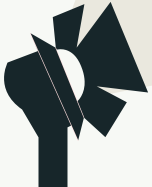
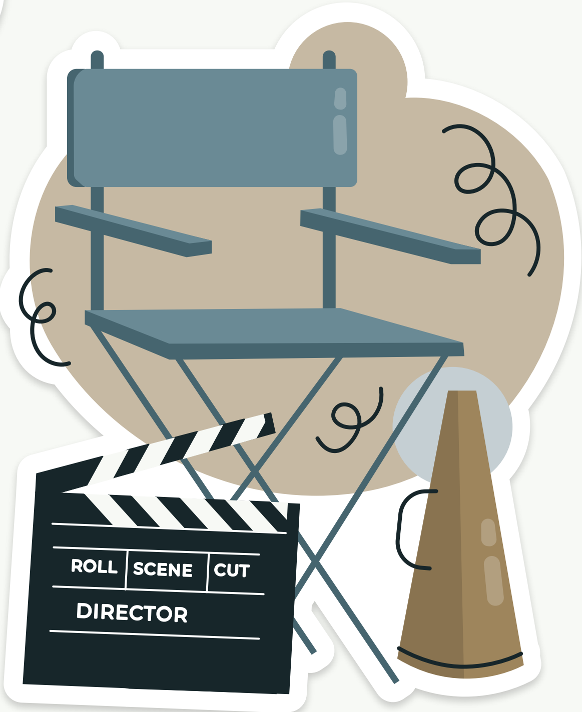
From the TV to the Big Screen

By: Purbasha, Orlyana, Kimberly, and Airin





01

Introduction

Opening Example

- Glass Onion had a limited, one-week theatrical release before it was released on Netflix for streaming
- Grossed 15 million dollars, highest for any Netflix film released theatrically
- Should more Netflix movies have theatrical releases before the typical streaming release?
- How would Netflix executives know if the movie they're releasing would perform well in theaters?

Question + Approach

Question: What features of the Netflix movies would maximize revenue if they were released in theaters?

We used the TF-IDF to identify the most similar theatrical movie, and then we used the associated theatrical movie's revenue to cluster the Netflix movies according to their features. This allowed us to identify the kind of Netflix movies that would do best if they were made available to view in theaters.

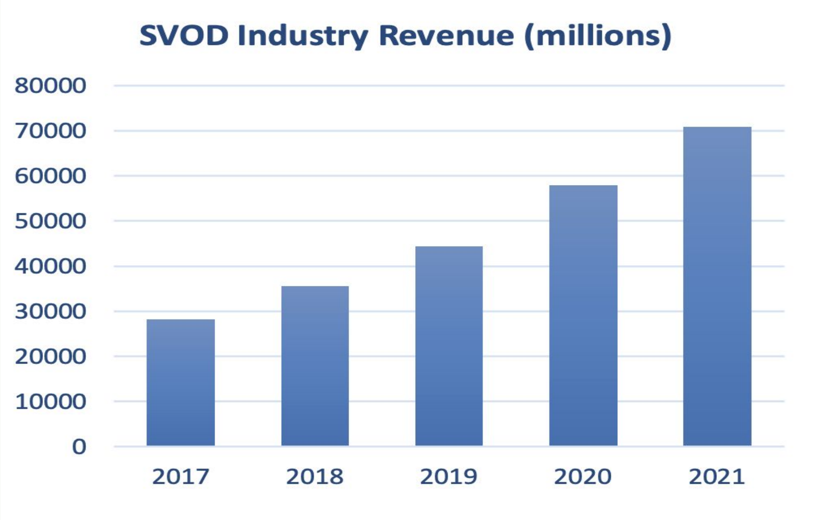
Project Context (Relevance)

The rapid rise of digital streaming services has fueled a radical shift in the entertainment scene in recent years, especially after the Covid-19 pandemic. This transformation has reshaped how people consume content, with the shift to on-demand services that provide tailored, easy access movies. Streaming services like Netflix have not only transformed viewing habits and behaviors, but also emphasized original content (Netflix Originals). Using techniques, such as TF-IDF and cosine similarity, we investigate the relationship between the features of Netflix movies and their economic potential in theaters. Using insights from the streaming data we're determining what makes movies successful in theaters. Then, we want to find different strategies that work for various ways of movie distribution.

The Growth of Streaming Subscriptions

According to Forbes (March 27, 2023)...

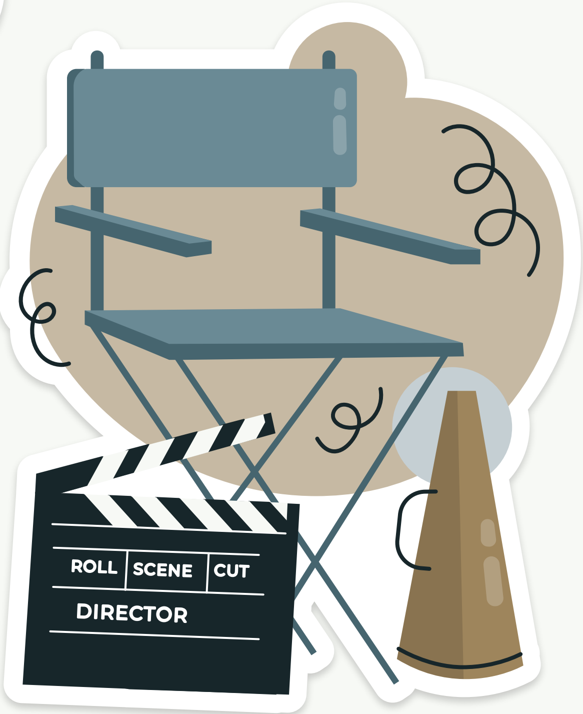
- Americans spend an average of 13 hours and 11 minutes a day using digital media
- 78% of all U.S. households subscribe to at least one or more streaming services
- Netflix still dominates as the most subscribed to video streaming service with 231 million subscribers



Stakeholders

Who They Are: This is extremely important for **stakeholders** in the entertainment industry, such as executives and producers. Netflix, more specifically their production company, is the primary stakeholder company, with a potential for other streaming platforms to benefit in the future.

Benefits for Them: This project allows for stakeholders to optimize another stream of revenue: ticket sales from the theatres. Our model will allow stakeholders to predict what their revenue from the theatres would be. It will also show stakeholders what attributes of a movie their production company should pay attention to if they want to optimize revenue from the theatres.



02

Dataset

Three Datasets (Kaggle)

Theatrical Movies

Updated daily and contains data from 700,000 movies with theatrical releases including cast, crew, genre, budget, revenue, etc.

Netflix Originals

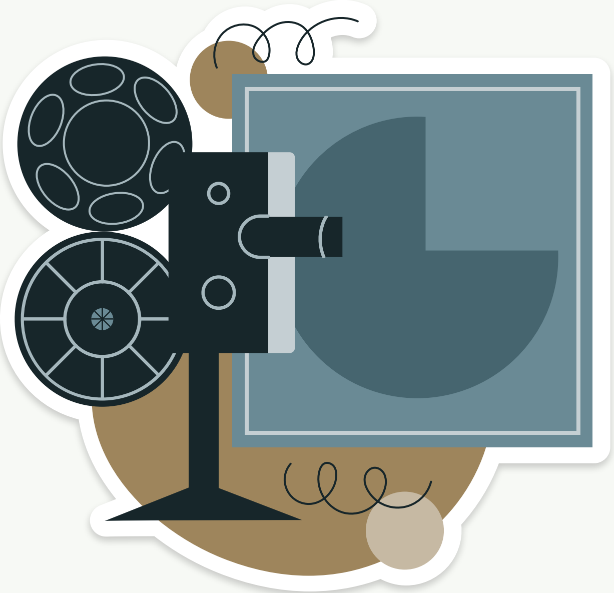
Has data of Netflix Original movies released since June 1st, 2021 including genre, premiere date, runtime, imdb score, and language

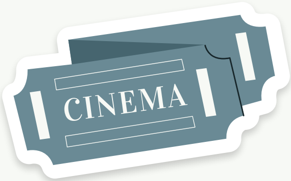
Netflix Movies + TV Shows

Contains data from all movies and TV shows that reside on Netflix including cast, directors, rating, release year, etc.

Filtering (Theatrical Movies)

- Released Movies
- Released between January 2000 to August 2023
- Original language is English
- Dropped unnecessary columns
- Dropped rows where revenue equals 0
- Kept first 5 (most popular) actors per movie





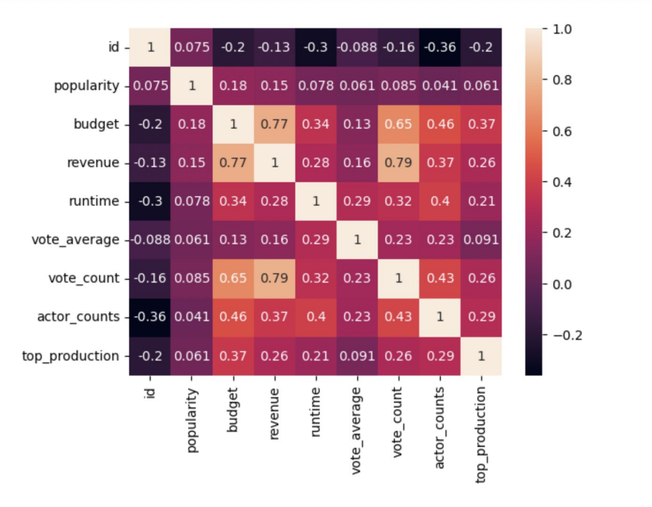
Feature Engineering & Filtering (Netflix Movies)

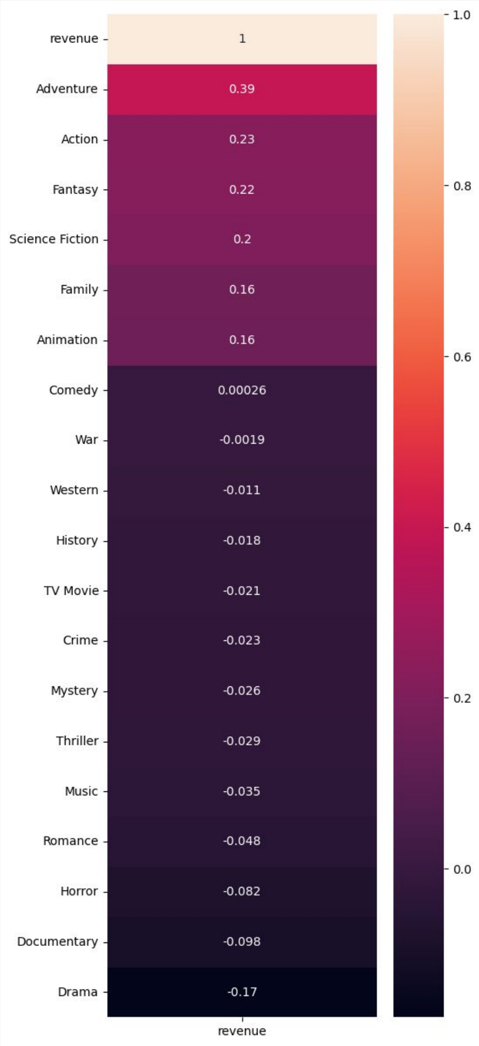
- Text Cleaning and Standardization
    - Lowercasing (e.g. title, genre, description, cast, director)
    - Removal of punctuation and special characters
- Filtering by Date
    - Selecting movies between January 2000 and August 2023
- Language Filtering
    - Only having movies that have English as the language of choice
- Null Values
    - Dropped rows with missing values

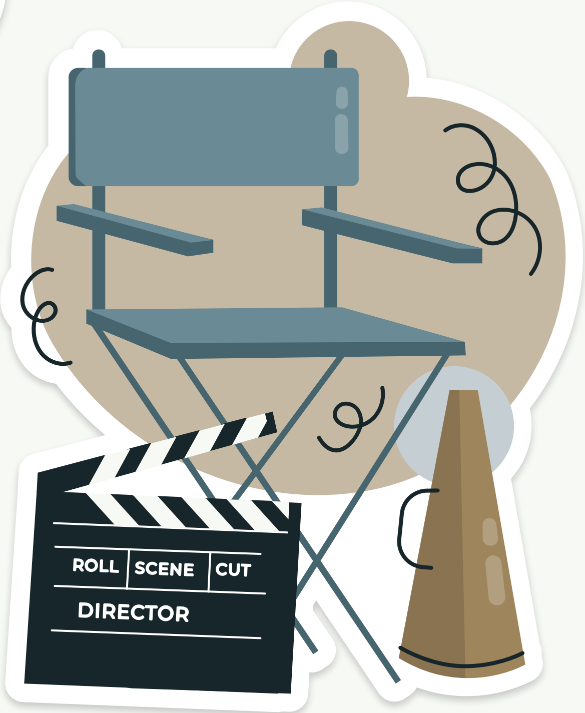
Feature Engineering (Theatrical Movies)

New Columns

- Actor Counts: Amount of times the top 5 actors appear in other movies in the data set
- Top Production: Created a list of the top 15 production companies that produced the most movies in the dataset and column represented if movie was produced by a top company

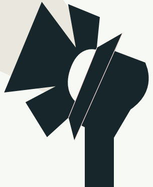


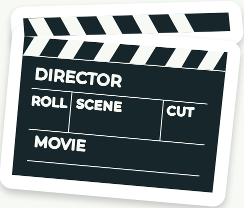




03

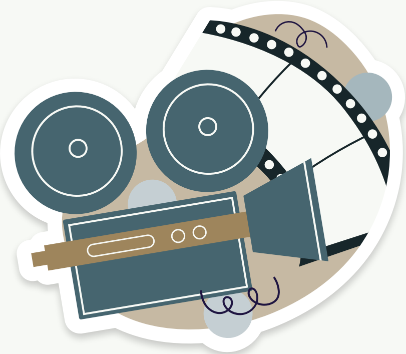
Methods



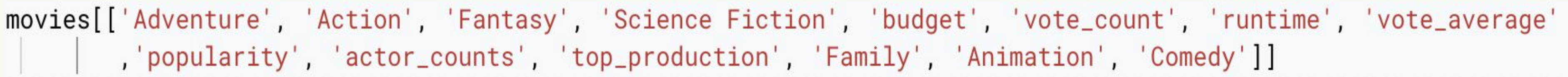


Method 1

Predicting revenue of theatrical movies

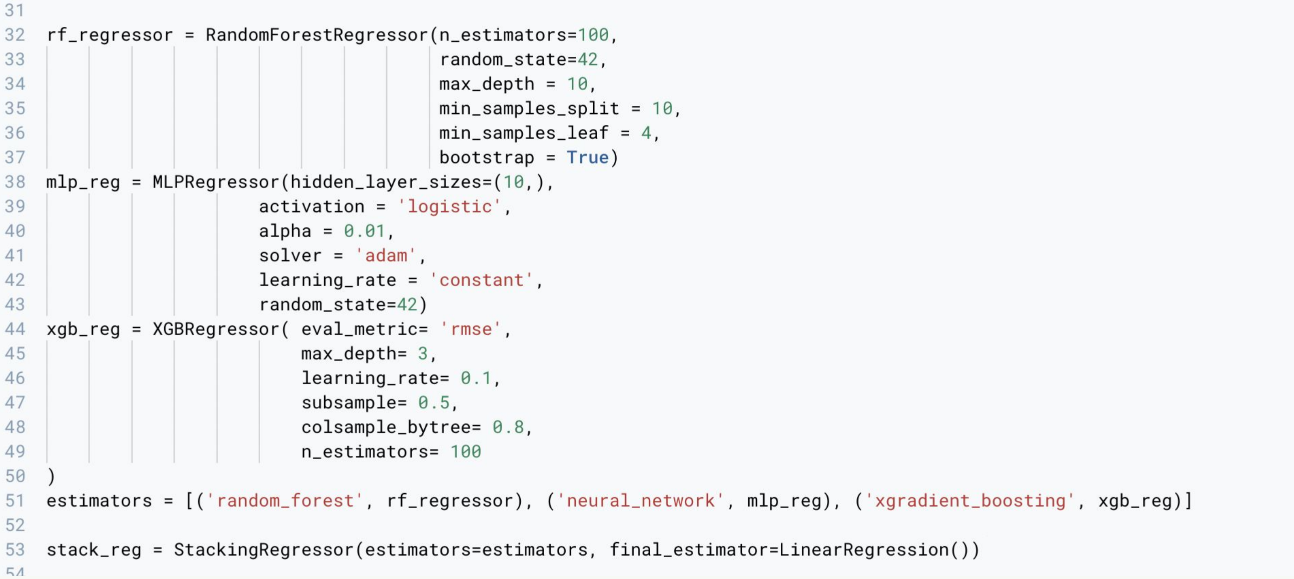


Features + ML Used



```
movies[['Adventure', 'Action', 'Fantasy', 'Science Fiction', 'budget', 'vote_count', 'runtime', 'vote_average'
            ,'popularity', 'actor_counts', 'top_production', 'Family', 'Animation', 'Comedy']]
```

- Random Forest
- Neural Network
- XGBoost
- Stacking



```

rf_regressor = RandomForestRegressor(n_estimators=100,
                                     random_state=42,
                                     max_depth = 10,
                                     min_samples_split = 10,
                                     min_samples_leaf = 4,
                                     bootstrap = True)
mlp_reg = MLPRegressor(hidden_layer_sizes=(10,),
                    activation = 'logistic',
                    alpha = 0.01,
                    solver = 'adam',
                    learning_rate = 'constant',
                    random_state=42)
xgb_reg = XGBRegressor( eval_metric= 'rmse',
                        max_depth= 3,
                        learning_rate= 0.1,
                        subsample= 0.5,
                        colsample_bytree= 0.8,
                        n_estimators= 100
)
estimators = [('random_forest', rf_regressor), ('neural_network', mlp_reg), ('xgradient_boosting', xgb_reg)]

stack_reg = StackingRegressor(estimators=estimators, final_estimator=LinearRegression())
```

Accuracy

Evaluation Metric

Mean Squared Error

Results

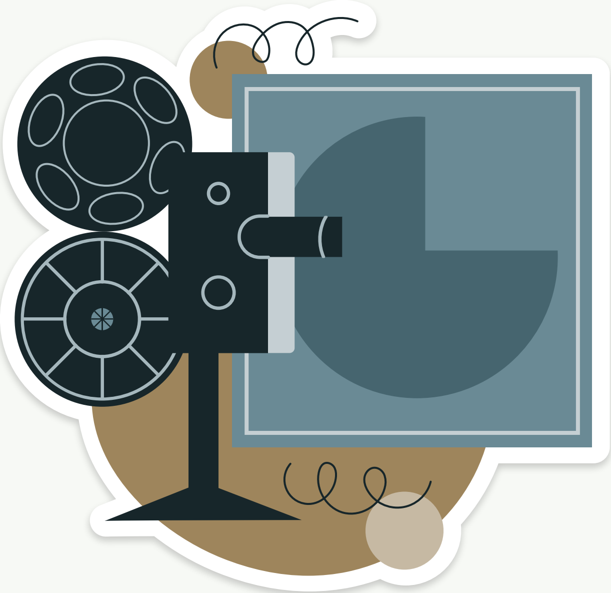
Using k-fold validation, error seemed to typically range from 0.16-0.20

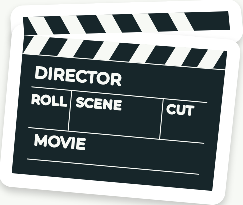
Interpretation

Since we were looking for an error close to 0, seems like our model is a good model

Revenue Model Concluding Thoughts

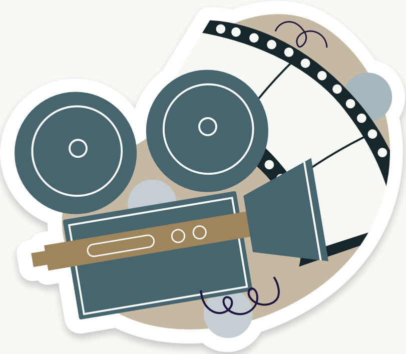
- Although a good model, doesn’t answer our original question
- But still insightful model that can help predict how much revenue a movie will accrue based on its features





Method 2

Most similar theatrical movie to a Netflix movie



For every Netflix Original Movie Find a Similar Theater-Released Movie

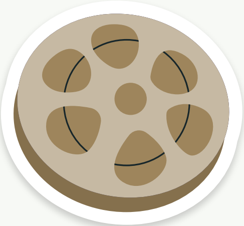
1. Make a “features” column that contains “title”, “genres”, and “description”

2. Concatenate Netflix and theatre-released movies datasets

3. Use TF-IDF Vectorizer on the Concatenated Dataframe

4. Use Cosine similarity to compare rows (movies). For every Netflix movie find similarity score with every theatre-released movie

5. Make a function that finds the most similar theatre-released movie by looking at the highest cosine similarity score.



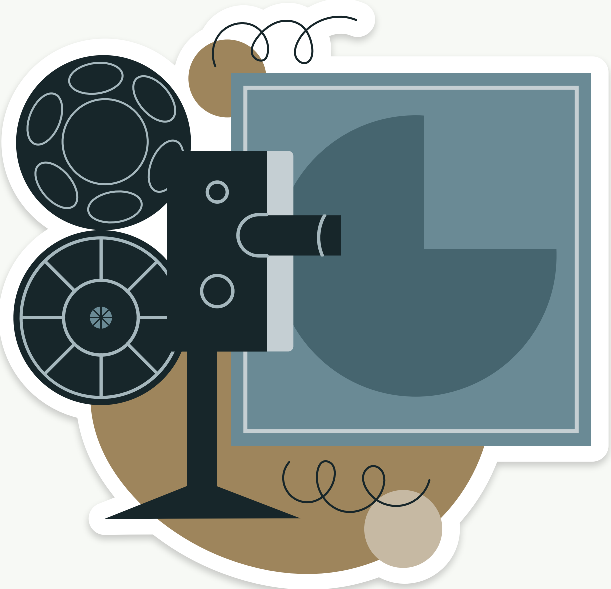
What is TF-IDF?

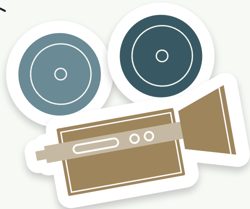
Term Frequency-Inverse Document Frequency

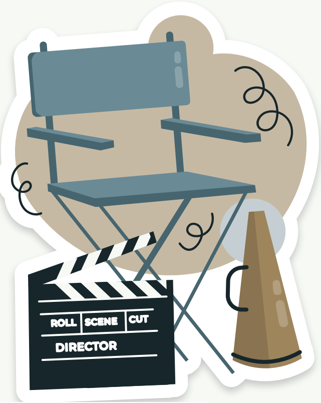
- A statistical method used in NLP and information retrieval.

- It transforms words within a text document into number by text vectorization process.

- It measures how important a term is within a document relative to a collection of documents (i.e., relative to a corpus)







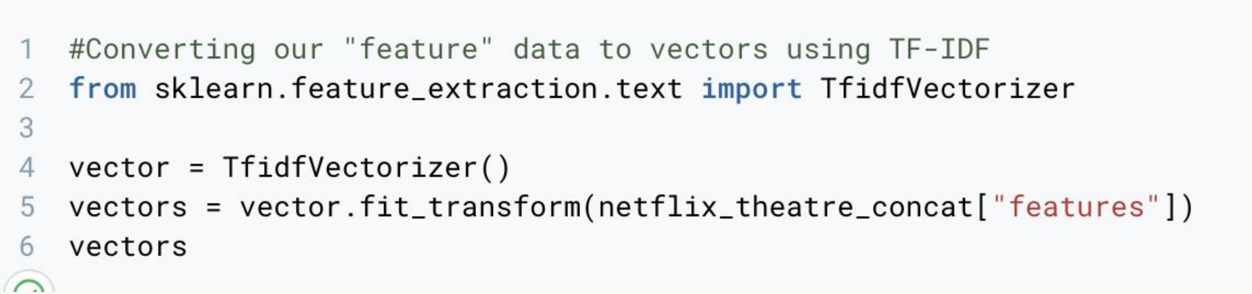
TF-IDF vectorizes a word by multiplying the word’s Term Frequency (TF) with the Inverse Document Frequency (IDF)

$$TF = \frac{\text{number of times the term appears in the document}}{\text{total number of terms in the document}}$$

$$IDF = log(\frac{\text{number of the documents in the corpus}}{\text{number of documents in the corpus contain the term}})$$

$$TF\text{-}IDF = TF * IDF$$

Using TF - IDF in our project

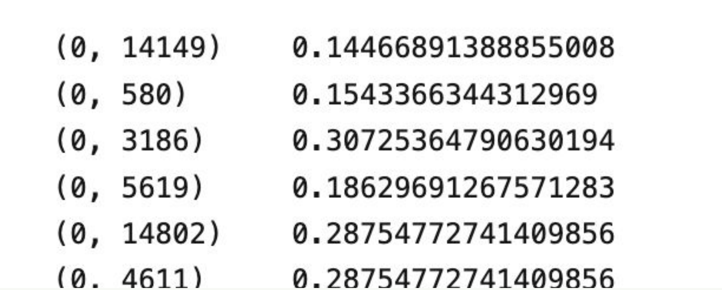


```python
#Converting our "feature" data to vectors using TF-IDF
from sklearn.feature_extraction.text import TfidfVectorizer

vector = TfidfVectorizer()
vectors = vector.fit_transform(netflix_theatre_concat["features"])
vectors
```

“vectors” is a sparse matrix that indicates tf-idf score for all non-zero values in the word vector for each document.

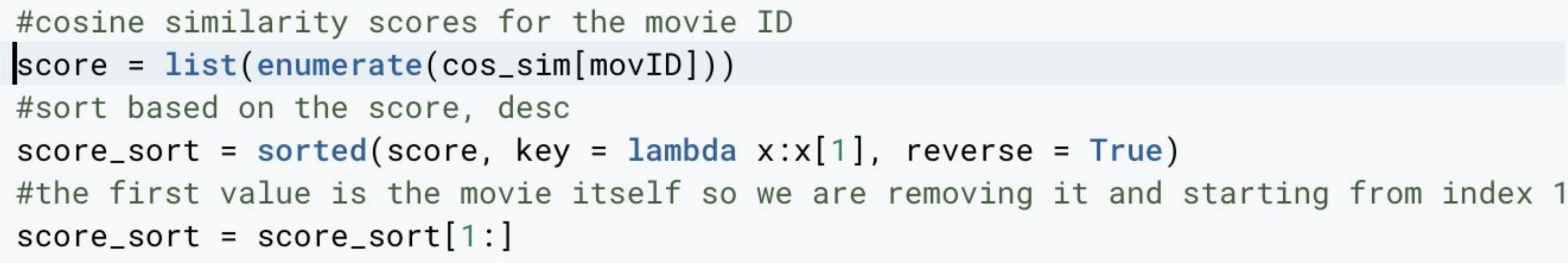
Output of “vectors”
(A,B) C
A: Document index
B: Specific word-vector index)
C TF-IDF score for word B in document A



```
  (0, 14149)	0.14466891388855008
  (0, 580)	0.154336634431296 9
  (0, 3186)	0.30725364790630194
  (0, 5619)	0.18629691267571283
  (0, 14802)	0.28754772741409856
  (0, 4611)	0.28754772741409856
```

movie_finder(name):

Given a Netflix Movie it outputs the most similar theatre-released movie



```
#cosine similarity scores for the movie ID
score = list(enumerate(cos_sim[movID]))
#sort based on the score, desc
score_sort = sorted(score, key = lambda x:x[1], reverse = True)
#the first value is the movie itself so we are removing it and starting from index 1
score_sort = score_sort[1:]
```

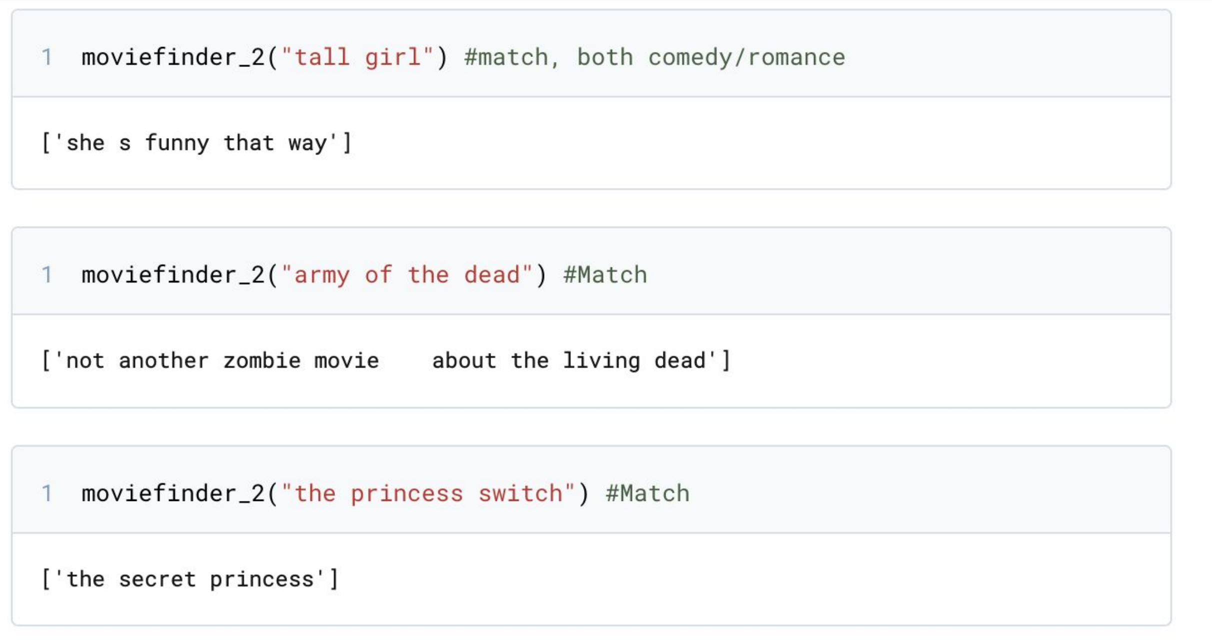
Finds similar movie based on the cosine similarity and makes sure that it is a theatre-released movies

Evaluating Our Movie Finder

Cosine Similarity between movies

Manual Check

Manual Check of movie_finder



```
moviefinder_2("tall girl") #match, both comedy/romance
```

```
['she s funny that way']
```

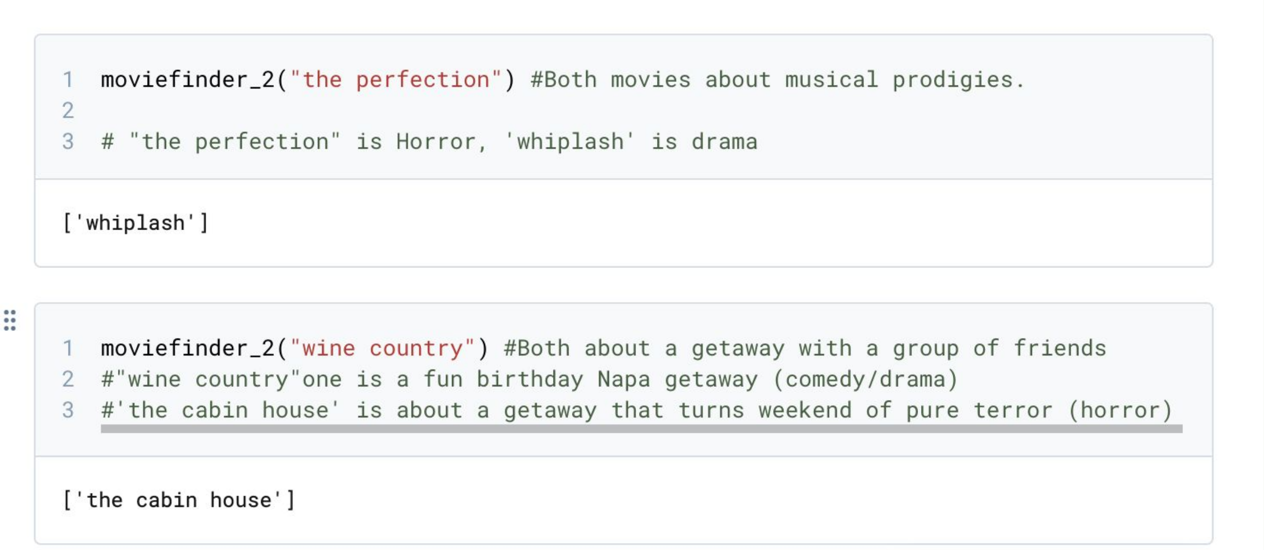
```
moviefinder_2("army of the dead") #Match
```

```
['not another zombie movie    about the living dead']
```

```
moviefinder_2("the princess switch") #Match
```

```
['the secret princess']
```

Interesting movie_finder results



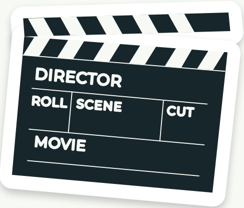
```
moviefinder_2("the perfection") #Both movies about musical prodigies.

# "the perfection" is Horror, 'whiplash' is drama
```

```
['whiplash']
```

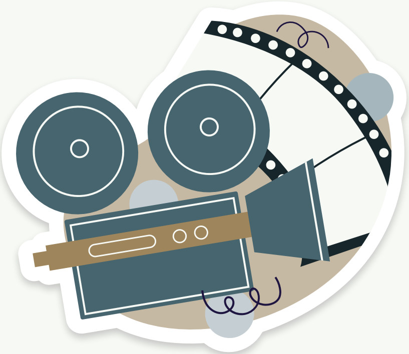
```
moviefinder_2("wine country") #Both about a getaway with a group of friends
#"wine country"one is a fun birthday Napa getaway (comedy/drama)
#'the cabin house' is about a getaway that turns weekend of pure terror (horror)
```

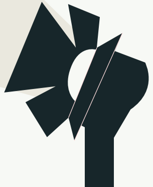
```
['the cabin house']
```



Method 3

Types of Netflix movies that would have successful theatrical releases





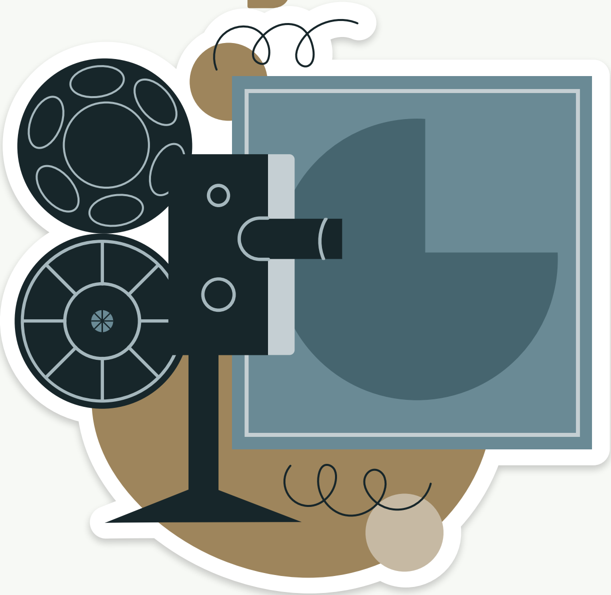
Two Clustering Methods

K-means

HDB Scanner

Features Used For Clustering

- Runtime
- Imdb Score
- Revenue
- Genre



UMAP and HDB Scanner

HDB Scanner

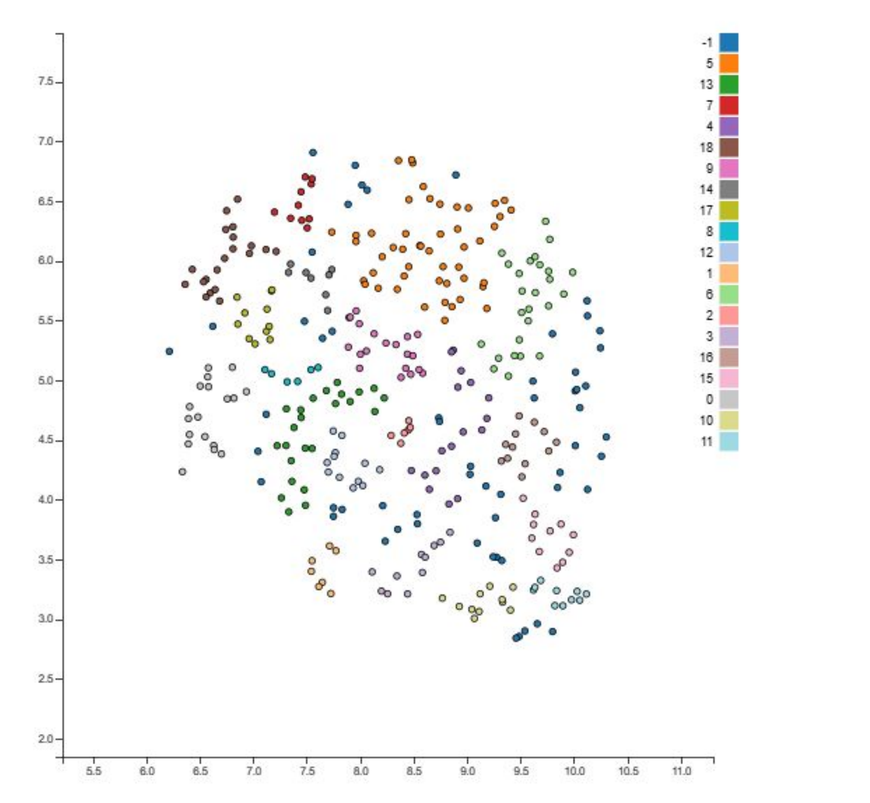
- A Hierarchical Clustering technique
- Saw that it performed better than K Means for this dataset
- We believe that because of the genres being related to one another that hierarchical clustering is applicable here

UMAP

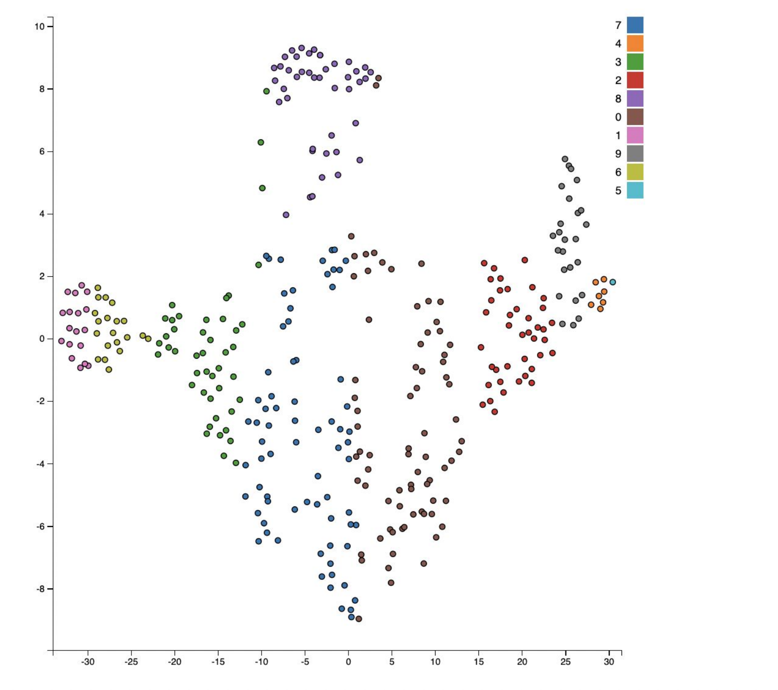
- A Dimensionality reduction technique that works well on mixed data
    - A lot like TSNE except it does a better job with preserving the data’s structure

Our Sanity Check

- Used UMAP and HDB Scanner to cluster the data set
- The vectors used here are the ones that were produced by the TF-IDF Model
- Visualization is a lot like lab 7!
- **Goal:**
  - Sanity check to see how well our embedding grouped movies together
    - For each cluster, was there a similarity in genre?
    - Did it make sense for the movies to be close together?
  - Use this visualization as a base model for our cluster visualizations
    - Are we seeing the same patterns be reproduced in our other visualizations, using only genre, revenue, imdb scores, and runtime as features

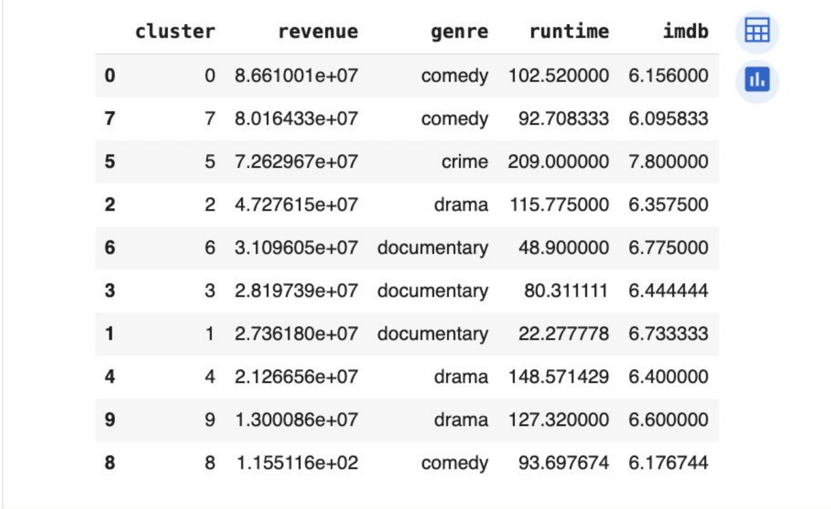


K-means Clustering



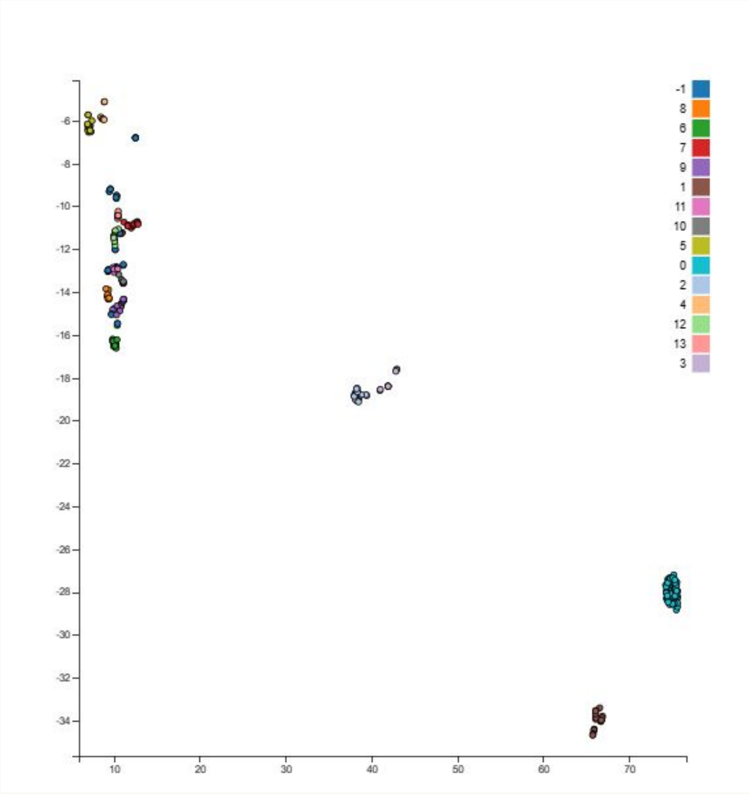
Results of Clustering

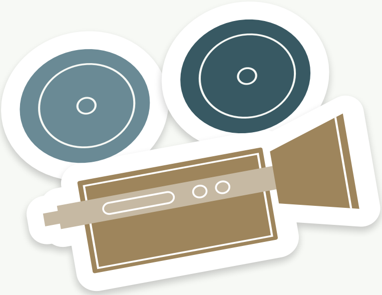
Argument could be made to Netflix executives based on this clustering that if they release a comedy movie with a 92-102 minute runtime and a 6.0-6.1 imdb score for a limited theatrical release, the movie should perform well



	cluster	revenue	genre	runtime	imdb
0	0	8.661001e+07	comedy	102.520000	6.156000
7	7	8.016433e+07	comedy	92.708333	6.095833
5	5	7.262967e+07	crime	209.000000	7.800000
2	2	4.727615e+07	drama	115.775000	6.357500
6	6	3.109605e+07	documentary	48.900000	6.775000
3	3	2.819739e+07	documentary	80.311111	6.444444
1	1	2.736180e+07	documentary	22.277778	6.733333
4	4	2.126656e+07	drama	148.571429	6.400000
9	9	1.300086e+07	drama	127.320000	6.600000
8	8	1.155116e+02	comedy	93.697674	6.176744

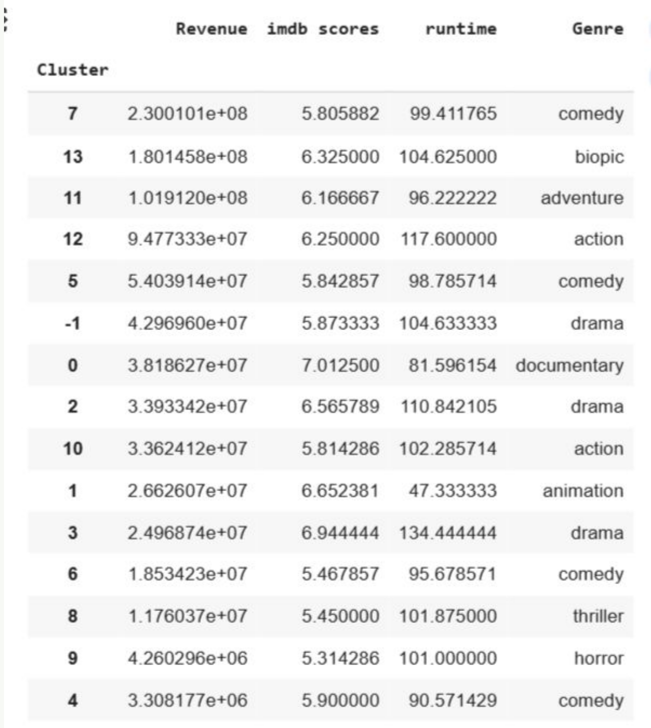
UMAP and HDB Scanner Cluster



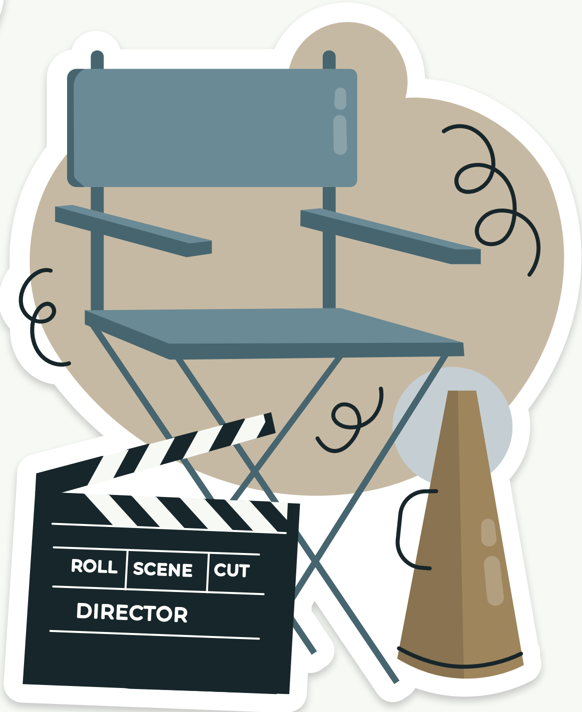


Results of the UMAP Clustering

- **Top 5 genres** that work well in theatres:
  - Comedy, biopics, adventure, action, drama
- Found that Cluster 7 had the highest average revenue, and that the most common Genre is Comedy
- Interestingly, Cluster 4 had the lowest average revenue and the genre was comedy
  - Both clusters had an average rating of about 5.8/5.9
  - Runtime is quite similar
  - How can it be different?
    - Lack of information prevented us to use important features
    - Lots of different flavors of genre, could be that it was different flavor of comedy that we can't see within the data



	Revenue	imdb scores	runtime	Genre
Cluster				
7	2.300101e+08	5.805882	99.411765	comedy
13	1.801458e+08	6.325000	104.625000	biopic
11	1.019120e+08	6.166667	96.222222	adventure
12	9.477333e+07	6.250000	117.600000	action
5	5.403914e+07	5.842857	98.785714	comedy
-1	4.296960e+07	5.873333	104.633333	drama
0	3.818627e+07	7.012500	81.596154	documentary
2	3.393342e+07	6.565789	110.842105	drama
10	3.362412e+07	5.814286	102.285714	action
1	2.662607e+07	6.652381	47.333333	animation
3	2.496874e+07	6.944444	134.444444	drama
6	1.853423e+07	5.467857	95.678571	comedy
8	1.176037e+07	5.450000	101.875000	thriller
9	4.260296e+06	5.314286	101.000000	horror
4	3.308177e+06	5.900000	90.571429	comedy

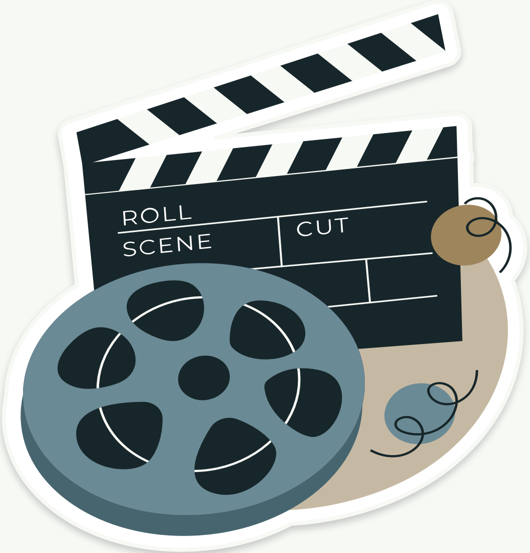


04

Conclusion

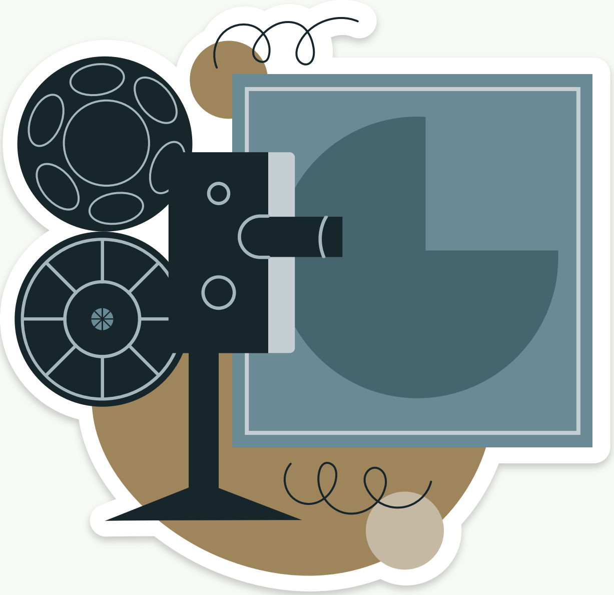
Trends

- Budget, Genre, Runtime, Vote Count (How well rated the movie was), and Actor_Counts (how many popular actors starred) were major contributing factors to how much revenue was made
    - *Found by EDA*
- Genres that seemed to do well in theatres : Comedy, Adventure, Action, and Drama
    - *Found by our TF-IDF model and a combination of both our UMAP and K-Means cluster visualizations*



Concluding Thoughts

- We can’t make assertive conclusions based on our models because of the few limitations
- Much of the data that could make our models better is privatized by Netflix
- If Netflix used our models and combined it with their privatized data they could make more accurate predictions that determine the amount of revenue a Netflix movie could make with a theatrical release



THANK YOU!

Do you have any questions?