IMDb Data Analysis

Sofia Alcazar, Dylan Jorling, Daniel Kwon, Ajay Patel

Introduction

In the movie industry, success can be defined in different ways. Some producers may only care about maximizing box office profits while others may deem critical acclaim and the respect of peers as a success. For the purposes of our analysis, we will focus on one of these definitions - box office profits. Our goal is to successfully model box office profits using data and information prior to a movie's release.

Data, Data Cleaning, Feature Engineering

Overview

Internet Movie Database (IMDb) is a popular website that provides information on over 10 million movies and tv shows. The IMDb API provides details about each of these movies and tv shows ranging from user-ratings and reviews to casting and budget. We determined the API contains sufficient information to create a predictive model.

Data Collection, Cleaning and Feature Engineering

First, we scraped data in JSON form for over 5,000 movies released between 1922 and 2022. Then, we processed each JSON file into a single pandas dataframe which we saved as a csv file. The data scraped from the API was consistently formatted and contained relatively few NAs, so data cleaning was fairly minimal.

Many of the raw data columns such as "writers" contained strings with multiple names. For these variables, we extracted each distinct name and created a list of lists. Since the lists contained thousands of unique entries, we decided to transform these variables into numerical scores based on each name's total frequency in the entire dataset. The "stars" column almost always contained three names per entry. For each entry, we summed each star's total frequency to create the star power variable. The "writers" variable varied in length between one and three names, so we took the mean frequency of each name, while for the "directors" variable, we took the max frequency value, because the entry contained one name often.

We used a similar methodology for companies (i.e. production studios), but instead of using frequency as a proxy for popularity, we used it as a proxy for the size of the production company. Then, we created a factor variable with four different levels. The genre variable also often contained more than one label, and we did not have a way to determine the best categorization for each film. So, for each observation, we randomly selected one genre from the

list of genres. From this point, we pared the categories down to 5 genres. Similarly, we reduced the film rating variable down to 3 categories.

Although our raw data included a year variable, we thought there may be some significance as to what time of the year the movie came out. From the raw date variable, we extracted the month the movie came out to create the release month variable.

For our financial analysis, we focused on more recent data to paint a clear picture of financial performance in modern moviemaking. Therefore, we decided to filter out any data before 1990. We also decided to filter out any data after 2019 to exclude the effect of the global pandemic on the film industry. Although this is an aggressive assumption, we believe that the industry will revert to pre-pandemic levels. “Top Gun: Maverick”’s recent success strongly supports this view. Additionally, we determined that financial success should not be based on gross revenue, but instead either gross profit (gross revenue - budget) or gross profit margin (gross revenue - budget) / gross revenue. We removed any observation missing a budget or a worldwide gross entry, and we removed any film with a budget recorded in a currency other than US dollars.

We recognized that we needed to normalize the gross profit variable for inflation. Using monthly CPI data from the Bureau of Labor Statistics, we calculated yearly inflation metrics. Then, we applied appropriate inflation multipliers to both budget and worldwide gross, before subtracting the adjusted budget from the adjusted gross to yield the adjusted gross profit variable.

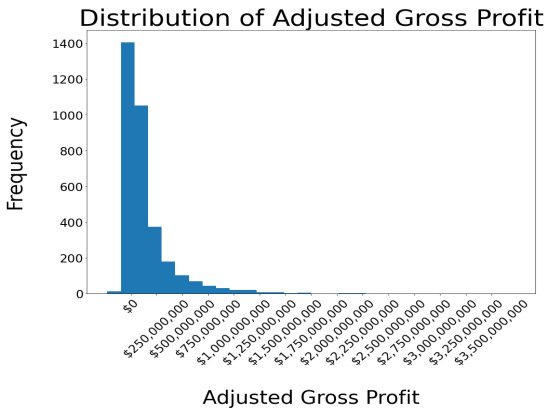
At the end of the data cleaning and data engineering, our dataset consisted of 3338 observations and 32 variables.

Variable Overview and EDA

Response Variable

1. Adjusted Gross Profit

   The adjusted gross profit distribution is heavily right skewed, so we investigated various transformations, such as log, square-root, and 1/4th power transformation. The distribution did not drastically improve, and we did not want to compromise the variable’s interpretability. According to the plot below, most movies earn less than $250 million in adjusted gross profit. However, there are a handful of movies that top the $1 billion mark.

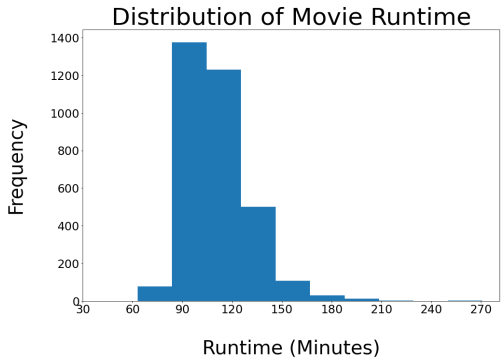


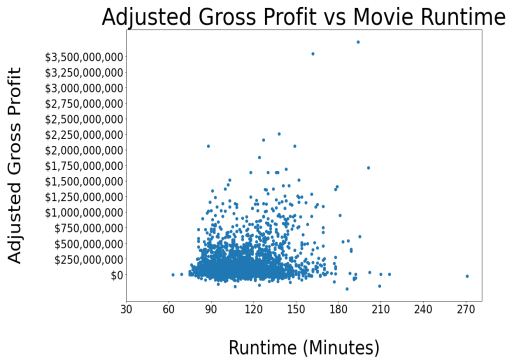
Predictor Variables

Our goal is to model adjusted box office profits with variables *prior* to a movie's release. Thus, we only used variables from the IMDb API that a producer can reasonably control. The following nine variables served as our predictor variables:

1. Runtime

   Most movies in our dataset have a runtime between 90 minutes and 150 minutes. But, there are cases where the runtime exceeds 3 hours causing the slight right skew in the runtime distribution below. According to the scatterplot below, there is a slight positive association between the runtime and adjusted gross profit.

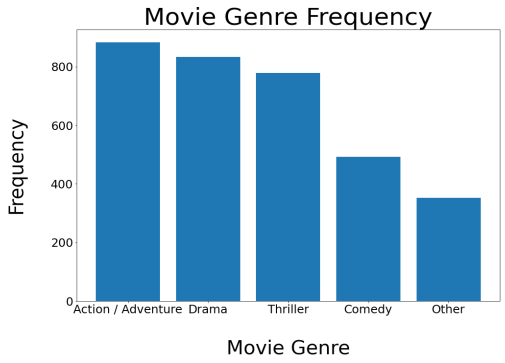


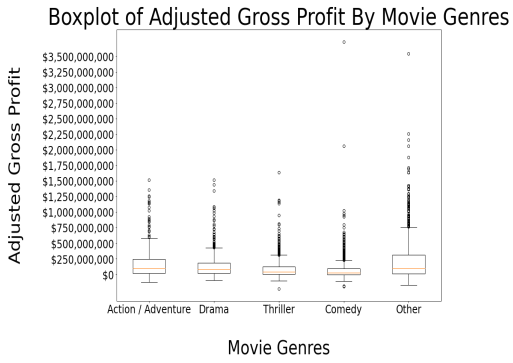


2. Genre

   The most common genres in our dataset are Action / Adventure, Drama, and Thriller. Comedy movies appear to have the smallest IQR, but surprisingly, the largest outlier movie belongs to the comedy genre. Additionally, the adjusted gross profit for Action / Adventure and Drama movies have relatively the same distribution. Although many more movie genres exist, we combined the genres with low frequencies into a new category called "Other." The adjusted gross profit of the "Other" category is on par with the other

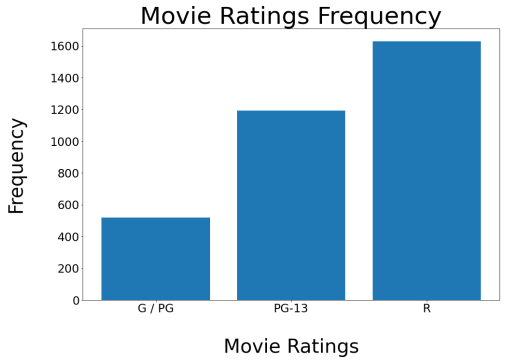
movie genres.

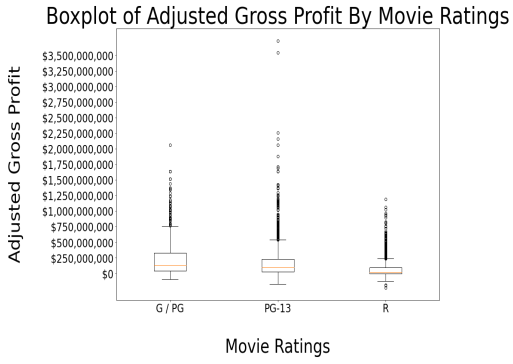




3. Rating

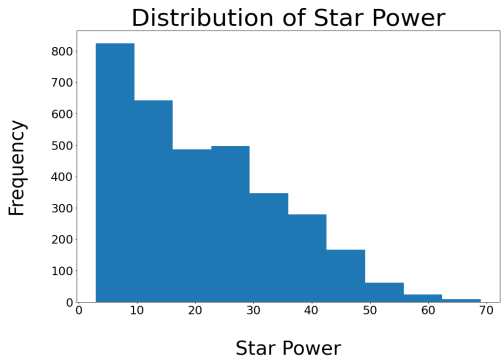
In our dataset, R rated movies make up nearly 50% of the movie ratings, whereas G and PG rated movies make up less than 25%. The G and PG rated movies have the largest range of adjusted gross profits, and R rated movies have the smallest. The largest outlier for an R rated movie is only $1.25 billion, whereas, for a PG-13 movie, the largest outlier is beyond $3.5 billion.

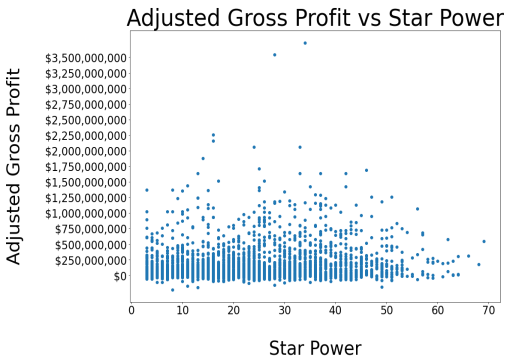




4. Star Power

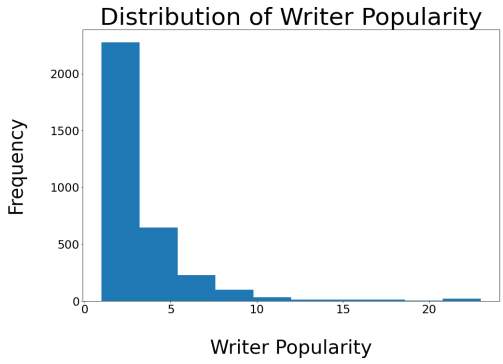
The star power variable is slightly right skewed. Again, we did not apply a transformation because we wanted to maintain the variable's interpretability. We can see in the scatterplot below, there is a slight positive association between star power and adjusted gross profit. Perhaps, writer and director popularity will have a stronger association with adjusted gross profit.

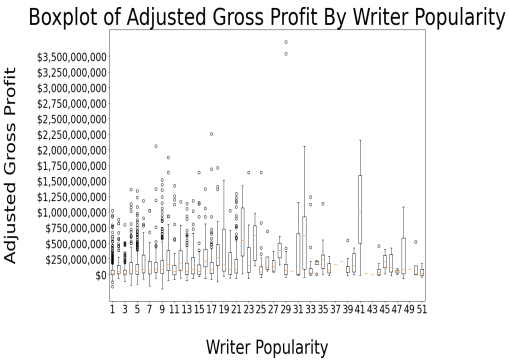




5. Writer Popularity

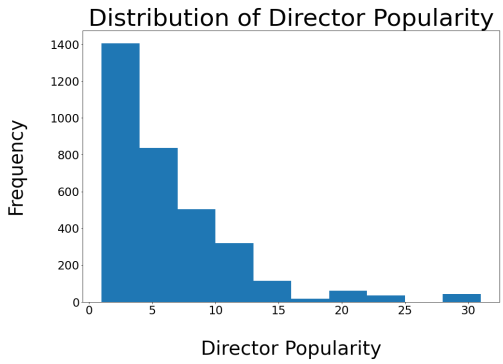
Similarly to star power, the writer popularity variable is right skewed. As before, we did not apply a transformation. There is no discernible difference in the adjusted gross profit across the smaller values of the writer popularity. The larger the writer popularity value, the more variance we see in the adjusted gross profit.

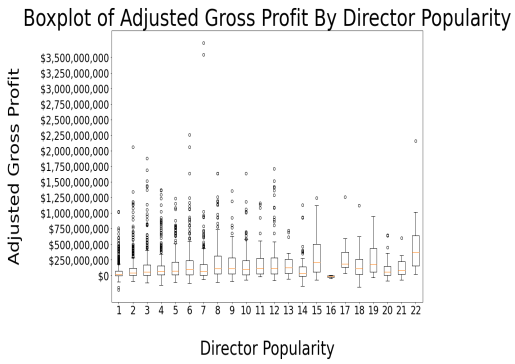




6. Director Popularity

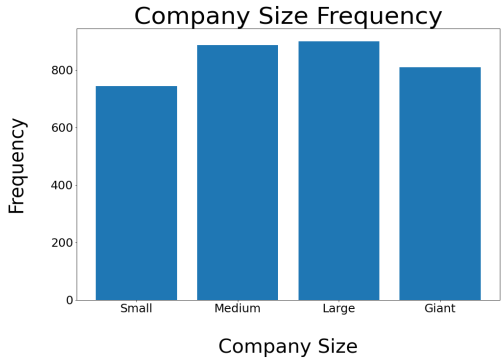
The director popularity variable is also right skewed which is no surprise given the writer popularity is also right skewed. Similar to the boxplots we saw above, the adjusted gross profit has larger variance as the director popularity increases.

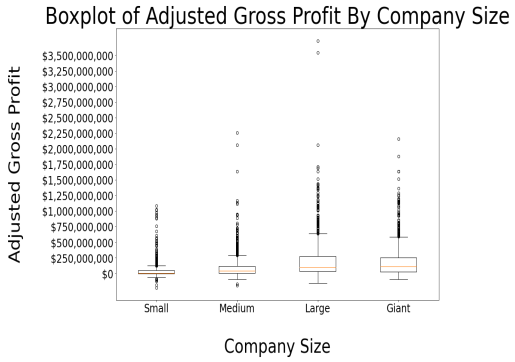




7. Production Company Size

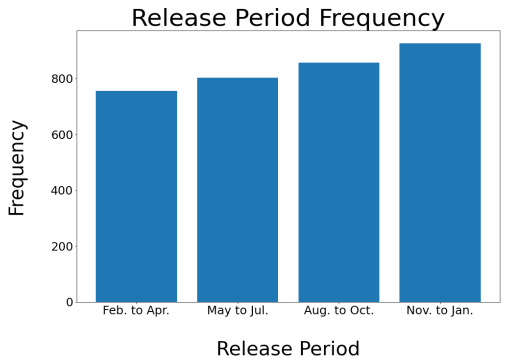
   In our dataset, we created four different categories for production company sizes so that each category is approximately equal in size. As the production company size increases, we would expect that the adjusted gross profit also increases. However, it appears that giant production companies do not necessarily produce more profitable films than large production companies.

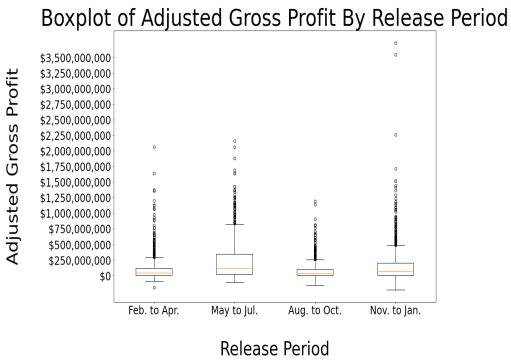




8. Release Period

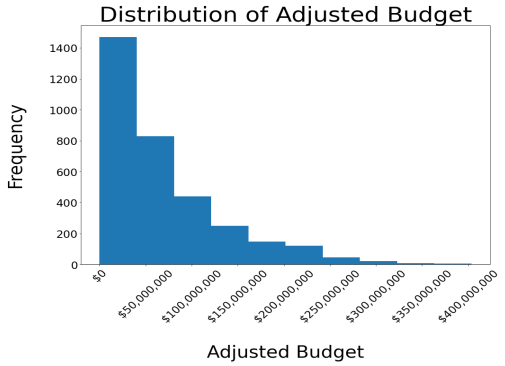
   Our original dataset contained each movie's release month. Rather than use 12 categories for each month, we reduced the months to four different seasons. We used November to January as the starting point to represent the holiday season. From there, every 3 month increment became a new category. This method created nearly equal sized categories according to the bar chart below. Interestingly, the summer time (May to July) has the largest variance in adjusted gross profit.

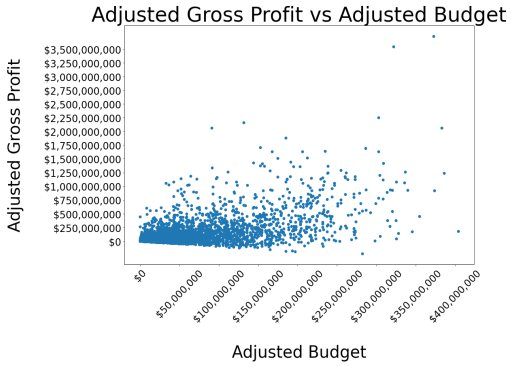




9. Inflation-Adjusted Budget

   The inflation-adjusted budget variable is skewed, but we did not apply a transformation. Most movies have an adjusted budget less than $100 million, but the good news is that many movies are profiting beyond that mark. Unfortunately, there are a handful of cases where the movies did not make profit. Typically these occurrences happen with large adjusted budget films.





Methodology and Results

Data Prep

Before we began the modeling process, we standardized the following numerical features: runtime, director popularity, writer popularity, star power, and adjusted budget. Additionally, we one-hot encoded our four categorical variables so that both our random forest and our neural network could properly utilize the data.

Neural Network

Using python's deep learning pytorch module, we built a fully connected model with two hidden layers. After accounting for one-hot encodings, the model input layer contained 21 features, the first hidden layer contained 50 nodes with *tanh* activation, and the second hidden layer

contained 100 nodes with *relu* activation. The output layer consisted of a linear layer that yielded our final predicted inflation-adjusted gross profit.

We split our data into a training set, consisting of 80% of our total data, and a testing set consisting of the training 20%. We tuned the hyper-parameters on the training set before using the unseen testing set on the tuned model. We trained our model with the Adam optimizer, a 0.0001 learning rate, a batch size of 64, and 150 epochs.

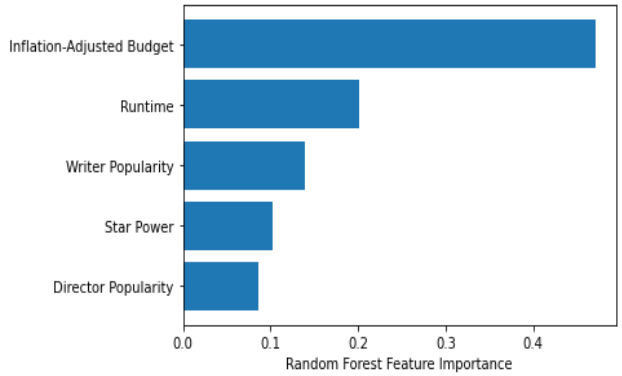
Our model produced an RMSE of 251M compared to the null model standard deviation of 291M. While not substantially predictive, we believe there is still value in the model and that it can be built upon utilizing more data to produce better results.

Random Forest

Using the scikit-learn package in Python, we used a random forest regression to extract feature importance when predicting a film's inflation-adjusted gross profit. The features we analyzed using random forest regression included film runtime, director popularity, writer popularity, star power, and inflation-adjusted budget.

Similar to our neural network, we split the data into an 80/20 train/test split and used hyper-parameter tuning to find the optimal number of trees and max depth. The optimal number of trees was 200 with each tree having a maximum depth of 11.

After optimizing our random forest model, the RMSE decreased to 190M. While this model's predictive power is not enormous, we can extract other important information. We hoped to better understand which features are most predictive of a film's inflation-adjusted gross profit, which in turn, tells us which features a producer should prioritize over others. This feature importance is computed using Gini importance - the average decrease in impurity for each feature across all 200 trees. Below is a graph of the random forest feature importance. It is no surprise that the inflation-adjusted budget is the best predictor of inflation-adjusted gross profit. Based on these results, a producer should also heavily focus on the runtime of the film and the popularity of the writer. Perhaps, the producer may not need to invest in a star studded cast or a popular director for their movie.



Conclusions, Limitations, Shortcomings / Areas for improvement

Overall, we are satisfied with the results of both gross profit models. Our models achieved RMSE and R-squared values that a hypothetical producer would surely find valuable, and we also identified the most important variables to optimize in order to achieve the highest expected gross profit possible.

In addition to our predictive analyses, we also achieved our goal of providing a recommendation system to producers to help them generate ideas for a movie they want to create based on films that have been made in the past. We envision producers being able to use any combination of the models and recommendations we generated to achieve their ultimate goals in creating a new film, knowing that these goals will vary between different producers.

In terms of model limitations, the movies in the data set were still generally well known, so collecting more relatively unknown movies may provide additional insights. Lastly, many variables we created based on frequency in the dataset are not completely accurate representations of how popular the stars, writers, or directors actually are or how large the production companies actually are, but we felt they worked as decent proxies given the correlations we saw with our response variables.

Natural Language for Movie Recommendations

Overview

We explored another application with the movie data. We matched a user's query with movie plot synopses in order to generate recommendations using natural language processing. We downloaded movie synopses from TMDb's API and saved them as data extracts in a python library uploaded to PyPI called **movienight**, along with the accompanying code for convenience. Our recommendations draw from the embeddings that are readily available in the **sentence-transformers** library, using cosine similarity as a method of measuring and ranking the quality of our recommendations.

Methodology

Sentence embeddings are the results of when sentences are mapped to vectors of real numbers. While we will not get into the details of specific sentence embeddings, the model we used by default, unless otherwise specified, was **all-MiniLM-L6-v2** which sentences to a 384 dimensional space.

Once our chosen query and movie synopses were all transformed from text to their corresponding embeddings, we calculated the query's cosine similarity with each movies' synopsis as defined below:

$$Cos(X,Y) = \frac{X \cdot Y}{||X||\,||Y||},\ where\ X\ and\ Y\ are\ embeddings$$

A completely orthogonal X and Y vectors result in a cosine similarity of 0, with a cosine similarity of 1 indicating perfectly identical sentences. In practical terms, we can interpret this as a way to find the similarity of two sentences, while also controlling for the length of the sentence.

The pre-trained **sentence-transformers** model, along with any relevant classes we created using the models, are imported when importing our PyPI package. Once imported, we deploy the model using **dash** to allow for a web app in which users can submit a query and resulting movie recommendations can be displayed.

Results

Results from the NLP movie recommendation search are promising at times, despite using a pre-trained model with no additional tuning, but has clear limitations. To illustrate its opportunities and pitfalls, we submit a few queries and analyze them below.

Query 1: “Intense story with a lot of gunfights and over the top action”

Results

Title	Overview
The Great Silence	A mute gunslinger fights in the defense of a group of outlaws and a vengeful young widow, agains...
Wild Tales	Six deadly stories that explore the extremities of human behaviour involving distressed people s...
Pulp Fiction	A burger-loving hit man, his philosophical partner, a drug-addled gangster's moll and a washed-u...
Shottas	A raw urban drama about two friends raised on the dangerous streets of Kingston, Jamaica. Biggs ...
Battleship Potemkin	A dramatized account of a great Russian naval mutiny and a resultant public demonstration, showi...

Results look decent, although filtering by different types of shooter-action films would likely surface more relevant results (e.g., westerns vs. more modern films).

Query 2: “A horror film that involves serial killers and has psychological thrills”

Results

Title	Overview
Scream	A killer known as Ghostface begins killing off teenagers, and as the body count begins rising, o...
The Enforcer	A noir thriller set in Miami, the film follows an enforcer who discovers his femme fatale boss h...
Wild Tales	Six deadly stories that explore the extremities of human behaviour involving distressed people s...
Se7en	Two homicide detectives are on a desperate hunt for a serial killer whose crimes are based on th...
Scary Movie	A familiar-looking group of teenagers find themselves being stalked by a more-than-vaguely recog...

Surprisingly, just based on NLP we are able to find some relevant horror films. This could be because the synopsis for horror films are often much more straightforward and explicit. There's still room for improvement by potentially splitting out by sub-genres.

Query 3: “A feel-good kids movie that teaches the importance of family”
Results

Title	Overview
Boyhood	The film tells a story of a divorced couple trying to raise their young son. The story follows t...
War Room	The family-friendly movie explores the transformational role prayer plays in the lives of the Jo...
Cinema Paradiso	A filmmaker recalls his childhood, when he fell in love with the movies at his village's theater...
Midnight in Paris	A romantic comedy about a family traveling to the French capital for business. The party include...
Au Revoir les Enfants	Au revoir les enfants tells a heartbreaking story of friendship and devastating loss concerning ...

Results are mixed, with *Boyhood* and *Cinema Paradiso* nailing the prompt while *Midnight in Paris* missing the mark. Opportunity for improvement here with an additional layer that allows for filtering by rating when keywords such as "children" or "kids" are present.

Query 4: “Story about immigrants pursuing the American Dream”
Results

Title	Overview
No One Gets Out Alive	An immigrant in search of the American dream is forced to take a room in a boarding house and so...
Everything Everywhere All at Once	An aging Chinese immigrant is swept up in an insane adventure, where she alone can save what's i...
The Banker	In the 1960s, two entrepreneurs hatch an ingenious business plan to fight for housing integratio...
Legends of the Fall	An epic tale of three brothers and their father living in the remote wilderness of 1900s USA and...
Summer of 85	What do you dream of when you're 16-years-old and in a seaside resort in Normandy in the 1980s? ...

Results here are completely hit or miss. The top pick *No One Gets Out Alive* is a horror film about an undocumented immigrant that encounters monsters in Cleveland. Shortcomings of this methodology are on full display here since using the synopsis fails to capture the broader context of the film. You could argue that our second pick Everything Everywhere All at Once is about immigrants but it's doubtful that our provided query would be the first way people would describe that movie. Still, other choices on our list seem to capture the essence of our query, which is promising.

Conclusion

Using NLP alone is likely inadequate for consistent, high-quality movie recommendations as the theme and context of the movie are rarely outlined clearly in a plot synopsis. Certain genres appear to do better than others, with horror films providing more explicit synopses that result in better recommendations and family-friendly and children's genres being more tricky. A filter that explicitly allows users to filter out certain genres would probably aid in improving the results.

Another area for opportunity would be to use movie reviews rather than the plot synopsis as reviews will often explicitly lay out the themes in a movie that are missing from the synopsis. Reviews would also contain how a viewer reacted to the movie, which would likely aid in the recommendations.