Springboard - DSC

Capstone Project II

Recommending the Right Movies for the Best User Experience

Final Report

Data Science Career Track

December 2020

Sathwik Kesappragada

Table of Contents

1. Introduction

Company XYZ is a video streaming platform that lets its members watch TV shows and movies without advertisements on any internet-connected device. They want to suggest their users to watch the best quality and most relevant content. They aim to do so by building a recommendation engine that matches on user preferences. Before that they want to know among the successful movies, which features lead to a highly rated film. Company XYZ wants to build a model based on movie attributes recorded such as duration, vote count, vote average, release date, revenue data that was collected from a movie database to recommend the right film. By identifying as many target variables as possible from all the features, the data science team in company XYZ could approach their suggested movies/tv shows more efficiently and effectively.

1.1 Objective

The objectives of this project are to:

- Explore and analyze film and ratings data for the XYZ video streaming platform
- Develop machine learning models that predict the rating
- Identify the key features that lead to a high rated film
- Identify the final model that captures most of the target variables
- Build a simple recommender system: collaborative filtering and content-based filtering

This report is divided into the following sections:

- Section 2: Dataset
- Section 3: Package Information
- Section 4: Data Wrangling
- Section 5: Exploratory Data Analysis
- Section 6: Machine Learning, Model Selection, & Hyperparameter Tuning
- Section 7: Recommendation Engines
- Section 8: Conclusion
- The programming codes for this report can be found at this Github Repository Movie-Recommender-System

1.2 Significance

By thoroughly exploring the dataset, we will identify key features that affect the rating of a film. We will also develop machine learning models that can be used by the software team of company XYZ to suggest accurate films to the users and approach them with better efficiency and high click rate success/accuracy.

2. Dataset

2.1 Data Description

The datasets are retrieved from the website kaggle, owned by Google, is an online platform of data science and machine learning professionals who compete, discover and post datasets, research and build models all in one setting. The movie dataset used for this project was last updated November 11, 2017. It consists of 45,466 rows and 23 columns, 6 of them numerical columns and 17 of them are categorical columns. A description of each of the columns is provided in *Table 1.1*. The rating dataset was made up of 2,600,000 rows and 4 columns.

Table 1.1 Description of dataset

No.	Variable Name	Variable Description	Data Summary
1	adult	Adult or Not	bool, False= Not Adult, True= Adult
2	belongs_to_collection	franchise/series a film belongs to	object, 4494 non-null
3	budget	cost of film	float, 1223 unique values
4	genres	categories associated with each film	object, 45466 non-null
5	homepage	official website of film	object, 7669 unique values
6	id	film ID	int, 45430 unique values
7	imdb_id	id of film associated in the IMDB database	int, 45413 unique values
8	original_language	language the film officially was shot in	object, 89 unique values
9	original_title	name of the film upon release	object, 43367 unique values
10	overview	description of film	object, 44303 unique values
11	popularity	score of how well know the film is	float, 43745 unique values
12	poster_path	url of the film poster image	object, 45021 unique values
13	production_companies	companies that contributed in the making of the film	object, 45463 non-null
14	production_countries	countries where the film was released/shot in	object, 45463 non-null
15	release_date	theatrical release date	datetime, 17333 unique values
16	revenue	total return of the film	float, 6863 unique values
17	runtime	duration of film in minutes	float, 353 unique values
18	spoken_languages	languages spoken in the film	object, 45460 non-null

19	status	state of film (released, rumored, post-production, etc.)	object, 6 unique values
20	tagline	punch line/catch phrase	object, 20283 unique values
21	video	indicates if TMDB has video of film	bool, True: it contains film, False: it does not have film
22	vote_average	average score of film	float, 92 unique values
23	vote_count	number of votes by users	int, 1820 unique values

3. Package Information

Here, we used JupyterNotebook (6.03) to run all the code. Pandas (1.0.5), Matplotlib (3.2.2), Numpy (1.18.5), Seaborn (0.11.0) were installed as basic package. Scikit-learn (0.24.0) and sklearn (0.0) were installed as the machine learning library.

4. Data Wrangling

4.1 Dataset Information

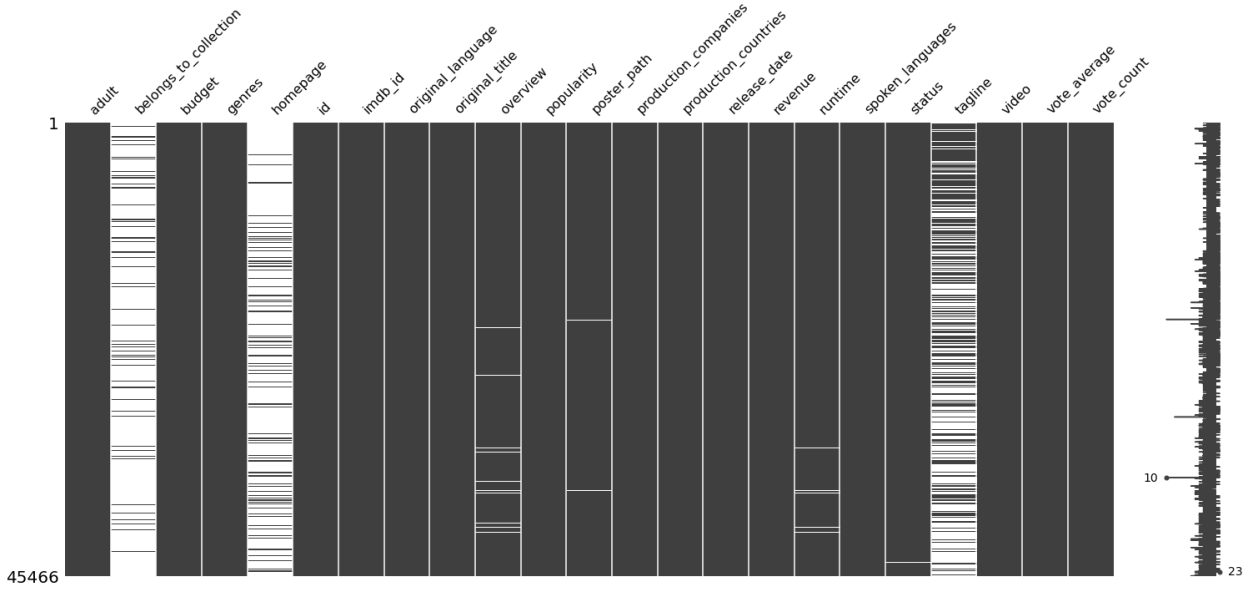
General information about the original dataset can be found in *Table 4.1*, sorted by the percentage of unique values by column. It contains columns such as unique value count, missing value percentage, zero value percentage, description, and datatype are provided in the table below.

Table 4.1 General information of dataset

Variable Name	Counts	Unique Value Percentage	Missing Value Percentage	Num of Zeros Percentage	Data Type
imdb_id	45499	99.9208	0.037391	0.000000	object
id	45466	99.9208	0.00000	0.000000	int64
poster_path	45080	99.8691	0.848986	0.000000	object
overview	44512	99.5303	2.098271	0.000000	object
tagline	20412	99.3680	55.104914	0.000000	object
homepage	7782	98.5479	82.883913	0.000000	object
popularity	45461	96.2253	0.013197	0.145278	float64
original_title	45466	95.3834	0.00000	0.000000	object

Variable Name	Counts	Unique Value Percentage	Missing Value Percentage	Num of Zeros Percentage	Data Type
release_date	45379	38.1961	0.191352	0.000000	datetime64[ns]
revenue	15460	15.0968	0.013197	83.715606	float64
vote_count	45460	4.0035	0.013197	6.374642	float64
budget	45466	2.6899	0.006598	80.453445	float64
runtime	15203	2.3219	0.578454	3.429452	float64
vote_average	45466	0.2023	0.013197	6.592560	float64
original_languages	45455	0.1958	0.024194	0.000000	object
status	45379	0.0132	0.191352	0.000000	object

*variable that has missing value percentage higher than 50% is highlighted in yellow



4.2 Data Cleaning

The shape of the dataset is 45466 rows by 23 columns. Some columns were in the form of a stringified JSON object that needed to be converted into lists. The data frame struggled with any analysis pertaining to these columns (genres, production_countries, belongs_to_collection) because they were unhashable lists. Additionally, all columns with misclassified data types were converted into proper types (integer, string, float) after removing unvaluable data. Release_date

was broken up into day, month, year to extract more findings. Duplicates were identified by using a combination of release_date and original_title as a key and dropped.

Variable Names	Missing Value Percentage	Process Method
genres	0.0	data type correction
production_companies	0.006598	data type correction, (one-hot-encoding)
production_countries	0.0	data type correction, (one-hot-encoding)
belongs_to_collections	0.0	data type correction
spoken_languages	0.0	data type correction, imputed with empty string
release_date	0.191352	data type correction
budget	0.006598	data type correction
popularity	0.013197	data type correction
rating (ratings)	0.0	multiplied by 2
timestamp (ratings)	0.0	data type correction
id (ratings)	0.0	data type correction
id	0.0	data type correction

* some features listed are from ratings dataset and are labeled

5. Exploratory Data Analysis

5.1 Summary Statistic

Statistics such as count, mean, standard deviation, percentiles of each numerical variable were compiled in *Table 5.1*.

Table 5.1 Summary statistics

variable name	count	mean	std	min	25%	50%	75%	max
budget	45430.0	4224828	17428530	0.0	0.000	0.0	0.000	380000000
popularity	45430.0	2.921206	6.006708	0.0	0.385872	1.127238	3.678128	547.4883
revenue	45430.0	11212880	64352130	0.0	0.000	0.0	0.00000	2787965000
runtime	45173.0	94.1243	38.41554	0.0	85.0000	95.000	107.000	1256.000
vote_average	45430.0	5.618329	1.924139	0.0	5.0000	6.000000	6.800000	10.000000
vote_count	45430.0	109.936	491.4663	0.0	3.0000	10.0000	34.00000	1.407500e+04
year	45346.0	1991.883	24.05304	1874.0	1978.000	2001.000	2010.000	2.020000e+03
day	45346.0	14.20948	9.283747	1.0	6.000000	14.00000	22.00000	31.00000
month	45346.0	6.459225	3.628039	1.0	3.000	7.0000	10.00000	12.00000
num_genres	45430.0	2.003214	1.130713	0.0	1.000	2.00000	3.000000	8.000000

5.2 Overall Distribution

The overall distribution of numerical variables is visualized in *Figure 5.2*.

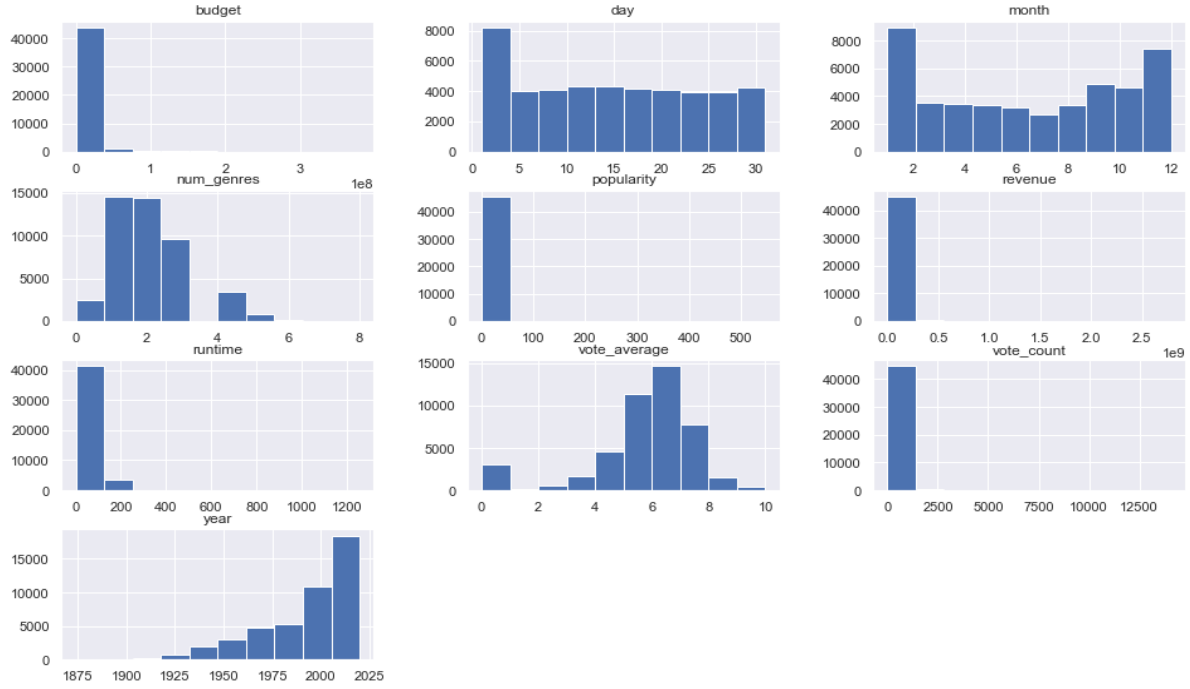


Figure 5.2 Distributions of numerical features

Please note that this was visualized after data type correction

5.3 Release Dates

First question that pops up in mind when thinking about an upcoming film is the release date. Movie makers are always concerned and meticulous about choosing an appropriate date for release, making sure the time periods do not overlap with the competition and anticipating significant amounts of attention surrounding the promotion.

- Oldest film in the dataset: Passage of Venus (1874-12-09)
- Latest film in the dataset: Avatar (2020-12-16) (yet to release)

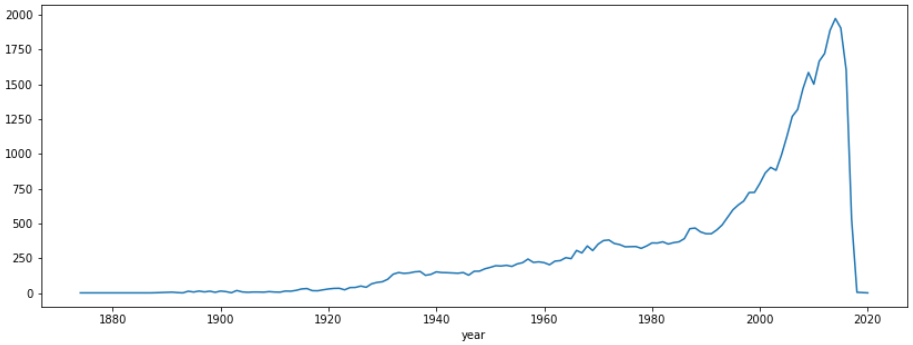


Figure 5.3, With technological advancements every year, the filmmaking process is becoming much easier that more movies are being produced at quicker speeds and this can be displayed by the graph on the left. In the age of video streaming platforms, content creators want to get their works out.

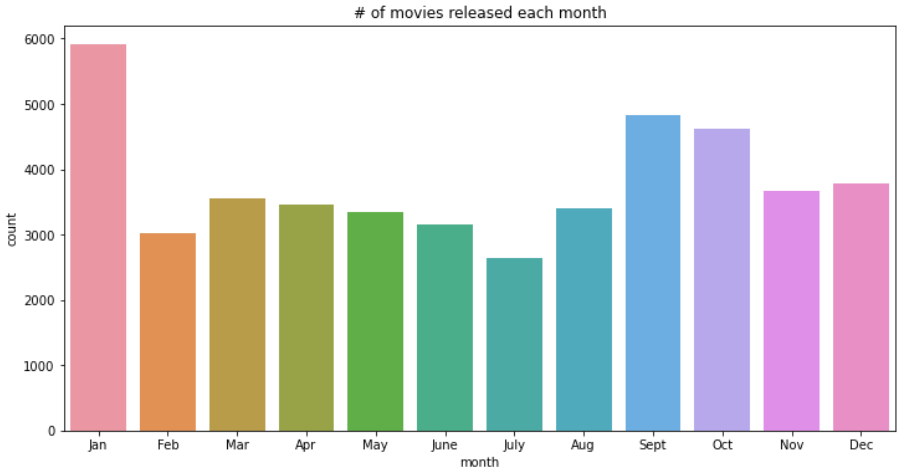


Figure 5.3, After splitting the release date feature into multiple subparts, the months that have the greatest number of movies released can be visualized. One may expect summer for having the most films announced, but the graph on the left does not agree. Most theatrical release dates of films fall in the months of January, September, and October. Subpar movies that were not released in the previous year are released at the beginning of a new year.

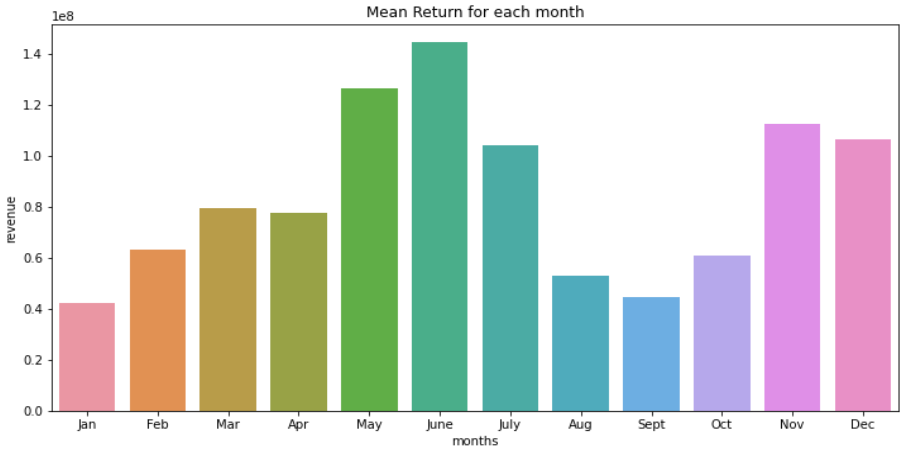


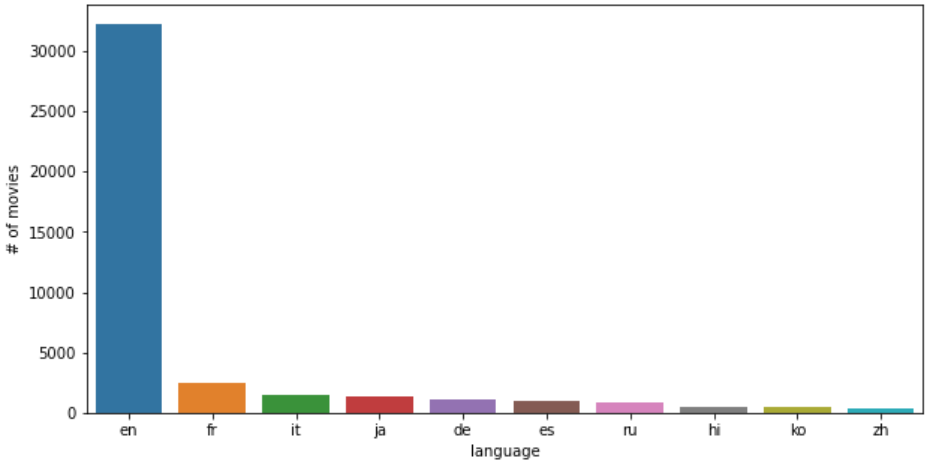
Figure 5.3, Continuing with examining the months feature, summer is usually the time when films make the most money in return. There is always hype surrounding a film that gets released around the time most people want to spend their time outside being active. May, June, and November are the top three months with the highest turnout. Movie theaters are expected to get busy during holiday season.

5.4 Films Released by Country and Language

Most of the films (~40%) in the dataset being inspected are from Hollywood. There exists a small subset (~14%) of films without a country label and British Cinema appears third on the list (~5%). Together with, English, French, and Italian dominating in the spoken languages section, much of the data set focuses on English flicks. Language and country of origin are imperative characteristics of any movie. Film has become a widely known form of communication and highly influential art.

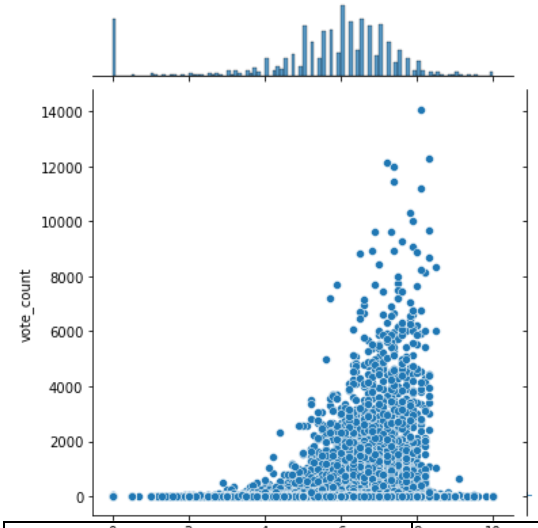
country	# of movies
United States of America	17841
-	6279
United Kingdom	2238
France	1653
Japan	1354
Italy	1030
Canada	840
Germany	748
Russia	735
India	735

Figure 5.4 Film Count by Language



5.5 Vote Count & Vote Average

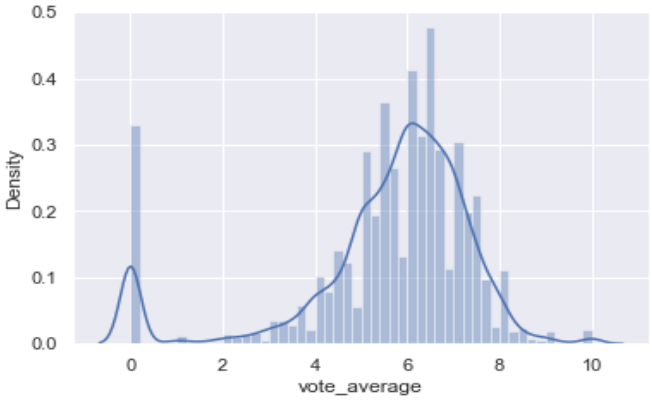
The chart on the left shows the films with the greatest number of votes from the movies dataset. Three out of the top then, Avengers, Deadpool, Guardians of the Galaxy, are from the Marvel franchise.



Relationship between vote count and vote average is visualized on the left. A dual sided histogram and scatterplot suggests that more votes are likely to yield to a higher vote average and resembles a true rating of a film.

Figure 5.5

Title	Vote Count
Fight Club	9678
The Hunger Games	9634



The distribution of vote average is shown on the left. The drawn line is not only attempting to symbolize a Gaussian distribution, but also indicates the outliers in the dataset. A huge subset of vote average falls in the range of 4.5 to 6.5.

Figure 5.5

5.6 Popularity, Budget & Revenue

After a film gets released, success metrics are usually monitored to determine the results.

Title	Popularity
Minions	547.488298
Wonder Woman	294.337037
Beauty and the Beast	287.253654
Baby Driver	228.032744
Big Hero 6	213.849907
Deadpool	187.860492
Guardians of the Galaxy Vol. 2	185.330992
Avatar	185.070892
John Wick	183.87034
Gone Girl	154.801009

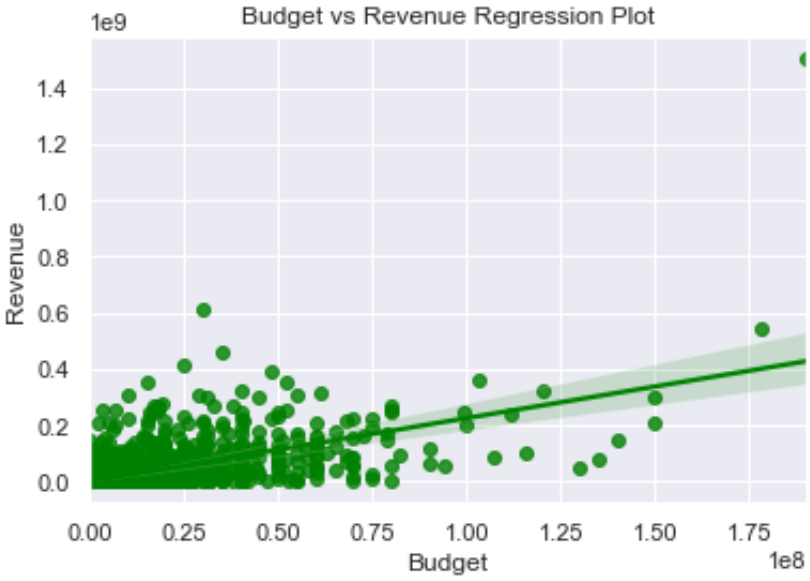
Minions, Wonder Woman, and Beauty and the Beast are the most popular films according to TMDB's Popularity Score.

Over a long period of time, Avatar, Star Wars: The Force Awakens, and Titanic made the most in return accumulating billions of dollars.

Title	Revenue
Avatar	2787965000
Star Wars: The Force Awakens	2068224000
Titanic	1845034000
The Avengers	1519557910
Jurassic World	1513529000
Furious 7	1506249000
Avengers: Age of Ultron	1405404000
Harry Potter and the Deathly Hallows: Part 2	1342000000
Frozen	1274219000
Beauty and the Beast	1262886000

Title	Budget
Pirates of the Caribbean: On Stranger Tides	380000000
Pirates of the Caribbean: At World's End	300000000
Avengers: Age of Ultron	280000000
Superman Returns	270000000
Transformers: The Last Knight	260000000
Tangled	260000000
John Carter	260000000
Spider-Man 3	258000000
The Lone Ranger	255000000
The Dark Knight Rises	250000000

Two out six of the Pirates of the Caribbean movies are the two highest budgeted films in the dataset. Avengers: Age of Ultron and Superman Returns follows.



This regression plot emphasizes on the relationship between budget and revenue and reveals that not every highly budgeted film result in a profit. Most film budgets are below 50 million.

Figure 5.6

*This graph was made with movies that have only one genre

5.7 Runtime

The average length of a film is about 1 hour and 34 minutes. The top ten longest and shortest films of the dataset are listed below. The shortest films made were outcomes of the initial spike in filmmaking during the late 1800s and early 1900s. The first ever films made lacked present day technology. Back then, one minute of pictures displayed sequentially was astonishing. Several of these long films are between 15 hours to 20 hours and are all one-episode TV shows.

Title	Runtime
Mr. Edison at Work in His Chemical Laboratory	1.0
Grandma's Reading Glass	1.0
What Happened on Twenty-Third Street, New York City	1.0
The Magician	1.0
Panorama pris d'un train en marche	1.0
Divers at Work on the Wreck of the "Maine"	1.0
After the Ball	1.0
Between Calals and Dover	1.0
The Surrender of Tournavos	1.0
Blacksmith Scene	1.0

Title	Runtime
Centennial	1256.0
Jazz	1140.0
Baseball	1140.0
Berlin Alexanderplatz	931.0
Heimat: A Chronicle of Germany	925.0
The Story of Film: An Odyssey	900.0
Taken	877.0
The War	874.0
The Roosevelts: An Intimate History	840.0
Seventenn Moments in Spring	840.0

5.8 Genres

Not only were 'genres' a tough feature to access because it was a unhashable list, but also there were 4065 different combinations of 'genres' entered, some ranging from one to eight genres per a movie. Observing films with one genre was much simpler to accomplish and draw analysis from. Drama had the highest vote average count and vote average mean. Comedy and Documentary follows. TV movie, History, and Foreign had the least number of films. Drama and Romance, Comedy and Drama, Comedy and Romance are the most popular pairs of genres.

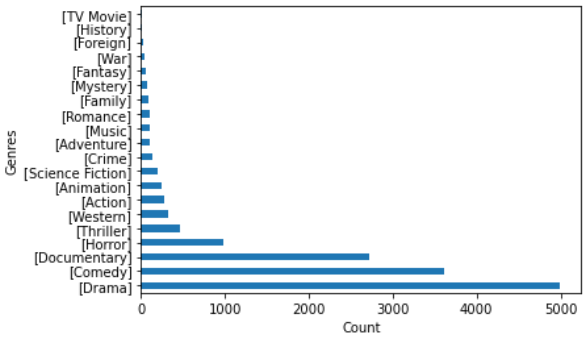
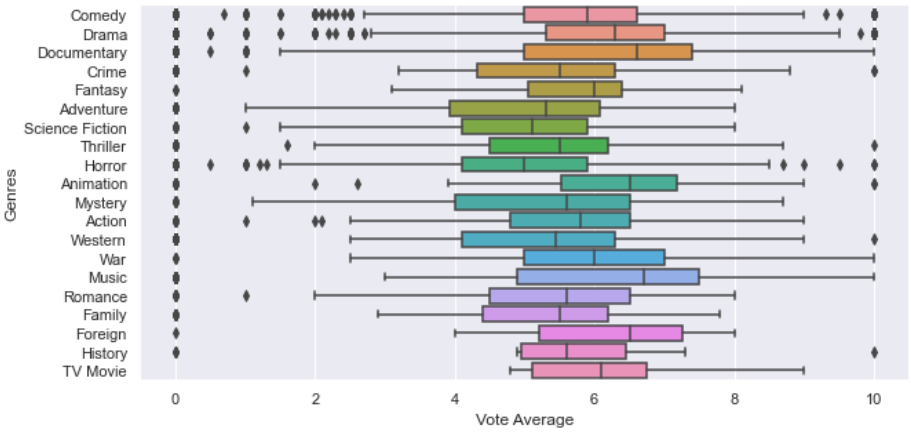
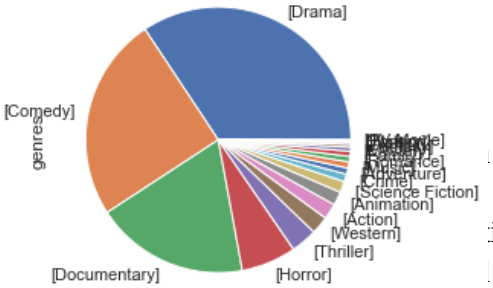


Figure 5.8



5.9 Merging Movies and



was to identify valuable features that affect the rating of a so it could help improve the recommendation system's ings and null user IDs in records represents users not having unimportant.

5.10 Variable Correlation Coefficient

After plotting heatmap *Figure 5.10*, not much correlation coefficient was found among the variables.

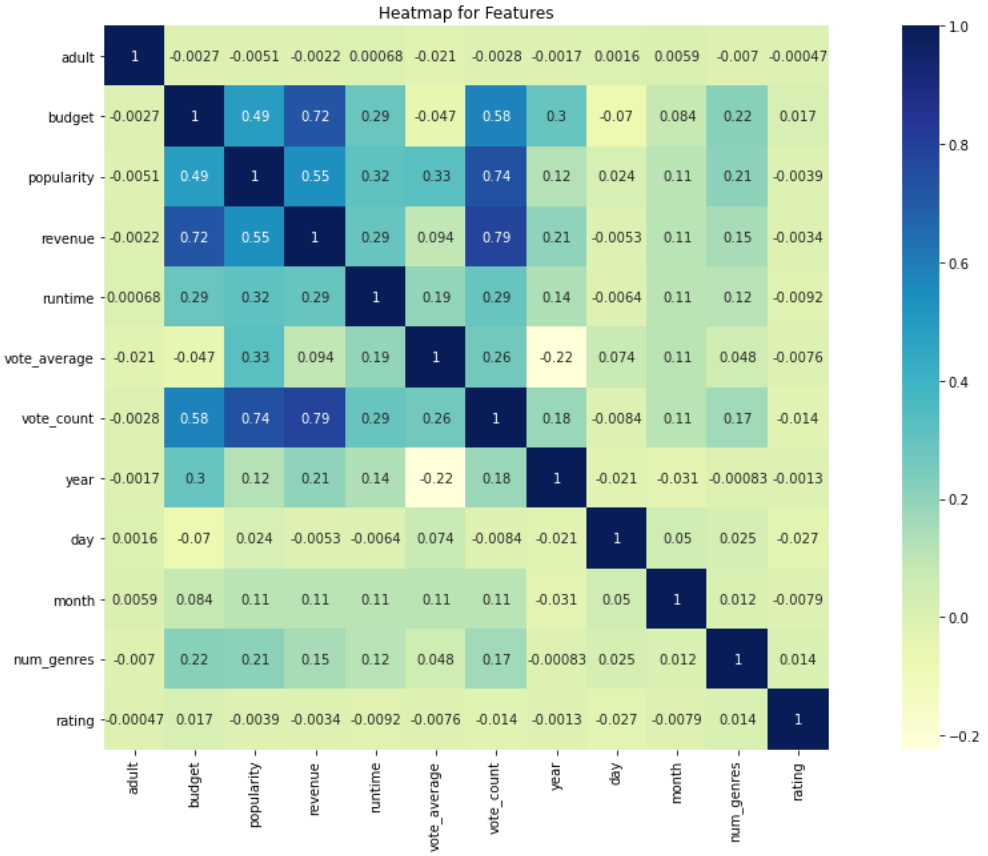


Figure 5.10 Heatmap of Correlation coefficients between variables

6. Machine Learning: Regression – Predicting Movie Ratings

Models ran on a small fraction (2%) of the merged data set due to computational power. Considering the objective of determining key features, variables such as ‘revenue’, ‘budget’, ‘vote_count’, and ‘runtime’ needed to contain values greater than 0. The data was split 75%/25% training/testing sets to run against ‘ratings’ variable. The process of feature engineering in total added 24 variables: 10 for each of the one hot encoded categorical variables, production countries and production companies, top 3 genres, and ‘is_Summer’.

Shape of Test set: (8130,35)

Shape of Train set: (2711, 35)

6.1 Data Preprocessing and Feature Selection/Engineering

- Categorical variables 'production_companies' and 'production_countries' were converted into dummy variables and only top 10 were selected
- 'Original_language' transformed into a binary feature, indicating 1 if the film was originally shot in English and 0 otherwise
- 'Is_Summer' is a binary feature added indicating if the film's release date month falls during the summertime (May – September)
- Any nulls in 'runtime' were imputed using the mean value
- Nulls of 'year', 'month', 'day' were dropped
- Categorical variables such as 'original_title', 'adult', 'genres', 'homepage', 'belongs_to_collection', 'production_companies', 'production_countires', 'spoken_languages' are all dropped
- "Is_(Drama, Comedy, Thriller)" is a binary feature added to check if the film listed as one of these top 3 genres

6.2 Model Metrics

Model Metrics using to measure model performance and their meaning:

- Explained Variance Score: measures the difference between the model and actual data
- Mean Absolute Error (MAE): mean of all absolute errors
- Mean Squared Error (MSE): distance between the regression line and predicted values (set of errors)
- Mean Squared Log Error (MSLE): percentual difference between the model and actual data
- R^2 score: (coefficient of determination) how well the data is to a fitted regression line, indicates the variance
- Median Absolute Error: median differences between predicted and actual observations
- Root Mean Squared Error (RMSE): square root of MSE, direct relationship to R^2, measures the differences between residuals

6.3 Model Performance, Hyperparameter Tuning, & Evaluation

Baseline Model Metrics:

At first, the models were trained using baseline implementation, in other words hyperparameters were left as default. The evaluations were all conducted over the same train/test split 75%/25% with 5folds Cross Validation. (best model is highlighted in yellow)

Hyperparameter Tuned Metrics:

On the contrary, hyperparameters were tweaked and given a range of values for its arguments. Model performance improved in all aspects, but the results remain similar.

	Model	Explained Variance Score	Mean Absolute Error (MAE)	Mean Squared Error (MSE)	Mean Squared Log Error (MSLE)	R^2 score	Median Absolute Error	RMSE	Final RMSE
1	RF	0.132785	1.570660	4.025107	0.100195	0.132032	1.256840	2.006267	2.004275
1	RF	0.131031	1.572565	4.033086	0.099487	0.130311	1.280914	2.008255	
2	GB	0.129824	1.580481	4.039149	0.101129	0.129003	1.255838	2.009763	2.004140
2	GB	0.089178	1.636326	4.228983	0.105470	0.088068	1.241732	2.043963	
3	XGB	0.129519	1.582284	4.052281	0.100322	0.126172	1.294407	2.013028	2.064004
3	XGB	0.123917	1.577920	4.066276	0.100084	0.123154	1.276165	2.106501	

6.4 Feature Importances

As displayed in the chart above, Random Forest outperforms the other models in all categories. Even though the R^2 score is low here, conclusions can still be drawn on statistically significant predictors. Regardless of the value, predictors that affect the model's performance are meaningful and can be displayed for each model by the figures below.

Random Forest

Figure 6.4 For Random Forest, Revenue, Runtime, Popularity are the top 3 most important features

Gradient Boosting

Figure 6.4 For Gradient Boosting, Revenue, Popularity, and Budget are the top 3 most important features

XGBoost

Figure 6.4, For XGBoost, these are the top 3 most important features:

production companies**:** 'Columbia Pictures','Intermedia Films', 'Warner Bros.', 'C-2 Pictures', 'IMF Internationale Medien und Film GmbH & Co. 3 Produktions KG', 'Mostow/Lieberman Productions',

production companies: 'Columbia Pictures', 'Amblin Entertainment', 'Columbia Pictures Corporation, 'Parkes_MacDonald Image Nation',

production companies: 'Village Roadshow Pictures', 'Robert Simonds Productions', 'Warner Bros.', 'Phoneix Pictures', 'Underground', 'Proposal Productions'

The idea behind this machine learning process was to identify which metrics have the most value when it comes to predicting the rating of a film. From the feature importances, revenue and popularity appeared to be in the top 3 for two of the higher performing models. Furthermore, those two variables affect movie reviews.

7. Recommender Systems

Movies data file is merged with credits and keywords to make a combined dataset.

Credits: credits for a particular film (Director, Cast, Characters, etc.)

Keywords: plot keywords of a film

7.1 Content Based Filtering

Taking a fraction of the combined dataset and filling in the null values for categorical columns such as overview and tagline with empty strings will help the system moving forward. In effect, create a column by combining overview and tagline to get a soup of details. Then, another column is created with a combination of keywords, cast, genres, and details after all the variables agree. Using TfidfVectorizer fit transform the combination of all aspects of a movie and cosine similarity to match films that are similar. Below are the recommendations for the movie Toy Story.

Figure 7.1

```
print(get_recs('Toy Story',cosine_sim, indices))
```

executed in 36ms, finished 03:32:18 2020-12-29

```
1                          Jumanji
2                 Grumpier Old Men
3                Waiting to Exhale
4       Father of the Bride Part II
5                             Heat
6                          Sabrina
7                      Tom and Huck
8                      Sudden Death
9                         GoldenEye
10           The American President
11      Dracula: Dead and Loving It
12                           Balto
13                           Nixon
14                Cutthroat Island
15                          Casino
16           Sense and Sensibility
17                       Four Rooms
18   Ace Ventura: When Nature Calls
19                     Money Train
```

7.2 Recommender using Correlation

Another recommendation technique is creating a matrix with user IDs as rows and film titles as columns, making it easier to see what rating each user has given to every movie. Null records indicate a user has yet to watch that movie. On the side, create another data frame with movie titles, rating, and number of ratings, sorted by the count. Then taking a specific movie (column of all ratings) from the first matrix create a variable and correlate it with the full movie matrix. With that result using the second data frame created, identify films that are highly correlated to each other, but also have a count of more than 100 ratings. Below are the results for Jumanji.

original_title	Correlation	num_ratings
EVA	1.0	1062
Shiloh	1.0	578
Saving Grace	1.0	263
Du rififi chez les hommes	1.0	483
Juste une question d'amour	1.0	972

Figure 7.2

7.3 Hybrid Technique: Scaled Weighted Average and Scaled Popularity Score

Like TMDB, IMDB has their own scaling method. They rank their films using a set formula and it is shown below. Using this ranking method and setting the cut off to be at the 90th percentile for vote counts, the weighted average calculated can help identify the top films. Now with the help of MinMaxScaler, 50% importance can be given to valuable features such as popularity and the weighted average just calculated to retrieve a new list of to be recommended based off ratings and popularity score. Below is the list of top 10 movies.

Weighted Rating (WR) =

$$(\frac{v}{v+m} * R) + (\frac{m}{v+m} * C)$$

where,

- v: number of votes for each film
- m: minimum number of votes
- R: the average rating of a fimn
- C: the mean vote throughout the dataset

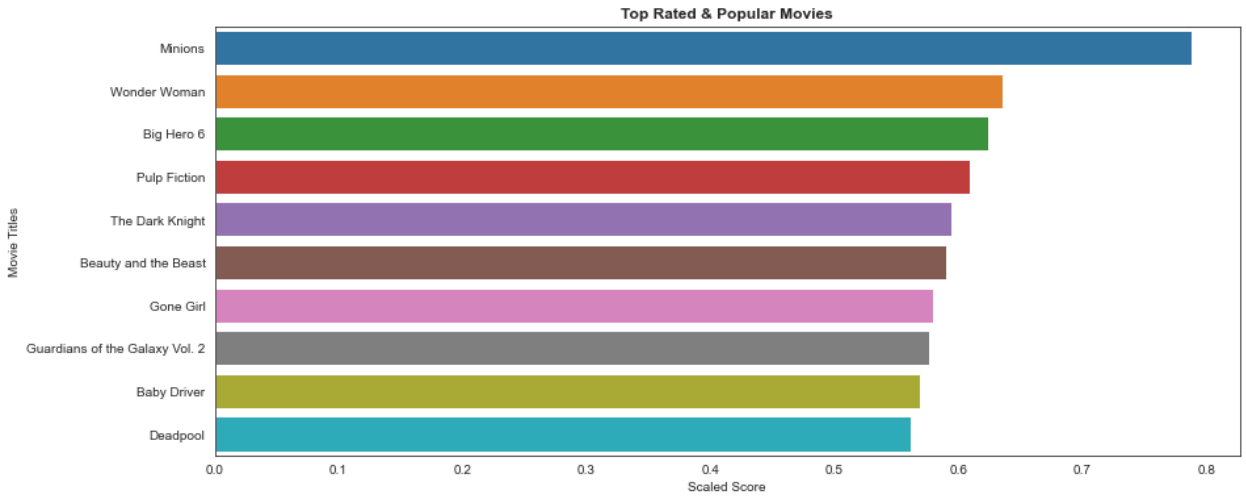


Figure 7.3, Minions, Wonder Woman, Big Hero 6 are the best according to the scaled variables.

7.4 Simple Collaborative Filtering using KNearest Neighbor

Similarly, to content-based filtering, another movie matrix can be created with movie titles as rows and user IDs as columns. Next, converting the pivot table into an array matrix using scipy library will help with calculating neighbors of observed films by cosine similarity

and Euclidean distance. Whichever films are the closest to the film entered, based on their distance five films will be recommended. Below is an example with the film 300.

```
Recommendations for 300:

1: Rocky Balboa, with distance of 0.6608025529091446:
2: The Prestige, with distance of 0.6872639724192523:
3: Madagascar, with distance of 0.6942923891642457:
4: Whale Rider, with distance of 0.6973719879556649:
5: Blood: The Last Vampire, with distance of 0.7076322952914149:
```

Figure 7.4

8. Conclusion

This report highlights and visualizes lot of metrics that are monitored surrounding film. After performing some data wrangling, processing, and feature engineering, the results give insights on which elements play a key part in a successful film.

Drama, Comedy, and Thriller are the most popular genres in the dataset. Minions and Wonder Woman are the top films respective to popularity and ratings. The Shawshank Redemption and Dilwale Dulhania Le Jayenga are the top films according to IMDB's ranking method. There exists a small correlation between vote count and vote average. A high vote count does not entail a good film. Inception and The Dark Knight have the most votes. Most movies get released in January, September, October while the most returns are made during the summer. Avatar and Star Wars: The Force Awakens both broke 2-billion-dollar mark for revenue.

Model performance can be improved with the addition of more features/variables such as figuring out the weekday based on the day of release. Classification can be executed to determine whether a film was a hit or not depending on relevant features. The hyperparameter tuned Random Forest was the best performing model with an R^2 of 13.3%. Revenue, runtime, popularity are influential predictors. Additionally, there were four recommender systems built with different ideas and algorithms. All these methods combined may be how production level engines are run. Future work involves deployment.