Movie Recommendation System Report

OVERVIEW

We have built this project by dividing it into two parts: -

1) Data Cleaning, EDA, and predictive models: - First, we have understood the structure of the database then done extensive exploratory data analysis on the data about languages spoken, most words used in the title, most production countries, release dates, different genres, budget, revenue, cast, etc. Later, we have built a predictive model, who can say that a movie will be a hit or a flop based on the features.
2) Recommendation Engine: - We have built four recommendation engines one by one as founding flaws in the previous recommender engine. The four recommender engines are simple, content-based, collaborative filtering, and Hybrid.

INTRODUCTION

With the rise of YouTube, Netflix, Amazon and many more other web services over the last few decades recommendation systems have gained more and more positions in our lives.

From ecommerce (suggesting customers products that might interest them) to online ads (suggesting users the right con tent, matching their preferences), recommendation systems are unavoidable in our every day online journeys today. Recommending systems are, in a very general way, algorithms aimed at recommending related things tousers (objects b eing movies to watch, goods to buy, or something else depending on industry).

AUDIENCES

- Customers whose interest lie in Movies
- Subscription-based Online Entertainment Businesses
- Movie Producers

GOALS

- To give better Experience to users so that they can spend more time watching than wasting time on searching.
- To provide recommender engine to streaming services like Netflix to attract movie lovers who will increase their revenue and Sales.

SCOPE

The scope of the project is to answer our questions, which define the reason why we have done this project. Questions:

- What are the features that contribute a particular movie to be a hit or a flop?
- Which genres are more successful as compared to others?
- What are the prediction ratings of user2 based on user1 using collaborative filtering?

DATA

Dataset is taken from Kaggle. It contains 26 million ratings and 750,000 tag applications applied to 45,000 movies by 270,000 users.

Also, there is a small dataset containing 10,000 ratings for 9000 movies from 700 users is also available. Files used in the project are

1. movie_metadata.csv:- The metadata file has the TMDB reviews of 45,000 films, with many features like genre, ratings, budget, revenue, released date, etc.
2. credits.csv:- Credits of any movie, i.e., cast, crew, producer, director, etc.
3. keywords.csv:- plot keywords associated with the film.
4. links_small.csv:- Contains a list of movies included in the small subset of Full Movie Dataset.
5. Ratings_small.csv :- This file contains 100,000 ratings on 9,000 movies from 700 users.

DATA CLEANING

Original Title: - In our dataset, the original title refers to the film title, which is in the native language. We would choose to use the translated, Anglicized name and will thus remove the original names entirely. By looking at the original language function, we would be able to deduce if the film is a foreign language film, and no meaningful knowledge is lost in doing so.

Revenue: - In the movie where revenue is 0, it means that we don't have overall sales for that movie. Thus, we are left with 7000 movies in our dataset.

Budget: - The feature budget has some unclean values, which we will be converting it into numeric values and also replacing all non-numeric values into NaN.

Adult: - In our Dataset, we have 0 adult movies. Hence, we will be removing the adult column from our dataset.

EXPLORATORY DATA ANALYSIS

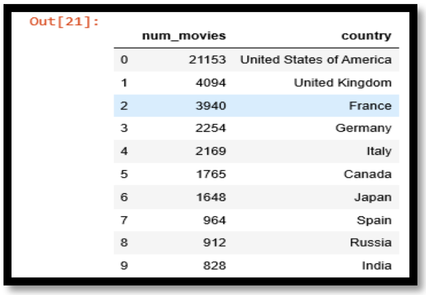
In EDA, we have done descriptive statistics and presented data visualization.

Title and Overview Word clouds

Love
Day
Man
Girl
Night
Life
Story
Time
Movie
Christmas
House
Black
Little
Dead
Last
World
Death
One
City
Dark
Blue
King
Red
Dream
Boy
Live
Men
Game
Blood
Heart
Secret
American
War
Family
Woman
Summer
Adventure

- The word Love is the term most commonly used in film titles. Girl, day, and man are among the most frequently used words, too.

Production Countries



Out[21]:

	num_movies	country
0	21153	United States of America
1	4094	United Kingdom
2	3940	France
3	2254	Germany
4	2169	Italy
5	1765	Canada
6	1648	Japan
7	964	Spain
8	912	Russia
9	828	India

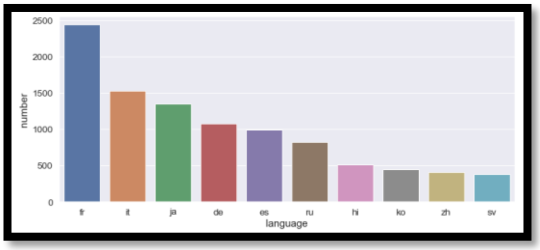
- The US is the most common movie production destination because our dataset consists mostly of English films. Europe is also very common with UK, France, Germany, and Italy among the top 5. In terms of film production, Japan and India are the most famous Asian countries.

Franchise Movies

- The **Harry Potter** Series is the most profitable film franchise ranking in more than $7.707 billion from 8 films. The **Star war** Movies are coming in a near second too with $7.403 billion from 8 movies. **James Bond** is sixth, but the franchise has a slightly higher gross average than the others on the list.
- While the **Avatar** Series currently consists of only one film, it is the most profitable franchise of all time, with the single film raking in nearly $3 billion. For at least five movies, the **Harry Potter** series is the most successful franchise.

Original Languages

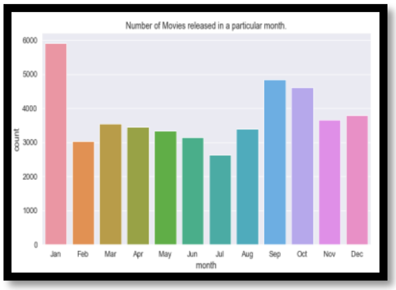
- Our dataset contains over 93 languages. English language films form the vast majority, as we had expected. Both French and Italian films come in very distant seconds and thirds.
- The bar plot we can say that after **English**, **French** and **Italian** are the languages that occur the most. As for Asian Languages, **Japanese** and **Hindi** make up the rest.

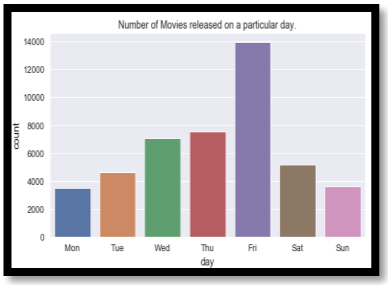


Movie Release Dates

- January continues to be the most common month when it comes to film releases. It is also known in Hollywood circles as the "the dump month" when the dozen release sub per films.

- The months of April, May, and June have the highest gross average of the top-grossing films. The hit movies are usually released in the summer when the children are out of school, and the parents are on holiday and, thus, the audience is more likely to spend their discretionary income on entertainment.
- Friday is probably the most famous day for film releases. That's understandable given the fact that it typically denotes the weekend started. Sunday and Monday are the least-popular days and can be due to the same cause.

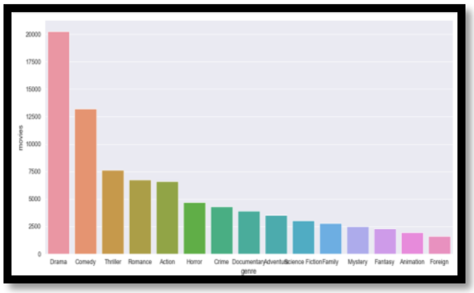




Budget & Revenue

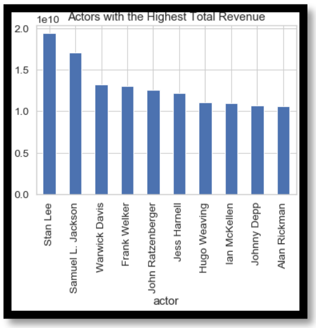
- Budget is always a crucial feature in predicting film's performance. Two Caribbean Pirates films hold the top slots in this list with a massive $300 million budget. All the top 10 most expensive movies made a return on their investment except for The Lone Ranger, who managed to raise less than 35 percent of their investment, pulling in a pitiful $90 million on a $255 million budget.
- The second most critical numerical number associated with a film is the revenue. Avatar is the highest-grossing movie of all time with revenue of $2.78 billion.

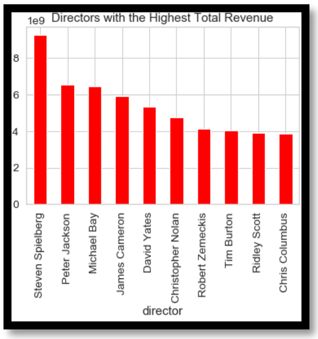
Genres



- Drama is the most prevalent genre, with almost half of the films describing themselves as a drama. Comedy comes in at a distant second, with 25% of the movie having sufficient comedy doses. Action, Horror, Crime, Mystery, Science Fiction, Animation, and Fantasy are other major genres included in the top 10.

Cast & Crew





PREDICTIVE MODEL

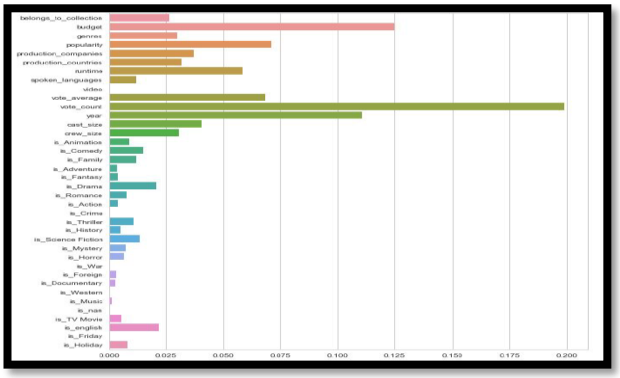
We will perform the following feature engineering tasks:

- belongs to the collection variable is transformed into a Boolean type. 1 states that a movie is part of a series, while 0 means that it is not.
- Genres will be converted into several genres.
- The homepage will be converted into a Boolean variable that will indicate if a movie has a website or not.
- original_language will be replaced by a feature called is_foreign to denote if a particular film is in English or a Foreign Language.

- production_companies will be replaced with just the number of production companies collaborating to make the movie.
- production_countries will be replaced with the number of countries the film was shot in.
- The day will be converted into a binary feature to indicate if the film was released on a Friday.
- The month will be converted into a variable that indicates if the month was a holiday season.

Classification

- Classification is the process of finding or discovering a model or function which helps in separating the data into multiple categorical classes, i.e., discrete values.
- The model we choose for classification is the Gradient Boosting Classifier. The model showcased an accuracy of 80% with unseen test cases.

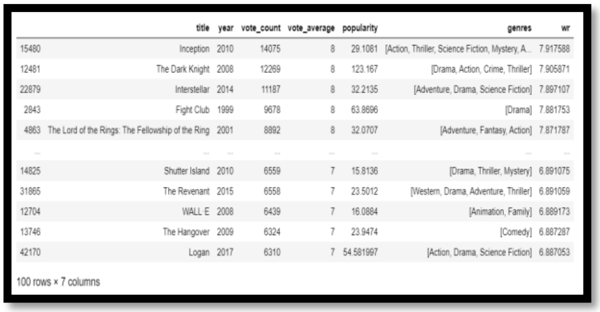


- We see that; **Vote Count** is the most essential feature our Classifier has found. Other highlights include **Budget**, **Popularity**, and **Year**.

RECOMMENDER ENGINE

Simple Recommender System

- Simple Recommender offers generic recommendations based on movie popularity and genre to every user. The fundamental idea behind this recommender is that more successful and critically acclaimed movies should have a higher chance of being enjoyed by the average viewer. This model does not offer user-based, custom recommendations.
- We have used the TMDB Ratings and IMDB's *weighted rating* formula to come up with our Top Movies Chart. The next step was to decide a suitable value for 'm', the minimum votes needed to appear in the chart. We have used 95th percentile as our limit. In other words, for a movie to be included in the charts, it must have more votes in the list than at least 95th percent of the movies.

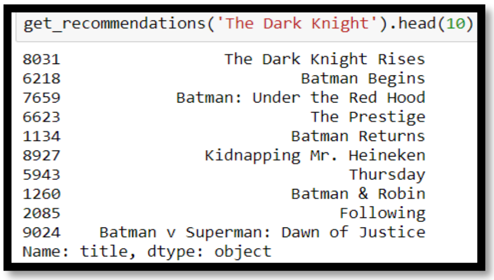


	title	year	vote_count	vote_average	popularity	genres	wr
15480	Inception	2010	14075	8	29.1081	[Action, Thriller, Science Fiction, Mystery, A...	7.917588
12481	The Dark Knight	2008	12269	8	123.167	[Drama, Action, Crime, Thriller]	7.905871
22879	Interstellar	2014	11187	8	32.2135	[Adventure, Drama, Science 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
...	...	...	...	...	...	...	...
14825	Shutter Island	2010	6559	7	15.8136	[Drama, Thriller, Mystery]	6.891075
31865	The Revenant	2015	6558	7	23.5012	[Western, Drama, Adventure, Thriller]	6.891059
12704	WALL·E	2008	6439	7	16.0884	[Animation, Family]	6.889173
13746	The Hangover	2009	6324	7	23.9474	[Comedy]	6.887287
42170	Logan	2017	6310	7	54.581997	[Action, Drama, Science Fiction]	6.887053

100 rows × 7 columns

Content Based Recommender System

- In Simple Recommender engine, we suffered from some significant limitations. It gave everybody the same suggestion, no matter the personal taste of the user. If a person who loves romantic films were to look at our top 15 list, they probably wouldn't like any of the movies.
- In a content-based recommender system, we have created an algorithm that will measure the similarity between films based on specific metrics and suggest movies that are more similar to a particular movie that a user liked. For that, we have used a TFIDF vectorizer to convert raw data into a count matrix. With the help of cosine similarities, we have measured a numerical quantity that denotes the similarity of two movies.
- We selected the top 25 movies based on the calculated similarity score and determined the vote of the 60th percentile movie. Then using this as the value of 'm', we measured each movie's weighted rating using IMDB's formula similar to what we did in Simple Recommender System.

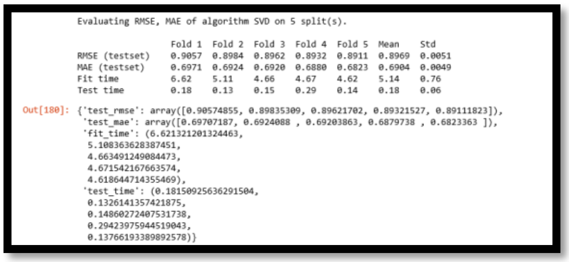


```
get_recommendations('The Dark Knight').head(10)

8031                          The Dark Knight Rises
6218                                  Batman Begins
7659                   Batman: Under the Red Hood
6623                                   The Prestige
1134                                 Batman Returns
8927                       Kidnapping Mr. Heineken
5943                                       Thursday
1260                                 Batman & Robin
2085                                      Following
9024     Batman v Superman: Dawn of Justice
Name: title, dtype: object
```

Collaborative Filtering

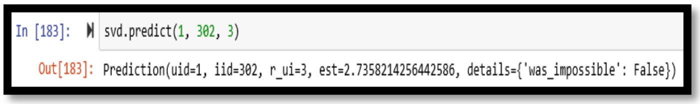
- Our content-based engine suffers from some significant constraints. It is only able to recommend movies that are similar to a particular film. This is, it cannot identify the user's tastes and make suggestions across genres. The engine we designed isn't personal in that it doesn't absorb a user's personal preferences and biases. Anyone who queries our engine for recommendations based on a movie will provide the same recommendations for that movie, no matter who he/she is.
- Collaborative filtering helps to make suggestions to movie fans, Watchers. It is based on the idea that users similar to me can be used to predict how much I will like a particular product or service those users have experienced, but I have not. We will use Surprise Library that uses Singular Value Decomposition Algorithm to minimize RMSE and give user's excellent recommendations.



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

                  Fold 1  Fold 2  Fold 3  Fold 4  Fold 5  Mean    Std
RMSE (testset)    0.9057  0.8984  0.8962  0.8932  0.8911  0.8969  0.0051
MAE (testset)     0.6971  0.6924  0.6920  0.6880  0.6823  0.6904  0.0049
Fit time          6.62    5.11    4.66    4.67    4.62    5.14    0.76
Test time         0.18    0.13    0.15    0.29    0.14    0.18    0.06

Out[180]: {'test_rmse': array([0.90574855, 0.89835309, 0.89621702, 0.89321527, 0.89111823]),
           'test_mae': array([0.69707187, 0.6924088 , 0.69203863, 0.6879738 , 0.6823363 ]),
           'fit_time': (6.621321201324463,
            5.108363628387451,
            4.663491249084473,
            4.671542167663574,
            4.618644714355469),
           'test_time': (0.18150925636291504,
            0.1326141357421875,
            0.14860272407531738,
            0.29423975944519043,
            0.13766193389892578)}
```



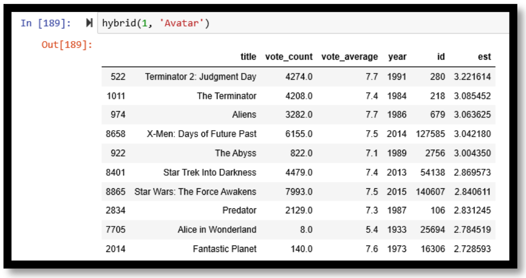
```
In [183]:  svd.predict(1, 302, 3)

Out[183]: Prediction(uid=1, iid=302, r_ui=3, est=2.7358214256442586, details={'was_impossible': False})
```

- We get a mean Root Mean Square Error of 0.8963, which is more than good enough.
- Prediction of 2.686 for a film with ID 302. Surprising aspect of this recommender method is that what the film is doesn't matter. It operates solely based on an allocated film ID and attempts to predict ratings based on how the other users predicted the film.

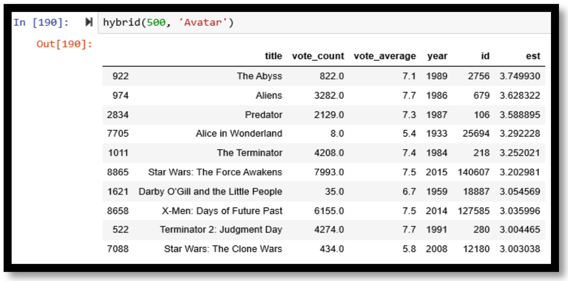
Hybrid Recommender

- Hybrid Recommender Combined the techniques of Content-based and Collaborative filtering and gave personalized hybrid recommendations to users based on their taste.



```
In [189]:  hybrid(1, 'Avatar')
Out[189]:
```

	title	vote_count	vote_average	year	id	est
522	Terminator 2: Judgment Day	4274.0	7.7	1991	280	3.221614
1011	The Terminator	4208.0	7.4	1984	218	3.085452
974	Aliens	3282.0	7.7	1986	679	3.063625
8658	X-Men: Days of Future Past	6155.0	7.5	2014	127585	3.042180
922	The Abyss	822.0	7.1	1989	2756	3.004350
8401	Star Trek Into Darkness	4479.0	7.4	2013	54138	2.869573
8865	Star Wars: The Force Awakens	7993.0	7.5	2015	140607	2.840611
2834	Predator	2129.0	7.3	1987	106	2.831245
7705	Alice in Wonderland	8.0	5.4	1933	25694	2.784519
2014	Fantastic Planet	140.0	7.6	1973	16306	2.728593



```
In [190]:  hybrid(500, 'Avatar')
Out[190]:
```

	title	vote_count	vote_average	year	id	est
922	The Abyss	822.0	7.1	1989	2756	3.749930
974	Aliens	3282.0	7.7	1986	679	3.628322
2834	Predator	2129.0	7.3	1987	106	3.588895
7705	Alice in Wonderland	8.0	5.4	1933	25694	3.292228
1011	The Terminator	4208.0	7.4	1984	218	3.252021
8865	Star Wars: The Force Awakens	7993.0	7.5	2015	140607	3.202981
1621	Darby O'Gill and the Little People	35.0	6.7	1959	18887	3.054569
8658	X-Men: Days of Future Past	6155.0	7.5	2014	127585	3.035996
522	Terminator 2: Judgment Day	4274.0	7.7	1991	280	3.004465
7088	Star Wars: The Clone Wars	434.0	5.8	2008	12180	3.003038

- We see that we get different suggestions for different users for our hybrid recommender while the film is the same. Our reviews are also more personalized and tailored to individual users.

Conclusion

The report mentioned all the data analysis techniques like cleaning, exploratory analysis, predictive modelling. Also, four recommendation engines are built using various algorithms like TFIDF, Singular Value Decomposition. At the end we brought ideas together from content and collaborative model and built our Hybrid model to give user a good recommender engine.