Rounak Banik
IIT Roorkee
rounakbanik@gmail.com
+91 84398 60325

Movies Final Report

24th October 2017

OVERVIEW

This project is divided into two parts:

1. **The Story of Film:** This section aims at narrating the history, trivia and facts behind the world of cinema through the lens of data. Extensive Exploratory Data Analysis is performed on Movie Metadata about Movie Revenues, Casts, Crews, Budgets, etc. through the years. Two predictive models are built to predict movie revenues and movie success. Through these models, we also aim at discovering what features have the most significant impact in determining revenue and success.
2. **Movie Recommender Systems:** This part is focused around building various kinds of recommendation engines; namely the Simple Generic Recommender, the Content Based Filter and the User Based Collaborative Filter. The performance of the systems are evaluated in both a qualitative and quantitative manner.

THE CLIENT

The first section of the project does not have a definitive client. But some of the analysis performed could be of use to anyone in the Movie Making Business (Streaming Providers, Producers, etc). The Movie Success and Revenue Prediction Models can give valuable insights into the features that actually determine the end class and value respectively.

The Movie Recommender System is useful to any business that makes money via recommendations. This includes Amazon, Netflix, Hotstar, etc. Giving good recommendations directly entails one or many of the following:

1. Customers buy a particular product or service leading to increased revenue or sales.
2. Customers use the platform more frequently due to the quality and relevance of content shown to them.
3. Better User Experience. Customers spend less time searching and more time watching. The pain of discovery is eliminated.

THE DATA

The data used in this project has been obtained from two sources: The Movie Database (TMDB) and MovieLens.

MovieLens has a publicly available dataset that contains 26 million ratings and 750,000 tag applications applied to 45,000 movies by 270,000 users. It also includes tag genome data with 12 million relevance scores across 1,100 tags. A small subset of this dataset, containing 10,000 ratings for 9000 movies from 700 users is also available.

One of the files contains the TMDB ID of every movie listed in the MovieLens dataset. Using this ID, the metadata, credits and keywords of all 45,000 movies were obtained by running a script that requested and parsed data from TMDB Open API. The data collected was initially in the JSON format but was converted into CSV files using Python's Pandas Library.

The following files were used in the project:

1. **movies_metadata.csv:** The file containing metadata collected from TMDB for over 45,000 movies. Data includes budget, revenue, date released, genres, etc.
2. **credits.csv:** Complete information on credits for a particular movie. Data includes Director, Producer, Actors, Characters, etc.
3. **keywords.csv:** Contains plot keywords associated with a movie.
4. **links_small.csv:** Contains the list of movies that are included in the small subset of the Full MovieLens Dataset.
5. **Ratings_small.csv:** The MovieLens Dataset containing 100,000 ratings on 9,000 movies from 700 users. The main dataset used for building the Collaborative Filter.

DATA COLLECTION

The MovieLens Full Dataset was readily available at the GroupLens Website (https://grouplens.org/datasets/movielens/). This dataset contained 26 million ratings from 270,000 users on 45,000 movies. One file in this dataset, links.csv, contained the TMDB and IMDB IDs for all the movies.

I signed up for an API Key with TMDB. This gave me access to data at 3 endpoints. Each endpoint gave me details about the movie, its cast and crew information and plot keywords. I wrote 3 separate scrapers to hit each endpoint and collect this data for all 45,000 movies. Since TMDB has a restriction of 40 requests every 10 seconds, this task took a day to execute.

All the data collected was in the form of stringified JSON which demanded more processing.

DATA WRANGLING

Overview

This section describes the various data cleaning and data wrangling methods applied on the Movie datasets to make it more suitable for further analysis. The following sections are divided based on the procedures followed.

Conversion to CSV Files

The data obtained from scraping was in the form of stringified JSON. This had to be converted into CSV Files to enable easier parsing and subsequent upload to public platforms such as Kaggle.

Removing Unnecessary Features

Some features such as the Backdrop Path, Adult and IMDB ID were unnecessary attributes and were dropped to reduce the dimensions of the dataset.

Cleaning

The dataset had a lot of features which had 0s for values it did not possess. These values were converted to NaN. Some features were still in the form of a Stringified JSON Object. They were converted into Python Dictionaries using Python's ast library. These were further reduced into lists since we did not have a need for ID, timestamp and other attributes.

The dataframe was exploded wherever the analysis demanded it (for instance, genres and production countries).

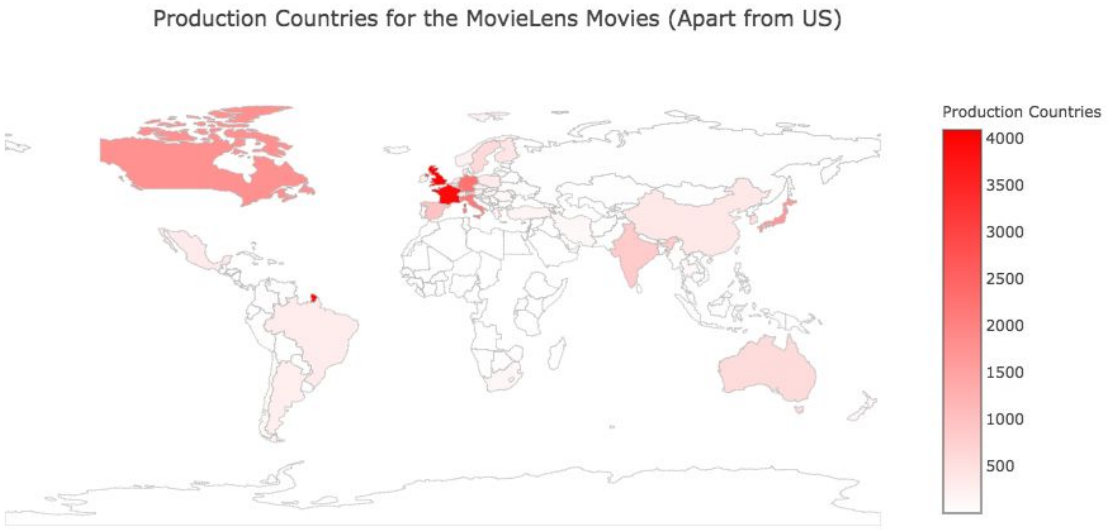
Finally, most of the features were converted into a Python basic type (integer, string, float) by removing all the unclean values. The date string was converted into a Pandas Datetime and from it, we extracted the month, year and day of release of every movie.

EXPLORATORY DATA VISUALIZATION AND ANALYSIS

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

This forms the crux of the first section of my Capstone Project.

Production Countries



1. The Movies in the dataset are overwhelmingly in the English Language and shot in the United States of America.
2. Europe is also an extremely popular location with the UK, France, Germany and Italy in the top five.
3. Japan and India are the most popular Asian countries when it comes to movie production.

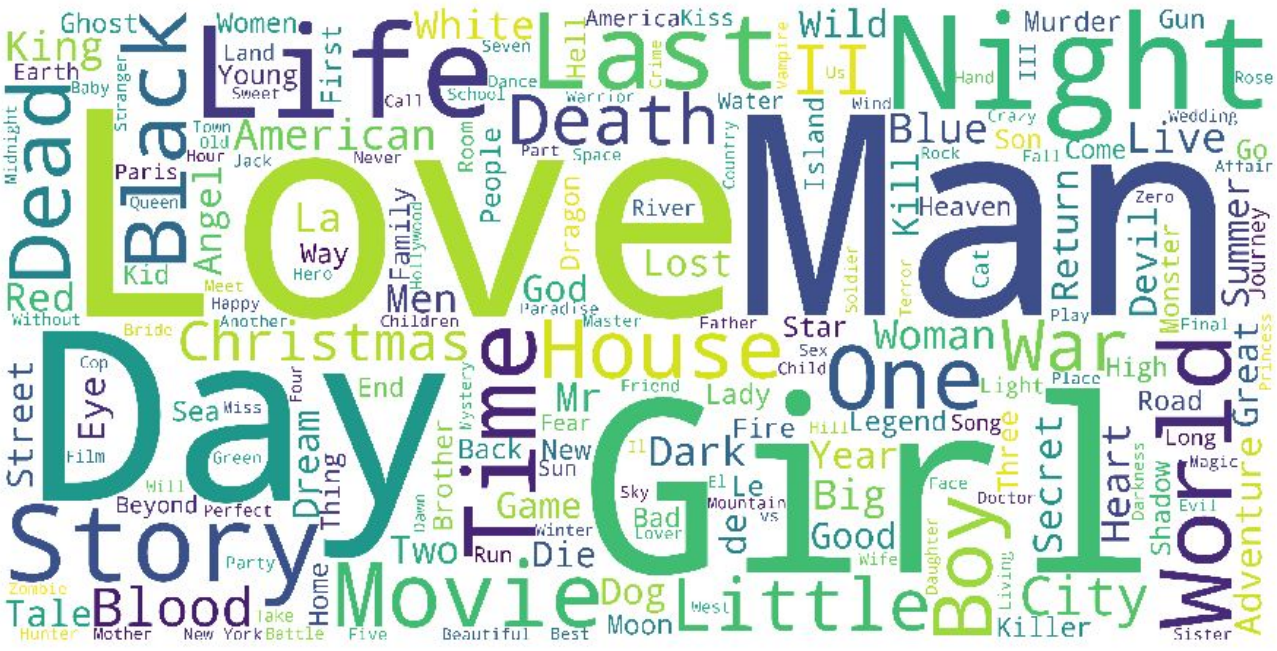
Franchise Movies

1. The Harry Potter Franchise is the most successful movie franchise raking in more than 7.707 billion dollars from 8 movies. The Star Wars Movies come in a close second with a 7.403 billion dollars from 8 movies too.
2. The Avatar Collection, although just consisting of one movie at the moment, is the most successful franchise of all time with the sole movie raking in close to 3 billion dollars. The Harry Potter franchise is still the most successful franchise with at least 5 movies.
3. The James Bond Movies is the largest franchise ever with over 26 movies released under the banner. Friday the 13th and Pokemon come in at a distant second and third with 12 and 11 movies respectively.

Production Companies

1. Warner Bros is the highest earning production company of all time earning a staggering 63.5 billion dollars from close to 500 movies. Universal Pictures and Paramount Pictures are the second and the third highest earning companies with 55 billion dollars and 48 billion dollars in revenue respectively.
2. Pixar Animation Studios has produced the most successful movies, on average. This is not surprising considering the amazing array of movies that it has produced in the last few decades: Up, Finding Nemo, Inside Out, Wall-E, Ratatouille, the Toy Story Franchise, Cars Franchise, etc. Marvel Studios with an average gross of 615 million dollars comes in second with movies such as Iron Man and The Avengers under its banner.

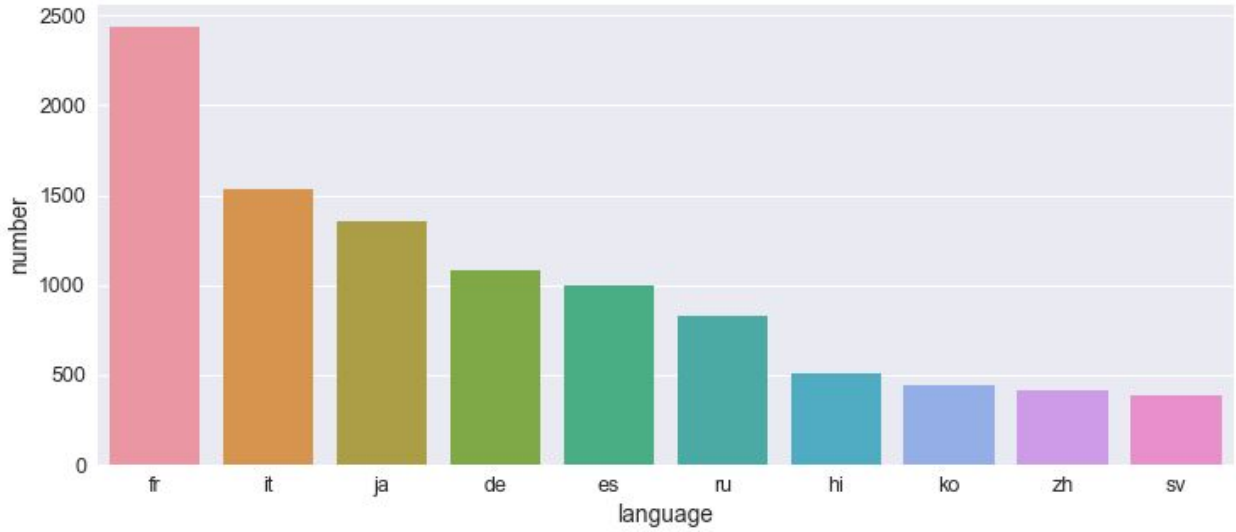
Movie Title Wordcloud



The word Love is the most commonly used word in movie titles. Girl, Day and Man are also among the most commonly occurring words. I think this encapsulates the idea of the ubiquitous presence of romance in movies pretty well.

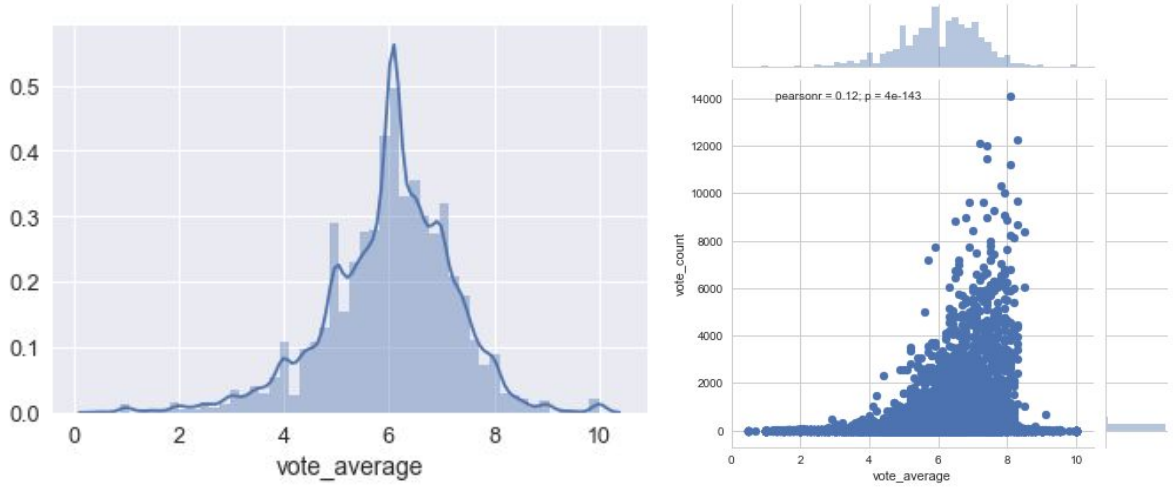
Original Languages

There are over 93 languages represented in our dataset. As we had expected, English language films form the overwhelmingly majority. French and Italian movies come at a very distant second and third respectively.



As mentioned earlier, French and Italian are the most commonly occurring languages after English. Japanese and Hindi form the majority as far as Asian Languages are concerned.

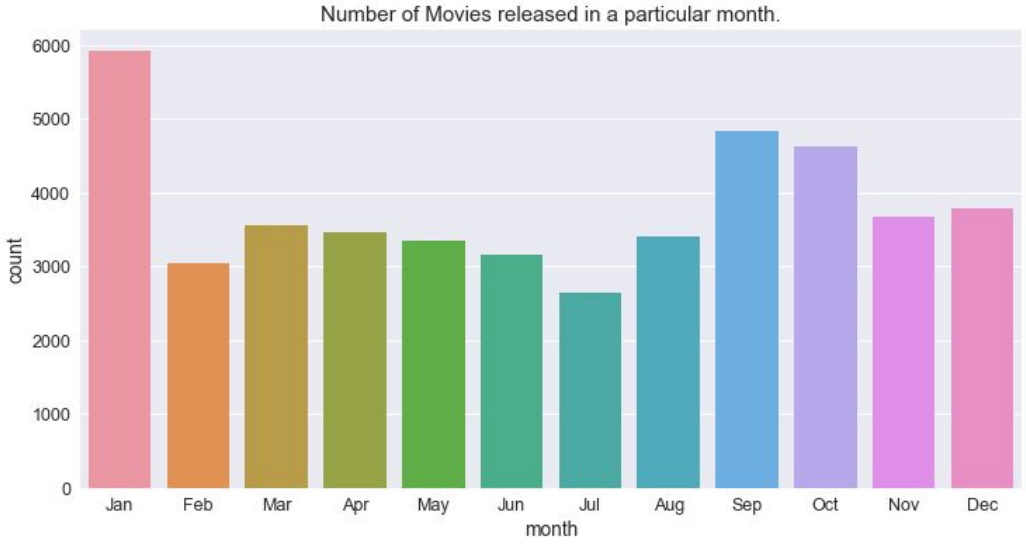
Popularity, Vote Average and Vote Count

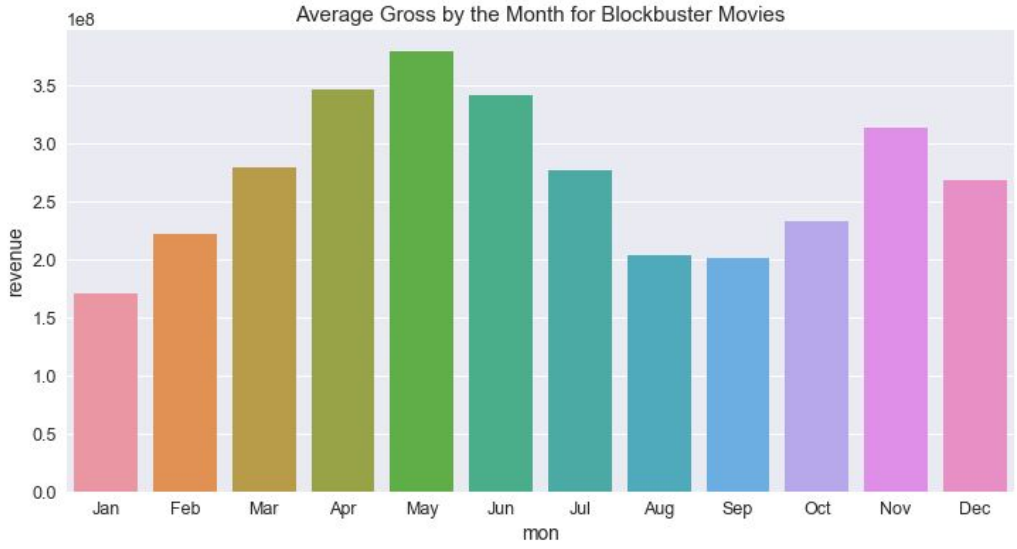


1. Minions is the most popular movie by the TMDB Popularity Score. Wonder Woman and Beauty and the Beast, two extremely successful woman centric movies come in second and third respectively.
2. Inception and The Dark Knight, two critically acclaimed and commercially successful Christopher Nolan movies figure at the top of The Most Voted On Movies Chart.
3. The Shawshank Redemption and The Godfather are the two most critically acclaimed movies in the TMDB Database. Interestingly, they are the top 2 movies in IMDB's Top 250 Movies list too. They have a rating of over 9 on IMDB as compared to their 8.5 TMDB Scores.

4. Surprisingly, the Pearson Coefficient of the two aforementioned quantities is a measly 0.097 which suggests that there is no tangible correlation. In other words, popularity and vote average and independent quantities. It would be interesting to discover how TMDB assigns numerical popularity scores to its movies.
5. There is a very small correlation between Vote Count and Vote Average. A large number of votes on a particular movie does not necessarily imply that the movie is good.

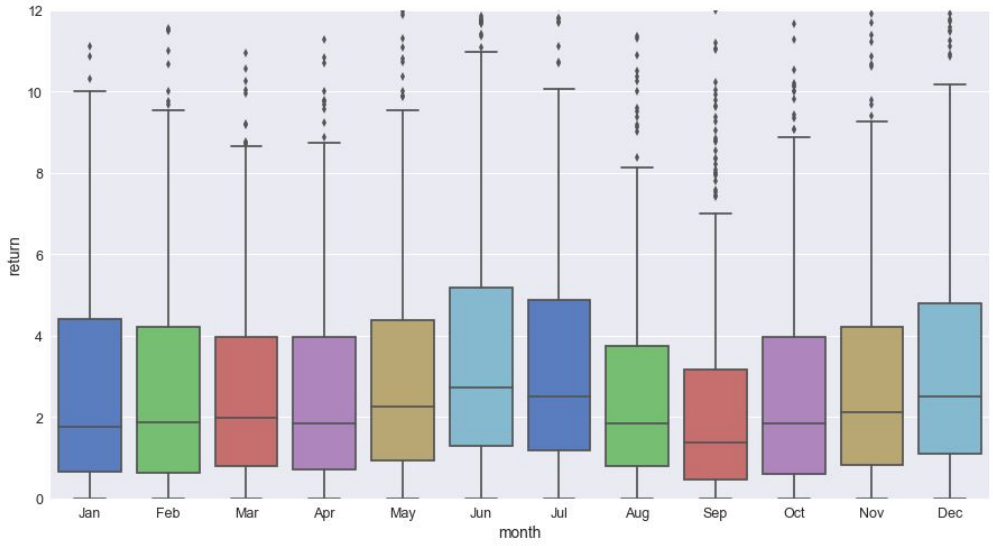
Movie Release Dates



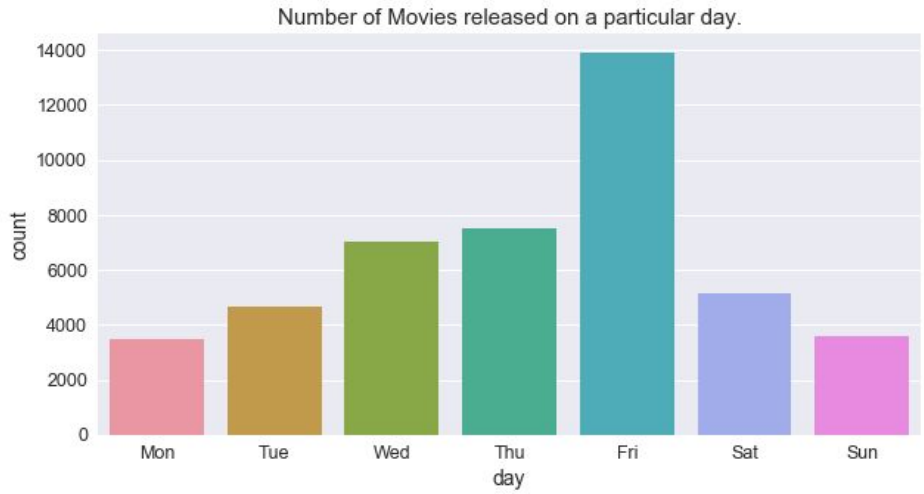


It appears that January is the most popular month when it comes to movie releases. In Hollywood circles, this is also known as the the dump month when sub par movies are released by the dozen.

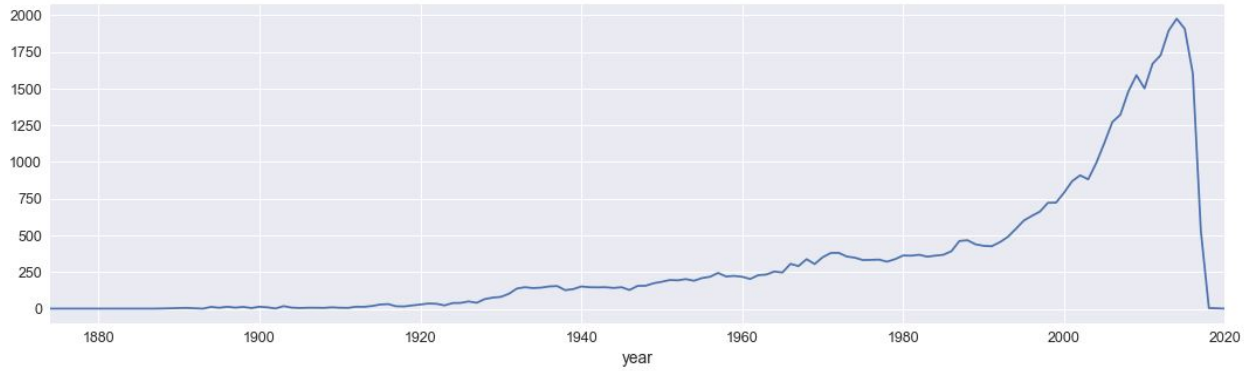
We see that the months of April, May and June have the highest average gross among high grossing movies. This can be attributed to the fact that blockbuster movies are usually released in the summer when the kids are out of school and the parents are on vacation and therefore, the audience is more likely to spend their disposable income on entertainment.



The months of June and July tend to yield the highest median returns. September is the least successful months on the aforementioned metrics. Again, the success of June and July movies can be attributed to them being summer months and times of vacation. September usually denotes the beginning of the school/college semester and hence a slight reduction in the consumption of movies.

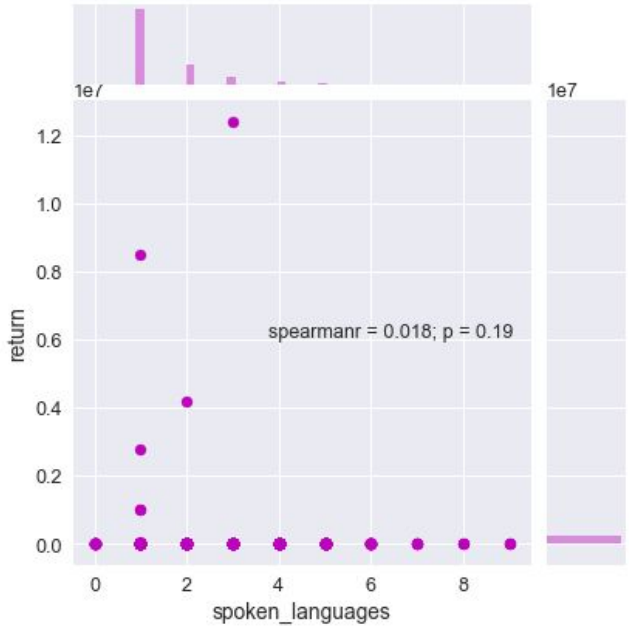


Friday is clearly the most popular day for movie releases. This is understandable considering the fact that it usually denotes the beginning of the weekend. Sunday and Monday are the least popular days and this can be attributed to the same aforementioned reason.



The oldest movie, Passage of Venus, was a series of photographs of the transit of the planet Venus across the Sun in 1874. They were taken in Japan by the French astronomer Pierre Janssen using his 'photographic revolver'. This is also the oldest movie on both IMDB and TMDB.

Spoken Languages

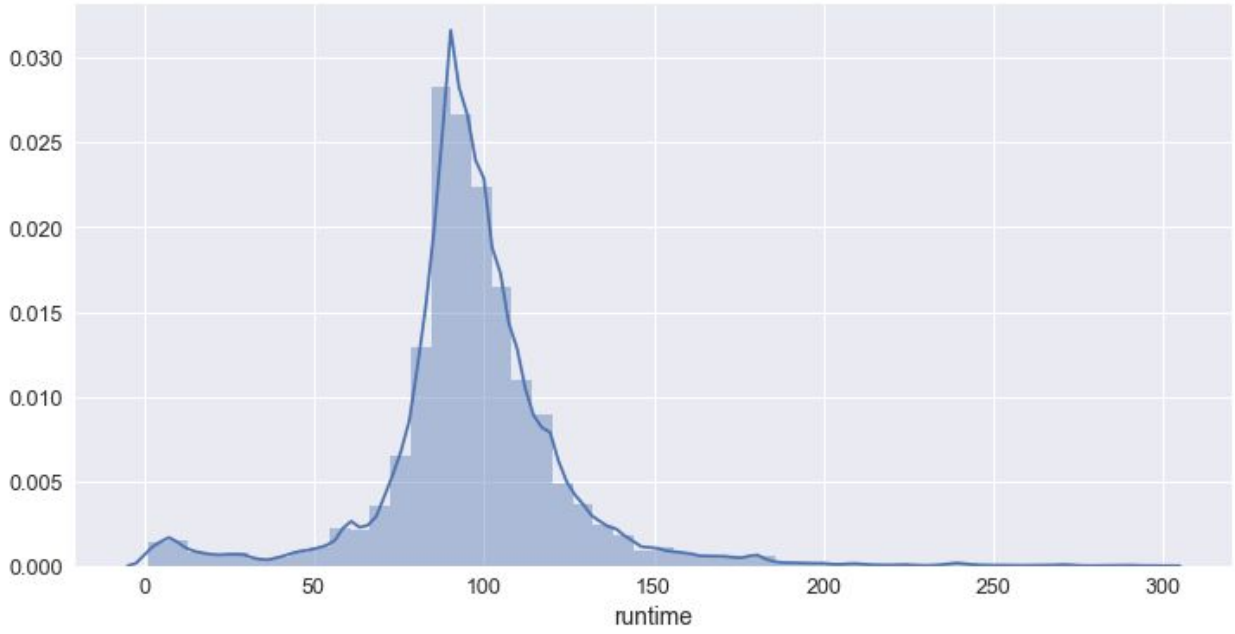


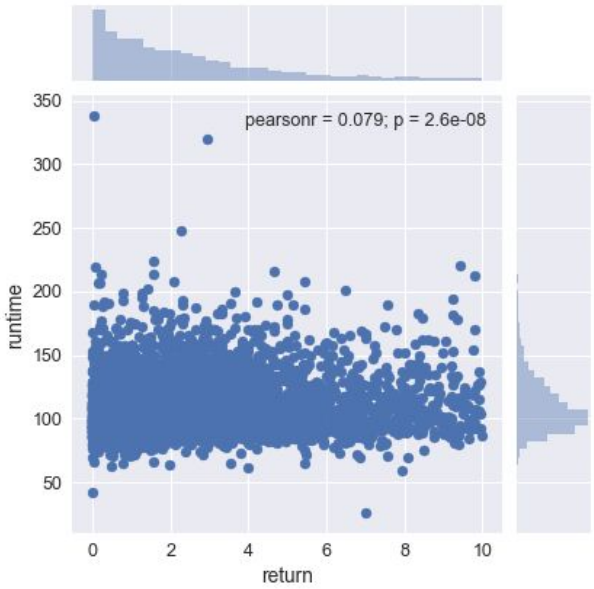
The movie with the most number of languages, Visions of Europe is actually a collection of 25 short films by 25 different European directors. This explains the sheer diversity of the movie in terms of language.

There is no correlation between the number of languages and returns of a movie.

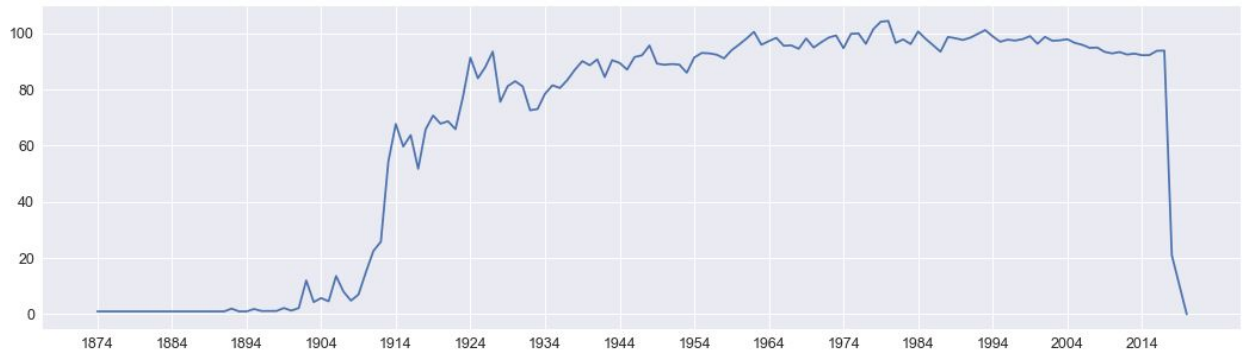
Runtime

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.



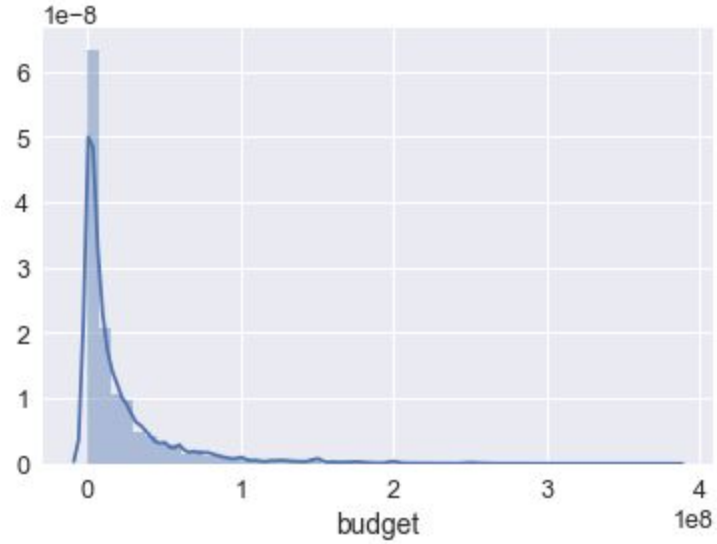


There seems to be no relationship between runtime and return. The duration of a movie is independent of its success.



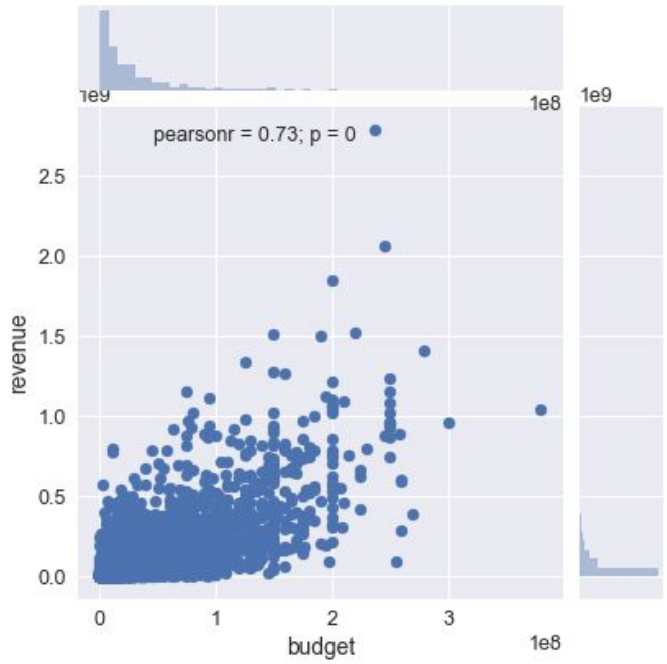
We notice that films started hitting the 60 minute mark as early as 1914. Starting 1924, films started having the traditional 90 minute duration and has remained more or less constant ever since.

Budget



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

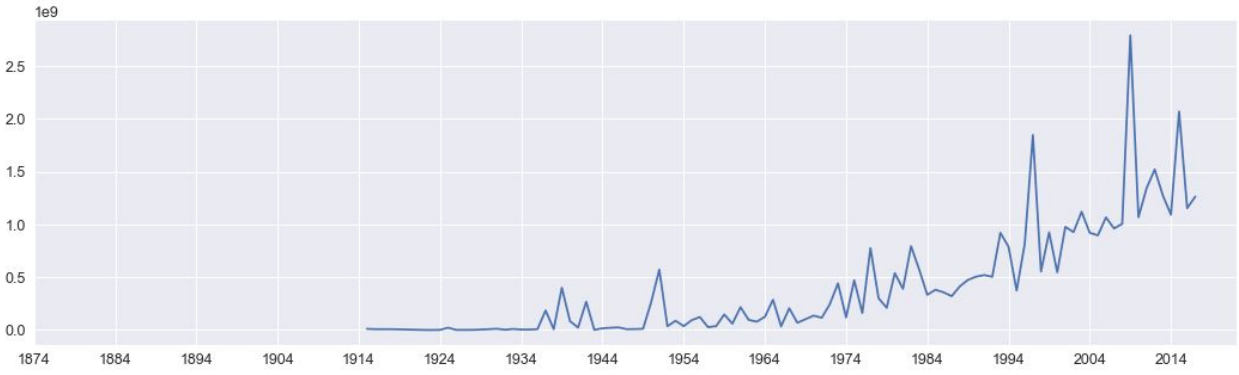
Two Pirates of the Caribbean films occupy the top spots in this list with a staggering budget of over 300 million dollars. All the top 10 most expensive films made a profit on their investment except for The Lone Ranger which managed to recoup less than 35% of its investment, taking in a paltry 90 million dollars on a 255 million dollar budget.



The pearson r value of 0.73 between the two quantities indicates a very strong correlation.

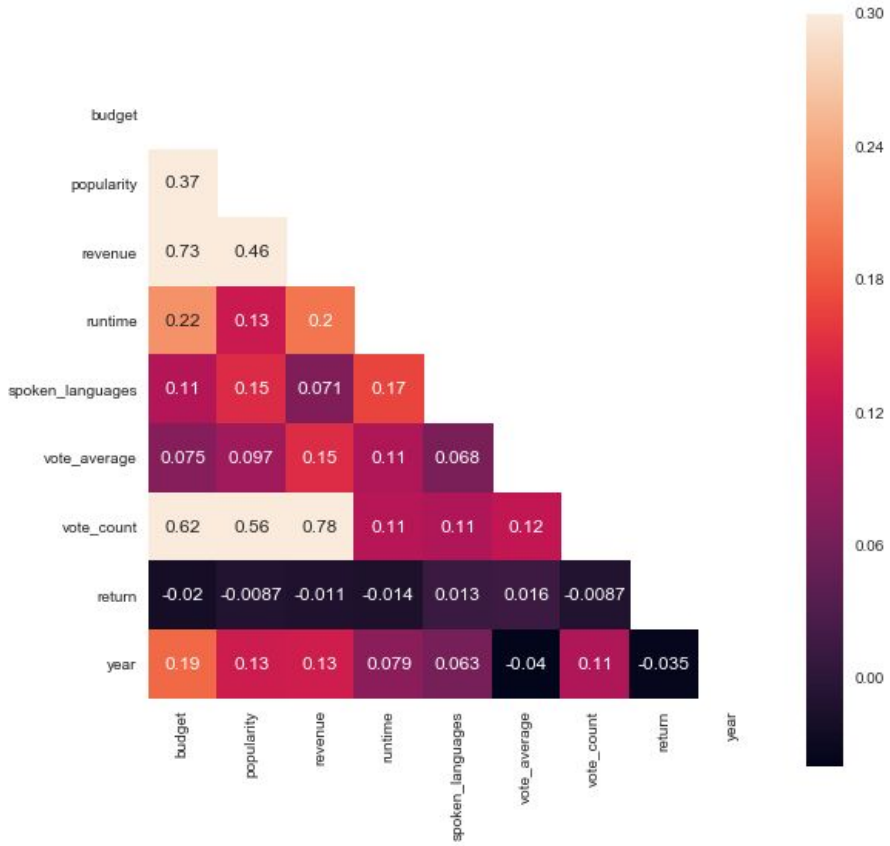
Revenue

The mean gross of a movie is 68.7 million dollars whereas the median gross is much lower at 16.8 million dollars, suggesting the skewed nature of revenue. The lowest revenue generated by a movie is just 1 dollar whereas the highest grossing movie of all time has raked in an astonishing 2.78 billion dollars.

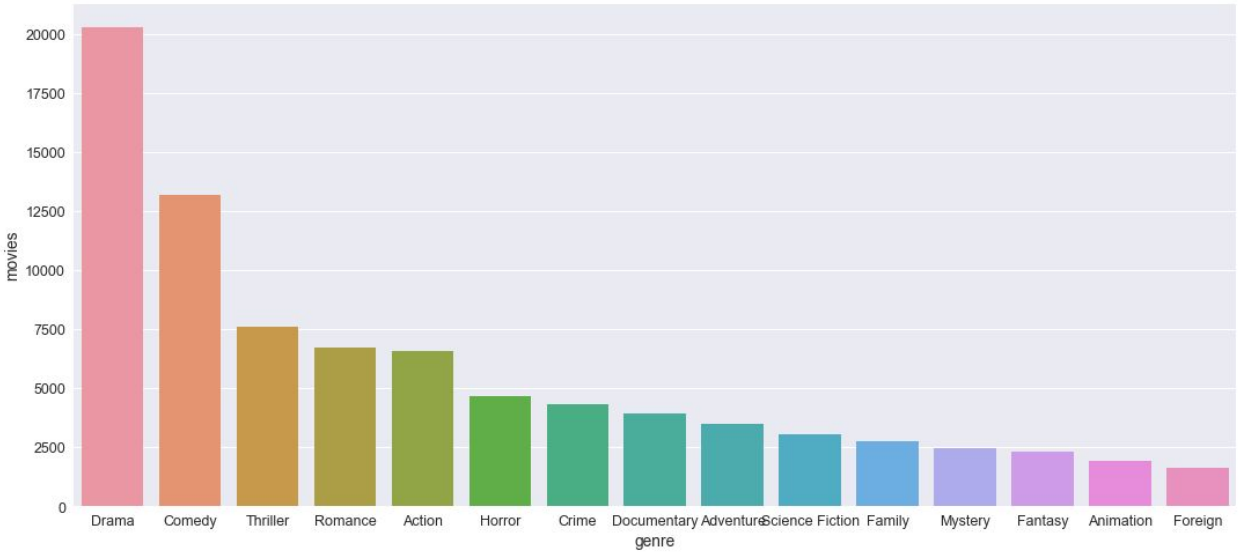


As can be seen from the figure, the maximum gross has steadily risen over the years. The world of movies broke the 1 billion dollar mark in 1997 with the release of Titanic. It took another 12 years to break the 2 billion dollar mark with Avatar. Both these movies were directed by James Cameron.

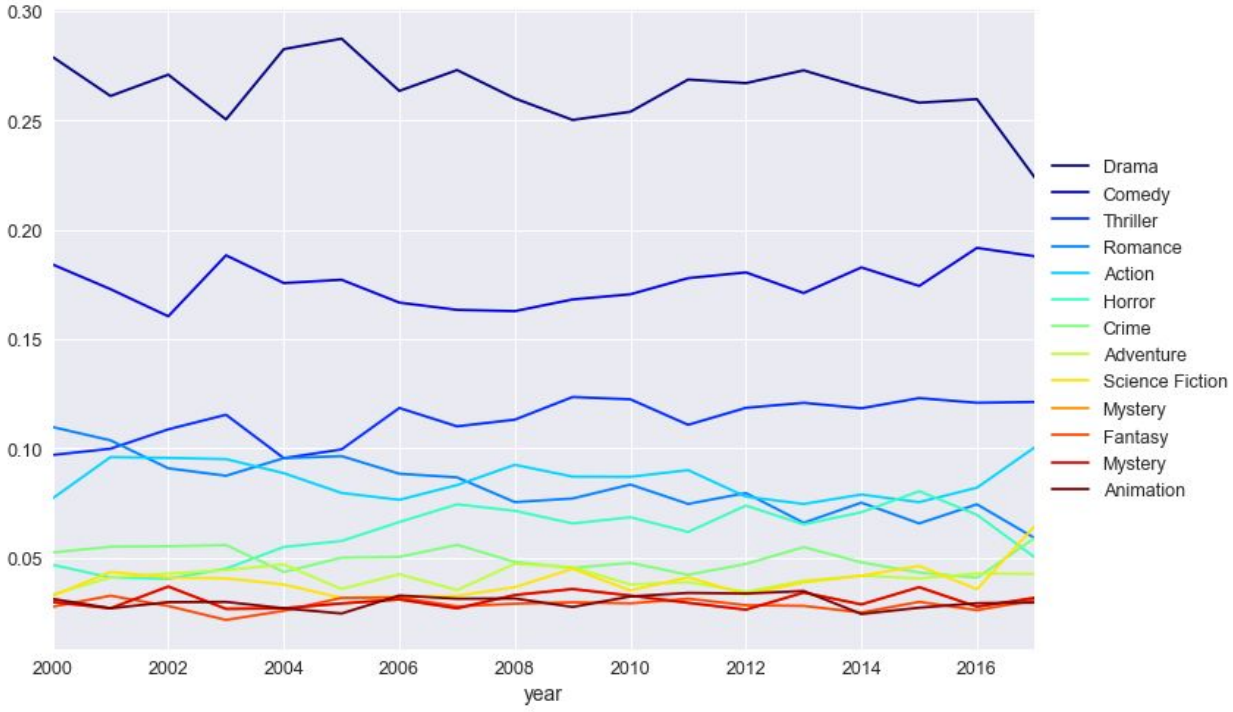
Correlation Matrix



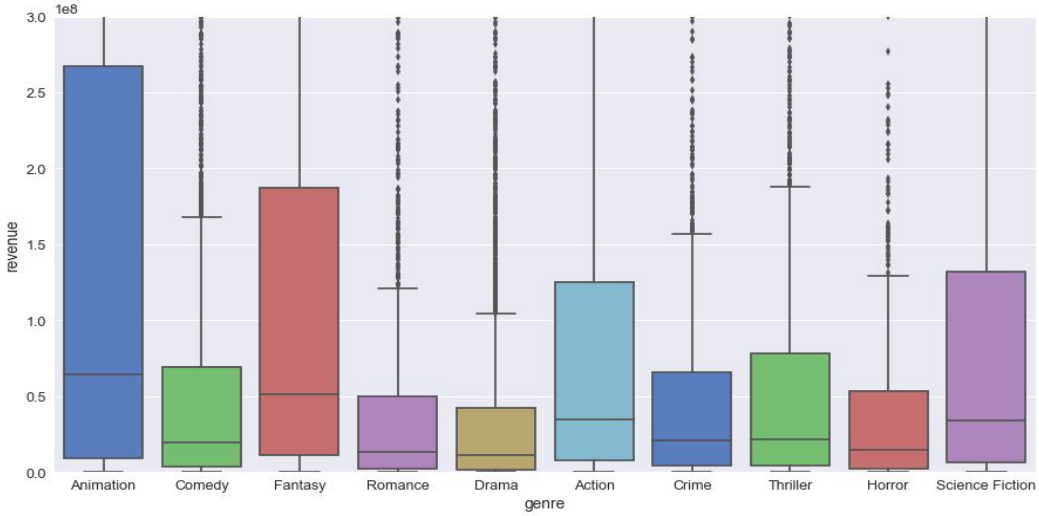
Genres



Drama is the most commonly occurring genre with almost half the movies identifying itself as a drama film. Comedy comes in at a distant second with 25% of the movies having adequate doses of humor. Other major genres represented in the top 10 are Action, Horror, Crime, Mystery, Science Fiction, Animation and Fantasy.

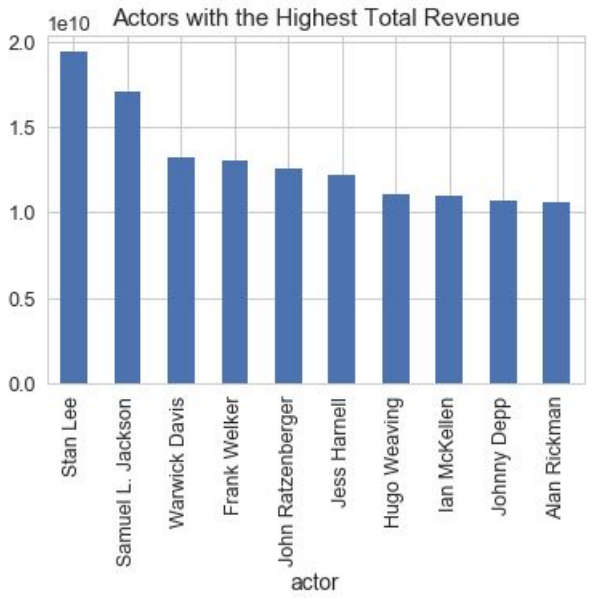


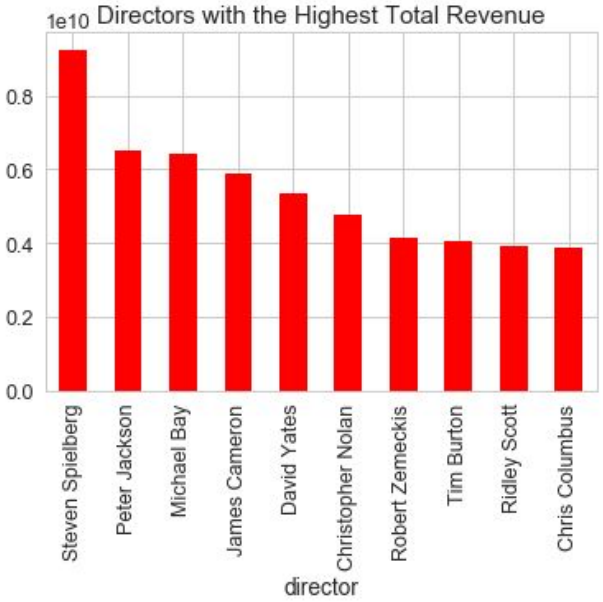
The proportion of movies of each genre has remained fairly constant since the beginning of this century except for Drama. The proportion of drama films has fallen by over 5%. Thriller movies have enjoyed a slight increase in their share.

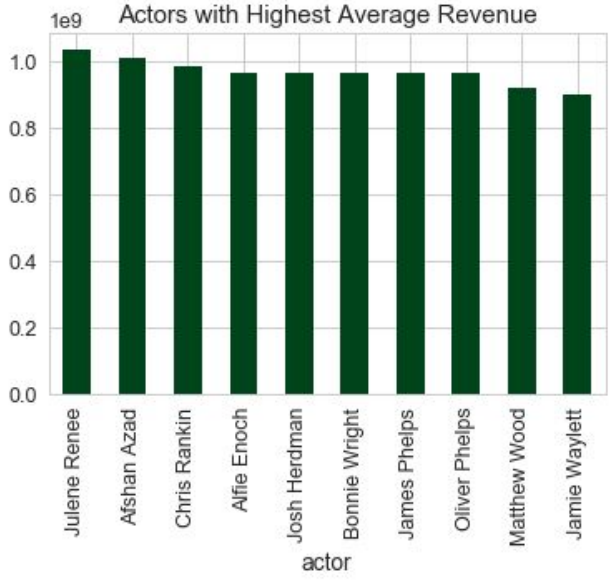


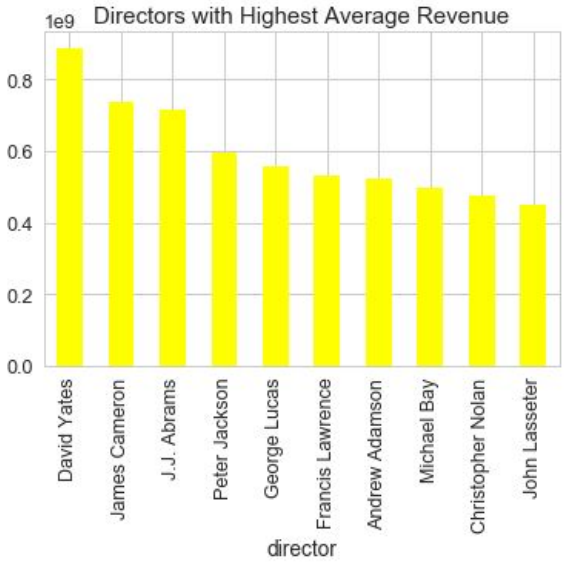
Animation movies has the largest 25-75 range as well as the median revenue among all the genres plotted. Fantasy and Science Fiction have the second and third highest median revenue respectively.

Cast and Crew









REGRESSION: PREDICTING MOVIE REVENUES

Predicting Movie Revenues is an extremely popular problem in Machine Learning which has created a huge amount of literature. Most of the models proposed in these papers use far more potent features than what we possess at the moment. These include Facebook Page Likes, Information on Tweets about the Movie, YouTube Trailer Reaction (Views, Likes, Dislikes, etc.), Movie Rating (MPCAA, CBIFC) among many others.

To compensate for the lack of these features, we are going to cheat a little. We will be using TMDB's Popularity Score and Vote Average as our features in our model to assign a numerical value to popularity. However, it must be kept in mind that these metrics will not be available when predicting movie revenues in the real world, when the movie has not been released yet.

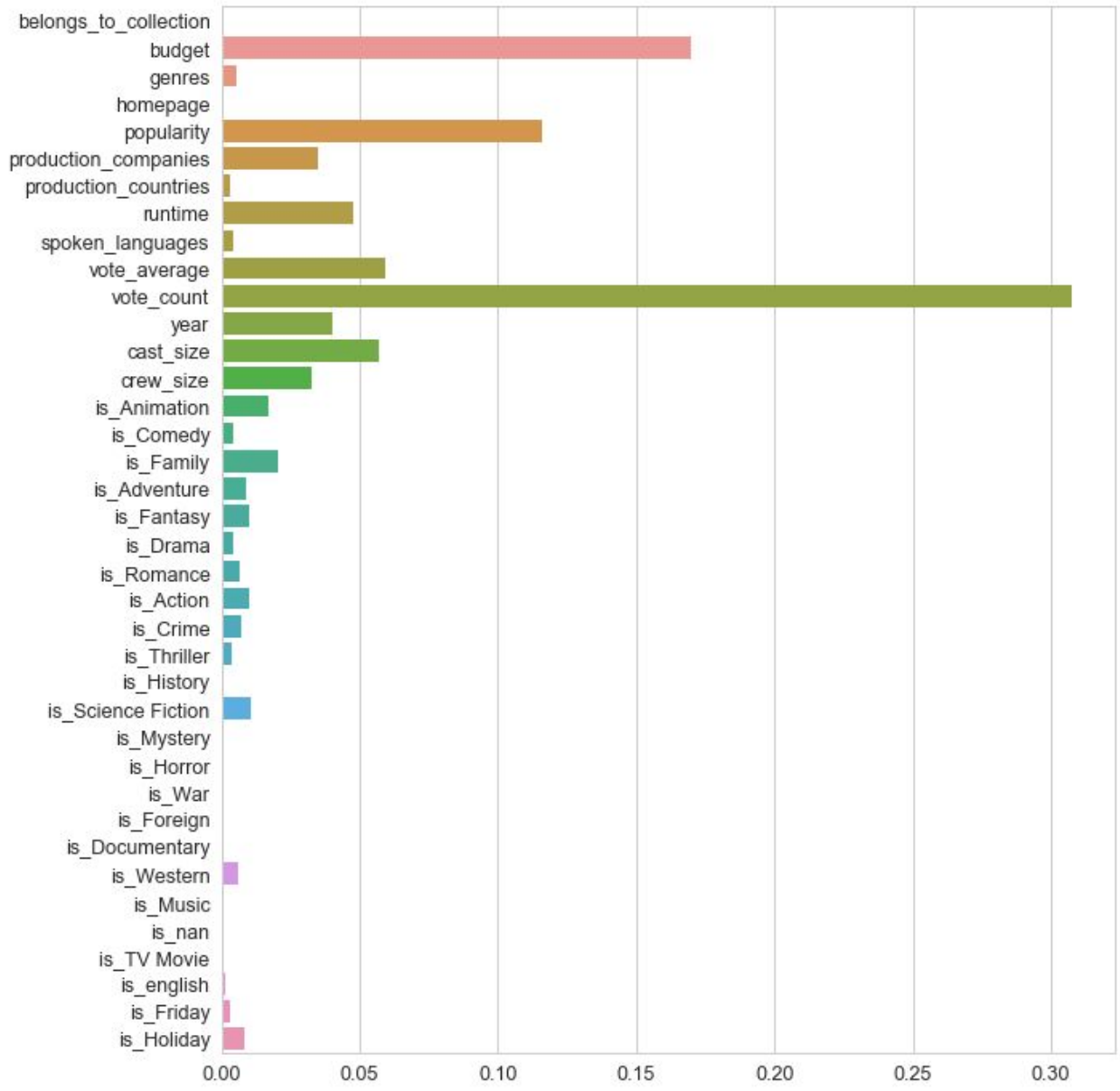
Feature Engineering

1. belongs_to_collection will be turned into a Boolean variable. 1 indicates a movie is a part of collection whereas 0 indicates it is not.
2. genres will be converted into number of genres.
3. homepage will be converted into a Boolean variable that will indicate if a movie has a homepage or not.
4. original_language will be replaced by a feature called is_foreign to denote if a particular film is in English or a Foreign Language.
5. production_companies will be replaced with just the number of production companies collaborating to make the movie.
6. production_countries will be replaced with the number of countries the film was shot in.
7. day will be converted into a binary feature to indicate if the film was released on a Friday.
8. month will be converted into a variable that indicates if the month was a holiday season.

Model

The model that I choose for regression is the Gradient Boosting Regression. The Coefficient of Determination Score obtained by the regressor was 0.78

Feature Importances



We notice that vote_count, a feature we cheated with, is the most important feature to our Gradient Boosting Model. This goes on to show the importance of popularity metrics in determining the revenue of a movie. Budget was the second most important feature followed by Popularity (Literally, a popularity metric) and Crew Size.

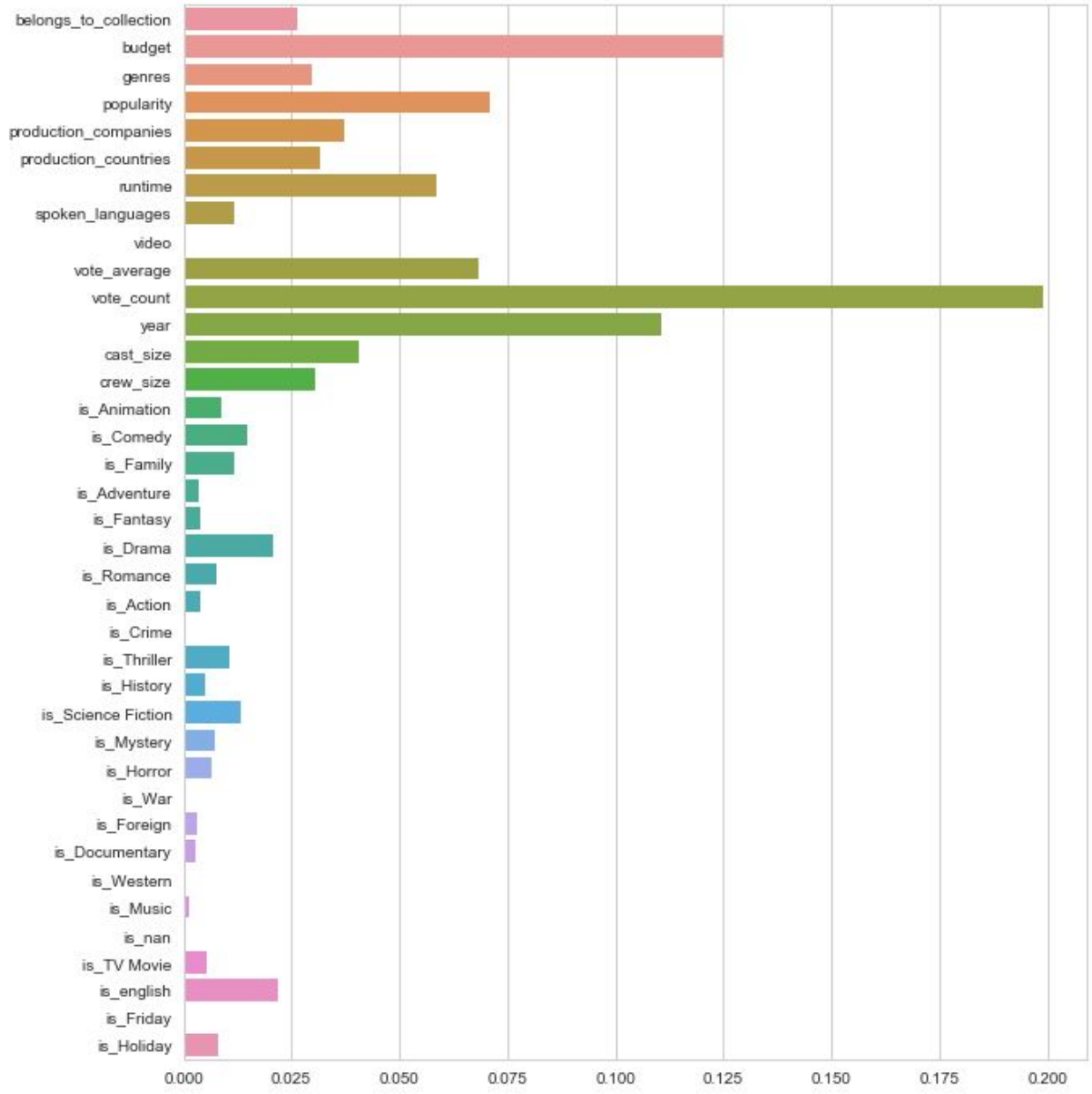
CLASSIFICATION: PREDICTING MOVIE SUCCESS

The Classification model uses the same Feature Engineering steps as those followed by the Regression Model built in the previous section.

Model

The model that I choose for classification is the Gradient Boosting Classifier. The model showcased an accuracy of 80% with unseen test cases.

Feature Importances



We see that Vote Count is once again the most significant feature identified by our Classifier. Other important features include Budget, Popularity and Year. With this, we will conclude our discussion on the classification model and move on to the main part of the project.

RECOMMENDATION SYSTEMS

The next step was to build a classifier to train the data on and then test its performance against the test data. With all the feature engineering already done in the previous step, applying machine learning was a fairly concise step.

The Simple Recommender

The Simple Recommender offers generalized recommendations to every user based on movie popularity and (sometimes) 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.

I used the TMDB Ratings to come up with our Top Movies Chart. I used IMDB's weighted rating formula to construct my chart.

The next step was to determine an appropriate value for m, the minimum votes required to be listed in the chart. I used 95th percentile as the 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.

	title	year	vote_count	vote_average	popularity	genres	wr
15480	Inception	2010	14075	8	29.1081	[Action, Thriller, Science Fiction, Mystery, A...	7.917588
12481	The Dark Knight	2008	12269	8	123.167	[Drama, Action, Crime, Thriller]	7.905871
22879	Interstellar	2014	11187	8	32.2135	[Adventure, Drama, Science Fiction]	7.897107
2843	Fight Club	1999	9678	8	63.8696	[Drama]	7.881753
4863	The Lord of the Rings: The Fellowship of the Ring	2001	8892	8	32.0707	[Adventure, Fantasy, Action]	7.871787
292	Pulp Fiction	1994	8670	8	140.95	[Thriller, Crime]	7.868660
314	The Shawshank Redemption	1994	8358	8	51.6454	[Drama, Crime]	7.864000
7000	The Lord of the Rings: The Return of the King	2003	8226	8	29.3244	[Adventure, Fantasy, Action]	7.861927
351	Forrest Gump	1994	8147	8	48.3072	[Comedy, Drama, Romance]	7.860656
5814	The Lord of the Rings: The Two Towers	2002	7641	8	29.4235	[Adventure, Fantasy, Action]	7.851924
256	Star Wars	1977	6778	8	42.1497	[Adventure, Action, Science Fiction]	7.834205
1225	Back to the Future	1985	6239	8	25.7785	[Adventure, Comedy, Science Fiction, Family]	7.820813
834	The Godfather	1972	6024	8	41.1093	[Drama, Crime]	7.814847
1154	The Empire Strikes Back	1980	5998	8	19.471	[Adventure, Action, Science Fiction]	7.814099
46	Se7en	1995	5915	8	18.4574	[Crime, Mystery, Thriller]	7.811669

Content Based Recommender

My approach to building the recommender was extremely hacky. What I did was create a metadata dump for every movie which consisted of genres, director, main actors and keywords. I then used a Countvectorizer to create a count matrix. I then calculated the cosine similarities and returned movies that are most similar.

I also added a mechanism to remove bad movies and return movies which are popular and have had a good critical response.

I took the top 25 movies based on similarity scores and calculate the vote of the 60th percentile movie. Then, using this as the value of m , I calculated the weighted rating of each movie using IMDB's formula like I did with the Simple Recommender.

```
In [53]: improved_recommendations('The Dark Knight')
```

Out[53]:

	title	vote_count	vote_average	year	wr
6623	The Prestige	4510	8	2006	7.758148
8031	The Dark Knight Rises	9263	7	2012	6.921448
6218	Batman Begins	7511	7	2005	6.904127
7659	Batman: Under the Red Hood	459	7	2010	6.147016
2085	Following	363	7	1998	6.044272
1134	Batman Returns	1706	6	1992	5.846862
7561	Harry Brown	351	6	2009	5.582529
8026	Bullet to the Head	490	5	2013	5.115027
9024	Batman v Superman: Dawn of Justice	7189	5	2016	5.013943
1260	Batman & Robin	1447	4	1997	4.287233

Collaborative Filtering

The content based engine suffers from some severe limitations. It is only capable of suggesting movies that are close to a certain movie. That is, it is not capable of capturing tastes and providing recommendations across genres.

Also, the engine that I built is not really personal in that it doesn't capture the personal tastes and biases of a user. Anyone querying our engine for recommendations based on a movie will receive the same recommendations for that movie, regardless of who s/he is.

Therefore, I used a technique called Collaborative Filtering to make recommendations to Movie Watchers. Collaborative Filtering is based on the idea that users similar to a me can be used to predict how much I will like a particular product or service those users have used/experienced but I have not.

I did not implement Collaborative Filtering from scratch. Instead, I used the Surprise library that provides extremely powerful algorithms like Singular Value Decomposition (SVD) to minimise RMSE (Root Mean Square Error) and give great recommendations.

```
Evaluating RMSE, MAE of algorithm SVD.

------------
Fold 1
RMSE: 0.8927
MAE:  0.6887
------------
Fold 2
RMSE: 0.9036
MAE:  0.6944
------------
Fold 3
RMSE: 0.8929
MAE:  0.6862
------------
Fold 4
RMSE: 0.9066
MAE:  0.6962
------------
Fold 5
RMSE: 0.8865
MAE:  0.6845
------------
------------
Mean RMSE: 0.8965
Mean MAE : 0.6900
------------
------------
```

Hybrid Recommender

The Hybrid Recommender brought together techniques from both Content Based and Collaborative Filtering Based engines to provide personalized Similar Movie Recommendations to Users based on their taste.

```
In [66]: hybrid(1, 'Avatar')
```

Out[66]:

	title	vote_count	vote_average	year	id	est
1011	The Terminator	4208.0	7.4	1984	218	3.111555
8401	Star Trek Into Darkness	4479.0	7.4	2013	54138	3.075929
8658	X-Men: Days of Future Past	6155.0	7.5	2014	127585	2.981067
974	Aliens	3282.0	7.7	1986	679	2.960471
522	Terminator 2: Judgment Day	4274.0	7.7	1991	280	2.941737
2834	Predator	2129.0	7.3	1987	106	2.843896
1621	Darby O'Gill and the Little People	35.0	6.7	1959	18887	2.772147
1668	Return from Witch Mountain	38.0	5.6	1978	14822	2.763146
922	The Abyss	822.0	7.1	1989	2756	2.729397
7705	Alice in Wonderland	8.0	5.4	1933	25694	2.728899

CONCLUSION

This report highlighted the processes of data wrangling, inferential statistics, data visualization, feature engineering and predictive modelling performed on the Movies Dataset. All the results and insights gained as part of these processes were also highlighted. With these insights, a Gradient Boosting Regressor and Classifier were built to predict Movie Revenue and Success respectively with a Score of 0.78 and 0.8 respectively.

In addition, four recommendation engines were built based on different ideas and algorithms:

1. **Simple Recommender**: This system used overall TMDB Vote Count and Vote Averages to build Top Movies Charts, in general and for a specific genre. The IMDB Weighted Rating System was used to calculate ratings on which the sorting was finally performed.
2. **Content Based Recommender**: We built two content based engines; one that took movie overview and taglines as input and the other which took metadata such as cast, crew, genre and keywords to come up with predictions. We also devised a simple filter to give greater preference to movies with more votes and higher ratings.
3. **Collaborative Filtering**: We used the powerful Surprise Library to build a collaborative filter based on singular value decomposition. The RMSE obtained was less than 1 and the engine gave estimated ratings for a given user and movie.
4. **Hybrid Engine**: We brought together ideas from content and collaborative filtering to build an engine that gave movie suggestions to a particular user based on the estimated ratings that it had internally calculated for that user.

The code associated with this report is available at: https://github.com/rounakbanik/movies