Capstone Project #2 – **Recommendation System for movies**

Problem:
It has been estimated that there are approximately 500,000 movies currently in existence. There are now over 135,000 cinema screens worldwide, on which about 8,000 movies are released each year internationally. In the evolutionary view, this situation creates `selection pressure' on individual movies. Not all movies are equally popular; industry-supply and audience-demand for specific movies (in fact, for particular movie stories) are asymmetrical(or at least seem so), as most movies lose money. The audience will not be spending their time watching every movie available to them. They will randomly pick something to watch. If a person selects a movie and does not enjoy it, there will be no positive word-of-mouth. This doesn't necessarily mean the movie was terrible. It might mean it was not fascinating to that individual.

This is where the recommendation system is helpful. The recommendation system helps the user find items of their interest and Helps the item provider deliver their items to the right user. It increases revenues for business through increased consumption. Movie Recommendation systems are becoming increasingly important in today's hectic world as it helps audiences make the right choices without having to expend their cognitive resources.
In this project, the goal is to Build Content-Based and Collaborative Filtering Based Recommendation Engines for movies.

Clients: Producers and distributors of movies.

Data: These files contain metadata for all 45,000 movies listed in the Full MovieLens Dataset. The dataset consists of movies released on or before July 2017. Data points include cast, crew, plot keywords, budget, revenue, posters, release dates, languages, production companies, countries, TMDB vote counts, and vote averages.
This dataset also has files containing 26 million ratings from 270,000 users for all 45,000 movies. Ratings are on a scale of 1-5 and have been obtained from the official GroupLens website. [1]

Acknowledgments: This dataset is an ensemble of data collected from TMDB and GroupLens.
The Movie Details, Credits, and Keywords have been collected from the TMDB Open API. This product uses the TMDb API but is not endorsed or certified by TMDb. Their API also provides data on many other movies, actors and actresses, crew members, and TV shows.
The dataset has a record of 45466 movies with 24 columns(features).

https://www.kaggle.com/rounakbanik/the-movies-dataset

Acknowledgements: This dataset is an ensemble of data collected from TMDB and GroupLens.

The Movie Details, Credits and Keywords have been collected from the TMDB Open API. This product uses the TMDb API but is not endorsed or certified by TMDb. Their API also provides access to data on many additional movies, actors and actresses, crew members, and TV shows.

Data Wrangling: After looking closer at the elements, I observed more than 50% of its data with null values. As revenue is the feature that I am interested in, I checked for Nan values in that column. Thirty-eight thousand fifty-two records of the movies have recorded revenue of 0, indicating that we do not have valuable information about the total revenue for these movies. Although this forms most of the movies available to us, we will still use revenue as a vital feature to advance from the remaining 7414 movies. The budget feature had some unclean values that make Pandas assign it as a generic object. I converted this into a numeric variable and replaced all the non-numeric values with NaN. Extracted feature "Release Year" from "Release Date." "Release Year" is the year in which the movie was released. I calculated the Net Profit/Loss using features "Revenue" and "budget." This feature is incredibly insightful as it will give us a more accurate picture of a movie's financial success. Presently, our data will not judge if a 200 million budget movie that earned 100 million did better than a 50,000 budget movie taking in 200,000. This feature will be able to capture that information. A value > 1 would indicate profit, whereas a return value < 1 would indicate a loss. A few features like adult, original_title, poster_path,

video does not provide useful information. I dropped these features from the data frame.

By exploring clean Movie metadata, we would try to answer the below questions.

1. Which Production companies make the most money in the movie business?
2. Which movies are more popular?
3. Which TMDB voters have the most voted movies?
4. Which movies are most Critically Acclaimed?
5. Does Release Month play a significant role in determining the success and the revenue generated by a particular movie?
6. Which are the most expensive movies of all time?
7. How strong a correlation does the budget hold with the revenue?
8. Which are the Highest Grossing Films of All Time?
9. Which are the least and the most successful movies of all time?

Let us find out which production companies have earned the most money from the movie-making business.

	Total Revenue	Average Revenue	Number Of Movies
Warner Bros.	6.352519e+10	1.293792e+08	491
Universal Pictures	5.525919e+10	1.193503e+08	463
Paramount Pictures	4.880819e+10	1.235650e+08	395
Twentieth Century Fox Film Corporation	4.768775e+10	1.398468e+08	341
Walt Disney Pictures	4.083727e+10	2.778046e+08	147
Columbia Pictures	3.227974e+10	1.367785e+08	236
New Line Cinema	2.217339e+10	1.119868e+08	198
Amblin Entertainment	1.734372e+10	2.550547e+08	68
DreamWorks SKG	1.547575e+10	1.984071e+08	78
Dune Entertainment	1.500379e+10	2.419966e+08	62

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.

As we are aware, Warner Bros and Universal Pictures are bigger studios compared to others on the list. Thus it would be more appropriate to look at the average revenue of studios. We will consider studios that have produced at least ten movies.

	Total Revenue	Average Revenue	Number Of Movies
Pixar Animation Studios	1.118853e+10	6.215852e+08	18
Marvel Studios	1.169964e+10	6.157703e+08	19
Heyday Films	7.920012e+09	6.092317e+08	13
WingNut Films	7.111004e+09	5.470003e+08	13
Revolution Sun Studios	8.120339e+09	5.413559e+08	15
Syncopy	5.359856e+09	5.359856e+08	10
Fuji Television Network	5.880444e+09	4.900370e+08	12
Blue Sky Studios	5.274028e+09	4.794570e+08	11
Walt Disney Animation Studios	6.053112e+09	4.656240e+08	13
Lucasfilm	9.898421e+09	4.499282e+08	22

Pixar Animation Studios has produced the most successful movies, on average. This is no surprise, though Pixar has made just 18 movies. It includes the Toy Story Franchise, Up, Finding Nemo, Inside Out, Wall-E, Ratatouille, Cars Franchise, Incredibles, etc. The audience has received well across the world as well as critically acclaimed. Marvel Studios, with an average gross of 615 million dollars, comes in second.

To answer questions like Which movies are more popular? And Which moves have been most voted by TMDB voters? We need to look at features popularity, vote_count, and vote_average. As these features are of type object, I converted them to float type.

First, let us find the answer to Which movies are the most popular?

	title	popularity	year
30700	Minions	547.488298	2015.0
33356	Wonder Woman	294.337037	2017.0
42222	Beauty and the Beast	287.253654	2017.0
43644	Baby Driver	228.032744	2017.0
24455	Big Hero 6	213.849907	2014.0
26564	Deadpool	187.860492	2016.0
26566	Guardians of the Galaxy Vol. 2	185.330992	2017.0
14551	Avatar	185.070892	2009.0
24351	John Wick	183.870374	2014.0
23675	Gone Girl	154.801009	2014.0

"Minions" is the most popular movie by the TMDB Popularity Score. I guess no arguing about liking cute minions. It is also interesting to note that Minions' title characters Talk less than a few words in the movie and yet most popular. Wonder Woman and Beauty and the Beast come in second and third respectively, both of which are women-centric stories almost tieing up for the second place.

Now that we know which movies are most popular, let us see which movies people most voted on in TMDB.

	title	vote_count	year
15480	Inception	14075.0	2010.0
12481	The Dark Knight	12269.0	2008.0
14551	Avatar	12114.0	2009.0
17818	The Avengers	12000.0	2012.0
26564	Deadpool	11444.0	2016.0
22879	Interstellar	11187.0	2014.0
20051	Django Unchained	10297.0	2012.0
23753	Guardians of the Galaxy	10014.0	2014.0
2843	Fight Club	9678.0	1999.0
18244	The Hunger Games	9634.0	2012.0

Inception and The Dark Knight, two critically acclaimed movies, are at the top of our chart. It is interesting to note that Christopher Nolan directed both of these.

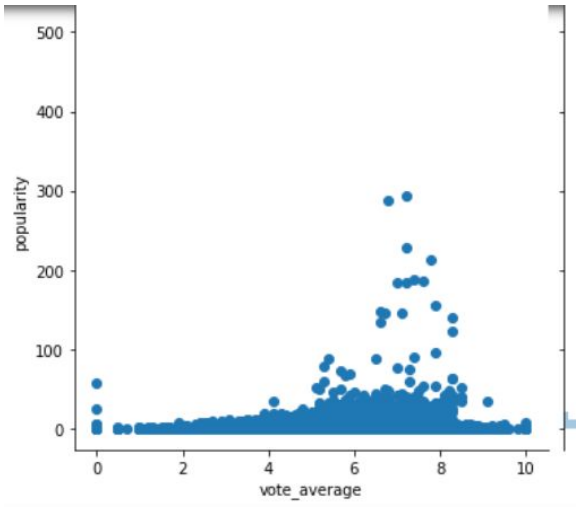
Let us check what the most critically acclaimed movies as per TMDB are. We will only consider those movies with more than 5000 votes (similar to IMDB's criteria of 5000 options in selecting its top 250).

	title	vote_average	vote_count	year
314	The Shawshank Redemption	8.5	8358.0	1994.0
834	The Godfather	8.5	6024.0	1972.0
292	Pulp Fiction	8.3	8670.0	1994.0
12481	The Dark Knight	8.3	12269.0	2008.0
2843	Fight Club	8.3	9678.0	1999.0
18465	The Intouchables	8.2	5410.0	2011.0
351	Forrest Gump	8.2	8147.0	1994.0
1154	The Empire Strikes Back	8.2	5998.0	1980.0
256	Star Wars	8.1	6778.0	1977.0
46	Se7en	8.1	5915.0	1995.0

The Shawshank Redemption and The Godfather are the two most critically acclaimed movies in the

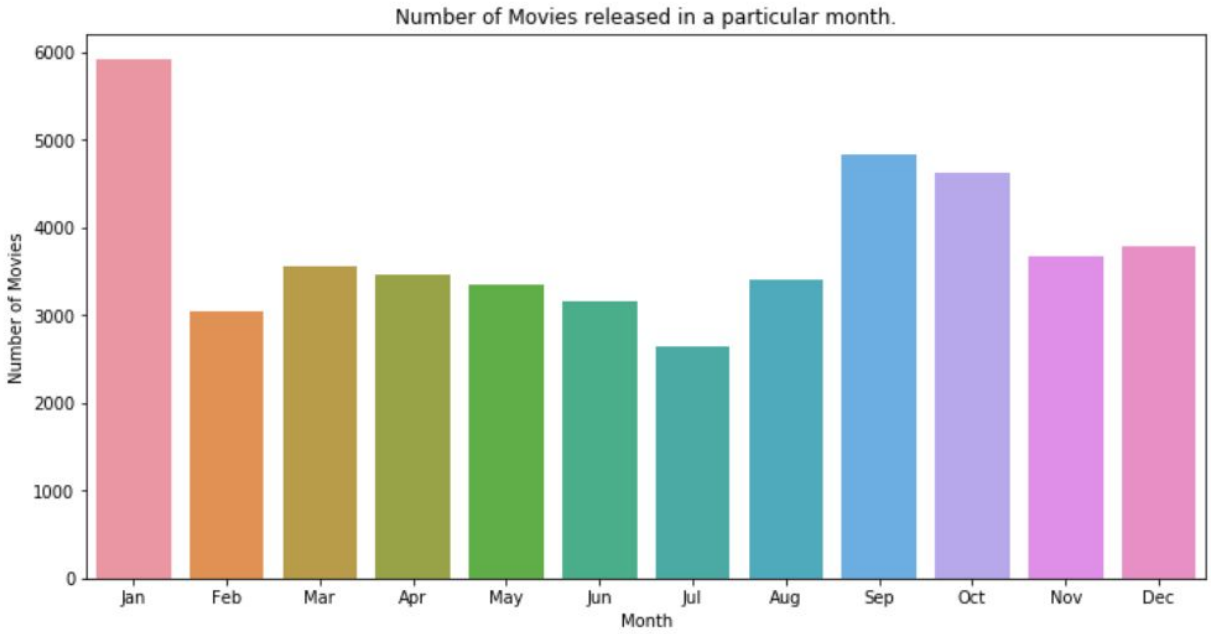
TMDB Database. The Academy Awards is going to agree with me on this.

Do popularity and vote average share a real relationship? In other words, is there a strong positive correlation between these two quantities?



Surprisingly, the Pearson Coefficient of the two quantities is 0.154, suggesting no definite correlation. In other words, popularity and vote average and independent quantities.

Release Dates can often play a significant role in determining the success and the revenue generated by a particular movie. This section will try and gain insights about release dates in terms of months. We have already constructed the year feature in our preliminary data wrangling step. Let us now extract the month for each movie with a release date.



It appears that January is the most popular month when it comes to movie releases. This is also known as the dump month in Hollywood circles when the dozen release bad movies.

```
movie_MetaData['budget'].describe()
```

```
count    8.890000e+03
mean     2.160428e+07
std      3.431063e+07
min      1.000000e+00
25%      2.000000e+06
50%      8.000000e+06
75%      2.500000e+07
max      3.800000e+08
Name: budget, dtype: float64
```

A film's mean budget is 21.6 million dollars, whereas the median budget is far smaller at 8 million dollars. This strongly suggests the mean being influenced by outliers.

```
movie_MetaData['revenue'].describe()
```

```
count    7.408000e+03
mean     6.878739e+07
std      1.464203e+08
min      1.000000e+00
25%      2.400000e+06
50%      1.682272e+07
75%      6.722707e+07
max      2.787965e+09
Name: revenue, dtype: float64
```

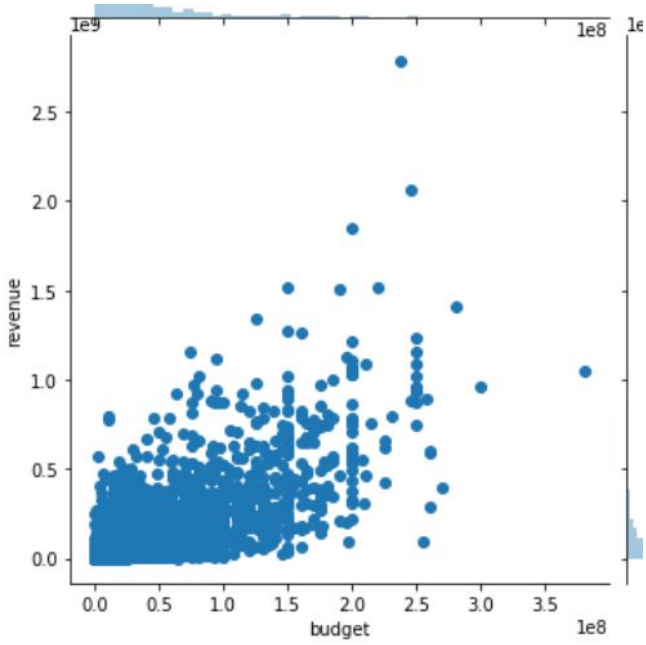
The mean gross of a movie is 68.7 million dollars, whereas the median gross is much lower at 16.8 million dollars, suggesting the skewed nature of revenue. The most insufficient revenue generated by a movie is just 1 dollar, whereas the highest-grossing film of all time has raked in an astonishing *2.78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

Let us take a look at the most expensive movies of all time and the revenue & returns.

	title	budget	revenue	Net Profit/Loss	year
17124	Pirates of the Caribbean: On Stranger Tides	380000000.0	1.045714e+09	2.751878	2011.0
11827	Pirates of the Caribbean: At World's End	300000000.0	9.610000e+08	3.203333	2007.0
26558	Avengers: Age of Ultron	280000000.0	1.405404e+09	5.019299	2015.0
11067	Superman Returns	270000000.0	3.910812e+08	1.448449	2006.0
44842	Transformers: The Last Knight	260000000.0	6.049421e+08	2.326701	2017.0
16130	Tangled	260000000.0	5.917949e+08	2.276134	2010.0
18685	John Carter	260000000.0	2.841391e+08	1.092843	2012.0
11780	Spider-Man 3	258000000.0	8.908716e+08	3.452991	2007.0
21175	The Lone Ranger	255000000.0	8.928991e+07	0.350157	2013.0
22059	The Hobbit: The Desolation of Smaug	250000000.0	9.584000e+08	3.833600	2013.0

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 movies 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.

How strong a correlation does the budget hold with the revenue? A stronger correlation would directly imply more accurate forecasts.



The scatterplot above shows a positive correlation between budget and revenue.

Let us see which are the Highest Grossing Films of All Time?

	title	budget	revenue	year
14551	Avatar	237000000.0	2.787965e+09	2009.0
26555	Star Wars: The Force Awakens	245000000.0	2.068224e+09	2015.0
1639	Titanic	200000000.0	1.845034e+09	1997.0
17818	The Avengers	220000000.0	1.519558e+09	2012.0
25084	Jurassic World	150000000.0	1.513529e+09	2015.0
28830	Furious 7	190000000.0	1.506249e+09	2015.0
26558	Avengers: Age of Ultron	280000000.0	1.405404e+09	2015.0
17437	Harry Potter and the Deathly Hallows: Part 2	125000000.0	1.342000e+09	2011.0
22110	Frozen	150000000.0	1.274219e+09	2013.0
42222	Beauty and the Beast	160000000.0	1.262886e+09	2017.0

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. James Cameron directed both these movies.

The highest-grossing movie does not necessarily mean the movie made the highest profit of all. Let us check the least and the most successful movies of all time. To do this, we will only consider those movies which have a budget greater than 5 million dollars.

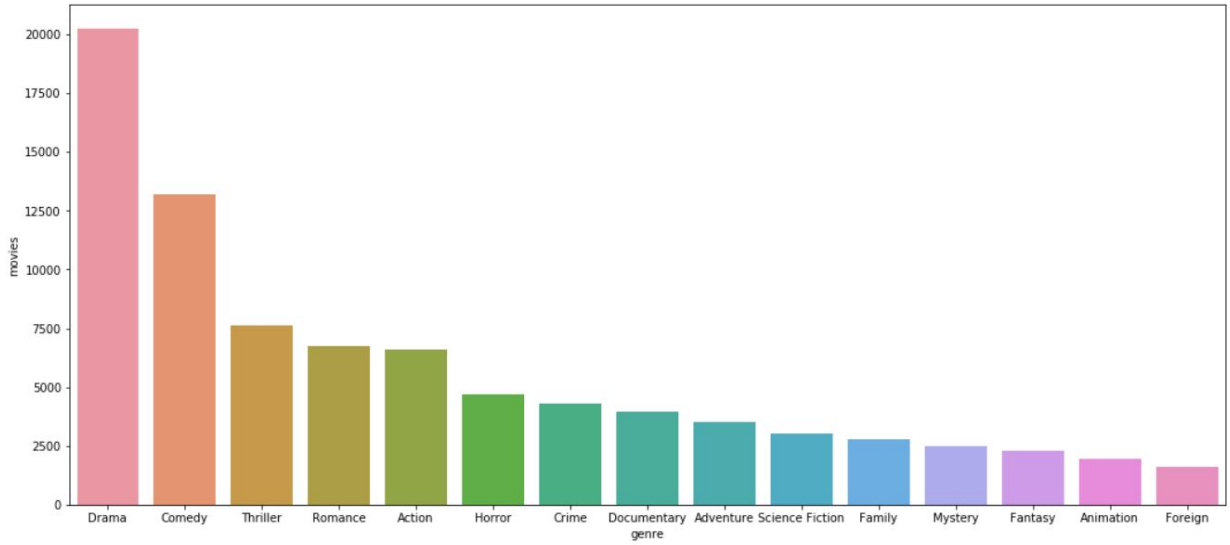
	title	budget	revenue	Net Profit/Loss	year
1065	E.T. the Extra-Terrestrial	10500000.0	792965326.0	75.520507	1982.0
256	Star Wars	11000000.0	775398007.0	70.490728	1977.0
1338	Jaws	7000000.0	470654000.0	67.236286	1975.0
1888	The Exorcist	8000000.0	441306145.0	55.163268	1973.0
352	Four Weddings and a Funeral	6000000.0	254700832.0	42.450139	1994.0
834	The Godfather	6000000.0	245066411.0	40.844402	1972.0
4492	Look Who's Talking	7500000.0	296000000.0	39.466667	1989.0
24258	Annabelle	6500000.0	255273813.0	39.272894	2014.0
1056	Dirty Dancing	6000000.0	213954274.0	35.659046	1987.0
1006	The Sound of Music	8200000.0	286214286.0	34.904181	1965.0

ET the Extra-Terrestrial is the most successful movie! Interestingly, the most successful movies in the top 10 list were released between 1965 - 1989.

	title	budget	revenue	Net Profit/Loss	year
11159	Chaos	20000000.0	10289.0	0.000514	2005.0
19027	5 Days of War	20000000.0	17479.0	0.000874	2011.0
21034	Special Forces	10000000.0	10759.0	0.001076	2011.0
25732	Foodfight!	65000000.0	73706.0	0.001134	2012.0
38388	Term Life	16500000.0	21256.0	0.001288	2016.0
19505	Laurence Anyways	9500000.0	12250.0	0.001289	2012.0
12038	The Good Night	15000000.0	20380.0	0.001359	2007.0
3966	Cherry 2000	10000000.0	14000.0	0.001400	1987.0
22097	Twice Born	13000000.0	18295.0	0.001407	2012.0
5651	All The Queen's Men	15000000.0	23000.0	0.001533	2001.0

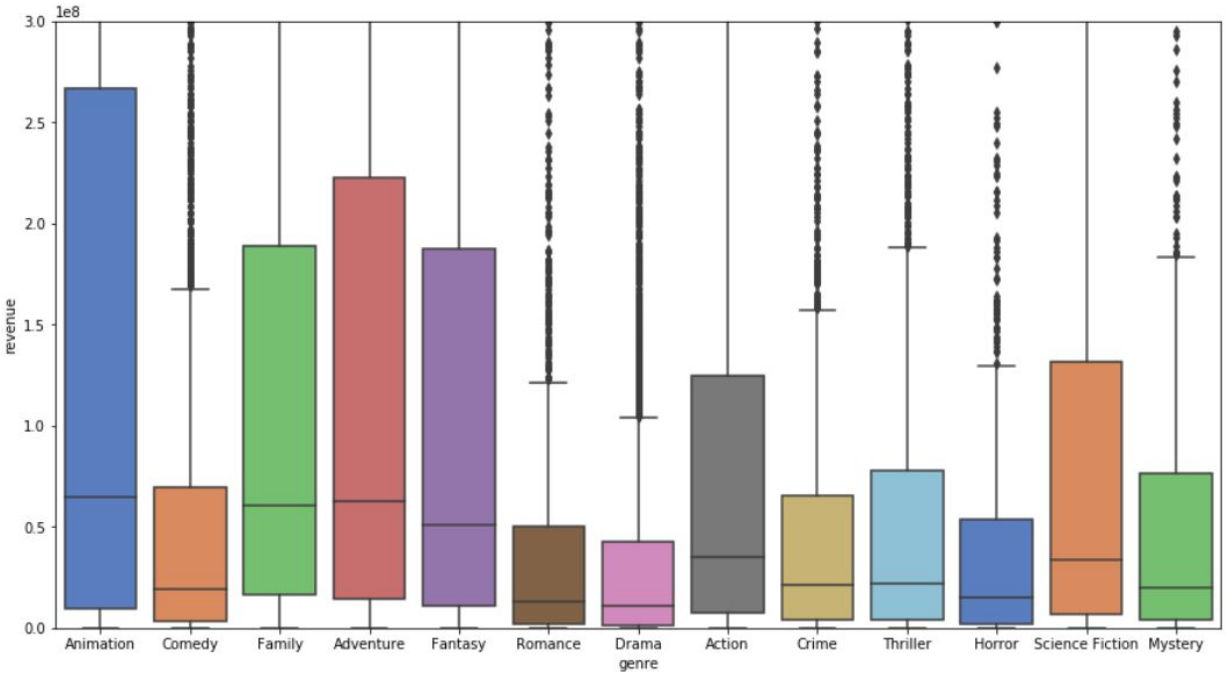
Chaos is the least successful movie. We can observe here that most of the movies listed in the top 10 are released between 2000-2012! So is it that older movies were more successful than newer ones? We cannot positively say so as these figures have not been adjusted for inflation.

Let us now have a look at the most commonly occurring genres in movies.TMDB defines 32 different genres for our set of 45,000 movies. The below screenshot shows the top ten genres.

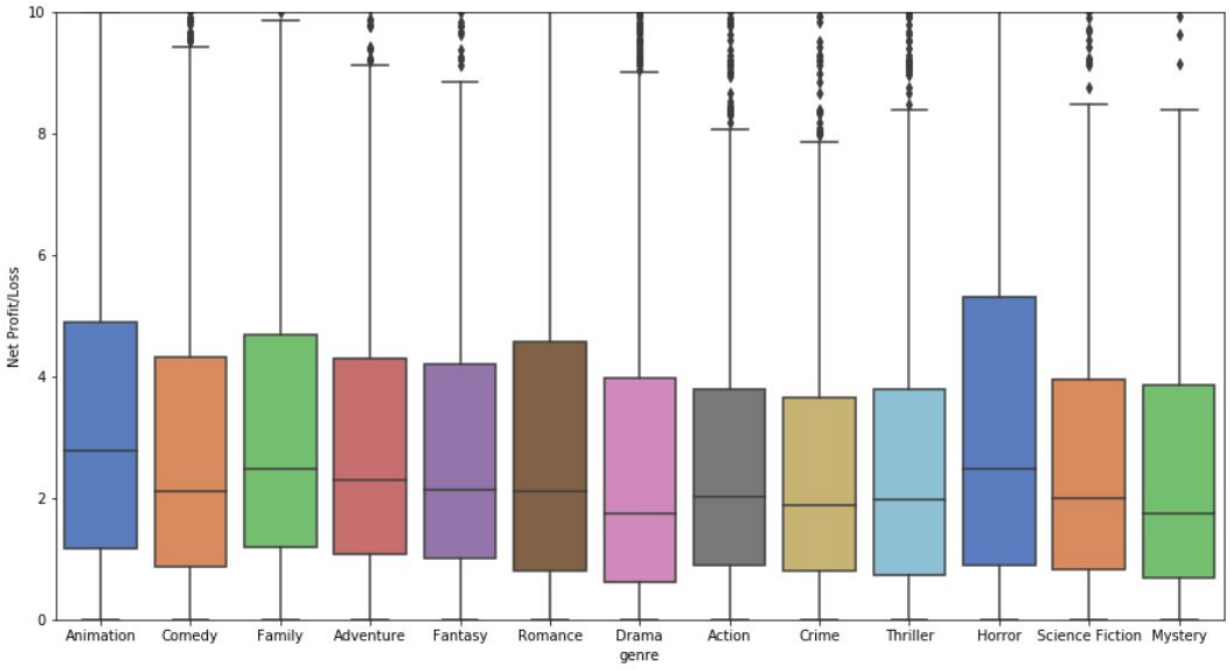


Drama is the most commonly occurring genre, with almost half the movies identifying itself as a drama film. Comedy comes in at a distant second, with 25% of the movies having adequate humor doses. Other significant genres represented in the top 10 are Action, Horror, Crime, Mystery, Science Fiction,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

Let us further see if some genres are particularly more successful than others. We expect Science Fiction and Fantasy Movies to bring in more revenue than other genres but when normalized with their budget, do they prove to be as successful?



Animation movies have the most extensive 25-75 range and the median revenue among all the genres plotted. Family and Adventure have the second and third highest median revenue, respectively. It is surreal to know that Romance and Drama have the lowest median revenue compared to other genres. From the boxplot shown below, it seems like Animation Movies tend to yield the highest returns on average. Horror Movies also tend to be a good yield.



Now that we analyzed the movie dataset, I will now move toward the project goal, which is to build a Content-Based and Collaborative Filtering based movie recommendation system.

First let us see what content based and Collaborative Filtering mean.

Content-based recommenders: suggest similar items based on a particular item. This system uses item metadata, such as genre, director, description, actors, etc. for movies, to make these recommendations. The general idea behind these recommender systems is that if a person likes a particular item, they will also enjoy an item that is similar to it. And to recommend that, it will make use of the user's past item metadata.

Collaborative Filtering recommenders: These systems are widely used. They try to predict the rating or preference that a user would give an item-based on past ratings and other users' preferences. Collaborative filters do not require item metadata like its content-based counterparts[2].

To build our standard metadata-based content recommender, I have merged our metadata dataset with the crew and the keyword datasets. We have two MovieLens datasets.

- **The Full Dataset:** Consists of 26,000,000 ratings and 750,000 tag applications applied to 45,000 movies by 270,000 users. Includes tag genome data with 12 million relevance scores across 1,100 tags.
- **The Small Dataset:** Comprises 100,000 ratings and 1,300 tag applications applied to 9,000 movies by 700 users.

Due to my laptop's computational limitations, I will be using the small dataset for a content-based and collaborative filtering-based recommendation system.

First, I will be building a simple recommendation system. The Simple Recommender offers generalized recommendations to every user based on movie popularity and genre. The basic idea behind this Recommender is that movies with more votes and more positive critic reception will have a higher probability of being liked by the average audience. This model does not give personalized recommendations based on the user. In simple terms, this is like searching "Top 10 Good romantic movies of all time" in any search engine.

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

Weighted Rating (WR) = $(v/v+m.R)+(m/v+m.C)$

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

For a movie to feature in the charts, it must have more votes than at least 80% of the movies on the list. Therefore, to qualify for the chart, a movie has to have at least 50 votes on TMDB. We also see that the average rating for a movie on TMDB is 5.244 on a scale of 10, and 9151 Movies qualify to be on our chart.

I will build our overall Top 250 movies Chart and define a function to build charts for a particular genre.

	title	year	vote_count	vote_average	popularity	genres	weighted_Rating
10309	Dilwale Dulhania Le Jayenge	1995	661	9	34.457	[Comedy, Drama, Romance]	8.735928
15480	Inception	2010	14075	8	29.1081	[Action, Thriller, Science Fiction, Mystery, A...	7.990247
12481	The Dark Knight	2008	12269	8	123.167	[Drama, Action, Crime, Thriller]	7.988818
22879	Interstellar	2014	11187	8	32.2135	[Adventure, Drama, Science Fiction]	7.987741
2843	Fight Club	1999	9678	8	63.8696	[Drama]	7.985839
4863	The Lord of the Rings: The Fellowship of the Ring	2001	8892	8	32.0707	[Adventure, Fantasy, Action]	7.984595
292	Pulp Fiction	1994	8670	8	140.95	[Thriller, Crime]	7.984202
314	The Shawshank Redemption	1994	8358	8	51.6454	[Drama, Crime]	7.983616
7000	The Lord of the Rings: The Return of the King	2003	8226	8	29.3244	[Adventure, Fantasy, Action]	7.983355
351	Forrest Gump	1994	8147	8	48.3072	[Comedy, Drama, Romance]	7.983194

It is interesting to see three Christopher Nolan Films, Inception, The Dark Knight, and Interstellar, which occur at the top of our chart. The chart also indicates a strong bias of TMDB Users towards particular genres and directors.

As I intended to build charts for particular genres and saw Drama is the most popular movie genre. I constructed a function that creates charts for specific genres and passed 'Drama' as an input parameter, and displayed the top 10 drama movies.

	title	year	vote_count	vote_average	popularity	weighted_Rating
12481	The Dark Knight	2008	12269	8	123.167	7.924623
22879	Interstellar	2014	11187	8	32.2135	7.917574
2843	Fight Club	1999	9678	8	63.8696	7.905213
314	The Shawshank Redemption	1994	8358	8	51.6454	7.890901
351	Forrest Gump	1994	8147	8	48.3072	7.888202
834	The Godfather	1972	6024	8	41.1093	7.851163
24860	The Imitation Game	2014	5895	8	31.5959	7.848105
359	The Lion King	1994	5520	8	21.6058	7.838458
18465	The Intouchables	2011	5410	8	16.0869	7.835391
22841	The Grand Budapest Hotel	2014	4644	8	14.442	7.810313

I built a chart for the 'Mystery' genre, which is less popular than my analysis.

	title	year	vote_count	vote_average	popularity	weighted_Rating
15480	Inception	2010	14075	8	29.1081	7.856221
46	Se7en	1995	5915	8	18.4574	7.682311
11354	The Prestige	2006	4510	8	16.9456	7.598743
4099	Memento	2000	4168	8	15.4508	7.571292
9430	Oldboy	2003	2000	8	10.6169	7.243008
877	Rear Window	1954	1531	8	17.9113	7.092712
896	Citizen Kane	1941	1244	8	15.8119	6.967234
876	Vertigo	1958	1162	8	18.2082	6.924746
14825	Shutter Island	2010	6559	7	15.8136	6.822468
23675	Gone Girl	2014	6023	7	154.801	6.808585

The Recommender I built in the previous section suffers some severe limitations. For one, it gives the same recommendation to everyone, regardless of the user's taste. If a person who loves romantic movies (and hates action) were to look at our Top 10 Chart, she/they/he wouldn't probably like most of the movies. If she/they/he were to go one step further and look at our charts by genre, she/they/he wouldn't still be getting the best recommendations.

The quality of our Recommender would be increased with the usage of better metadata. We will build a recommender based on the following metadata: the 3 top actors, the director, related genres, and the movie plot keywords. In this recommender system, the movie's content (cast, crew, keyword, etc.) is used to find its similarity with other movies. Then the movies that are most likely to be similar are recommended.This is our content-based recommender.

I have used a subset of all the movies available to us due to the limited computing power available. Thiis small dataset Comprises 100,000 ratings and 1,300 tag applications applied to 9,000 movies by 700 users. To achieve this, I wrote functions.

As Required data was present in the form of "stringified" lists, I converted it into a safe and usable structure. I also converted the names and keyword instances into lowercase and stripped all the spaces between them. This is done so that our vectorizer doesn't count the Jennifer of "Jennifer Garner" and "Jennifer Aniston" as the same. As We do not have a quantitative metric to judge our machine's performance, it will have to be done qualitatively. I have Mentioned Director 3 times to give it more weight relative to the entire cast. I created a "metadata soup," a string containing all the metadata that we want to feed to our vectorizer (namely actors, director, and keywords).

To use textual data for predictive modeling, the text must be parsed to remove certain words – this process is called tokenization. These words need to be encoded as integers, or floating-point values, for inputs in machine learning algorithms. This process is called feature extraction (or vectorization). [3].

Using cosine similarity, I am calculating a numeric quantity that denotes the similarity between two movies. I used the cosine similarity score since it is independent of magnitude and is relatively easy and fast to calculate. Mathematically, it is defined as follows:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

Below is the algorithm for the content-based recommendation system:

- Get the index of the movie given its title.
- Get the list of cosine similarity scores for that particular movie with all movies.
- Convert it into a list of tuples where the first element is its position, and the second is the similarity score.
- Sort the list mentioned above of tuples based on the similarity scores, the second element.
- Get the top 10 elements of this list. Ignore the first element as it refers to self (the movie most similar to a particular movie is the movie itself).
- Return the titles corresponding to the indices of the top elements[2].

The function **get_recommendations()** will take the movie title as the first input argument and the **cosine_sim** matrix as your second input argument. It will output a list of the 10 most similar movies. For this, we need a reverse mapping of movie titles and DataFrame indices. In other words, we need a mechanism to identify the index of a movie in our metadata DataFrame, given its title.

```
get_recommendations('The Dark Knight', cosine_sim)
```

```
8031    The Dark Knight Rises
6218            Batman Begins
6623             The Prestige
2085                Following
4145                 Insomnia
7648                Inception
3381                  Memento
8613             Interstellar
6645              Harsh Times
6902                   Hitman
Name: title, dtype: object
```

The recommendations seem to have recognized other Christopher Nolan movies (due to the high weightage given to the director) and put them as top recommendations. It is recommended 8 out of 10 movies based on the director. One thing that we can notice about our recommendation system is that it recommends movies regardless of ratings and popularity. To address this, I have added a mechanism to remove movies with less rating and return movies that are popular and have had an excellent critical response.

So I took the top 25 movies based on similarity scores and calculated the 60th percentile movie's vote. Using this as the value of **m**, we will calculate the weighted rating of each movie using IMDB's formula as we did in the Simple Recommender section.

	title	vote_count	vote_average	year	weighted_rating
7648	Inception	14075	8	2010	7.990247
8613	Interstellar	11187	8	2014	7.987741
6623	The Prestige	4510	8	2006	7.969791
3381	Memento	4168	8	2000	7.967341
8031	The Dark Knight Rises	9263	7	2012	6.990577
6218	Batman Begins	7511	7	2005	6.988394
2839	American Psycho	2128	7	2000	6.959708
4145	Insomnia	1181	6	2002	5.969330
7912	Takers	399	6	2010	5.915913
6902	Hitman	982	5	2007	5.011865

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

Also, the engine that we built is not personal in that it doesn't capture the individual 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 she/he is[4].

Therefore, next, we will use a technique called Collaborative Filtering to make recommendations to Movie Watchers. To better interpret the data, we first pivot the data frame to have userId as rows and movieId as columns, filling the null values with 0.0. I have used the scipy library in Python to implement algorithms like Singular Value Decomposition (SVD) to give great recommendations and build a function that uses factorized matrices to recommend movies to a user user_id. This function evaluates all the movies and their ratings, rated so far by the user(implying the user watched this movie) and recommends other movies in the dataset by predicting its rating.

```
User 44 has already rated 25 movies.
Recommending the highest 10 predicted ratings movies not already rated.
```

User 44 has already rated 25 movies.

Below are 10 movies which user 44 has already rated.

```
already_rated.dropna().head(10)
```

	userID	MovieID	Rating	timestamp	Title	Genre
19	44	780	5.0	858707138	The Passion of Joan of Arc	[drama, history]
7	44	62	5.0	858707138	2001: A Space Odyssey	[sciencefiction, mystery, adventure]
23	44	805	4.0	858707310	Rosemary's Baby	[horror, drama, mystery]
9	44	104	4.0	858707248	Run Lola Run	[action, drama, thriller]
10	44	135	4.0	858707310	Dont Look Back	[documentary, music]
14	44	628	3.0	858707310	Interview with the Vampire	[horror, romance]
22	44	802	3.0	858707310	Lolita	[drama, romance]
21	44	788	3.0	858707248	Mrs. Doubtfire	[comedy, drama, family]
20	44	786	3.0	858707194	Almost Famous	[drama, music]
16	44	648	3.0	858707138	Beauty and the Beast	[drama, fantasy, romance]

Recommending the highest 10 predicted ratings movies not already rated by user 44.

```
predictions
```

	MovieID	Title	Genre
4168	608	Men in Black II	[action, adventure, comedy, sciencefiction]
2172	1073	Arlington Road	[drama, thriller, mystery]
1025	832	M	[drama, action, thriller, crime]
3211	708	The Living Daylights	[action, adventure, thriller]
1103	653	Nosferatu	[fantasy, horror]
5024	79	Hero	[drama, adventure, action, history]
923	762	Monty Python and the Holy Grail	[adventure, comedy, fantasy]
5437	673	Harry Potter and the Prisoner of Azkaban	[adventure, fantasy, family]
6287	647	Final Fantasy VII: Advent Children	[action, adventure, animation, fantasy]
6590	86	The Elementary Particles	[drama, romance]

To compare movie recommendations of a content-based system with a collaborative filtering-based system, I chose the movie "2001: A Space Odyssey" which is highly rated by user' 44' and passed it as a parameter to our content-based system. By looking at the result, I would say I would be happier with the recommendation given by the collaborative filtering based system if I was user' 44'.

	title	vote_count	vote_average	year	weighted_rating
8613	Interstellar	11187	8	2014	7.987741
1029	The Shining	3890	8	1980	7.965037
979	A Clockwork Orange	3432	8	1971	7.960438
995	Full Metal Jacket	2595	7	1987	6.966822
7284	Moon	1831	7	2009	6.953347
8132	Prometheus	5152	6	2012	5.992742
7907	Transformers: Dark of the Moon	3351	6	2011	5.988899
7764	TRON: Legacy	2895	6	2010	5.987180
1497	Armageddon	2540	6	1998	5.985423
1349	Starship Troopers	1584	6	1997	5.976894

Conclusion: I have built 3 different recommendation engines based on different ideas and algorithms. They are as follows:

- **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[1]. This Recommender does not account for any user preference.

- **Content-Based Recommender:** I built a that took movie metadata such as cast, crew, genre, and keywords to make predictions. We also devised a simple filter to provide further preference to movies with more votes and higher ratings. However, It is only capable of suggesting movies that are close to a particular movie. That is, it is not capable of capturing tastes and providing recommendations across genres.

- **Collaborative Filtering:** We used the powerful Surprise Library to build a collaborative filter based on single value decomposition. One con about this Recommender is that it doesn't necessarily succeed in automatically matching content to one's preferences. These collaborative filtering systems require a substantial number of users to rate a new item before that item can be recommended.

Recommendation:

We can use content-based recommenders to suggest movies based on a particular movie user watched.

Collaborative filter-based recommender would be advisable to provide recommendations to users not only based on their watched movie list but also their likings towards it.

But with a hybrid model that brings together ideas from content and collaborative filtering to build an engine, we may be able to give better movie suggestions to a particular user in the future.

References: The sources are listed in the order in which they are cited in the report, as in the following book/article/website.

[1] https://www.kaggle.com/rounakbanik/the-movies-dataset
[2] https://www.datacamp.com/community/tutorials/recommender-systems-python
[3] https://www.educative.io/edpresso/countvectorizer-in-python.
[4] https://medium.com/the-owl/recommender-systems-f62ad843f70c
[5]https://heartbeat.fritz.ai/recommender-systems-with-python-part-iii-collaborative-filtering-singular-value-decomposition-5b5dcb3f242b
[6] https://www.kaggle.com/ibtesama/getting-started-with-a-movie-recommendation-system

Below is the link to the Github repository of jupyter notebook files with Data wrangling and statistical analysis code.

https://github.com/lasyabheemendra/Sprigboard-DatascienceProjects/tree/master/Capstone2_Recommendation%20System%20for%20movies

PowerPoint presentation of the above report can be found in the link below.

https://github.com/lasyabheemendra/Sprigboard-DatascienceProjects/blob/master/Capstone2_Recommendation%20System%20for%20movies/Recommendation%20system.pptx