorite genre based on the movie choices.

Data set:

To create the model, I used the 100k MovieLense ratings data set. The data was collected through the MovieLens web site (movielens.umn.edu) during the seven-month period from September 19th, 1997 through April 22nd, 1998. The data set contains about 100,000 ratings
(1-5) from 943 users on 1664 movies. Movie metadata is also provided in MovieLenseMeta.

Algorithms used:

I used the following 3 algorithms to create the movie recommendations present in recommenderlab package in R.

1. Most Popular recommendation
2. Item Based Collaborative Filtering.
3. User Based Collaborative Filtering.

Future Improvement:

I have not used following algorithms but using these we can further improve the recommendations.

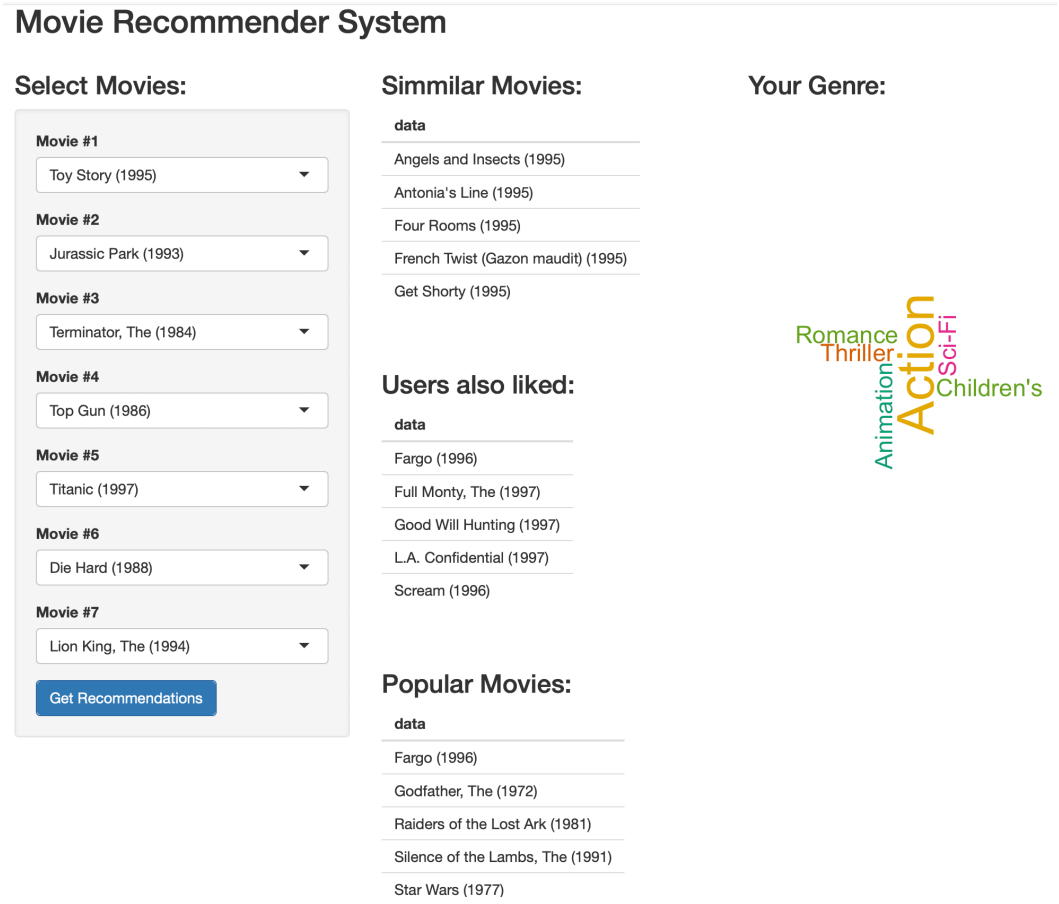
1. SVD
2. Matrix Factorization

Shiny App:

I deployed the app on the web, here is the link

https://animeshkansal.shinyapps.io/MovieRecommender/

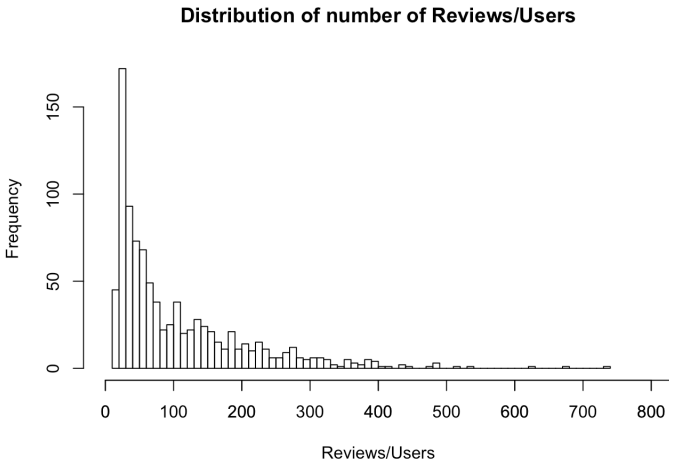
Screenshot of Shiny App:



Exploratory Data Analysis:

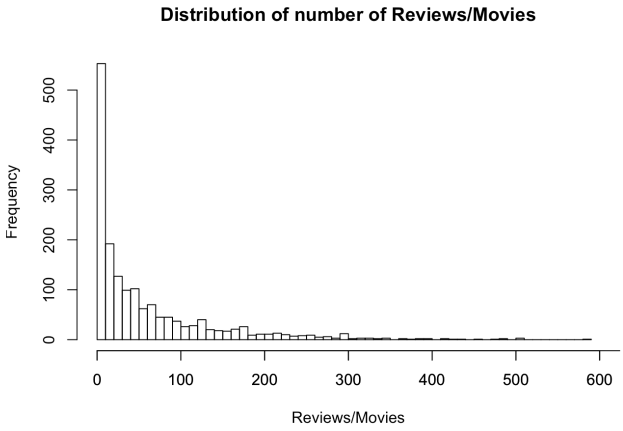
In report, I have only covered major EDA work. The remaining EDA can be found on the markdown HTML output.

1.) Filter the large ratings matrix.



Range of Reviews/Users - [19,735]

I did not remove any user from final model based on number of movies they reviewed.



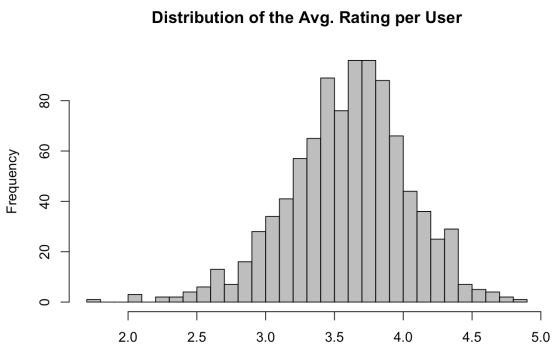
Range of Reviews/Movies – [1,583]

I removed movies with any less than 20 reviews. Since, these movies are very less popular and also, reviews might be biased.

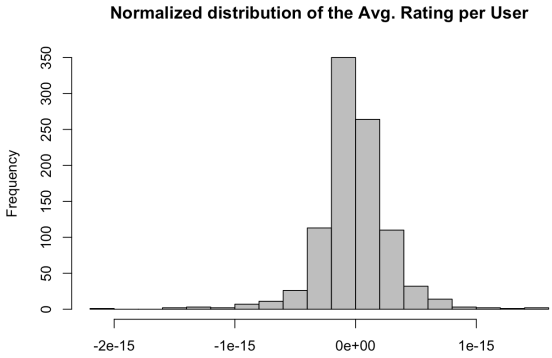
2.) Normalized User ratings:

There is very different type of users, one who give high (or low) ratings to all their movies. To tackle this bias problem, I normalized the data in such a way that the average rating of each user is 0.

Before



After



Recommender System Evaluation/performance:

Dataset was divided into

Train set :75% Test set:25%

Error metric: RMSE

Algorithm	RMSE
1. Most Popular recommendation	0.9673421
2. Item Based Collaborative Filtering	1.037869
3. User Based Collaborative Filtering	1.102027

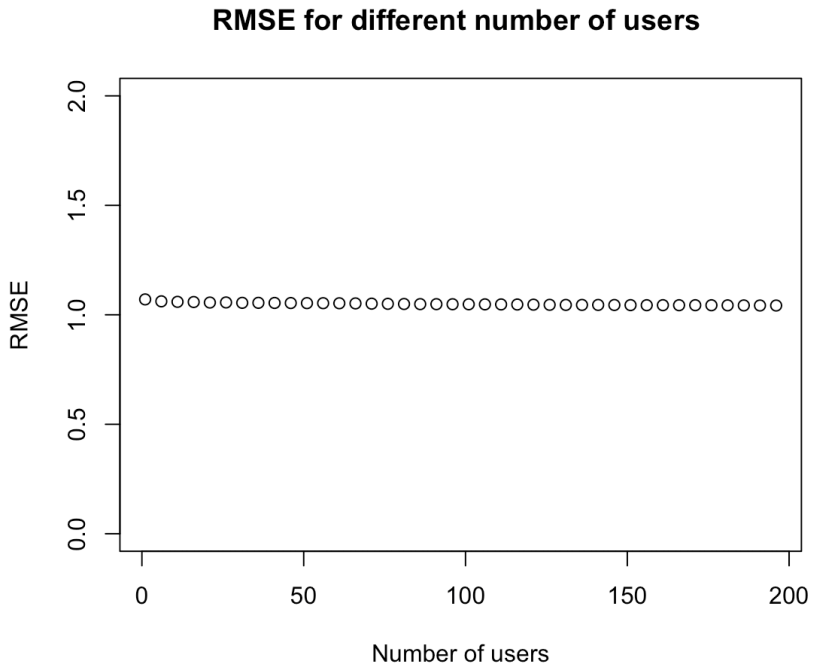
Hyper parameter tuning:

User Based Collaborative Filtering

Algorithm:
1. Calculate similarity between user *u* and all other users. For this could be used any preferred similarity measure;
2. Select top *n* users with the highest similarity to users *u*;
3. Calculate predictions for unknown ratings for user *u* as average of available ratings from *n* closest users or as weighed (on similarity distance) ratings of *n* closest users.

1) I used cosine similarity for similarity criteria, since all these ratings are comparable and not just binary rating.

2) I tuned top n users



Item Based Collaborative Filtering

Algorithm:
1. Calculate similarity matrix between all items based on available users' ratings. For this could be used any preferred similarity measure;
2. For user *u*:
2.1 Store only *n* closest items to each item;
2.2 Calculate predicted rating for each item based on available ratings of user *u* by weighting available ratings of users on similarities.

1) I used cosine similarity for similarity criteria, since all these ratings are comparable and not just binary rating.

2) I tuned number of closest items (movies in this case)