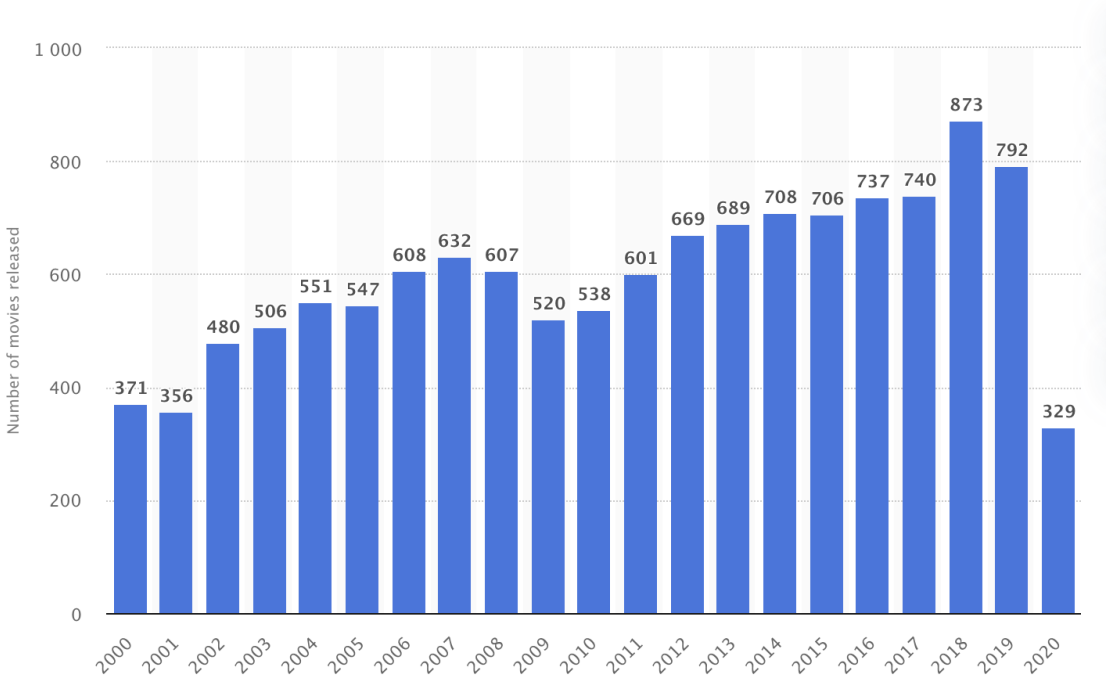
Building a Movie Recommender System

By Tianyang Shen

Abstract

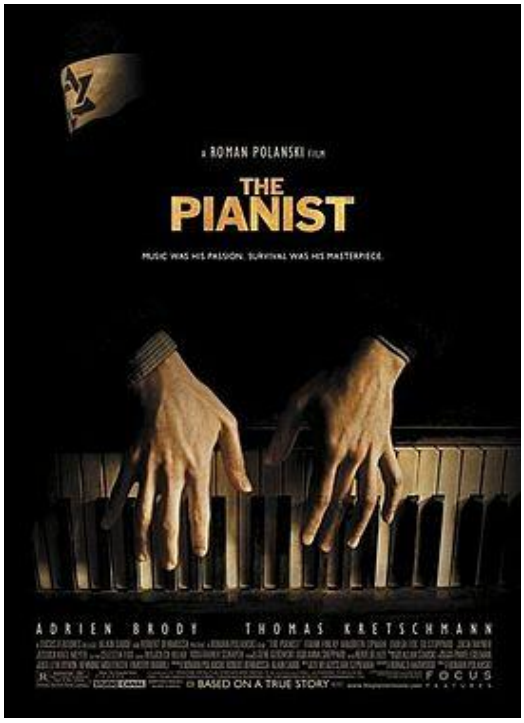
We built a collaborative-based recommender system of movies by SVD algorithm. On average, the final trained model predicts the user's rating on a movie to within 0.68 points of the actual rating. The overall statistics reveals that for movies with lesser reviewers, our model tends to overpredict, while for relatively popular movies, the model tends to slightly underpredict. It is also revealed that for almost every genre of movies, 75% of the median prediction error is within the range of -0.1 to 0.3. We also conclude that, for a user who has large number of review records, typically our recommendation system has the average error around 0.6.

Since the invention of films more than a century ago, movies have become one of the most important forms of art and entertainment. In 2019, the total gross revenue for movies is estimated to be $11.4 billion. In the year 2011-2019, more than 600 movies in the US and Canada were released each year. With so many options, how can viewers decide which movies to watch? Using data science techniques, we can build a recommendation system to suggest films viewers may like. Viewers may prefer certain genres, actors, and directors when searching for a movie (called content-based filtering). Another approach is to check other users' rating records (collaborative filtering). It is reasonable to assume that people with similar taste and history of movie ratings tend to prefer the same movies.



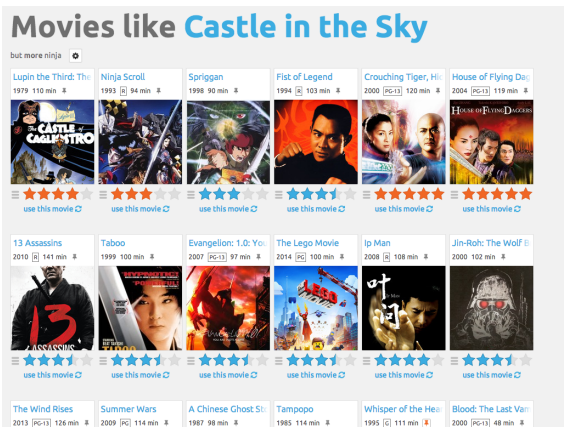
New movie released yearly in US and Canada

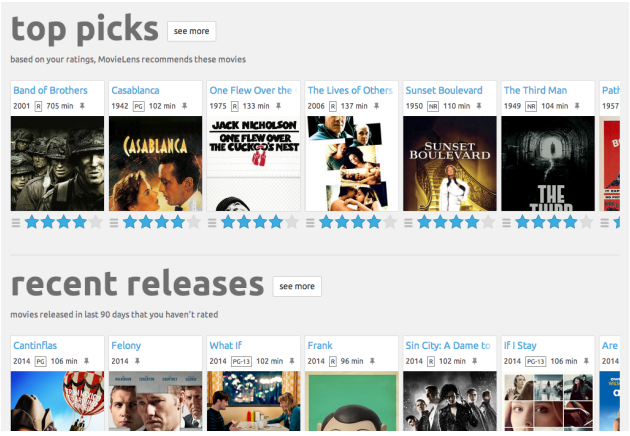
Movie databases collect ratings for each movie. This information can be used to build a recommendation system to recommend new movies to viewers. The goal of the system is to input a user ID and output the top movies the machine learning algorithm suggests.



.

This project focuses mainly on collaborative-based filtering. In this method, the system will provide estimated ratings for movies unrated by certain users based on rating records from other similar users. For example, if Alice and Bob share more than 20 same movies that they both rated for 5, then for a movie Alice rated top that Bob hasn't watched, it is highly potential that this movie will be in the recommendation list of Bob. This filtering method will rely on model-based approaches, such as clustering algorithms and matrix factorization algorithms, to predict ratings for unrated movies.

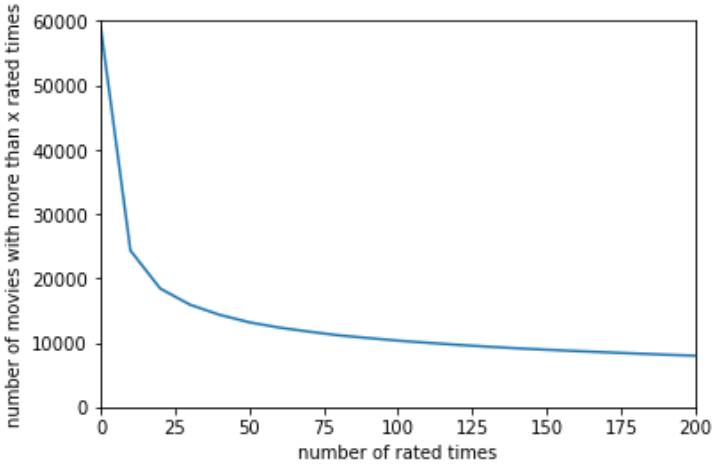




Data

The data set is from MovieLens. It describes 5-star rating (lowest 0.5, highest 5.0 with step 0.5) and free-text tagging activity from MovieLens, a movie recommendation service. It contains 25 million ratings and 1.1 million tag applications across 62,000 movies. These data were created by 16,3000 users between 1995 - 2019. Users were selected at random for inclusion. All selected users had rated at least 20 movies. No demographic information is included. Each user is represented by an ID, and no other information is provided. The movie dataset has the information for the movie title, release year and the genre.

IMDB provides other basic information for the movies, allowing us to include the director, actor and actress lists for each movie. Though information is not used for building up the recommendation system, it is important for a real movie searching engine and it is therefore kept. About 3.6% of movies don't have records in IMDB, so they are dropped from the data set.



The distribution of the rating numbers for the movies shows that there are more than 30000 movies that have less than 10 reviewers. It is difficult to make accurate predictions on movies with few reviews, therefore only movies with at least 10 reviewers were studied. In all, 24430 movies are kept in the dataset. These movies represent 25 million rating records.

Statistics by genre

Each movie in the dataset can belong to multiple genres. We reshape the movie dataset to represent genre with multiple columns of boolean parameters. In the following example, we can see that movieId 1 is Toy Story released in 1995, with the average rate 3.893708, and it belongs to “Adventure”, “Animation”, “Children”,and “Comedy” genres (and possibly other unshown columns of genres).

movieId	title	rating numbers	average rate	Action	Adventure	Animation	Children	Comedy	Crime	Documentary	...
1	Toy Story (1995)	57309	3.893708	False	True	True	True	True	False	False	...
2	Jumanji (1995)	24228	3.251527	False	True	False	True	False	False	False	...
3	Grumpier Old Men (1995)	11804	3.142028	False	False	False	False	True	False	False	...
4	Waiting to Exhale (1995)	2523	2.853547	False	False	False	False	True	False	False	...
5	Father of the Bride Part II (1995)	11714	3.058434	False	False	False	False	True	False	False	...

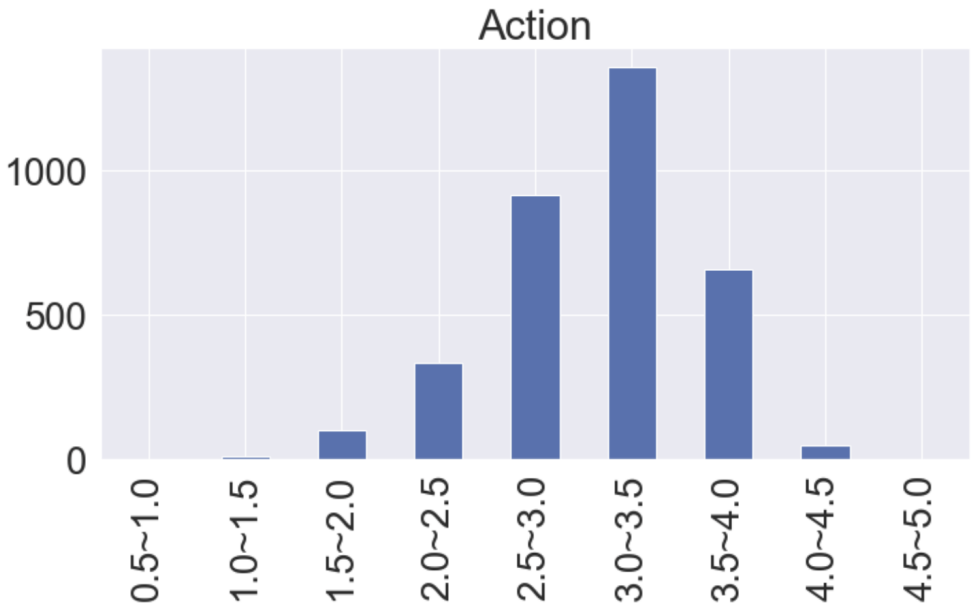
The dataset of movies, with genres in boolean parameters

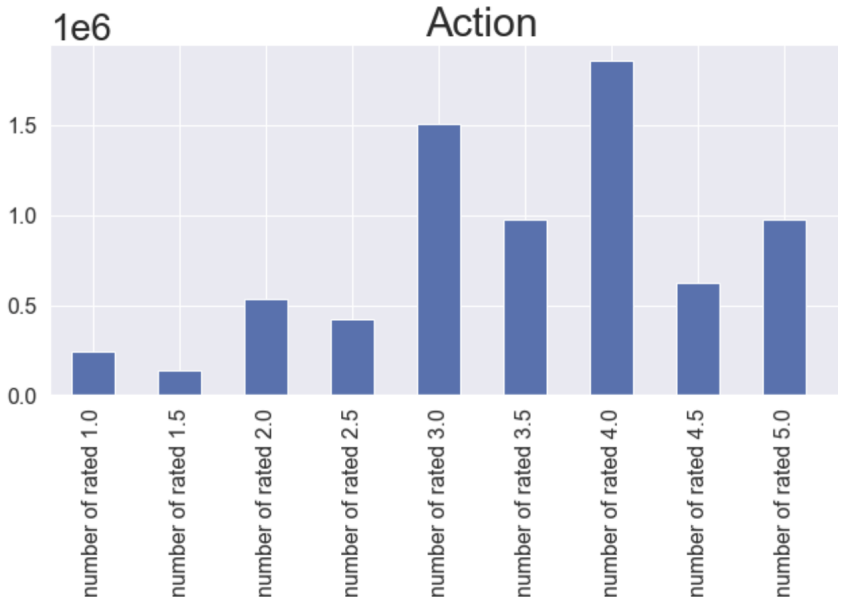
In order to understand how the genre may affect the rating tends, we plot the distribution of ratings in a certain genre, and the distribution of the average ratings of movies in a certain genre.

The first group of histograms in the next page shows how many users rated movies belong to a certain genre in a certain genre.(The “action” genre is given for a concrete example.) For example, there are more than 1400,000 records of “Adventure” movies rated between 3.5 and 4.0 shown in the second plot. We can conclude that people are more likely to rate movies with integer values (1,2,3,4,5) rather than half points (1.5, 2.5, etc). The modal value for ratings is 4 for all genres. That histograms also reveal that ratings are skewed positively that a rating of 5.0 is more common than 1.0.

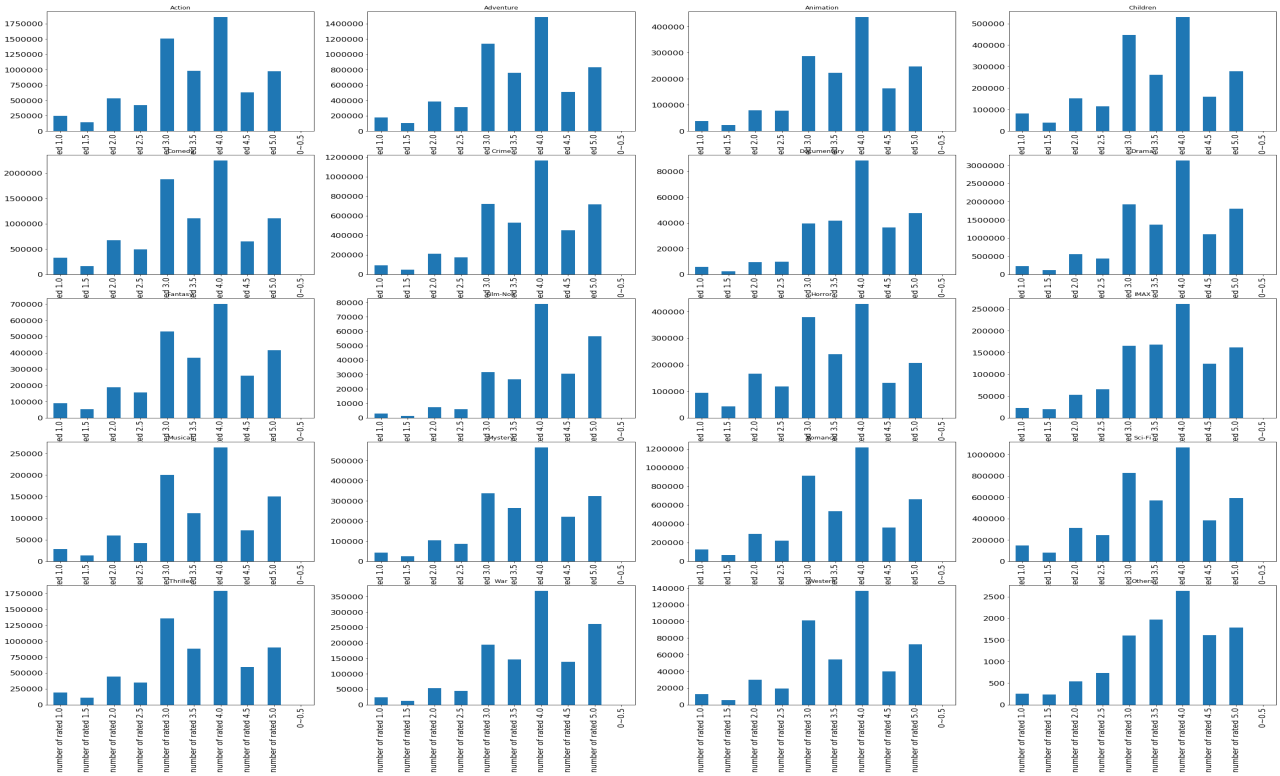
The second plot of the next page shows the distribution of the average ratings of a movie by genre. (The “action” genre is given for a concrete example.) For example, there are about 900 movies in the genre “action” with average rate 3.0~3.5 shown in the first plot.

We also plot the box diagrams of the average rating of movies by genre. The plots suggest that almost every genre has the centre value rate between 3.0 and 3.5, and the most centered 75% rates are within 1 point of maximum difference. It also suggests that the extreme ratings are most from low ratings.

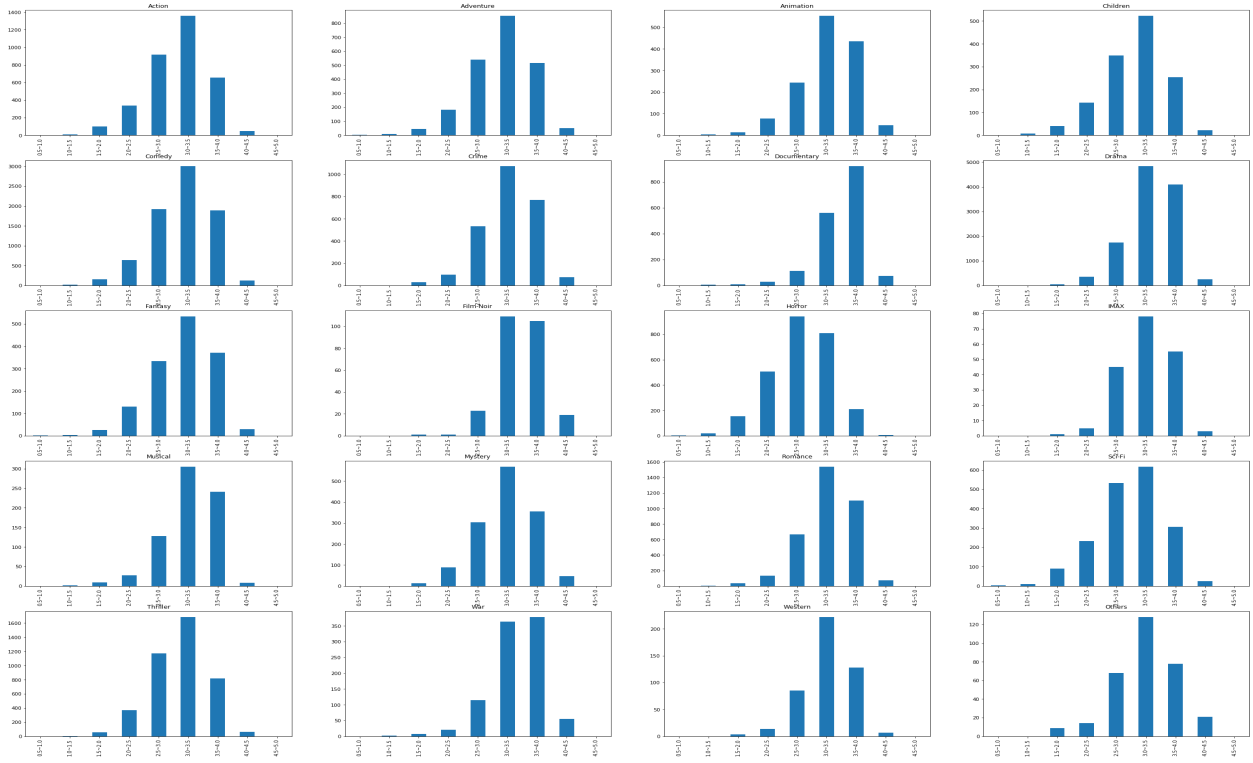


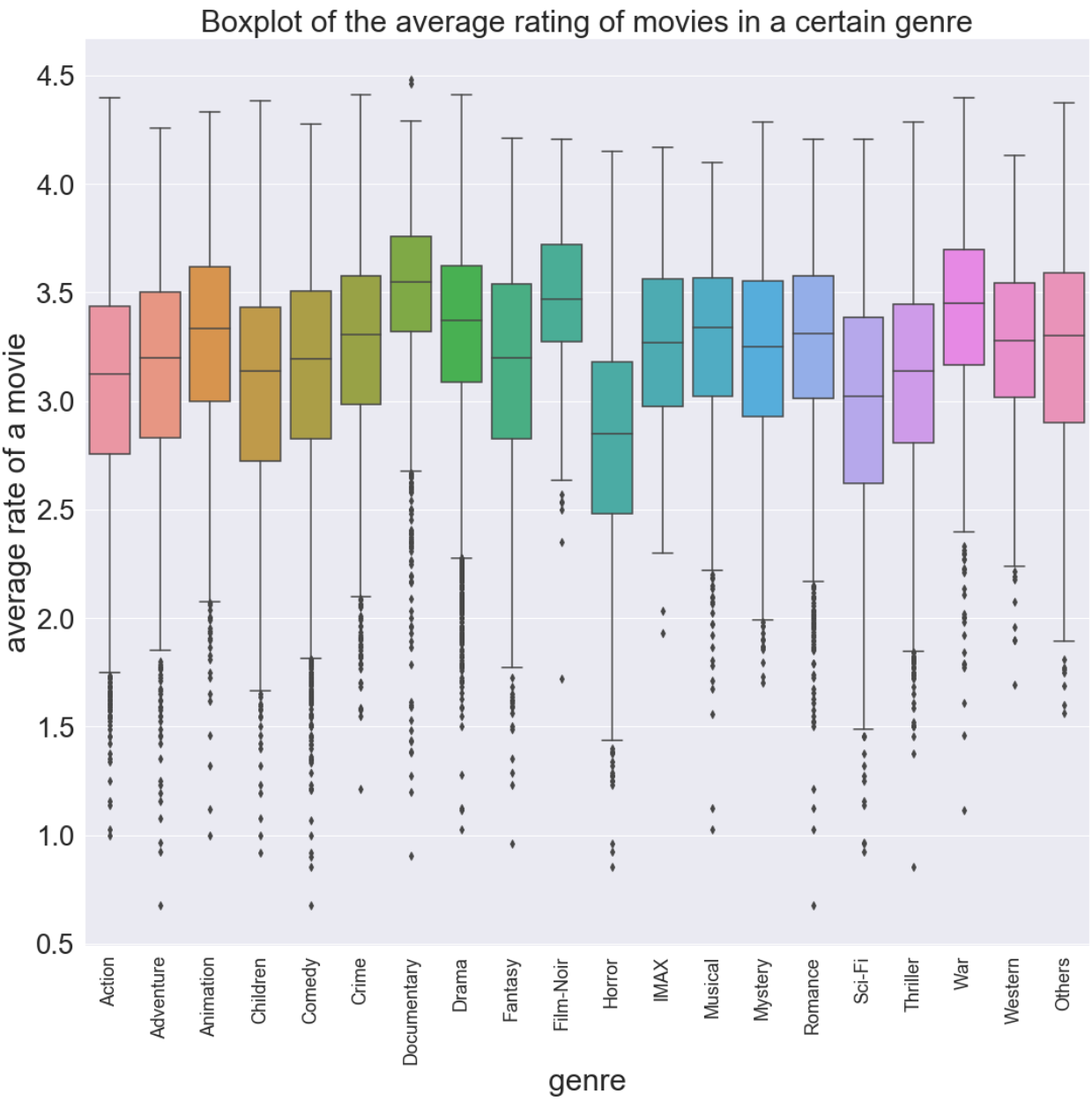


distributions of different ratings in each genre



distributions of numbers of movie in different range of ratings for each genre




Boxplot of the average rating of movies in a certain genre
average rate of a movie
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
Action
Adventure
Animation
Children
Comedy
Crime
Documentary
Drama
Fantasy
Film-Noir
Horror
IMAX
Musical
Mystery
Romance
Sci-Fi
Thriller
War
Western
Others
genre

movieId	title	rating numbers	average rate
171011	Planet Earth II (2016)	1124	4.483096
159817	Planet Earth (2006)	1747	4.464797

Here are example of highly rated documentaries. The documentary series Planet Earth gains the highest rating.

movieId	title	rating numbers	average rate
171479	Kidnapping, Caucasian Style (2014)	14	0.678571
103869	Bigfoot (2012)	13	0.923077
139761	Sharktopus vs. Whalewolf (2015)	14	0.964286

Examples of worst rated Adventure

movieId	title	rating numbers	average rate
120222	Foodfight! (2012)	24	1.000000
145096	Barbie & Her Sisters in the Great Puppy Advent...	59	1.118644
6371	Pokémon Heroes (2003)	355	1.321127

Examples of worst rated Animations

User Rating Trends

We analyze the features of ratings by users, and it is shown that only about 0.2% users have the difference of high rate between low rate that is less than 1pt, so that they will not have a

significant influence on the recommendation system built. We just simply do not consider the range of ratings of a certain user as a factor.

The model

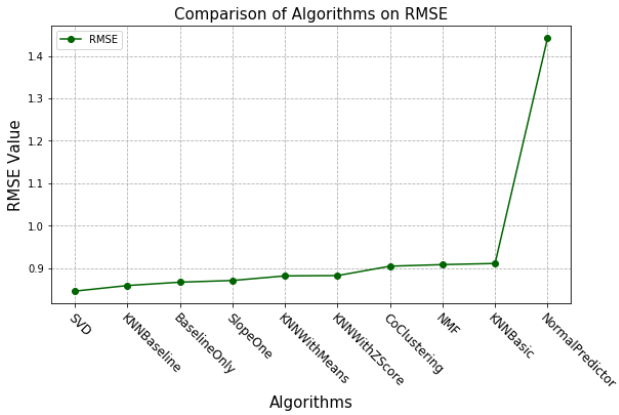
The sci-kit surprise package is used to build the collaborative-based method model. The general idea of the algorithm can be described in the following formula. For a given user i with unrated movie j, the prediction r_{ij} is the weighted average of ratings from other users (j in the formular) who have rating records for this movie, with similarity function between user i and j being the weight factor. The formula also includes bias terms to reduce the bias of ratings from specific users. The similarity function can be the Euclidean distance, Pearson's coefficient and cosine similarity.

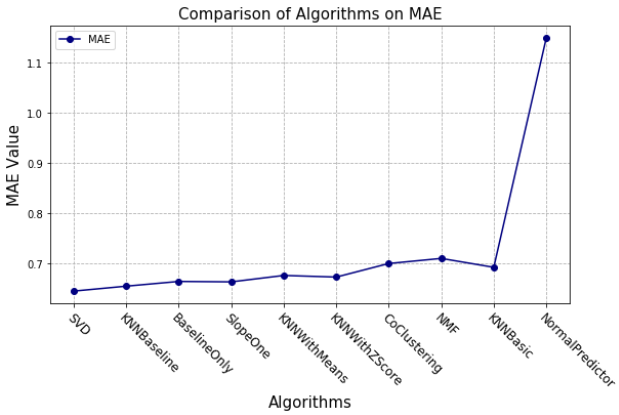
$$r_{ij} = \frac{\sum_k Similarity(u_k, u_i) * (r_{kj} - \bar{r_k})}{number\ of\ ratings} + \bar{r_i}$$

The algorithms of SVD,SVDpp, SlopeOne, NMF, NormalPredictor, KNNBaseline, KNNBasic, KNNWithMeans, KNNWithZScore, BaselineOnly, CoClustering are tested in the first round of selection with 10k rating records. It turns out that SVDpp is a very time-consuming algorithm, so that it is not suitable for a dataset of 24.88 million rows.

Algorithm	test_rmse	test_mae	fit_time	test_time
BaselineOnly	0.900157	0.701801	0.020949	0.020054
SVDpp	0.907694	0.702333	28.881190	0.896235
SVD	0.908462	0.704477	0.389803	0.018576

In the second round, we test models with a random sample of 1M records, using each algorithm. The SVD algorithm yields the lowest RMSE and MAE. Since a user has only a very small portion of movies rated, and also a movie has rating records from a very small portion of users, the problem of scarcity and sparsity rise here. Therefore, the matrix-factorization method is realized by this SVD (singular value decomposition) algorithm to decompose the original sparse matrix to low-dimensional matrices with latent factors/features and less sparsity.

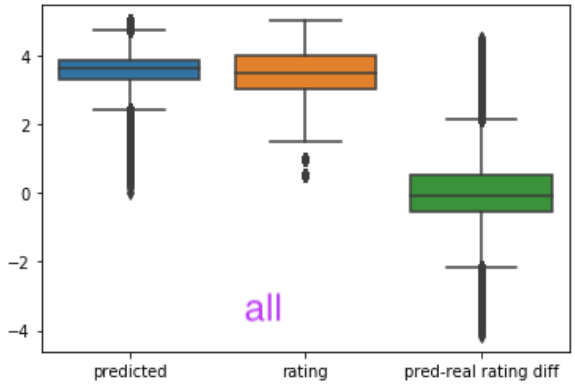


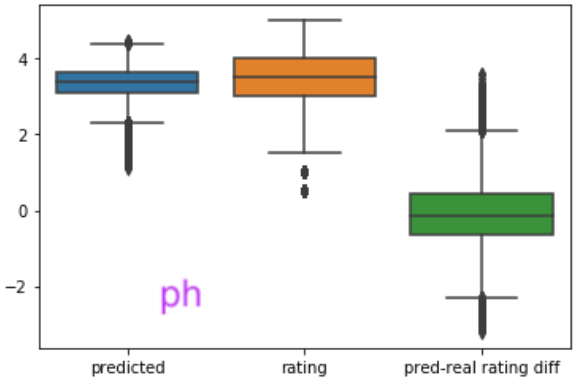


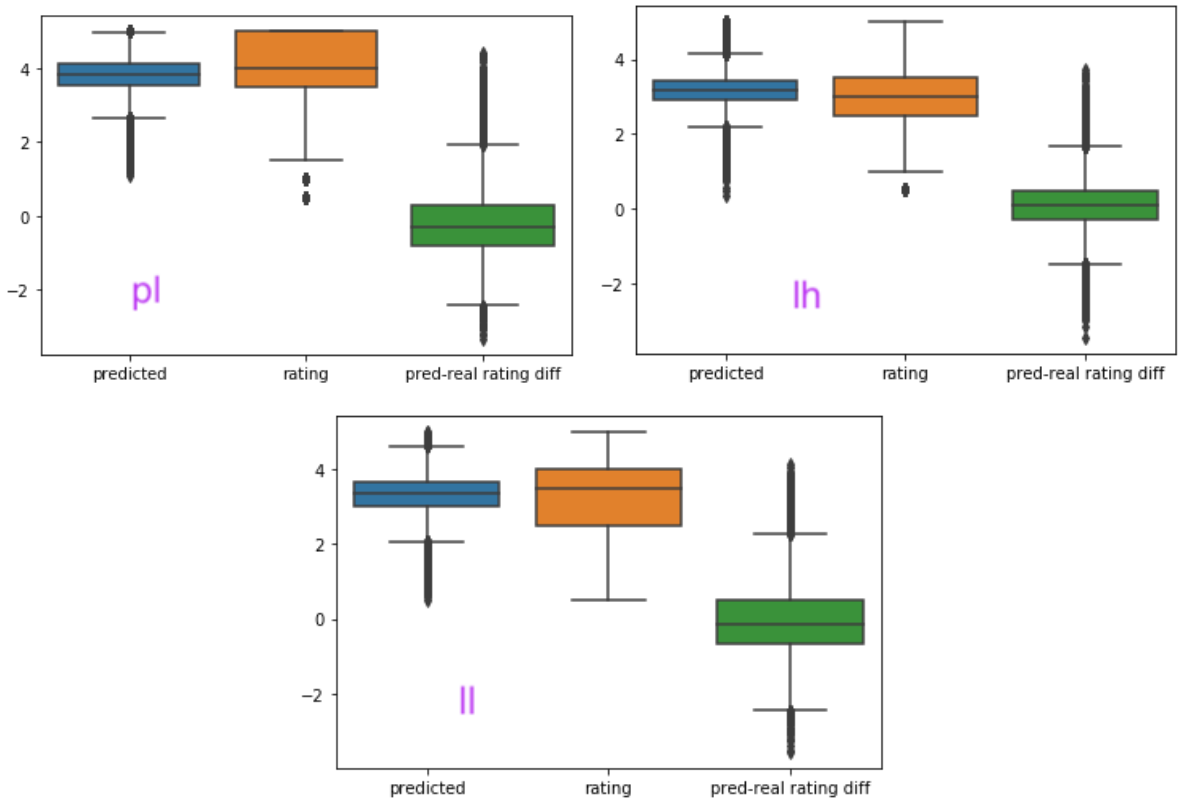
In the third round, we conduct a grid search of hyperparameter space for the SVD algorithm and determine to use the best performing parameters. The final trained model has a MAE of 0.6801 and RMSE of 0.8846. In other words, on average the model predicts the user's rating on a movie within 0.68 points of the actual rating.

Model Performance Analysis

In order to understand how the recommendation system model works, we consider two special groups of movies: popular movies with more than 10,000 reviews and lesser-known movies with less than 50 reviews; as well as two groups of users: heavy users with more than 500 movies rated, and light users with less than 10 movies rated. To simplify the text, we use "p" and "l" for popular and lesser-known movies, "h" and "l" for heavy and light users. For example "ph" stands for popular movies rated by heavy users, while "ll" stands for lesser-known movies rated by light users.

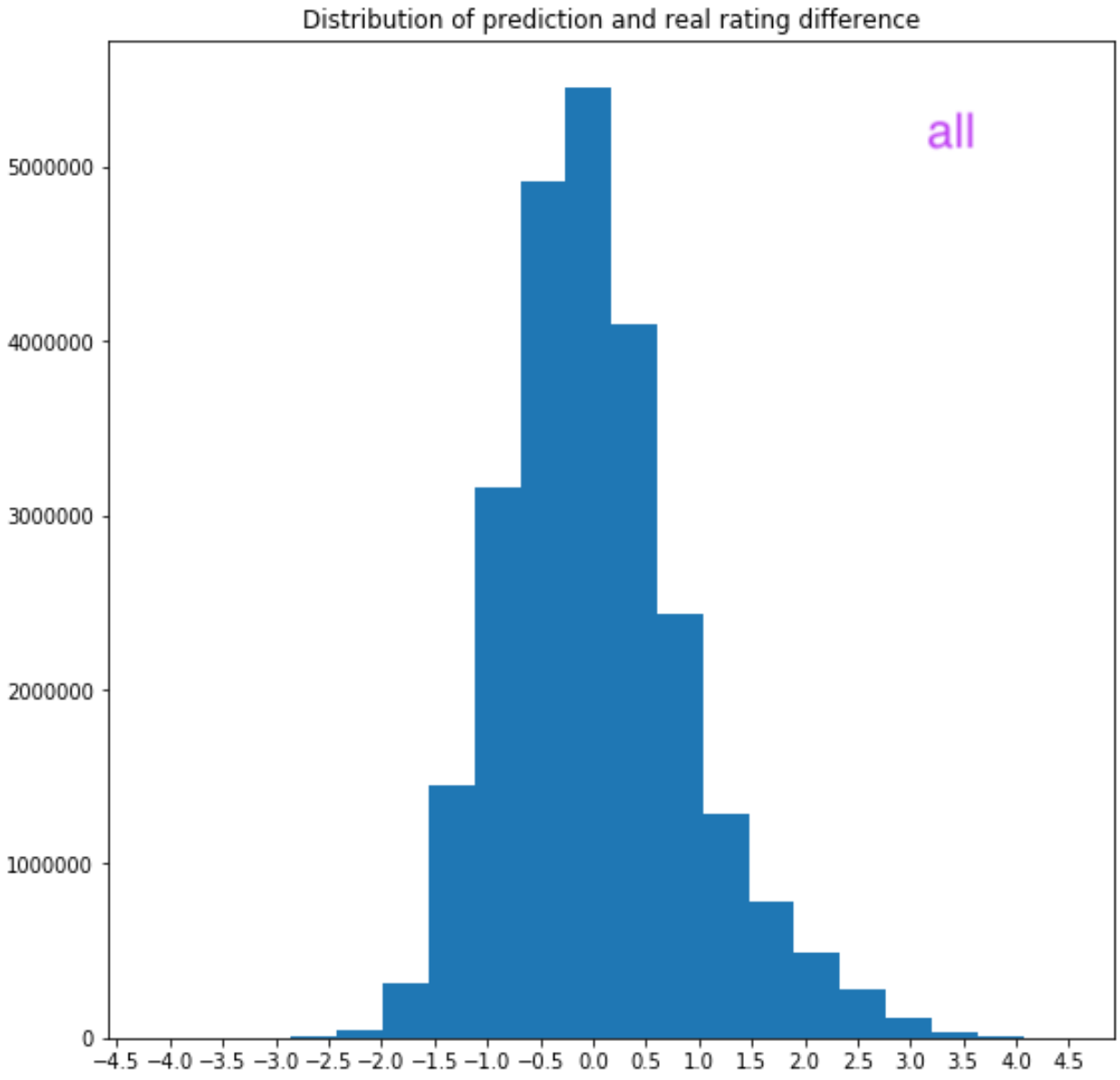


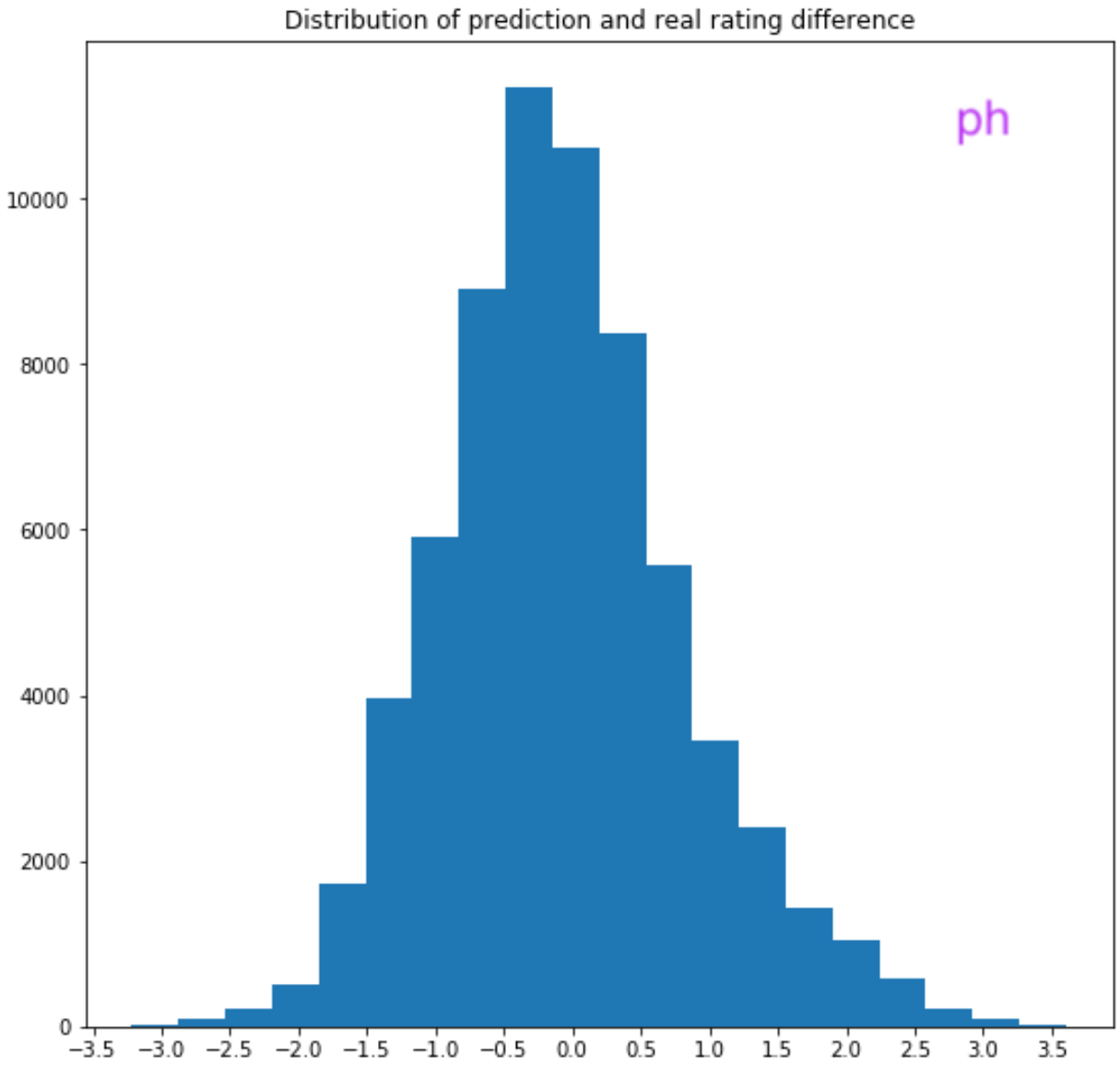


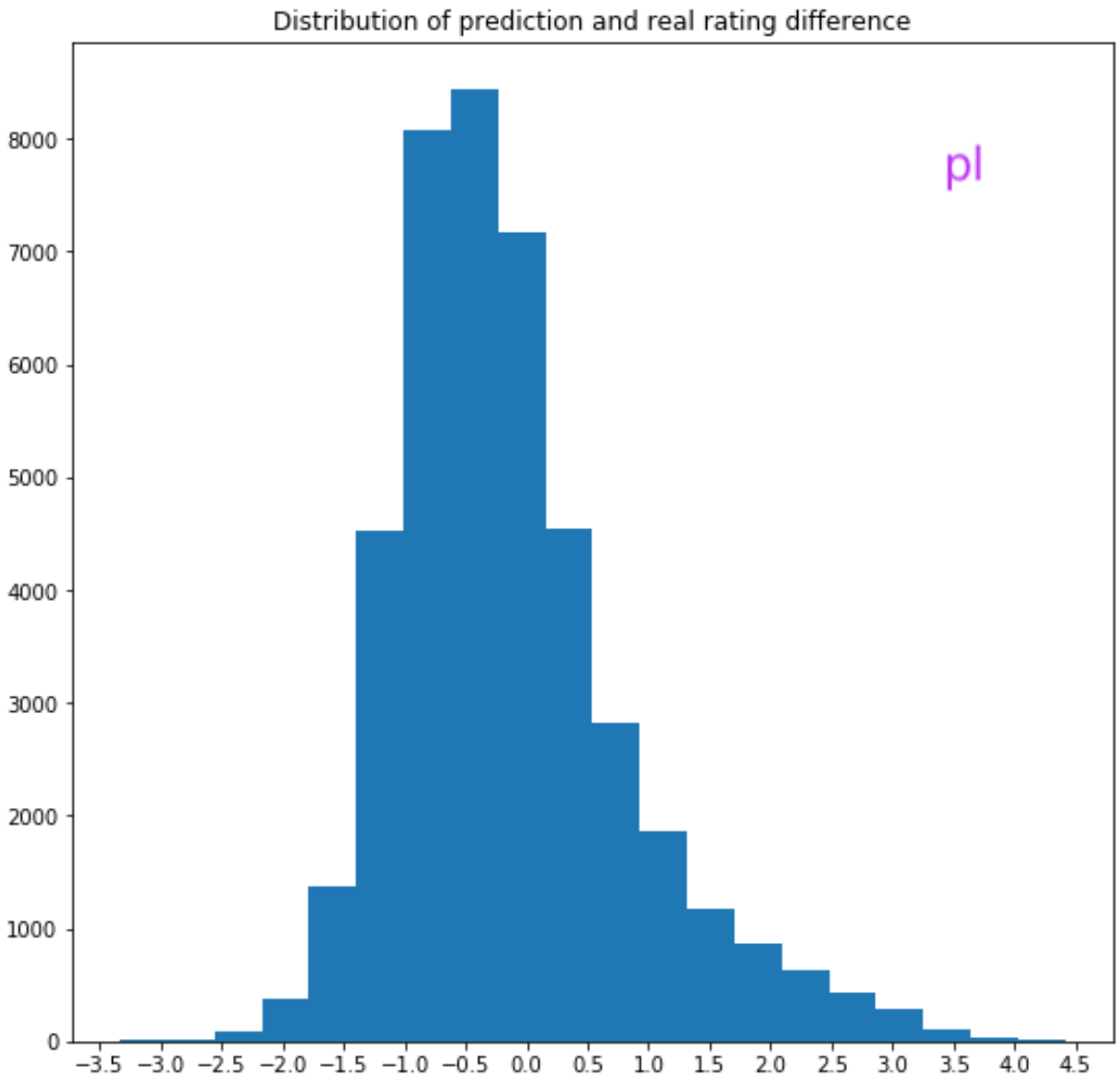


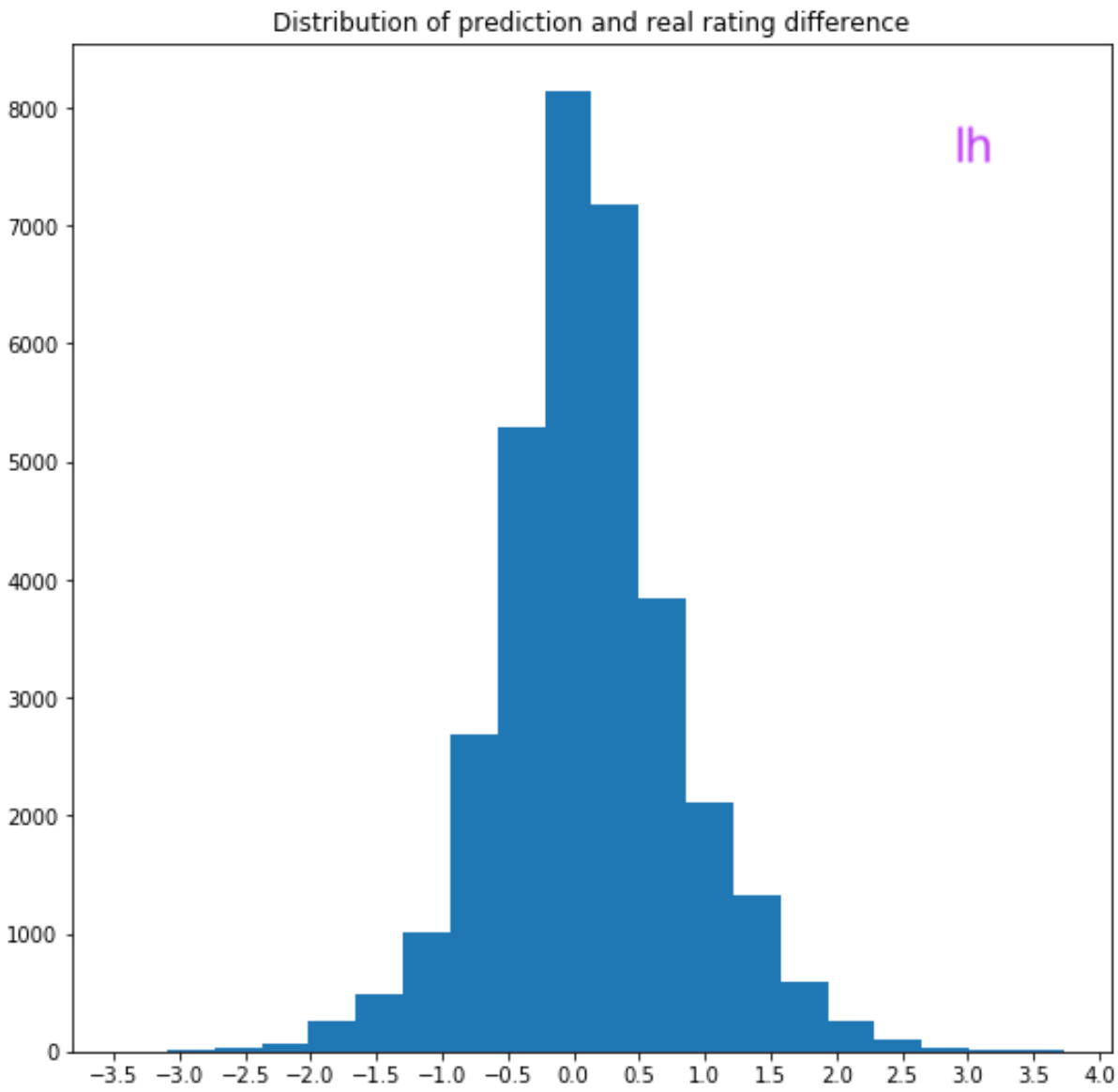
The box diagrams of the predicted ratings of each case suggest that our model tends to provide a narrower range of ratings than the real ratings. In addition, the model tends to slightly underpredict popular movies and over-predict lesser-known movies.

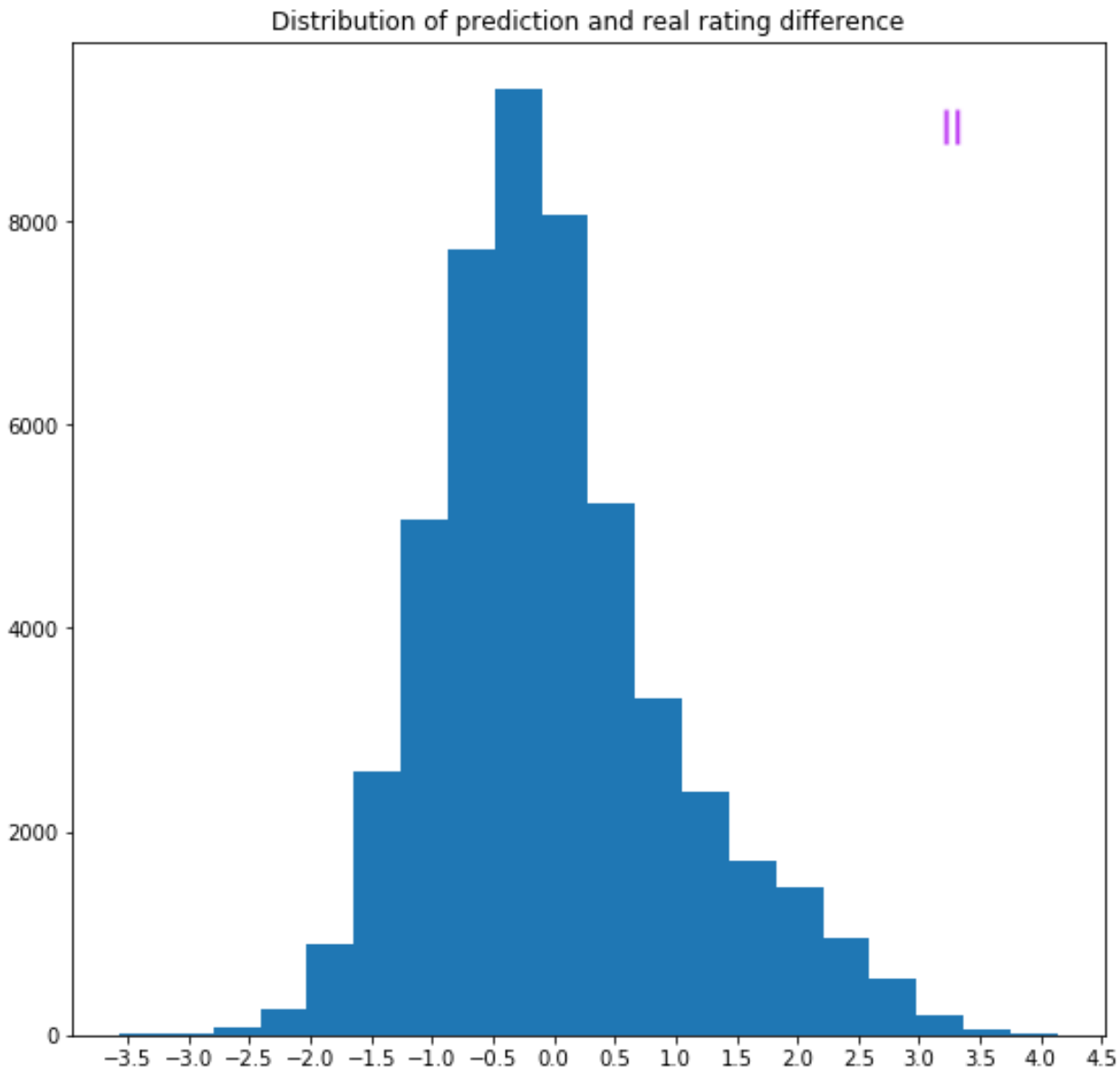
The histograms of the distribution of prediction-real rating difference suggest that in all categories, the most frequent case is that the rating difference is around -0.5 to 0.0.


Distribution of prediction and real rating difference
all
5000000
4000000
3000000
2000000
1000000
0
−4.5 −4.0 −3.5 −3.0 −2.5 −2.0 −1.5 −1.0 −0.5 0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5

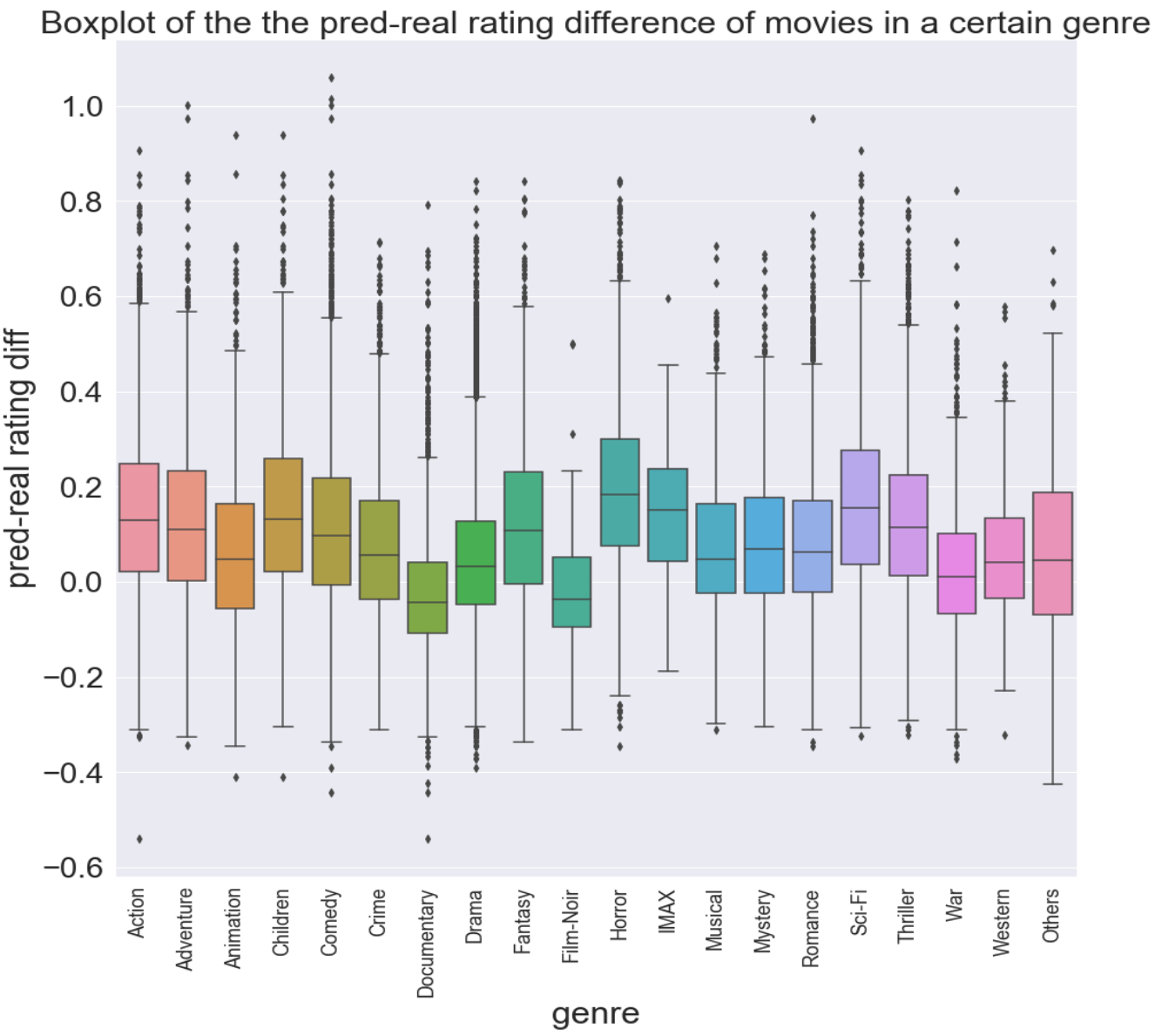

Distribution of prediction and real rating difference
ph
10000
8000
6000
4000
2000
0
−3.5
−3.0
−2.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5


Distribution of prediction and real rating difference
pl
8000
7000
6000
5000
4000
3000
2000
1000
0
−3.5
−3.0
−2.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5


Distribution of prediction and real rating difference
lh
8000
7000
6000
5000
4000
3000
2000
1000
0
−3.5
−3.0
−2.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0



Histograms the distribution of the prediction-real rating difference in each category



We provide the box plot of the difference of the average predicted and real ratings of movies in certain genres. It suggests that for almost every genre 75% of the median prediction error is within -0.1 and 0.3(well within the 0.5-point steps in the rating scores). It also suggests that the recommendation system tends to slightly overpredict the ratings. However, if we plot the scatter plot of the prediction-real difference and the rating record numbers of movies in the following graph, it is shown that movies with many large rating records (more than 50k) are not so overpredicted, they are slightly underpredicted. As movies have more ratings, the prediction error tends to decrease.

	title	rating numbers	average rate	average pred rate	pred-real rating diff	abs
108	Braveheart (1995)	59184	4.002273	4.008778	0.006505	0.006505
475	Jurassic Park (1993)	64144	3.679175	3.666863	-0.012312	0.012312

r

Two Popular Movies that have Low Prediction Error

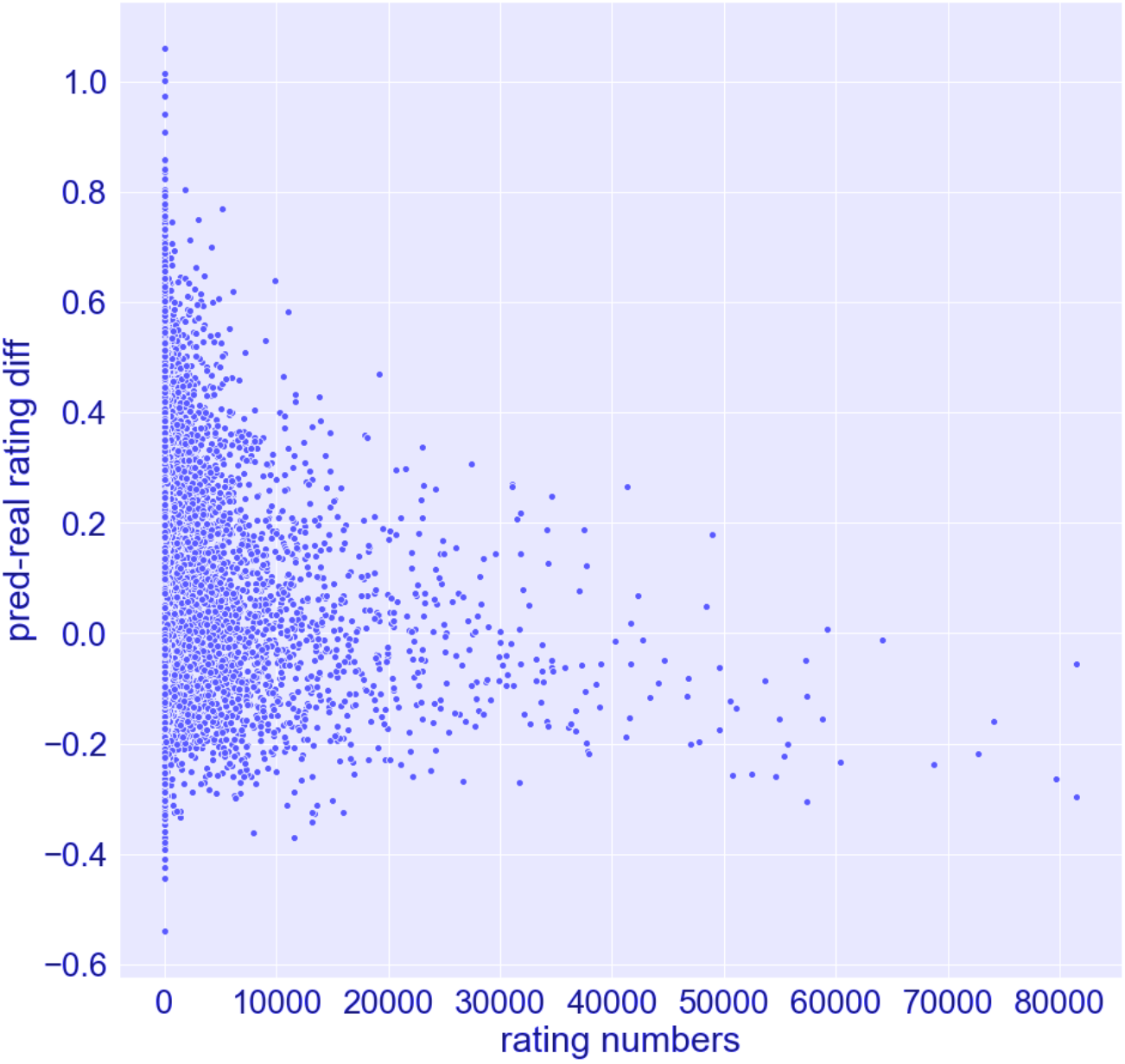
	title	rating numbers	average rate	average pred rate	pred-real rating diff
22668	What Men Do! (2013)	11	1.545455	2.605522	1.060068
19804	The Coed and the Zombie Stoner (2014)	12	1.416667	2.430434	1.013767

Two Most Overrated Movies

	title	rating numbers	average rate	average pred rate	pred-real rating diff
20670	BaadAsssss Cinema (2002)	10	4.15	3.610038	-0.539962
22274	George Carlin: What Am I Doing in New Jersey? ...	10	3.75	3.307036	-0.442964

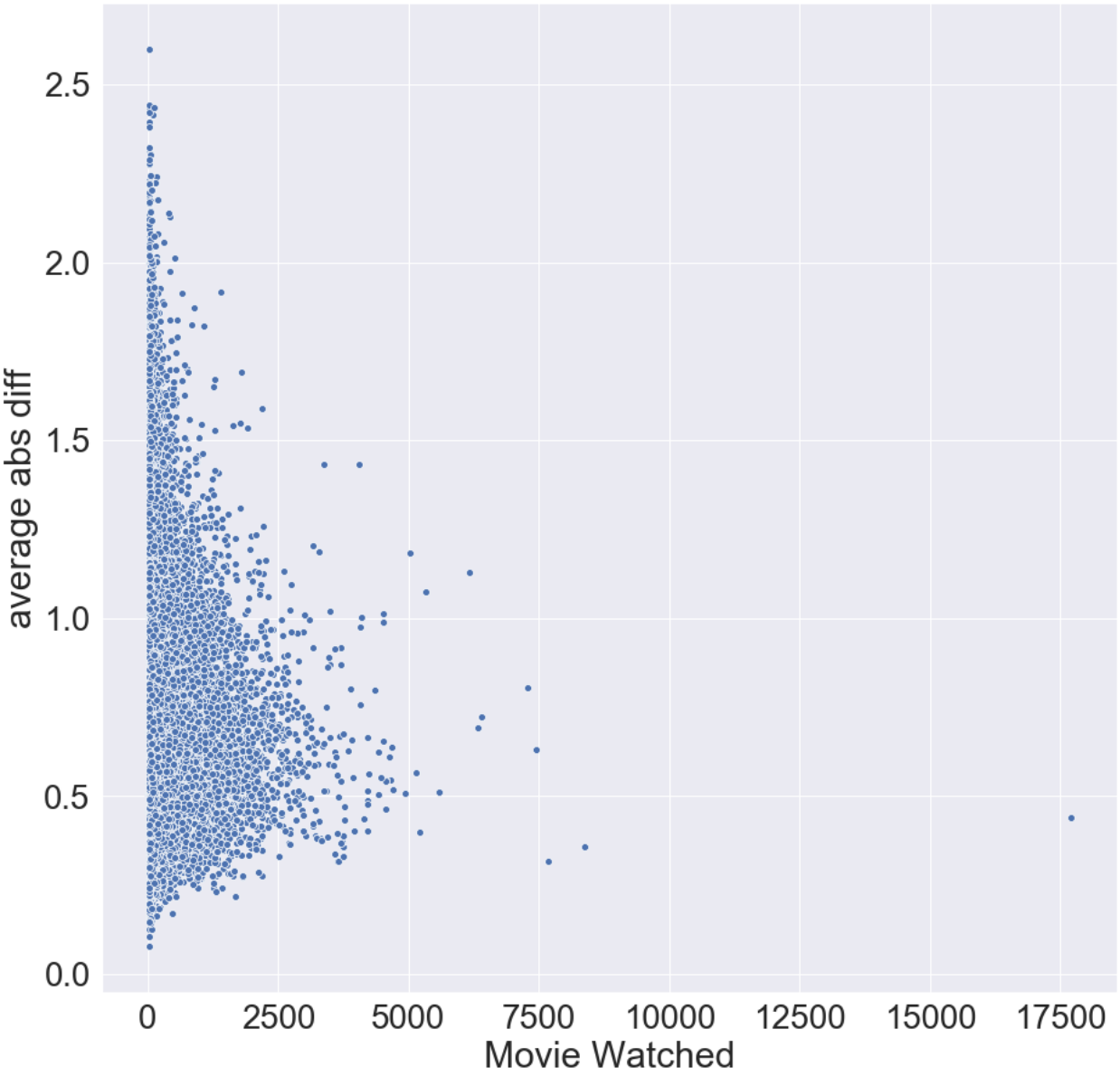
Two Most Underrated Movies

Scattered plot of pred-real rating difference vs. number of reviews of movies



The following scatter plot shows the relation between the MAE of ratings by a certain user and his total number of rated movies. It suggests that for a typical heavy user, the MAE of the recommendation system is around 0.6

Scattered plot of pred-real rating difference vs. number of movies watched by users



User Interaction

We built a function that if provided with a user id, it will give a recommendation list of the top N movies that the user is most likely to enjoy. It is also possible to recommend a movie to a list of potential interested users, but due to the large number of users causing long computing time, this function would not be very practical.

Conclusion

We build a movie recommendation system by collaborative-based method. Our study shows that the SVD algorithm can provide relatively better prediction within reasonable time consumption. It gives the MAE of 0.68. The overall statistics reveals that for movies with lesser reviewers, our model tends to overpredict, while for relatively popular movies, the model tends to slightly underpredict. It is also revealed that for almost every genre of movies, 75% of the median prediction error is within the range of -0.1 to 0.3. We can also conclude that, for a user who has a large number of review records, typically our recommendation system has the average error around 0.6.