Second Dataset (MetaData)

The second dataset which is from Kaggle competition, provides 24 features and 45,466 samples. The information about each feature is provided below:

1. **adult:** Indicates if the movie is X-Rated or Adult.
2. **belongs_to_collection:** A stringified dictionary that gives information on the movie series the particular film belongs to.
3. **budget:** The budget of the movie in dollars.
4. **genres:** A stringified list of dictionaries that list out all the genres associated with the movie.
5. **homepage:** The Official Homepage of the move.
6. **id:** The ID of the move.
7. **imdb_id:** The IMDB ID of the movie.
8. **original_language:** The language in which the movie was originally shot in.
9. **original_title:** The original title of the movie.
10. **overview:** A brief blurb of the movie.
11. **popularity:** The Popularity Score assigned by TMDB.
12. **poster_path:** The URL of the poster image.
13. **production_companies:** A stringified list of production companies involved with the making of the movie.
14. **production_countries:** A stringified list of countries where the movie was shot/produced in.
15. **release_date:** Theatrical Release Date of the movie.
16. **revenue:** The total revenue of the movie in dollars.
17. **runtime:** The runtime of the movie in minutes.
18. **spoken_languages:** A stringified list of spoken languages in the film.
19. **status:** The status of the movie (Released, To Be Released, Announced, etc.)
20. **tagline:** The tagline of the movie.
21. **title:** The Official Title of the movie.
22. **video:** Indicates if there is a video presentation of the movie with TMDB.
23. **vote_average:** The average rating of the movie.
24. **vote_count:** The number of votes by users, as counted by TMDB.

Data Wrangling

For cleaning the data, I started with:

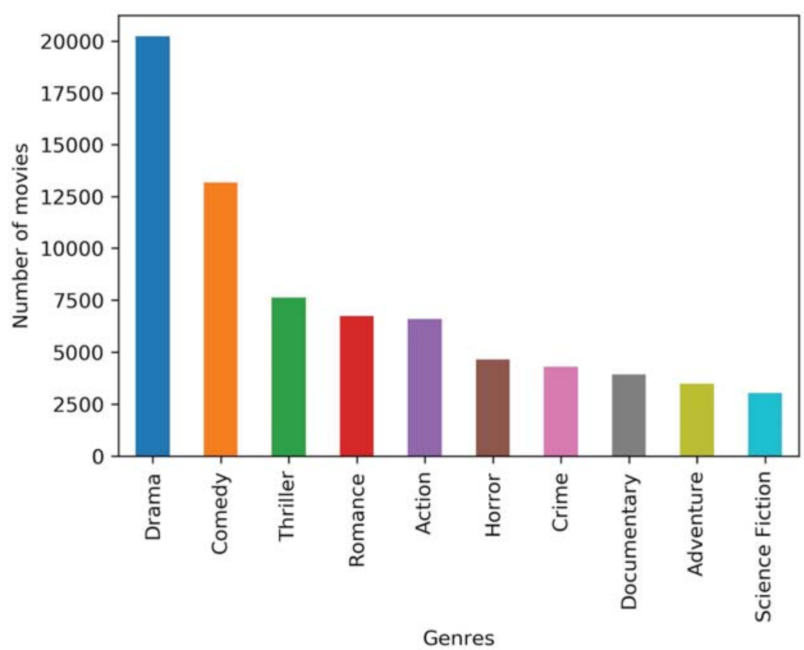
- dropping the duplicated rows.
- Just kept the rows with False and True in the adult column.
- I dropped the columns 'status' ,'adult', 'homepage', 'imdb_id', 'original_title', 'poster_path', 'tagline','video','spoken_languages', since they are not providing important information for this project.
- The ‘budget’ and ‘popularity’ columns have been changed to numeric.
- The ‘release_date’ has been changed to DateTime format.
- ‘year’ column is added for future use.
- I used the literal_eval, apply and lambda to change the format of the ‘production_companies’, ‘genres’, and ‘production_countries’ columns.

Please check the second ipython notebook.

Exploratory Data Analysis (EDA) and Storytelling

In this section, the various insights produced through descriptive statistics and data visualization is presented.

- Total number of movies of each genre:

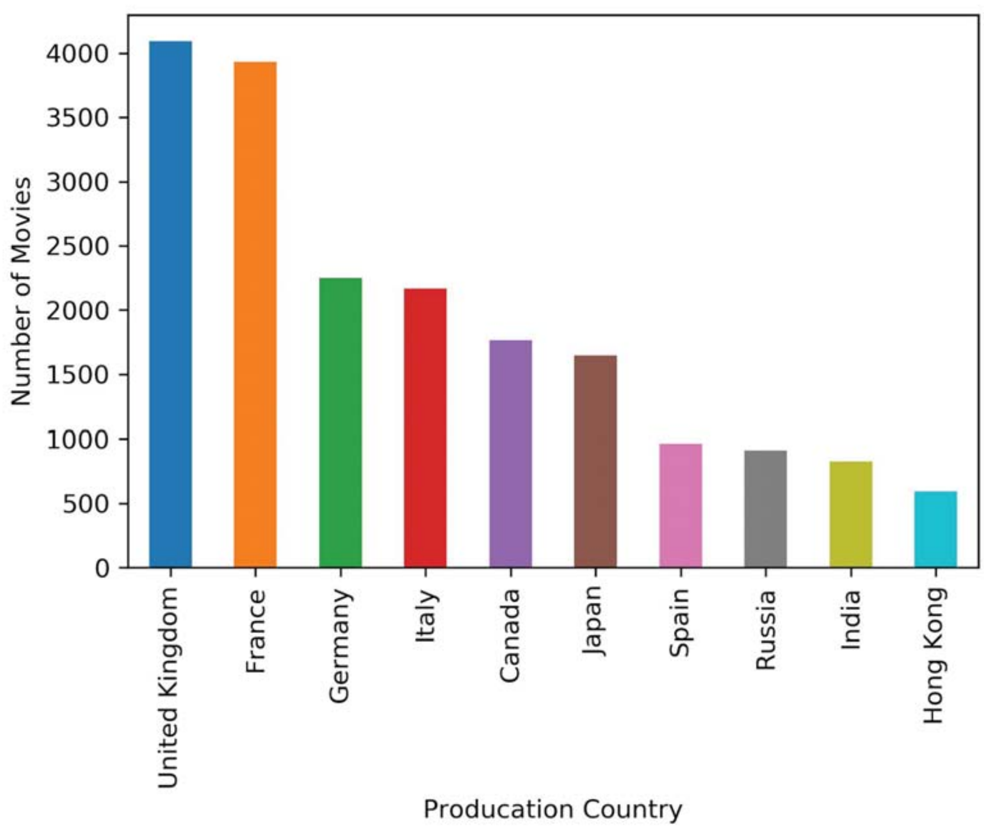


- Total production of each country is:

United States of America	21140
United Kingdom	4091
France	3932
Germany	2249
Italy	2166
Canada	1765
Japan	1645
Spain	964
Russia	912
India	827

To get a better bar plot, I have ignored the USA from the list, since the USA has produced 5 times more than the United Kingdom.

After the **USA, UK, France**, and **Germany** have the highest number of production.



- **Revenue**

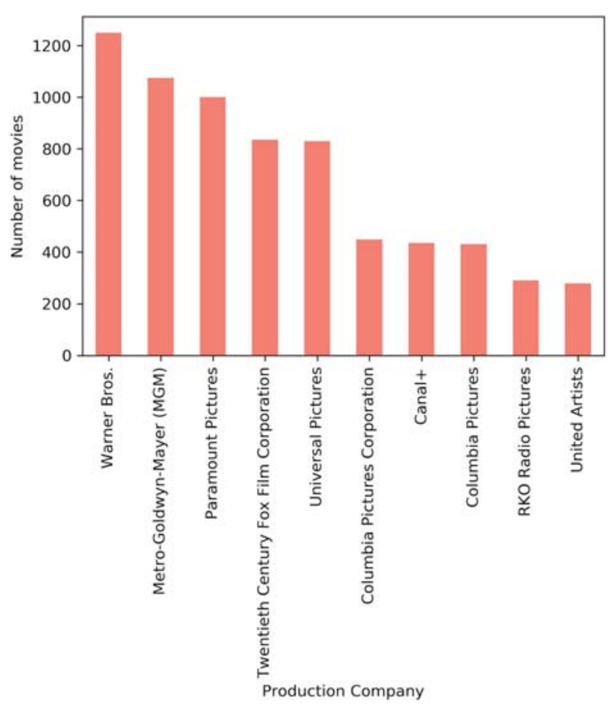
  USA average revenue is: 22,534,331.039

  UK average revenue is: 17,949,958.81

  France average revenue is: 5,156,112.68

- **Production companies**
  Since some movies have more than one production company, I used stack function to get the right value counts of each company.
  Warner Bros., Metro-Goldwyn-Mayer (MGM), and **Paramount Pictures** have the highest number of production.

Warner Bros.	1250
Metro-Goldwyn-Mayer (MGM)	1074
Paramount Pictures	1001
Twentieth Century Fox Film Corporation	836
Universal Pictures	830
Columbia Pictures Corporation	448
Canal+	436
Columbia Pictures	431
RKO Radio Pictures	290
United Artists	279



- **Warner Bros., Universal Pictures,** and **Paramount Pictures** have the highest total revenue respectively.

	Total	Average	Number
Warner Bros.	6.352519e+10	5.082015e+07	1250
Universal Pictures	5.525919e+10	6.657734e+07	830
Paramount Pictures	4.876940e+10	4.872068e+07	1001
Twentieth Century Fox Film Corporation	4.768775e+10	5.704276e+07	836
Walt Disney Pictures	4.083727e+10	1.552748e+08	263
Columbia Pictures	3.227974e+10	7.489498e+07	431
New Line Cinema	2.217339e+10	8.004834e+07	277
Amblin Entertainment	1.734372e+10	2.282068e+08	76
DreamWorks SKG	1.547575e+10	1.629027e+08	95
Dune Entertainment	1.500379e+10	2.308275e+08	65

- From the companies with a higher production number of 500, **Universal Pictures, Twentieth Century Fox Film Corporation,** and **Warner Bros**. have the highest average revenue respectively.

	Total	Average	Number
Universal Pictures	5.525919e+10	6.657734e+07	830
Twentieth Century Fox Film Corporation	4.768775e+10	5.704276e+07	836
Warner Bros.	6.352519e+10	5.082015e+07	1250
Paramount Pictures	4.876940e+10	4.872068e+07	1001
Metro-Goldwyn-Mayer (MGM)	1.237679e+10	1.152401e+07	1074

- Comparing revenue with budget

  5775 Movies have higher revenue than budget
  5108 Movies have lower revenue than budget

  34547 movies have equal revenue and budget

- **Avatar, Star Wars: The force awakens**, and **Titanic are** the movies with the maximum revenue.
- The movies with the highest revenue do not have the highest popularity and vote_average. Though the total number of votes for each movie is important.

title	popularity	vote_average	revenue
Avatar	185.070892	7.2	2.787965e+09
Star Wars: The Force Awakens	31.626013	7.5	2.068224e+09
Titanic	26.889070	7.5	1.845034e+09
The Avengers	89.887648	7.4	1.519558e+09
Jurassic World	32.790475	6.5	1.513529e+09
Furious 7	27.275687	7.3	1.506249e+09
Avengers: Age of Ultron	37.379420	7.3	1.405404e+09
Harry Potter and the Deathly Hallows: Part 2	24.990737	7.9	1.342000e+09
Frozen	24.248243	7.3	1.274219e+09
Beauty and the Beast	287.253654	6.8	1.262886e+09

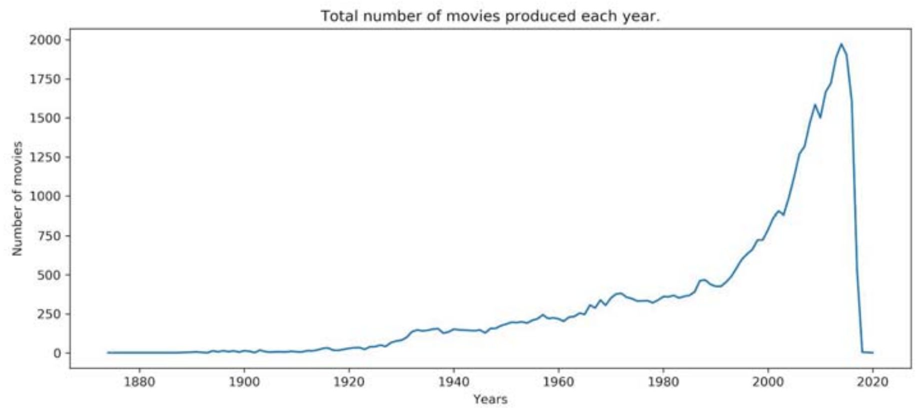
- The most voted movies are **Inception, The Dark Knight**, and **Avatar.**

title	vote_count	year
Inception	14075.0	2010.0
The Dark Knight	12269.0	2008.0
Avatar	12114.0	2009.0
The Avengers	12000.0	2012.0
Deadpool	11444.0	2016.0
Interstellar	11187.0	2014.0
Django Unchained	10297.0	2012.0
Guardians of the Galaxy	10014.0	2014.0
Fight Club	9678.0	1999.0
The Hunger Games	9634.0	2012.0

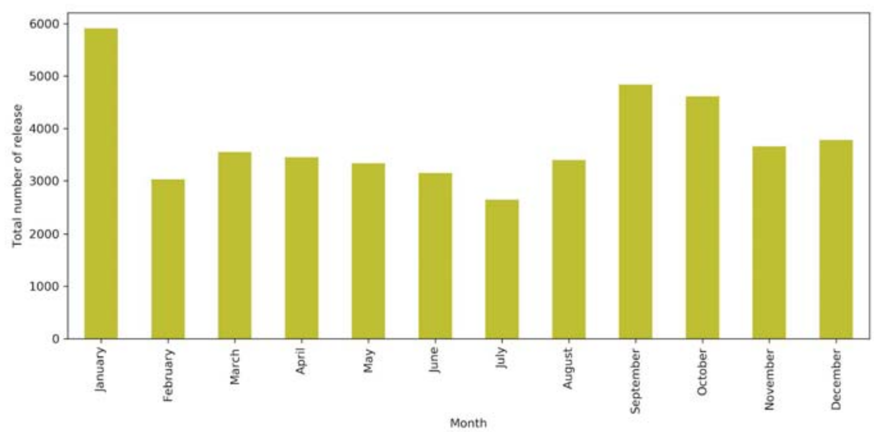
- The first three most popular movies are **Minions, Wonder Woman**, and **Beauty** and the beast.

title	popularity	vote_average	revenue
Minions	547.488298	6.4	1.156731e+09
Wonder Woman	294.337037	7.2	8.205804e+08
Beauty and the Beast	287.253654	6.8	1.262886e+09
Baby Driver	228.032744	7.2	2.245113e+08
Big Hero 6	213.849907	7.8	6.521054e+08
Deadpool	187.860492	7.4	7.831130e+08
Guardians of the Galaxy Vol. 2	185.330992	7.6	8.634161e+08
Avatar	185.070892	7.2	2.787965e+09
John Wick	183.870374	7.0	8.876166e+07
Gone Girl	154.801009	7.9	3.693304e+08

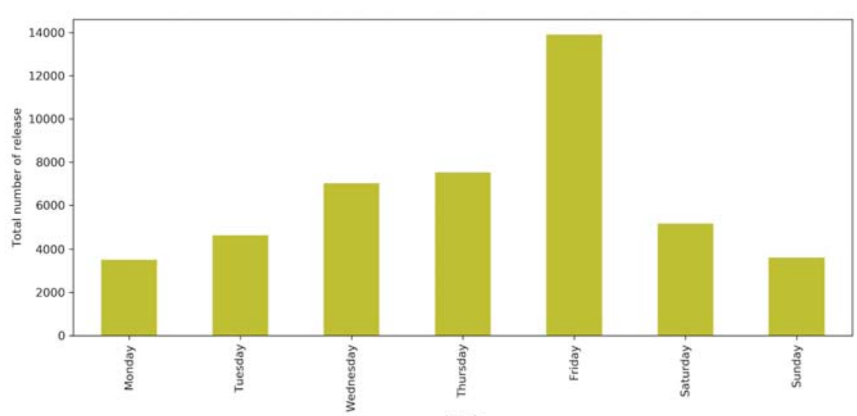
- Movies with more than 2000 votes and higher vote average are: **The Shawshank Redemption, Godfather, and Life is beautiful**.
- Plotting the total number of movies produced each year.



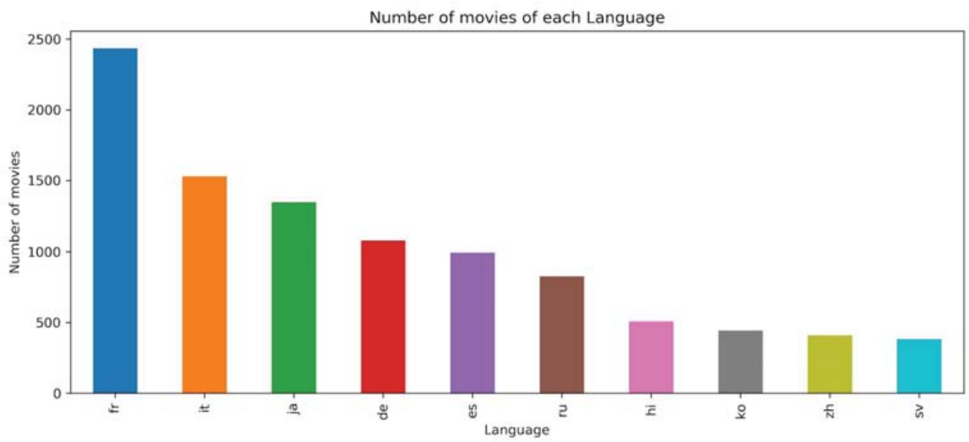
- Total number of movies release each month. (Most movies released in **January**)



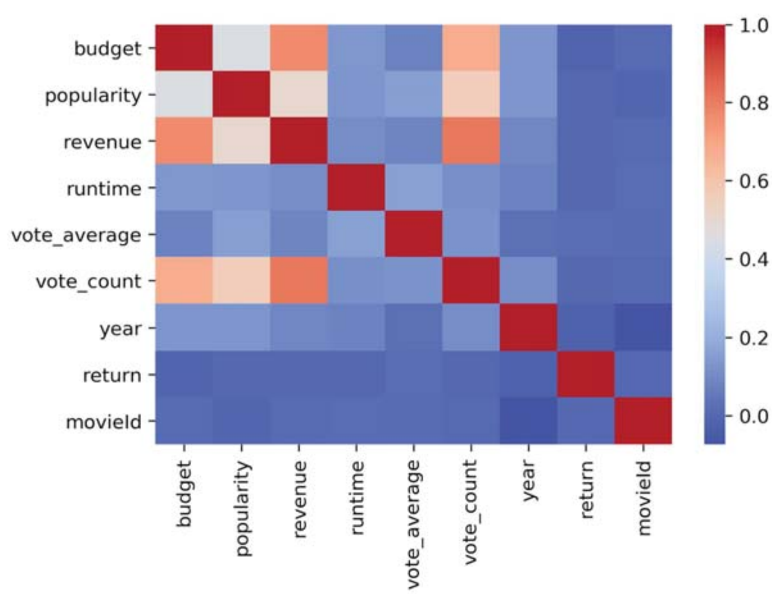
- Most movies are released on **Friday.** Most movies release on the first day of each month.



- **French, Italian, and Japanese** are the second, third and fourth languages.
- Total number of movies in **English** are: **45430**



- **Correlation**
  Budget and revenue are 77% correlated.
  Budget and vote_count are around 67% correlated.
  popularity and revenue are 50% correlated.
  popularity and vote_count are 56% correlated.
  Revenue and vote_count are 81% correlated.

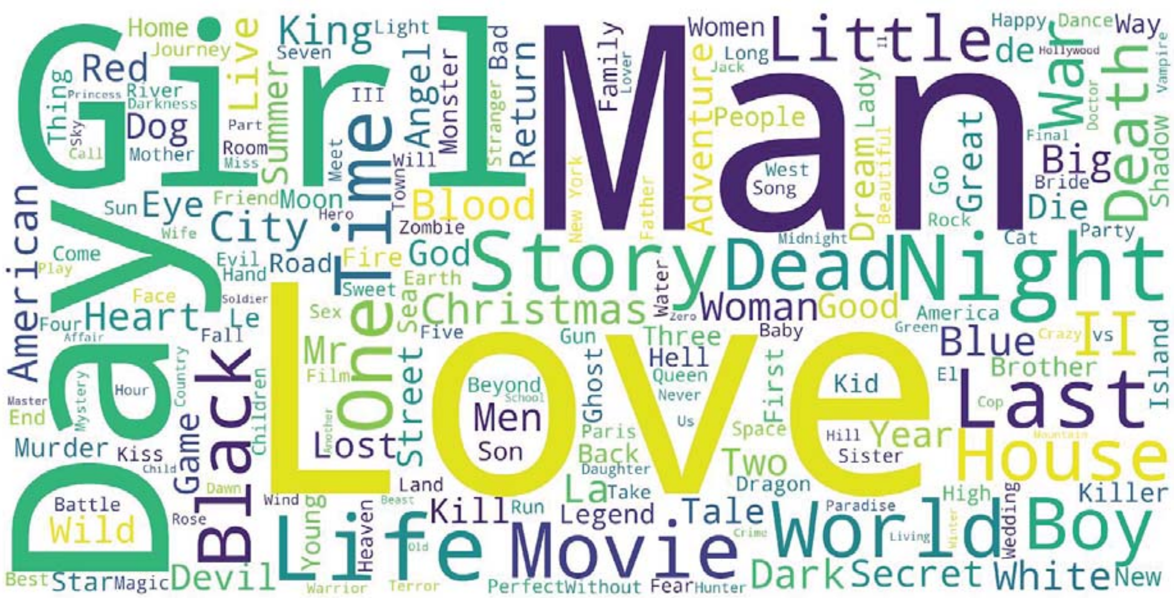


	budget	popularity	revenue	runtime	vote_average	vote_count	year	return	movieId
budget	1.000000	0.449682	0.768825	0.134700	0.073496	0.676699	0.131647	-0.012572	0.010009
popularity	0.449682	1.000000	0.506221	0.129912	0.154357	0.559995	0.131634	-0.003946	-0.009621
revenue	0.768825	0.506221	1.000000	0.103948	0.083883	0.812031	0.088358	-0.005515	0.007910
runtime	0.134700	0.129912	0.103948	1.000000	0.158192	0.113555	0.078714	-0.005189	0.012001
vote_average	0.073496	0.154357	0.083883	0.158192	1.000000	0.123611	0.025829	0.013161	0.015532
vote_count	0.676699	0.559995	0.812031	0.113555	0.123611	1.000000	0.106797	-0.003041	0.005362
year	0.131647	0.131634	0.088358	0.078714	0.025829	0.106797	1.000000	-0.024818	-0.073434
return	-0.012572	-0.003946	-0.005515	-0.005189	0.013161	-0.003041	-0.024818	1.000000	0.000810
movieId	0.010009	-0.009621	0.007910	0.012001	0.015532	0.005362	-0.073434	0.000810	1.000000

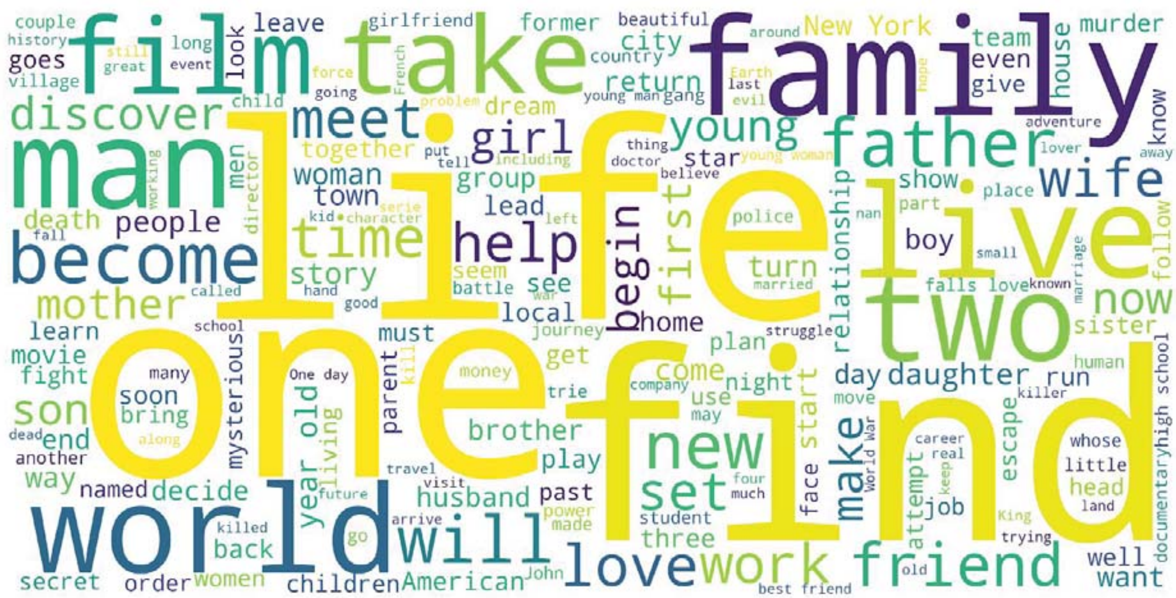
- **Title and Overview Word Clouds**

  There are certain words that use more often in titles and overviews. I use WordCloud library to find out what are these words.

  The word **Love** is the most commonly used word in movie titles. **Girl**, **Day** and **Man** are also among the most commonly occurring words.



Life is the most commonly used word in Movie titles. **One** and **Find** are also popular.

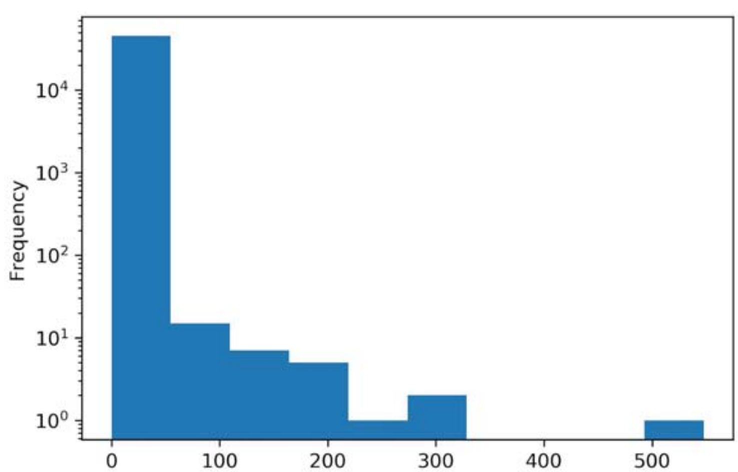


Statistics

Popularity

The Popularity score seems to be an extremely skewed quantity with a mean of only 2.9 but maximum values reaching as high as 547, which is almost 1800% greater than the mean. However, as can be seen from the distribution plot, almost all movies have a popularity score of less than 10 (the 75th percentile is at 3.678902).

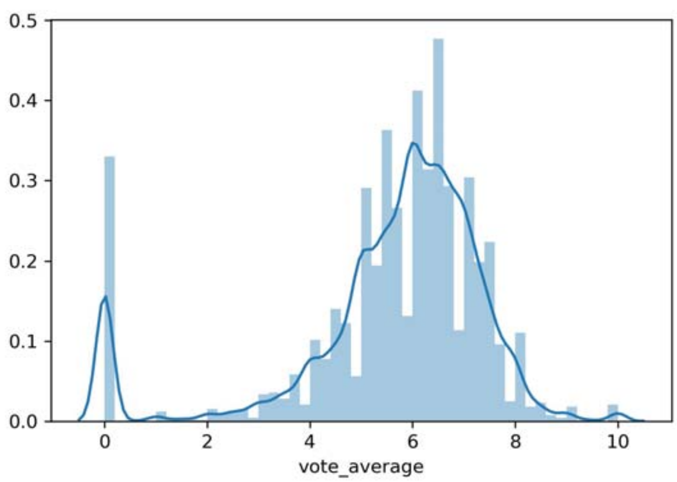
Minions is the most popular movie by the TMDB Popularity Score. **Wonder Woman** and **Beauty and the Beast**, two extremely successful woman-centric movies come in second and third respectively.



	title	popularity	year
30700	Minions	547.488298	2015
33356	Wonder Woman	294.337037	2017
42222	Beauty and the Beast	287.253654	2017
43644	Baby Driver	228.032744	2017
24455	Big Hero 6	213.849907	2014
26564	Deadpool	187.860492	2016
26566	Guardians of the Galaxy Vol. 2	185.330992	2017
14551	Avatar	185.070892	2009
24351	John Wick	183.870374	2014
23675	Gone Girl	154.801009	2014

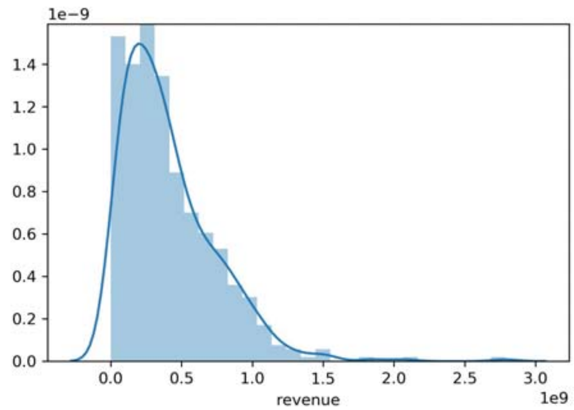
Vote average¶

The mean rating is only a 5.6 on a scale of 10. Half the movies have a rating of less than or equal to 6.



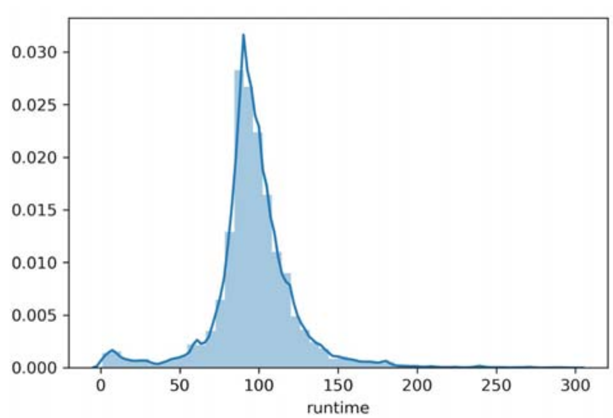
Revenue¶

The revenue statistics for movies with more than 2000 votes. The distribution plot is skewed to the right.



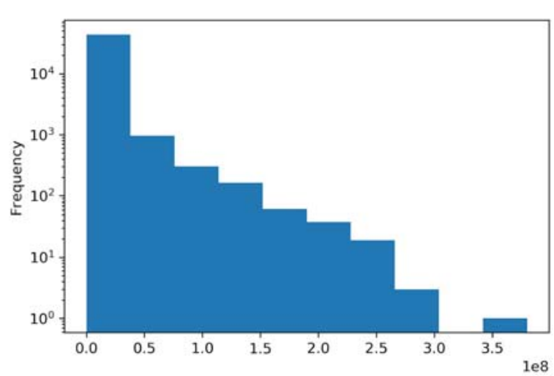
Runtime¶

The average length of a movie is about 1 hour and 30 minutes. The longest movie on record in this dataset is a staggering 1256 minutes (or 20 hours) long.



Budget¶

The distribution of movie budgets shows an exponential decay. More than 75% of the movies have a budget smaller than 25 million dollars.



Shortest and longest movies¶

- We can see there is several one minute length movies.

runtime	title	year
1.0	Mr. Edison at Work in His Chemical Laboratory	1897.0
1.0	Grandma's Reading Glass	1900.0
1.0	What Happened on Twenty-Third Street, New York...	1901.0
1.0	The Magician	1898.0
1.0	Panorama pris d'un train en marche	1898.0
1.0	Divers at Work on the Wreck of the "Maine"	1898.0
1.0	After the Ball	1897.0
1.0	Between Calais and Dover	1897.0
1.0	The Surrender of Tournavos	1897.0
1.0	Blacksmith Scene	1893.0

- All these long time films are TV series and we don't have access to the runtime of each episode in this dataset.

runtime	title	year
1256.0	Centennial	1978.0
1140.0	Jazz	2001.0
1140.0	Baseball	1994.0
931.0	Berlin Alexanderplatz	1980.0
925.0	Heimat: A Chronicle of Germany	1984.0
900.0	The Story of Film: An Odyssey	2011.0
877.0	Taken	2002.0
874.0	The War	2007.0
840.0	The Roosevelts: An Intimate History	2014.0
840.0	Seventeen Moments in Spring	1973.0

Recommender Systems

1. Simple Recommender

The Simple Recommender offers generalized recommendations to every user based on movie popularity, ratings, number of votes and genre. The basic idea behind this recommender is that movies that are more popular and more critically acclaimed will have a higher probability of being liked by the average audience. This model does not give personalized recommendations based on the user.

So I sort the movies based on their genre, and I recommend the ones with the highest number of votes and the higher average rankings to the new audience.

- The list of the **15 best Drama** movies to recommend are:

0 The Dark Knight
1 Interstellar
2 Fight Club
3 The Shawshank Redemption
4 Forrest Gump
5 The Godfather
6 The Intouchables
7 Schindler's List
8 Whiplash
9 Leon: The Professional
10 The Green Mile
11 Life Is Beautiful
12 The Godfather: Part II
13 The Usual Suspects
14 GoodFellas

- The list of the **15 best Romance** movies to recommend are:

0 Forrest Gump
1 Titanic
2 La La Land
3 Her
4 The Great Gatsby
5 The Fault in Our Stars
6 Eternal Sunshine of the Spotless Mind
7 Edward Scissorhands

8 Aladdin
9 Amélie
10 The Theory of Everything
11 The Curious Case of Benjamin Button
12 The Notebook
13 A Beautiful Mind
14 The Perks of Being a Wallflower

- The list of the **15 best Action** movies to recommend are:

0 Inception
1 The Dark Knight
2 Avatar
3 The Avengers
4 Deadpool
5 Guardians of the Galaxy
6 Mad Max: Fury Road
7 The Dark Knight Rises
8 The Matrix
9 Iron Man
10 The Lord of the Rings: The Fellowship of the Ring
11 The Lord of the Rings: The Return of the King
12 Star Wars: The Force Awakens
13 The Lord of the Rings: The Two Towers
14 Batman Begins

- The list of the **15 best Fantasy** movies to recommend are:

0 Avatar
1 The Lord of the Rings: The Fellowship of the Ring
2 The Lord of the Rings: The Return of the King
3 Star Wars: The Force Awakens
4 The Lord of the Rings: The Two Towers
5 Pirates of the Caribbean: The Curse of the Bla...
6 Harry Potter and the Philosopher's Stone
7 X-Men: Days of Future Past
8 Harry Potter and the Deathly Hallows: Part 2
9 Harry Potter and the Prisoner of Azkaban
10 Harry Potter and the Chamber of Secrets

11	Doctor Strange
12	Harry Potter and the Goblet of Fire
13	Harry Potter and the Deathly Hallows: Part 1
14	Harry Potter and the Order of the Phoenix

2. IMDB Weighted Rating Formula

Another technique could be IMDB's **weighted rating** formula which is mathematically represented as follows:

$$WR = \frac{v}{v+m}R + \frac{m}{v+m}C$$

where,

v is the number of votes for the movie

m is the minimum votes required to be listed in the chart

R is the average rating of the movie

C is the mean vote across the whole report

The next step is to determine an appropriate value for m, the minimum votes required to be listed in the chart. We will use **95th percentile** as our cutoff. In other words, for a movie to feature in the charts, it must have more votes than at least 95% of the movies in the list.

- mean vote across the whole report (C): 5.245
- minimum votes required to be listed in the chart (m): 434.0
- 2181 movies are qualified

The **best 30 movies** to recommend base on IMDB weighting are:

15480	Inception
12481	The Dark Knight
22879	Interstellar
2843	Fight Club
4863	The Lord of the Rings: The Fellowship of the Ring
292	Pulp Fiction
314	The Shawshank Redemption
7000	The Lord of the Rings: The Return of the King
351	Forrest Gump
5814	The Lord of the Rings: The Two Towers
256	Star Wars
1225	Back to the Future
834	The Godfather
1154	The Empire Strikes Back
46	Se7en

24860	The Imitation Game
359	The Lion King
18465	The Intouchables
22841	The Grand Budapest Hotel
586	The Silence of the Lambs
11354	The Prestige
522	Schindler's List
23673	Whiplash
289	Leon: The Professional
4099	Memento
3030	The Green Mile
5481	Spirited Away
1213	The Shining
1057	Reservoir Dogs
2211	Life Is Beautiful

3. Matrix Factorization-based algorithms

The idea is basically to take a large (or potentially huge) matrix and factor it into some smaller representation of the original matrix. You can think of it in the same way as we would take a large number and factor it into two much smaller primes. We end up with two or more lower dimensional matrices whose product equals the original one.

When we talk about collaborative filtering for recommender systems we want to solve the problem of our original matrix having millions of different dimensions, but our “tastes” not being nearly as complex. Even if i’ve viewed hundreds of items they might just express a couple of different tastes. Here we can actually use matrix factorization to mathematically reduce the dimensionality of our original “all users by all items” matrix into something much smaller that represents “all items by some taste dimensions” and “all users by some taste dimensions”. These dimensions are called latent or hidden features and we learn them from our data.

Doing this reduction and working with fewer dimensions makes it both much more computationally efficient and but also gives us better results since we can reason about items in this more compact “taste space”.

If we can express each user as a vector of their taste values, and at the same time express each item as a vector of what tastes they represent. You can see we can quite easily make a recommendation. This also gives us the ability to find connections between users who have no specific items in common but share common tastes.

SVD is a matrix factorization technique that is usually used to reduce the number of features of a data set by reducing space dimensions from N to K where K < N. For the purpose of the recommendation systems however, we are only interested in the matrix factorization part keeping same dimensionality. The matrix factorization is done on the user-item ratings matrix. From a high level, matrix factorization can be thought of as finding 2 matrices whose product is the original matrix.

(https://medium.com/@m_n_malaeb/singular-value-decomposition-svd-in-recommender-systems-for-non-math-statistics-programming-4a622de653e9)

I used **Surprise** library which has several powerful algorithms like **Singular Value Decomposition (SVD)** to minimise RMSE (Root Mean Square Error) and give recommendations.

In the first part, the **vote_average is predicted from the vote_average and the popularity.**

Evaluating RMSE, MAE of algorithm **SVD** on 5 split(s).

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE (testset)	1.5982	1.5953	1.5983	1.6038	1.6091	1.6009	0.0049
MAE (testset)	1.3276	1.3299	1.3291	1.3335	1.3344	1.3309	0.0026
Fit time	9.44	4.75	5.50	4.78	6.24	6.14	1.74
Test time	0.19	0.09	0.09	0.09	0.19	0.13	0.05

Total Elapsed time with the model is: 33.275503158569336

- In the second part, the **vote_average is predicted from the vote_count and the revenue.**

Evaluating RMSE, MAE of algorithm SVD on 5 split(s).

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean	Std
RMSE (testset)	1.6023	1.5977	1.6068	1.6030	1.5844	1.5988	0.0078
MAE (testset)	1.3284	1.3255	1.3387	1.3321	1.3228	1.3295	0.0055
Fit time	5.28	4.37	4.08	5.88	4.09	4.74	0.72
Test time	0.08	0.08	0.08	0.08	0.08	0.08	0.00

Total Elapsed time with the model is: 24.629876613616943

Machine Learning

In this section, I applied different machine learning techniques to find the best model with the lowest RMSE.

4. Random Forest

Random forest is a type of supervised machine learning algorithm based on ensemble learning. The random forest algorithm combines multiple algorithms of the same type i.e. multiple decision trees, resulting in a forest of trees, hence the name "Random Forest".

I check random forest regressor to find the vote_average base on some features 'id', 'popularity', 'movieId', 'vote_count', 'revenue', 'budget'.

I have checked a different number of estimators to get the minimum RMSE.

Total Elapsed time with the model is: 23.899521589279175
RMSE value for n_estimator= 100 is: 1.230501506418753

Total Elapsed time with the model is: 32.894049882888794
RMSE value for n_estimator= 110 is: 1.2297851910704445

Total Elapsed time with the model is: 34.64072108268738
RMSE value for n_estimator= 120 is: 1.2291184320901682

Total Elapsed time with the model is: 36.02364206314087
RMSE value for n_estimator= 130 is: 1.2285653671442673

Total Elapsed time with the model is: 36.642759799957275
RMSE value for n_estimator= 140 is: 1.228152851883824

Total Elapsed time with the model is: 42.90695285797119
RMSE value for n_estimator= 150 is: 1.2278656497690312

Total Elapsed time with the model is: 43.093995809555054
RMSE value for n_estimator= 160 is: 1.2273173943356 89

Total Elapsed time with the model is: 48.45515489578247
RMSE value for n_estimator= 170 is: 1.2268291900261832

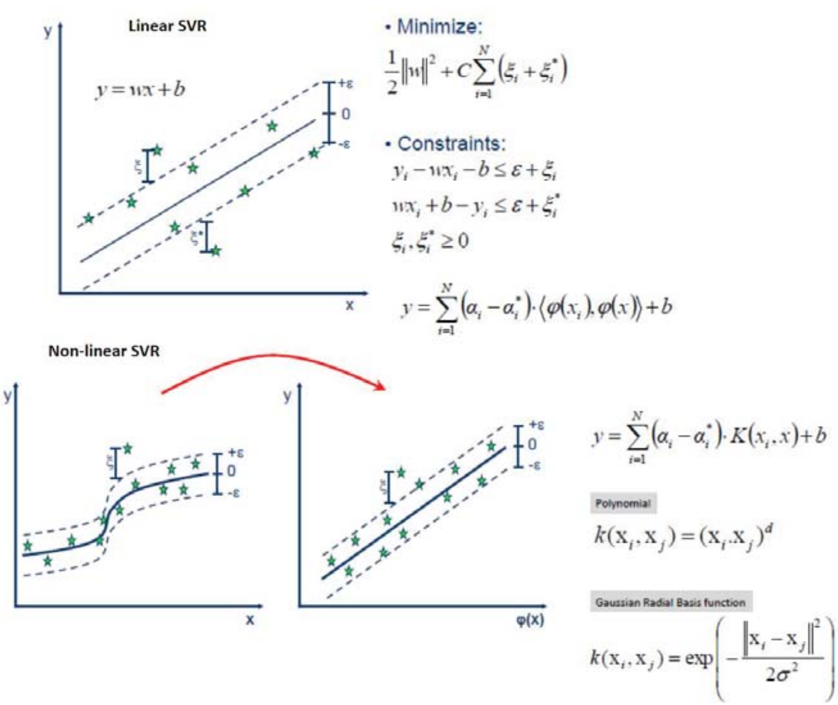
Total Elapsed time with the model is: 47.292954206466675
RMSE value for n_estimator= 180 is: 1.2270316755432094

Total Elapsed time with the model is: 53.971041202545166
RMSE value for n_estimator= 190 is: 1.22688515828762

The minimum RMSE **(1.268)** is achieved with n_estimator=**170**.

5. Support Vector Regression

Support Vector Machine can also be used as a regression method, maintaining all the main features that characterize the algorithm (maximal margin). The Support Vector Regression (SVR) uses the same principles as the SVM for classification, with only a few minor differences. First of all, because output is a real number it becomes very difficult to predict the information at hand, which has infinite possibilities. In the case of regression, a margin of tolerance (epsilon) is set in approximation to the SVM which would have already requested from the problem. But besides this fact, there is also a more complicated reason, the algorithm is more complicated therefore to be taken in consideration. However, the main idea is always the same: to minimize error, individualizing the hyperplane which maximizes the margin, keeping in mind that part of the error is tolerated.



Total Elapsed time with model is: 84.37661409378052
SVR RMSE (gamma=.01, C=100) is: 1.8604387694736897

Total Elapsed time with model is: 68.05094337463379
SVR RMSE(gamma=0.001, C=100) is: 1.8819087974081985

Total Elapsed time with model is: 63.12529253959656
SVR RMSE(gamma=0.001, C=10) is: 1.888942925254787

As it can see from the results, SVR is slow and the RMSE is not as small as random forest.

6. K-Nearest Neighbor

KNN is fast but the RMSE is not as small as random forest.

```
RMSE value for k= 11 is: 1.8038207834268334
RMSE value for k= 13 is: 1.7934131089891299
RMSE value for k= 15 is: 1.78986557123273
RMSE value for k= 17 is: 1.786579572380617
RMSE value for k= 19 is: 1.783226039661783
```

Conclusion

This report highlighted the processes of data wrangling, storytelling, EDA, data visualization, inferential statistics, recommender systems and machine learning techniques to perform on two movie datasets.

In this work, several recommended systems are implemented as:

1. Simple Recommender used the overall Vote Count and Vote Averages to build Top Movies Charts, in general and for a specific genre.

2. IMDB Weighted Rating system was used to calculate ratings on which the sorting was finally performed and recommend the best 30 movies of the datasets.

3. Content Based Recommender was built to recommend based on the user's rating history.

4. Collaborative Filtering in a very simple format could recommend based on other user's rating histories. It could also be base on matrix factorization techniques like SVD or ALS.

5. Machine learning: Random Forest is a fast and good technique to predict when we have several features in our dataset. Support Vector Machines are strong machine learning tools for classification and regression however tuning its parameters need optimization techniques and the training process is slow.

Which technique is the best to utilize for the stakeholder?

Picking the best technique for utilizing is not very easy. There is not one techniques suitable for all data sets. But As it is shown in the previous sections, When we don't have many features (Movielens dataset), SVD could be the easiest and fastest techniques to recommend.

For datasets with several features (MetaData dataset) we can predict the rating or popularity of movies based with different machine learning techniques. Random Forest worked fine and fast for predicting the target for this dataset.

Future Work

For future, other recommender techniques like Hybrid techniques or Deep learning methods could be implemented. Also A/B testing technique could be combined with previous techniques to improve the result.