PDS REPORT

Movie Recommendation System.

GROUP-01

Team Members :

NAME	ROLL NO.	EMAIL ID
BHAVESH KUMAR	S20190010034	bhaveshkumar.c19@iiits.in
BHARGHAV SAI P	S20190010017	bharghavsai.p19@iiits.in
CHETAN REDDY O	S20190010129	chetanreddy.o19@iiits.in
KHADYOTHAN	S20190010040	khadyothanchoudari.d19@iiits.in

DATE OF SUBMISSION: 28-04-2022

ABSTRACT AND MOTIVATION

Over the last decade, there has been a burgeoning of data due to social media, e-commerce and overall digitization of enterprises. The data is exploited to make informed choices, and predict marketplace trends and patterns in consumer preferences. Recommendation systems have become ubiquitous after the penetration of internet services among the masses. The idea is to make use of filtering and clustering techniques to suggest items of interest to users. For a media commodity like movies, suggestions are made to users by finding user profiles of individuals with similar tastes. Initially, user preference is obtained by letting them rate movies of their choice. Upon usage, the recommender system will be able to understand the user better and suggest movies that are more likely to be rated higher. The experiment results on the MovieLens dataset provides a reliable model which is precise and generates more personalized movie recommendations compared to other models. **So we are recommending the movie based on genre.**

PROCEDURE

- ➔ Dataset description
- ➔ Perform EDA
- ➔ Do visualization on most viewed movies, movie ratings, etc...
- ➔ Find similarities in people's ratings
- ➔ Interpret the people's reviews based on genre.

DATASET DESCRIPTION

There are two datasets given for this project. The name of the first dataset file is movies.csv which can be seen below.

Dataset Columns:

- movieId: unique number given to each movie.
- title : Title of the movie.
- genres : List genres given for each movie.

	movieId	title	genres
2	1	Toy Story	Adventure|Animation|Children|Comedy|Fantasy
3	2	Jumanji (1	Adventure|Children|Fantasy
4	3	Grumpier	Comedy|Romance
5	4	Waiting to	Comedy|Drama|Romance

The name of the second dataset file is ratings.csv which can be seen below.

Dataset Columns:

- userId: unique number given to each user.
- movieId : Title of the movie for which the particular user has given a review.
- rating : Rating given by the user.
- timestamp : Time at which the user has given the review.

	userId	movieId	rating	timestamp
2	1	16	4	1.22E+09
3	1	24	1.5	1.22E+09
4	1	32	4	1.22E+09
5	1	47	4	1.22E+09

PREPROCESSING

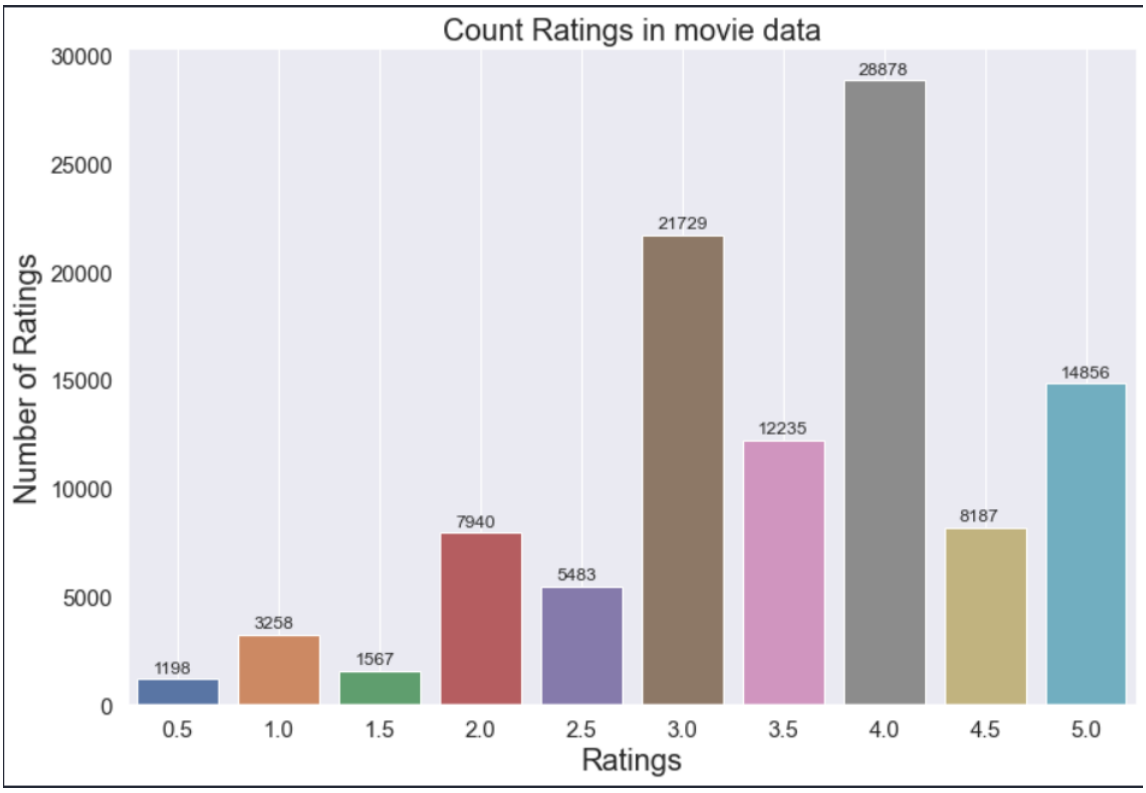
For movies.csv:

- The first preprocessing step we did was to create a separate column for year of release which was mentioned in the title itself for all movies. This step is done using regex.
- The second step was to extract genres given in the columns as python list from strings separated by ‘ | ’ using sets and regex expressions.
- The next step is to separate the list of genres given in the dataset to multiple columns through one hot encoding.
- The last step was to watch-out for movies with null attributes, fortunately there were only a few entries with null-values which gave us the option to safely drop them.

For ratings.csv:

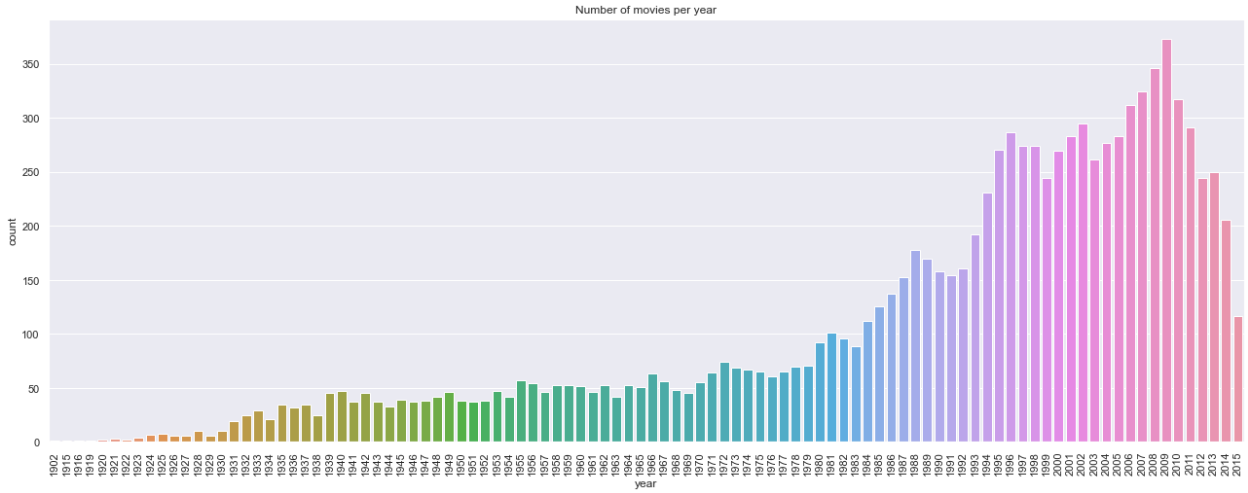
- The timestamp in the original data set was given as seconds count from Jan 1 st of 1970, which is not convenient for our analysis so we converted to pandas datetime object.
- The last step was to merge these two datasets to form one pandas dataframe.

Exploratory data analysis(EDA)

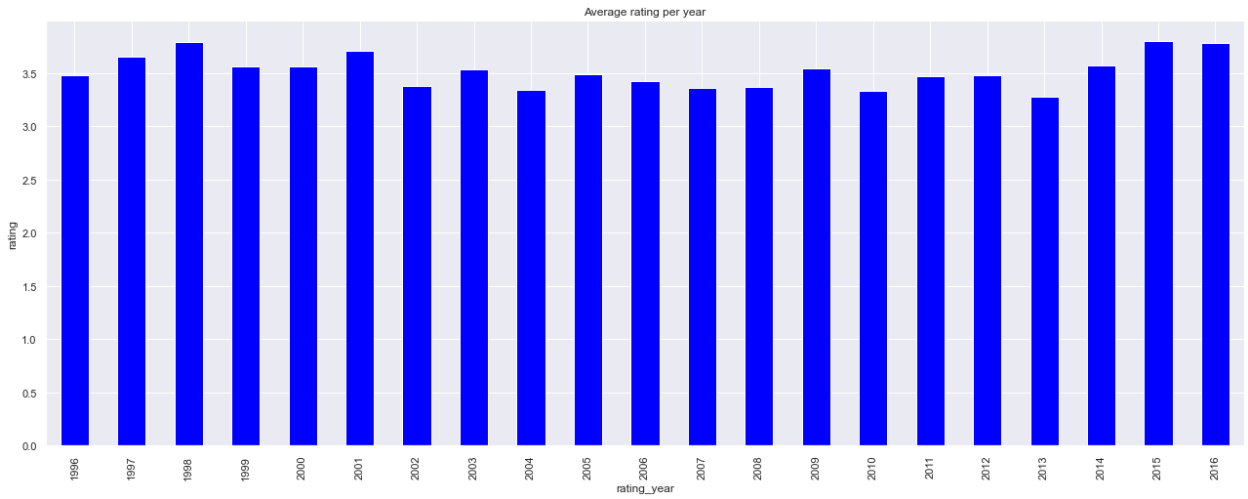


The plot is about finding the distribution of the count of the reviews given by the users.

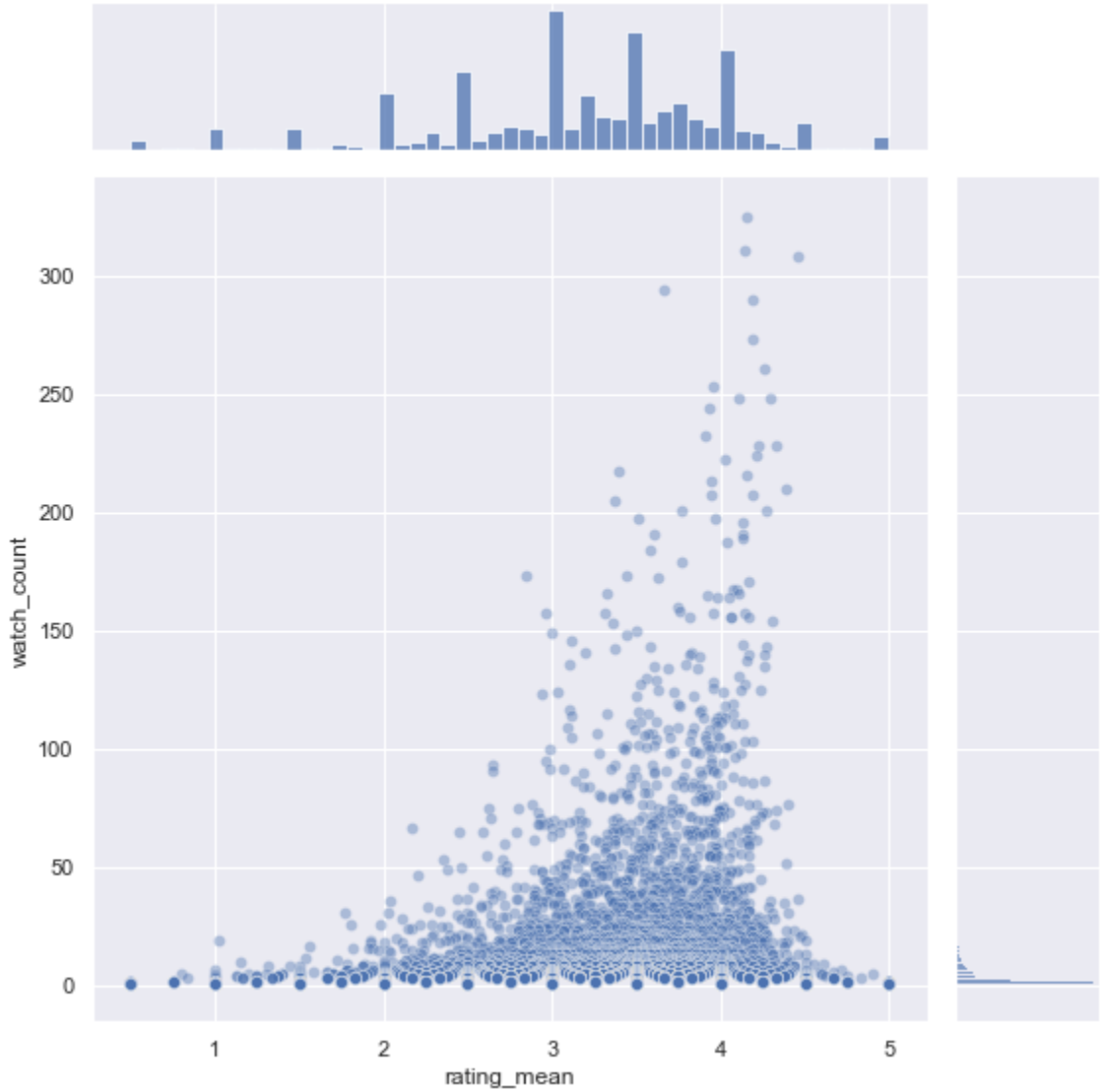
We can see that the highest count is for the rating 4.0 and the total count of that review is 28878. That means, around 28878 of the total ratings were 4.0. The least rating 1.0 has a count of 1198. Looking at this, we can almost approximate the average of the ratings would be around 3.5 to 4.0.



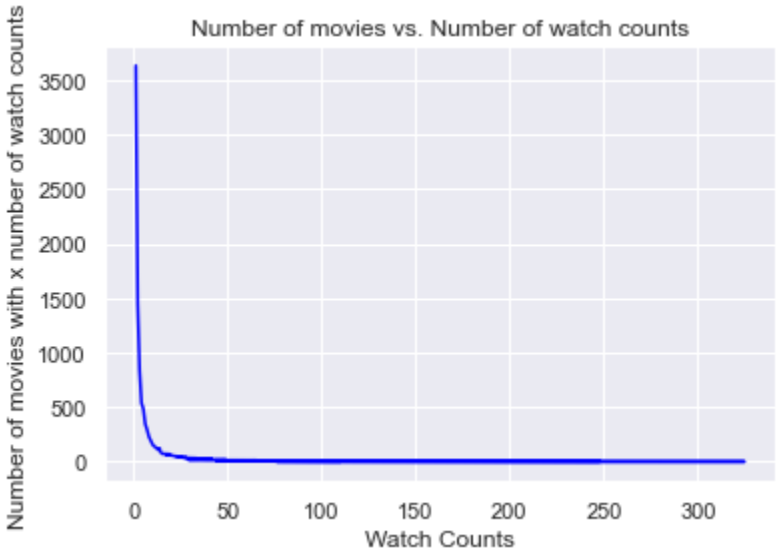
The above plot is for the total movies per year. The year 2009 witnessed the highest movie count. The plot is kind of increasing yearly and this is quite expected with the growth of the technology and the craze of the film industry. Also, one particularly interesting observation is the sudden decline in the movies count.



The above plot is for the average rating for that particular year of all movies. As mentioned above, most of the averages were around 3.5.

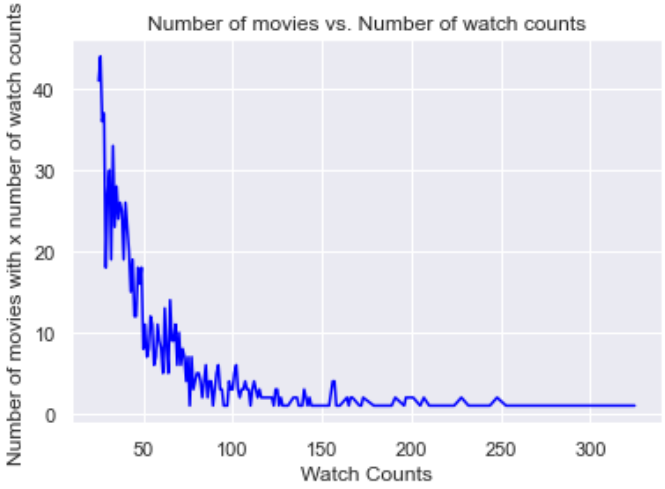


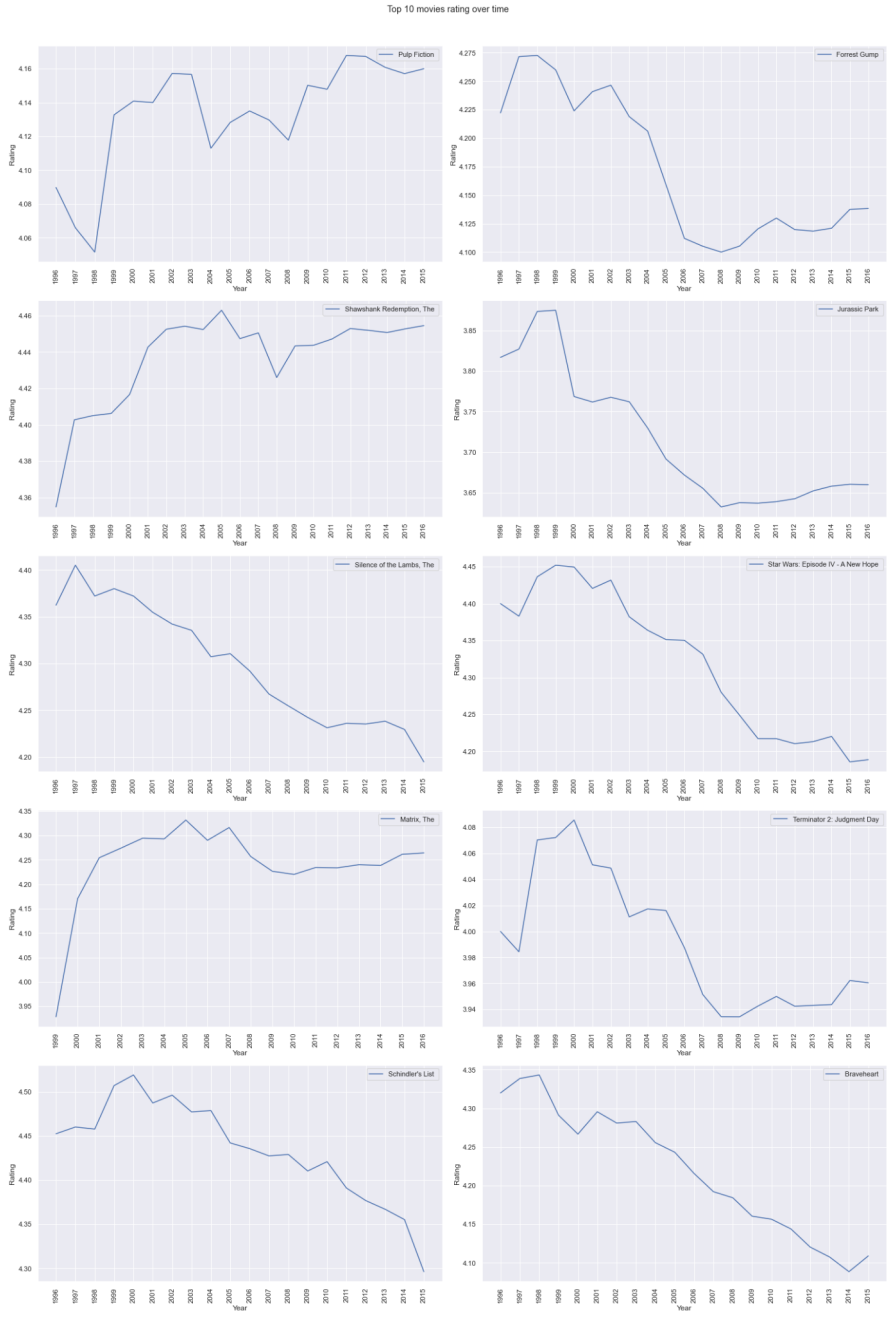
The above plot is to check the relationship between watch_count and rating_mean. The rating_mean is increasing with the increase in watch_count. This is expected as the movie with more watch count would typically be given a good rating.



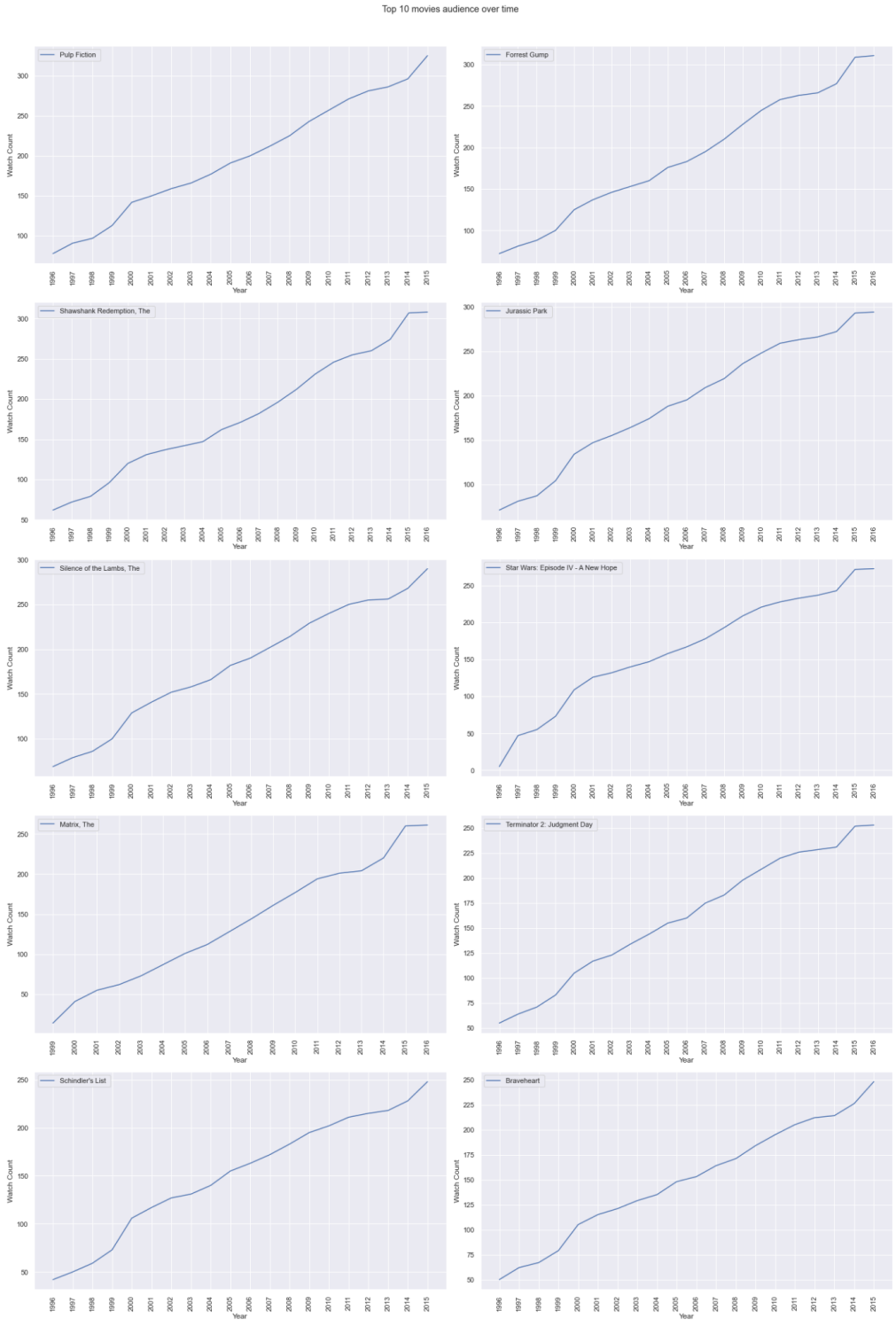
The above plot is to check the number of movies with various watch counts. We can see that almost all of the movies have a watch count between 1 and 100. There were few movies that had a watch count above 100 and 150.

The above plots and data analysis shows that there are over 3600 movies which are watched only once, 1455 movies that are watched only twice, and so on. This data confuses us in the rating average and needs to be removed. This distribution for refined data is shown below

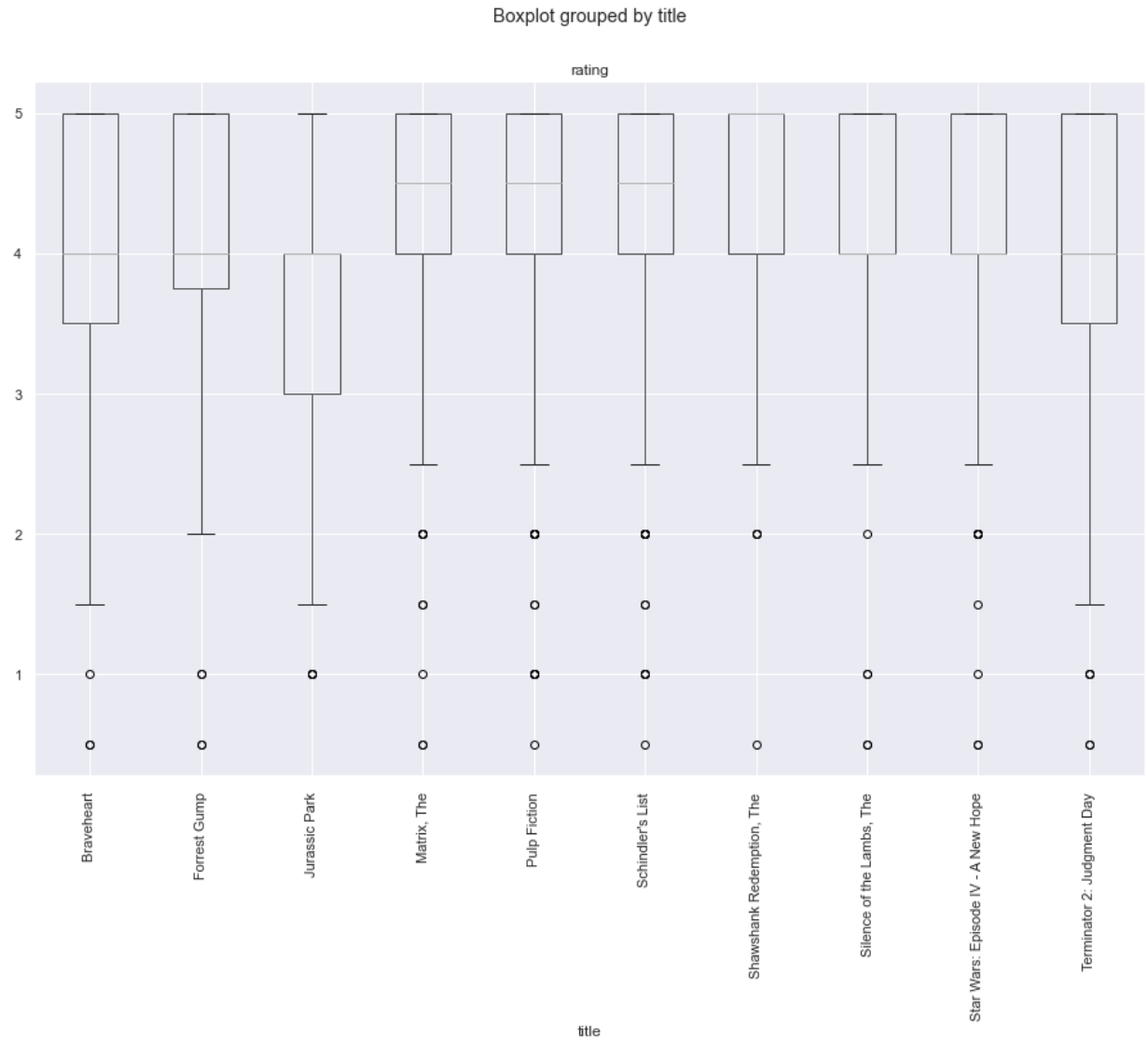




The above plot shows the trend of the top 10 movies over the years. Few movies have a rating that goes on increasing as the years pass by. But most of the movies have seen a decline in the rating over time.



The above plot shows the trend of watch count over years. It is quite obvious that the viewers would increase over time even though it is a bad movie.



The above plot shows the insights about the rating of the top 10 movies over all the years.

SIMILARITY ON PEOPLE'S RATING

We used cosine similarity to find the similarities between two people's ratings. First we make a pivot table where we gather the information of each user's rating on every movie. If there are any missing values we replace them with 0. And then apply cosine similarity between two users for every user existing in the database.

Similarities Found:

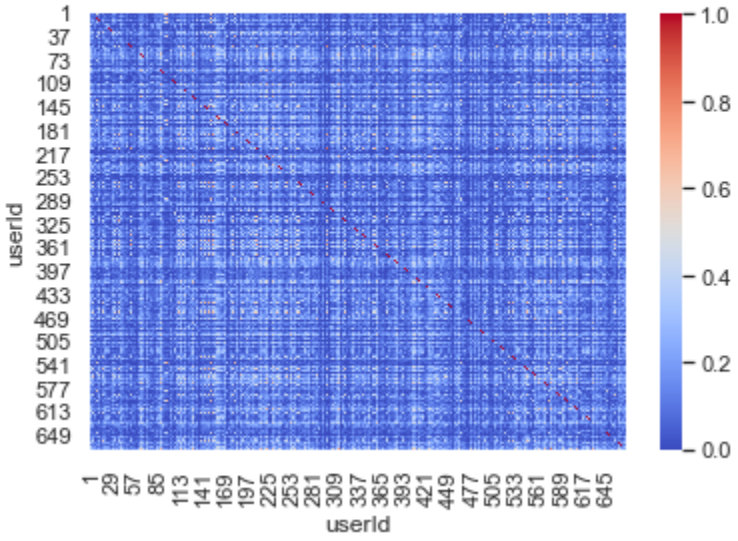
userId	1	2	3	4	5	6	7	8	9	10	...	659	660	661	662	663	664	665
userId																		
1	1.000000	0.105666	0.228638	0.172227	0.085252	0.087028	0.386759	0.113548	0.256933	0.031017	...	0.324944	0.171796	0.139104	0.113158	0.249760	0.182606	0.356979
2	0.105666	1.000000	0.127303	0.046849	0.048738	0.032321	0.068769	0.556589	0.211152	0.000000	...	0.077171	0.000000	0.590615	0.167260	0.174577	0.073837	0.075106
3	0.228638	0.127303	1.000000	0.082071	0.068961	0.015244	0.096234	0.081844	0.472026	0.083946	...	0.178917	0.345090	0.112303	0.120798	0.319497	0.098981	0.142529
4	0.172227	0.046849	0.082071	1.000000	0.036808	0.008449	0.082666	0.036863	0.070366	0.000000	...	0.080751	0.037405	0.068411	0.151783	0.104158	0.021447	0.080458
5	0.085252	0.048738	0.068961	0.036808	1.000000	0.085886	0.104342	0.069028	0.035489	0.019765	...	0.093023	0.012381	0.045986	0.067637	0.079842	0.082565	0.087450
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
664	0.182606	0.073837	0.098981	0.021447	0.082565	0.015105	0.256570	0.094409	0.121297	0.049906	...	0.193439	0.087925	0.088294	0.116773	0.113399	1.000000	0.168595
665	0.356979	0.075106	0.142529	0.080458	0.087450	0.208535	0.257539	0.055844	0.176217	0.042847	...	0.304883	0.102401	0.036841	0.091864	0.186251	0.168595	1.000000
666	0.304940	0.066495	0.109020	0.277486	0.044540	0.095608	0.231282	0.057553	0.111795	0.071911	...	0.191121	0.056308	0.049670	0.187189	0.103258	0.100253	0.170681
667	0.301087	0.039327	0.181326	0.086457	0.133749	0.037318	0.220762	0.145618	0.176389	0.037526	...	0.313925	0.143981	0.060631	0.118172	0.238436	0.113514	0.323301
668	0.351659	0.153252	0.255294	0.307264	0.143400	0.201363	0.324823	0.169327	0.259092	0.099043	...	0.458227	0.132090	0.203048	0.315275	0.177797	0.230451	0.317310

668 rows × 668 columns

Heatmap on similarities:

```
sns.heatmap(people_rating_similarity, cmap="coolwarm")
```

```
<AxesSubplot:xlabel='userId', ylabel='userId'>
```



RELATION BETWEEN PEOPLE'S REVIEWS AND GENRES

For this step we use two components

➔ Ordinary least squares regression

- In statistics, ordinary least squares (OLS) or linear least squares is a method for estimating the unknown parameters in a linear regression model.
- This method minimizes the sum of squared vertical distances between the observed responses in the dataset and the responses predicted by the linear approximation.

For our project we used Ordinary least squares regression to find the relationship between rating and all other features represented in the dataset. Assumptions are R^2 is computed without centering (uncentered) since the model does not contain a constant and Standard Errors assume that the covariance matrix of the errors is correctly specified. These regression parameters are used for calculating the part worth values in the next step.

➔ Conjoint analysis
- Conjoint analysis is a form of quantitative research.
- Respondents are asked to complete surveys with a number of product concepts which are presented in choice sets.
- Conjoint analysis is a survey-based statistical technique used in market research that helps determine how people value different attributes (feature, function, benefits) that make up an individual product or service.

OBSERVATIONS AND RESULTS

OLS REGRESSION RESULTS :

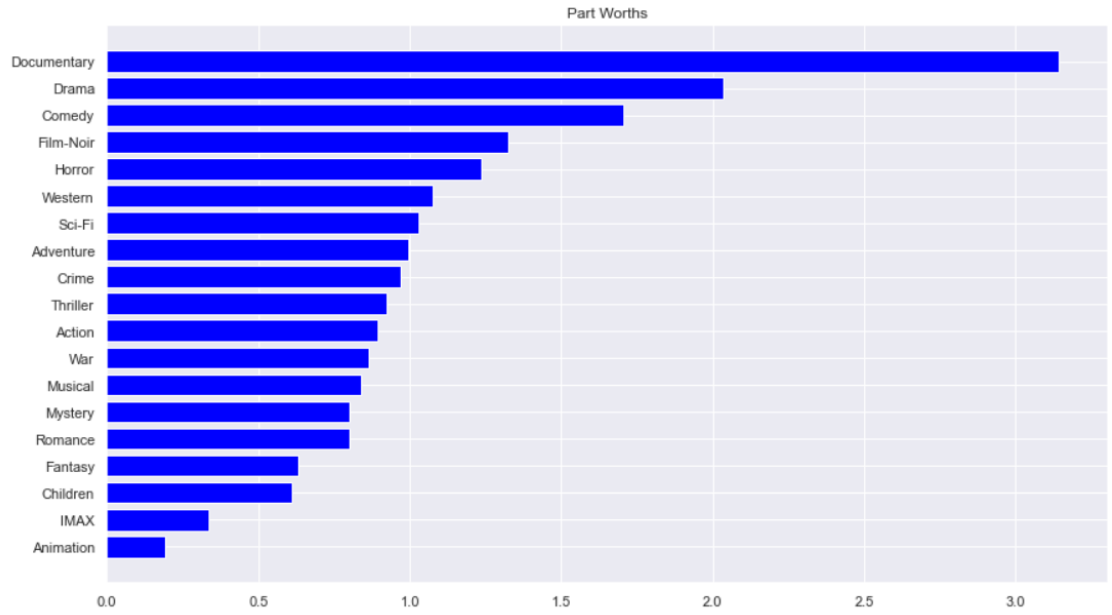
Dep. Variable:	rating	**R-squared (uncentered):**	0.839
Model:	OLS	**Adj. R-squared (uncentered):**	0.839
Method:	Least Squares	**F-statistic:**	1.777e+04
Date:	Thu, 28 Apr 2022	**Prob (F-statistic):**	0.00
Time:	22:36:46	**Log-Likelihood:**	-1.1938e+05
No. Observations:	64873	**AIC:**	2.388e+05
Df Residuals:	64854	**BIC:**	2.390e+05
Df Model:	19		
Covariance Type:	nonrobust		

OLS REGRESSION RESULTS :

	coef	std err	t	P>|t|	[0.025	0.975]
Action	0.8942	0.015	59.616	0.000	0.865	0.924
Adventure	0.9953	0.016	62.511	0.000	0.964	1.026
Animation	0.1926	0.037	5.191	0.000	0.120	0.265
Children	0.6103	0.033	18.234	0.000	0.545	0.676
Comedy	1.7052	0.012	142.789	0.000	1.682	1.729
Crime	0.9699	0.017	56.456	0.000	0.936	1.004
Documentary	3.1440	0.111	28.274	0.000	2.926	3.362
Drama	2.0352	0.011	178.085	0.000	2.013	2.058
Fantasy	0.6338	0.020	31.232	0.000	0.594	0.674
Film-Noir	1.3258	0.061	21.812	0.000	1.207	1.445
Horror	1.2383	0.025	49.521	0.000	1.189	1.287
IMAX	0.3380	0.035	9.751	0.000	0.270	0.406
Musical	0.8390	0.034	24.851	0.000	0.773	0.905
Mystery	0.8010	0.023	34.232	0.000	0.755	0.847
Romance	0.8009	0.016	48.656	0.000	0.769	0.833
Sci-Fi	1.0292	0.017	62.141	0.000	0.997	1.062
Thriller	0.9238	0.015	60.696	0.000	0.894	0.954
War	0.8636	0.027	32.584	0.000	0.812	0.916
Western	1.0741	0.040	27.186	0.000	0.997	1.151

Omnibus:	1402.430	**Durbin-Watson:**	1.658
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	1495.358
Skew:	-0.372	**Prob(JB):**	0.00
Kurtosis:	2.991	**Cond. No.**	17.2

PART WORTHS OF INDIVIDUAL GENRE:



CONCLUSION

“Users Gave good reviews for documentaries, Drama, comedy and didn't prefer Animation.” As we can see the individual part worths of some genre like Documentaries, Drama and comedy are higher compared to the other and also we can see that Animation has the least part worth which means many of the users didn’t prefer the animation genre.

REFERENCES

1. https://www.geeksforgeeks.org/ml-content-based-recommender-system/?ref=lbp
2. Subramaniyaswamy, V., Logesh, R., Chandrashekhar, M., Challa, A., & Vijayakumar, V. (2017). A personalized movie recommendation system based on collaborative filtering. International Journal of High-Performance Computing and Networking, 10(1-2), 54-63.
3. Reddy, S. R. S., Nalluri, S., Kunisetti, S., Ashok, S., & Venkatesh, B. (2019). Content-based movie recommendation system using genre correlation. In *Smart Intelligent Computing and Applications* (pp. 391-397). Springer, Singapore.
4. Wang, Z., Yu, X., Feng, N., & Wang, Z. (2014). An improved collaborative movie recommendation system using computational intelligence. *Journal of Visual Languages & Computing*, *25*(6), 667-675.
5. https://www.immagic.com/eLibrary/ARCHIVES/GENERAL/WIKIPEDI/W120529O.pdf