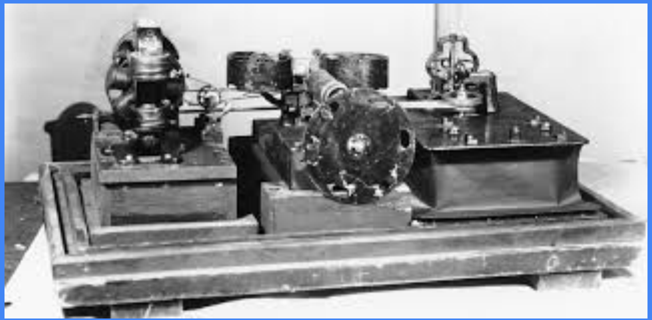
Recommending the Right Movies for the Best User Experience

By: Sathwik Kesappragada

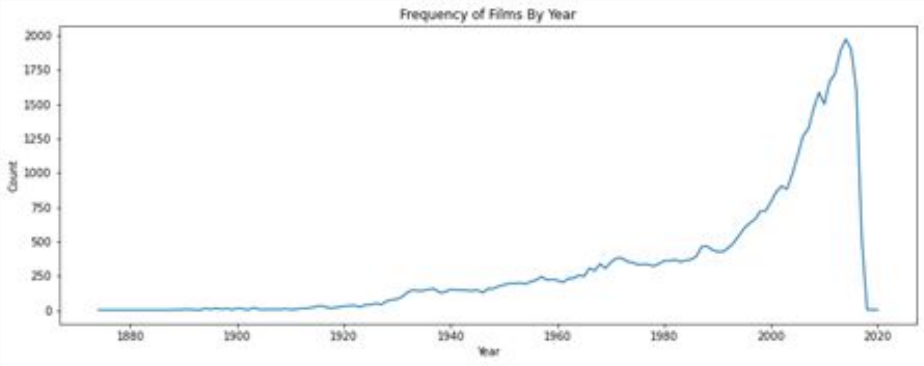
Problem Statement: Company X is a video streaming platform that lets its members watch TV shows and movies. They aim to build a recommendation engine that matches on user preferences by building a model that identifies important features of a film which influence rating.

- Explore and Analyze the movies and ratings datasets from database
- Develop Machine Learning models to predict on the rating
- Identify key features
- Build Recommender System

Film History



- Cinema was an accidental art that evolved as photography developed.
- Thomas Edison and William Dickson brought the world motion picture with kinetograph and kinetoscope.
- Lumiere Brothers brought the cinematographe (all in one camera)
- George Melles started editing, filmmaking has arrived



How does the data look?

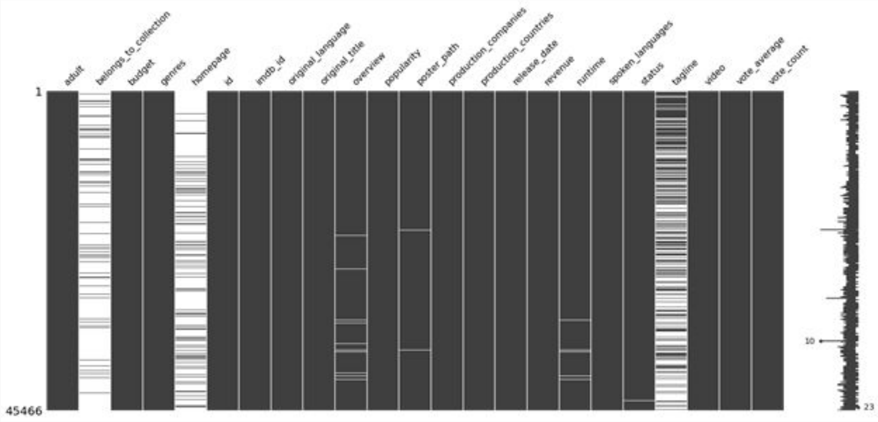
```
RangeIndex: 2600000 entries, 0 to 2599999
Data columns (total 4 columns):
 #   Column     Dtype
---  ------     -----
 0   userId     int64
 1   id         int32
 2   rating     float64
 3   timestamp  datetime64[ns]
dtypes: datetime64[ns](1), float64(1), int32(1), int64(1)
```

- Data Wrangling and Exploratory Data Analysis was done on movies
  - 45466 rows, 23 columns (Movies, Characteristics)
- Models with movies and ratings
  - 2.6 million rows, 4 columns
  - Combined: (1883206,30) (Ratings, Features)
- Recommender working with credits, ratings, and keywords
  - Combined: (1144418, 26)

```
Index: 45466 entries, Toy Story to Queerama
Data columns (total 23 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   adult                  45466 non-null  object
 1   belongs_to_collection  4494 non-null   object
 2   budget                 45466 non-null  object
 3   genres                 45466 non-null  object
 4   homepage               7782 non-null   object
 5   id                     45466 non-null  object
 6   imdb_id                45449 non-null  object
 7   original_language      45455 non-null  object
 8   original_title         45466 non-null  object
 9   overview               44512 non-null  object
 10  popularity             45461 non-null  object
 11  poster_path            45080 non-null  object
 12  production_companies   45463 non-null  object
 13  production_countries   45463 non-null  object
 14  release_date           45379 non-null  object
 15  revenue                45460 non-null  float64
 16  runtime                45203 non-null  float64
 17  spoken_languages       45460 non-null  object
 18  status                 45379 non-null  object
 19  tagline                20412 non-null  object
 20  video                  45460 non-null  object
 21  vote_average           45460 non-null  float64
 22  vote_count             45460 non-null  float64
dtypes: float64(4), object(19)
```

Data Cleaning

- Variables that were misclassified data types were converted into their respective data types (string, float, int)
- Runtime, Tagline, Homepage had the most missing data
- Duplicate values were found only using release date and title as a combination
- Video, Revenue, Budget had the most zeros in their columns
- Each film had a different number of genres listed

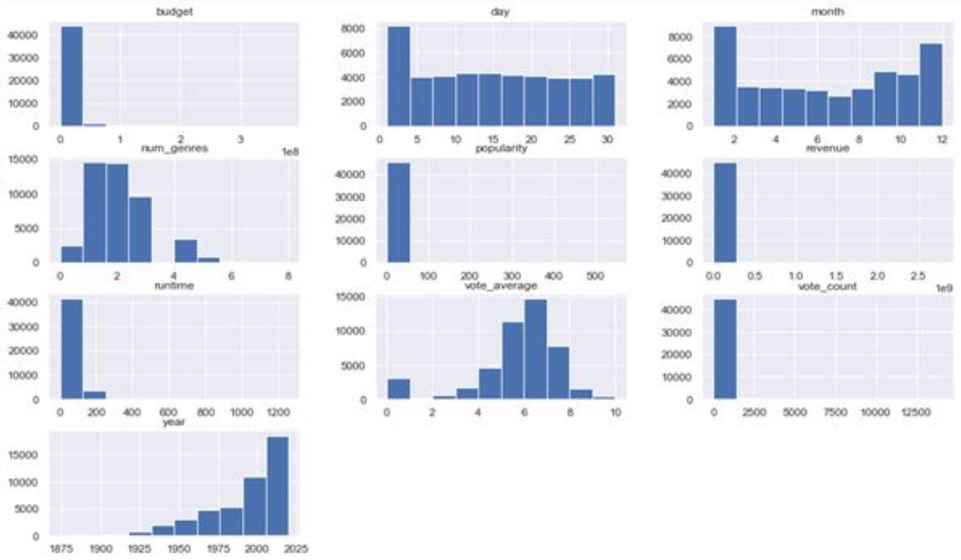


Variable Names	Missing Value %	Process Method
genres	0.00	Data type correction
production_companies	0.01	Data type correction
production_countries	0.00	Data type correction
belongs_to_collection	0.00	Data type correction
spoken_languages	0.00	Data type correction
release_date	0.19	Data type correction
budget	0.01	Data type correction
popularity	0.01	Data type correction
rating (ratings)	0.00	Multiplied by 2
timestamp (ratings)	0.00	Data type correction
id (ratings)	0.00	Data type correction
id	0.0	Data type correction

Exploring the data

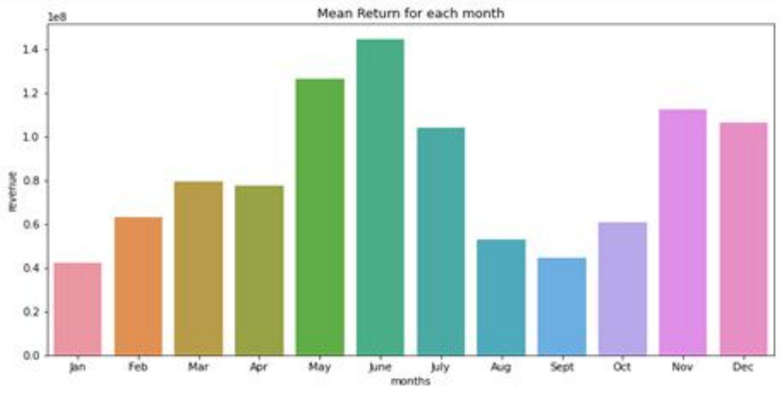
- Summary statistics such as count, min, and max and the distributions of every numerical variable
- Release date column was split into three subparts: day, month, year
- Clearly not all counts are aligned and demonstrate missing data
- Irrelevant records were dropped

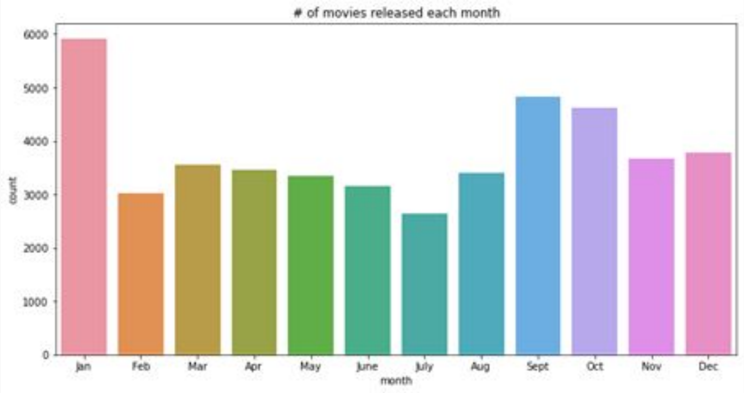
variable name	count	mean	std	min	25%	50%	75%	max
budget	45430.0	4224828	17428530	0.0	0.000	0.0	0.000	380000000
popularity	45430.0	2.921206	6.006708	0.0	0.385872	1.127238	3.678128	547.4883
revenue	45430.0	11212880	64352130	0.0	0.000	0.0	0.00000	2787965000
runtime	45173.0	94.1243	38.41554	0.0	85.0000	95.000	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	14075.00
year	45346.0	1991.883	24.05304	1874.0	1978.000	2001.000	2010.000	2020.00
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



Release Dates

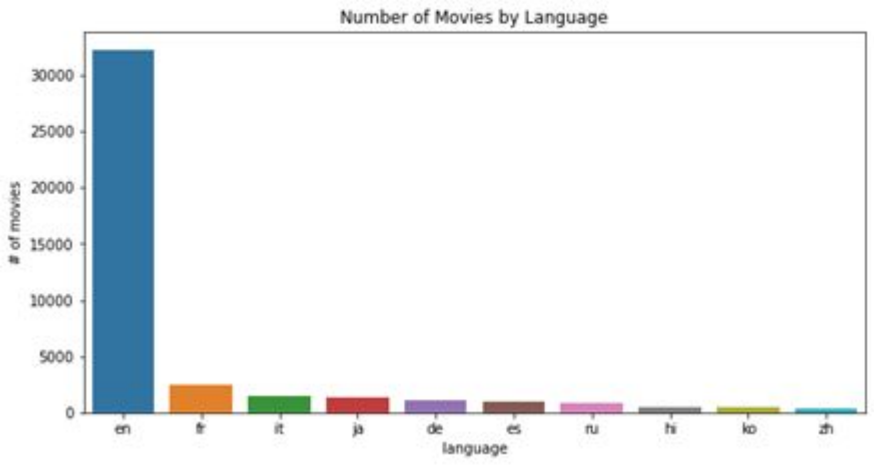
- Most theatrical release dates of films fall in the months of January, September, October
- January is called the dump month where all the subpar films from the previous year gets released
- May, June, November are the top 3 months with the highest turnout
- June and July have the highest median returns, mostly because summer vacation
- September is the worst, beginning of school





Countries and Languages

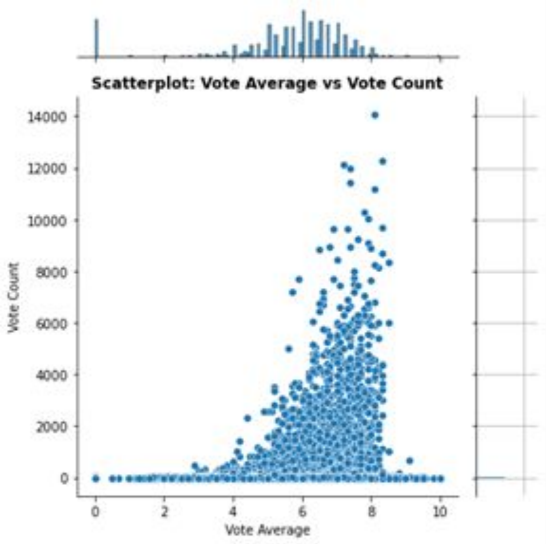
- 40% of films in the dataset are from Hollywood
- British Cinema appears 3rd on the list at 5%
- English, French, Italian are the most appearing languages in the dataset
- Together, much of the data focuses on English flicks.

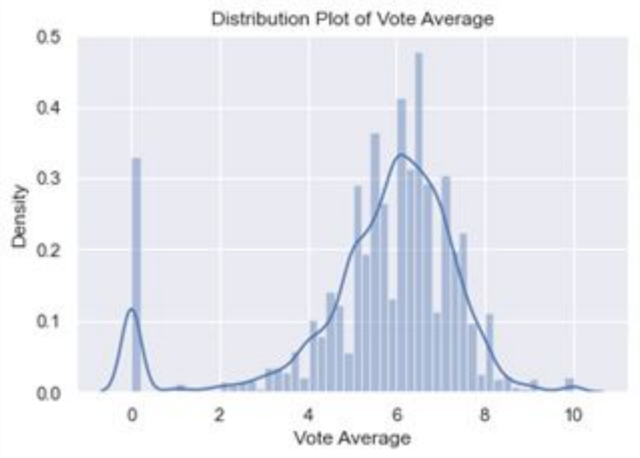


country	# of movies
United States of America	17841
-	6279
United Kingdom	2238
France	1653
Japan	1354
Italy	1030
Canada	840
Germany	748
Russia	735

Vote Count & Vote Average

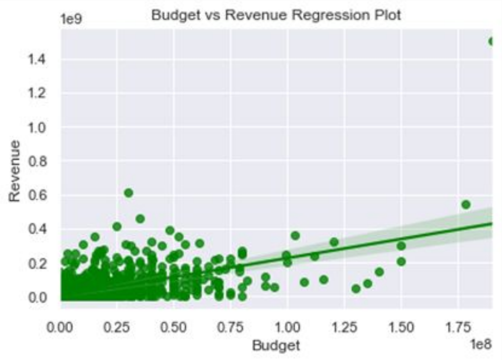
- The more votes are likely to yield to a higher vote average and resembles a true rating of a film
- Distribution of vote average points out the outliers and most of the data falls in the range of 4.5 to 6.5





Popularity, Budget & Revenue

- Top 5 charts of every variable
- 2 out of the 6 Pirates of the Carribean films were the most expensive to produce
- Avatar, Star Wars, & Titanic made the most in return
- Not every highly budgeted film results in a profit



Title	Popularity
Minions	547.49
Wonder Woman	294.34
Beauty and the Beast	287.25
Baby Driver	228.03
Big Hero 6	2.13.85

Title	Budget
Pirates of the Carribean: On Stranger Tides	380000000
Pirates of the Carribean: At World's End	300000000
Avenges: Age of Ultron	280000000
Superman Returns	270000000
Transformers: The Last Knight	260000000

Title	Revenue
Avatar	2787965000
Star Wars: The Force Awakens	2068224000
Titanic	1845034000
The Avengers	1519558000
Jurassic World	1513529000

Runtime

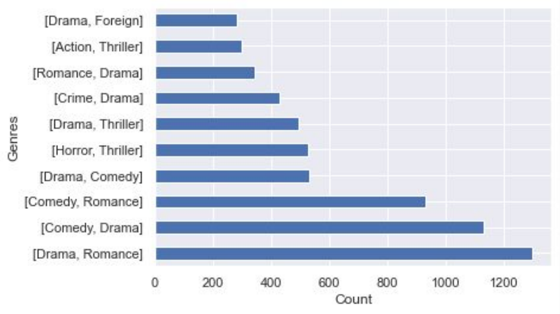
- Average length of a film is about 1 hour and 34 minutes
- Shortest films were made during the initial spike in filmmake
- The longest films in the dataset are one hour episode TV shows

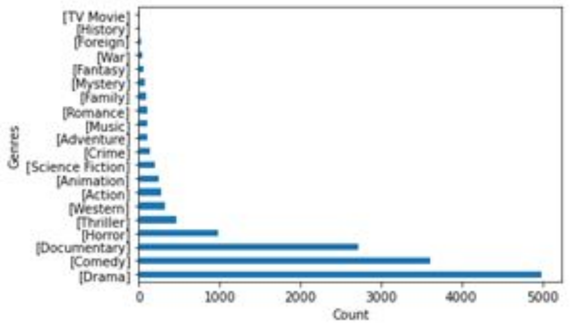
Title	Runtime
Mr. Edison at Work in His Chemical Laboratory	1.0
Grandma's Reading Class	1.0
What Happened on Twenty-Third Street, New York City	1.0
The Magician	1.0
Panorama pris d'un train en marche	1.0

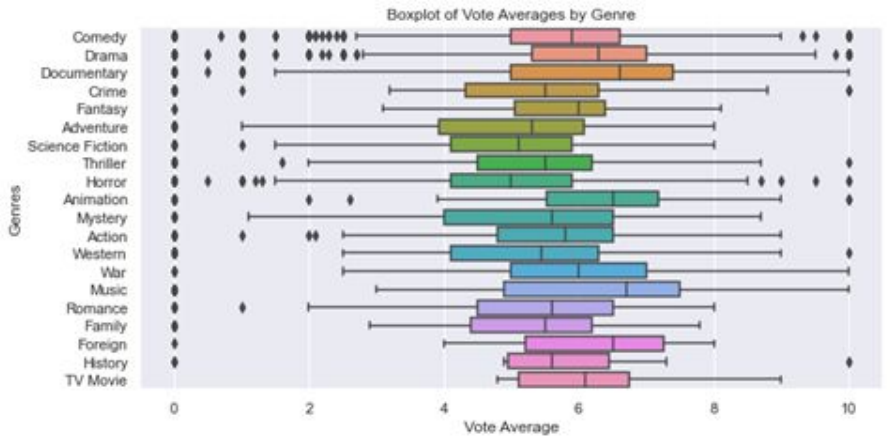
Title	Runtime
Centennial	1256.0
Jazz	1140.0
Baseball	1140.0
Berlin Alexanderplatz	931.0
Heimat: A Chronicle of Germany	925.0

Genres

- Drama has the highest vote average, then follows Comedy and Thriller
- Drama and Romance, Comedy and Drama, Comedy and Romance are the most popular pairs of genres

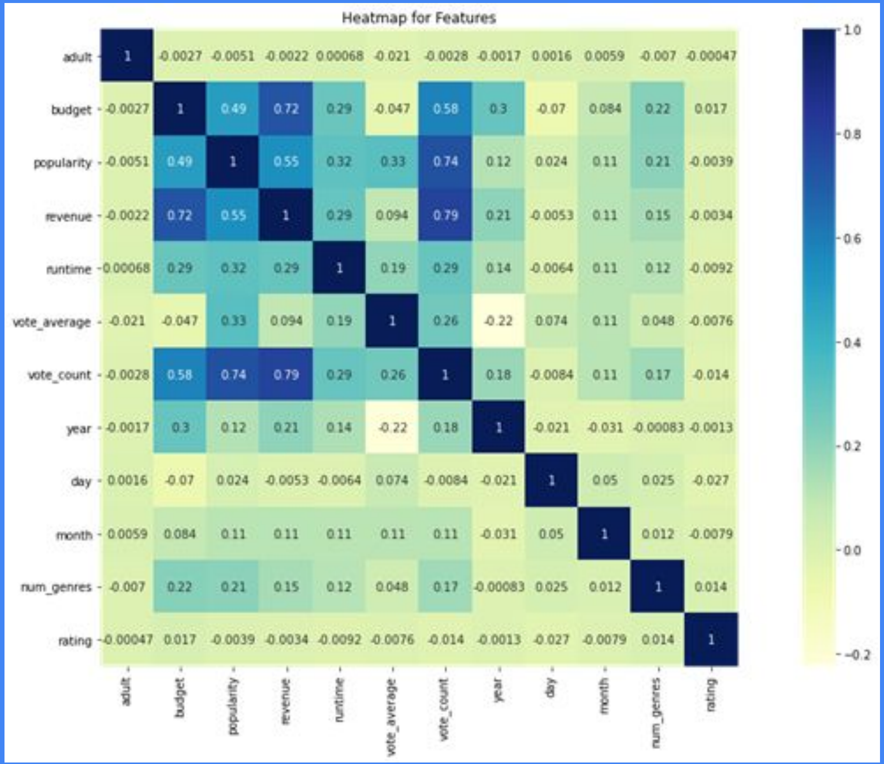






Merging Datasets, Machine Learning

- Motivation: identify valuable features that affect the rating of a film = improve recommendation system
- Not too much correlation coefficient was found among variables
- Models ran on a fraction of the data
- The data was split 75%/25% training/test sets
- The process of feature engineering involved:
    - One hot encoding categorical variables
    - Adding/converting features into binary, 0: true, 1: false
    - Nulls in runtime were imputed with mean



Metrics

- Baseline Implementation with all hyperparameters set a default
- Using Randomized Search hyperparameter Tuned models with a range of arguments
- Discussing the metrics

Model	Explained Variance Score	Mean Absolute Error (MAE)	Mean Squared Error (MSE)	Mean Squared Log Error (MSLE)	R^2 score	Median Absolute Error	RMSE
RF	0.13	1.57	4.03	0.10	0.13	1.28	2.01
GB	0.09	1.64	4.23	0.11	0.09	1.24	2.04
XGB	0.12	1.58	4.07	0.10	0.12	1.28	2.11

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
RF	0.13	1.57	4.03	0.10	0.13	1.26	2.01	2.00
GB	0.13	1.58	4.04	0.10	0.13	1.26	2.01	2.00
XGB	0.13	1.58	4.05	0.10	0.13	1.29	2.01	2.06

Recommender Systems (Pt. 1)

- Content Based FIltering
  - A mix of all categorical variables: overview, tagline, keywords, cast, genres
  - TfidfVectorizer + cosine similarity
  - Match films that are similar in content

- Correlation based Recommender
  - Store user IDs, titles, and ratings as a pivot table
  - Store title, rating, number of ratings as a dataframe
  - Correlate with table and suggest films that have are highly correlated and have more than 100 ratings.

```
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
```

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

original_title	Correlation	num_ratings
Judex	1.0	202
La fonte des neiges	1.0	127
Lord of Illusions	1.0	181
Brubaker	1.0	106
Bridesmaids	1.0	173

Recommender Systems (Pt. 2)

- Hybrid: Weighted Average + Popularity
  - IMDB Top 250 movies ranking formula
  - Set vote count to 90th percentile as cut off
  - Assign 50% importance to weighted average and popularity using MinMaxScaler
- Collaborative Filtering using KNearest Neighbor
  - Similar to content based
  - Convert pivot table into array matrix
  - Calculate neighbors using euclidean distance

Weighted Rating (WR) =

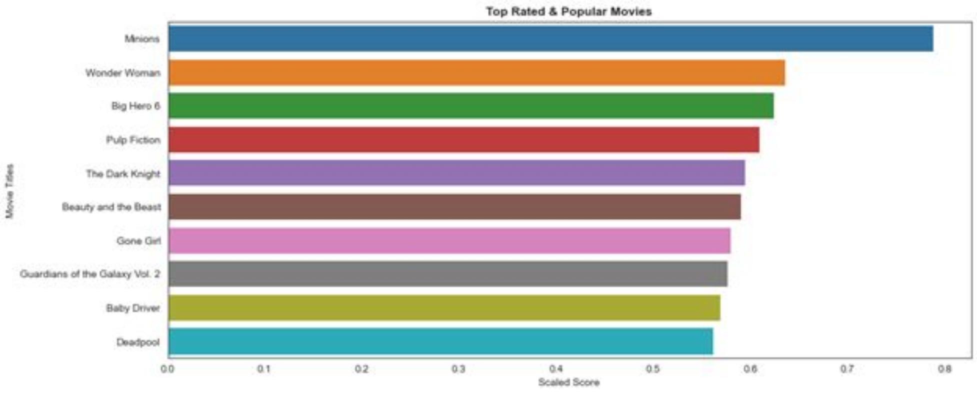
$$(\frac{v}{v+m} * R) + (\frac{m}{v+m} * C)$$

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

```
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:
```



Conclusion

- 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.
- Inception and The Dark Knight have the most votes
- Model performance can be improved with the addition of more features/variables such as figuring out the weekday based on the day of release.
- The hyperparameter tuned Random Forest was the best performing model with an R^2 of 13.3%.
- Revenue, runtime, popularity are influential predictors.
- Four baseline recommender systems