A Needle in a Data Haystack

Reel Patterns

A Deep Dive into the Data Behind the Scenes

Noam Kimhi
Or Forshmit
Adir Tuval

Table of Contents

Reel Patterns: Project Introduction

A Needle in a Data Haystack – Final Project

General Information

Title: **Reel Patterns** – A deep dive into the data behind the scenes of cinema.

Team Member Info:

	Noam Kimhi	**Or Forshmit**	**Adir Tuval**
CS id	noam.kimhi	or_forshmit8	adirt
Email	noam.kimhi@mail.huji.ac.il	or.forshmit@mail.huji.ac.il	adir.tuval@mail.huji.ac.il

Problem Description

In Reel Patterns, we explore the hidden and often surprising relationships within movie data. Rather than answering traditional questions like "What makes a movie successful?", we dive into more unconventional hypotheses and questions. We defined 3 main ideas we wanted to explore:

1. **What can I say, we cliqued: Finding communities in the film industry**
   Which actors tend to play together the most? Can we find cliques or patterns and identify movies they played together in? Which unexpected communities can we find? By analyzing co-casting patterns, we aim to reveal the cliques and on-screen partnerships that quietly shape the industry.
2. **Curtain call, please: When the audience is ready to say goodbye, but studios are not**
   We've all been there – The first film dazzles, the second might keep up with the hype, but somewhere around the third the magic fades. Our question is simple: As franchises create another sequel, how successful might that sequel turn out compared to the best movie in the franchise? By exploring this pattern, we aim to demonstrate that brand fatigue is real, and that, over time, it can erode the very franchises it tries to sustain.
3. **Reel hits meet real hits: Do ticket sales dance to the movie's track?**
   Music is an integral part of some movies, elevating them from a great film to a complete masterpiece. In this part, we will examine the role of music in cinematic success. We ask whether the popularity of a soundtrack correlates with box office performance or audience ratings. We hypothesize that movies with more popular soundtracks will also achieve higher audience ratings and revenues.

Data

We obtained a dataset containing over 1.2 million title records collected from ***Kaggle***, weighing 532 MB. The dataset includes key metadata for each movie, such as titles, release dates, genres, cast, etc. From ***IMDb*** we gathered tsv files for different categories (such as basic title information, title ratings, title cast etc.), weighing 8.12 GB.

For **Curtain call, please** we used multiple data sources. The first dataset comes from ***Kaggle*** with critic and audience ratings for movies (17 MB). The second was created using queries to the ***Wikidata database*** including franchise, revenue and budget data (1.6 MB).

For **Reel hits meet real hits**, using Spotify's API, we managed to collect data regarding 1,000 movie soundtracks weighing 254KB, since it required manual verification as well (See ***Reel hits meet real hits*** impediments).

What can I say, we cliqued

Finding communities in the film industry

Our Solution

We used community detection in order to find relationships in the film industry. Relationships between actors and filmmakers can be presented as undirected graphs through different algorithms we have seen in class:

1. Louvain: Highlights large-scale modular structures, showing clusters of actors that collaborate frequently across the industry (maximizing modularity).
2. Girvan-Newman: Reveals hierarchical splits by removing key "bridge actors", uncovering how communities break apart into subgroups (based on either edge betweenness, centrality or participation), while identifying key nodes.
3. Clique-Percolation: Focuses on tightly knit cliques by capturing overlapping groups of actors. Might lead to finding actors who often co-star in smaller, more exclusive circles.

There are various parameters to consider when detecting communities, that also depend on the kind of community algorithm we choose to use: number of nodes (actors and filmmakers), number of cliques, minimal weight of edges and many more. An easy way we found to manipulate these parameters is using `Streamlit` in Python.

≫ *Link to a video demo* ≫ *Link to the site*

Evaluation

Evaluation Criteria

First, we expect to be able to identify communities (cliques) of actors and filmmakers who tend to play together in movie franchises, or famous actors and directors. However, a better evaluation will include some kind of numerical value. Therefore, we present the value of modularity in the site for the users to see, and higher modularity values (closer to 1) can indicate a strong community structure.

In GN, we wish to identify *key nodes*, and look for actors and filmmakers who "bridge" between communities (marked with a ⋆ on the site). Then, evaluate if those key nodes make sense as bridges, and between which communities.

With Clique-Percolation, we want to see if we can dive into deeper resolutions inside larger scale communities, and whether we can make sense of these divisions.

We note that Louvain algorithm is non-deterministic and is based on a random node order. We made sure there was no chance in our findings by seeding the algorithm with a constant seed and comparing results using different seeds.

Setup

The setup involved cleaning the datasets (handling missing values, etc.) and merging them based on different types of information, such as titles, actors, and region. We then constructed the graph and set up the site using `Streamlit`, selecting which parameters to include, defining their minimum and maximum values, and adding features (3D visualizations) for improved readability.

Results & Visualizations

There are many possible results to share since the scale of the data we managed to find is large. In this section, we will share some interesting results we found using different algorithms. We chose to visualize our results using graphs and dendrograms, since these are the most suitable ways to present detected communities, as seen in class.

Avengers!!! Have already... assembled? Using Louvain, we managed to find and identify interesting Cliques of highly popular Hollywood actors, such as actors who participated in "Avengers" movies (Robert Downey Jr., Chris Evans, etc.). This partition yielded a modularity value of 0.784.

Note: Some snapshots may be crowded since it is hard to capture the 3D graph as a picture. For better quality use the site.

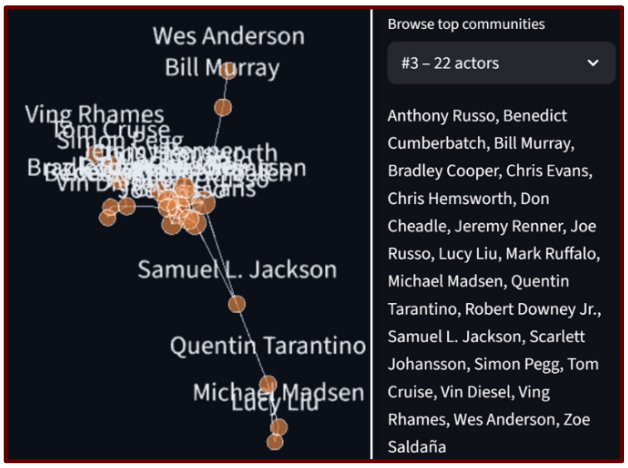

***Figure 1** – Hollywood Clique*

Now say it in Japanese: An interesting and surprising relation we found is how Japanese voice actors shaped the graphs because of their voice acting in movies. Using a feature we added that highlights ***bridges*** in GN algorithm, we found a JP voice actor called Natsuki Hanae who voiced many characters in Japanese and is also a bridge between American and Japanese VAs. This partition yielded a modularity value of 0.676.

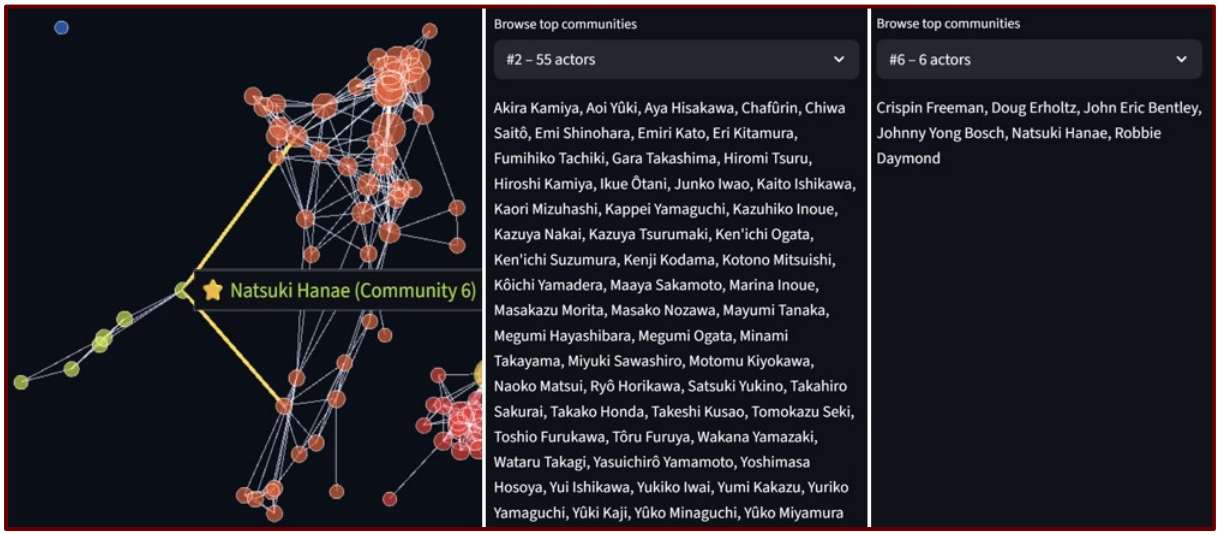

***Figure 2** – Japanese Voice Actor Bridge*

Guardians of the ~~Galaxy~~ Clique: Using Clique-Percolation, we managed to get refined communities that were previously grouped as a single community (using Louvain). We broke the community we previously detected as "famous Hollywood actors" into smaller, more precise communities according to franchises. In ***figure 3.1*** we identified Samuel L. Jackson as a bridge between "Kill

Bill" and "Avengers". This makes sense since Tarantino tends to cast Jackson in his movies often, and Jackson also plays the role of *Nick Fury* in "The Avengers" franchise. This partition, with clique size of 3, yielded a modularity value of 0.679. In the following figures, you can see some of the communities we mentioned:

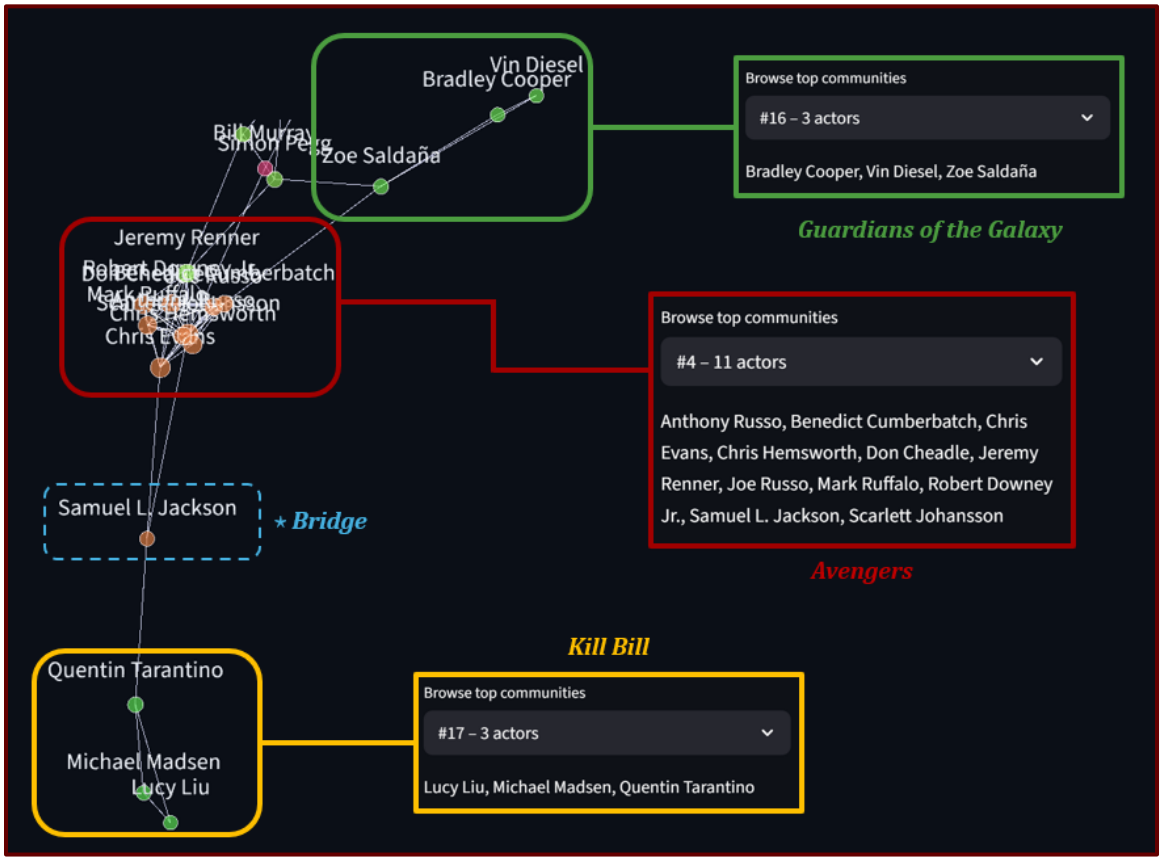

Figure 3.1 – *Refined Communities Using Clique-Percolation*

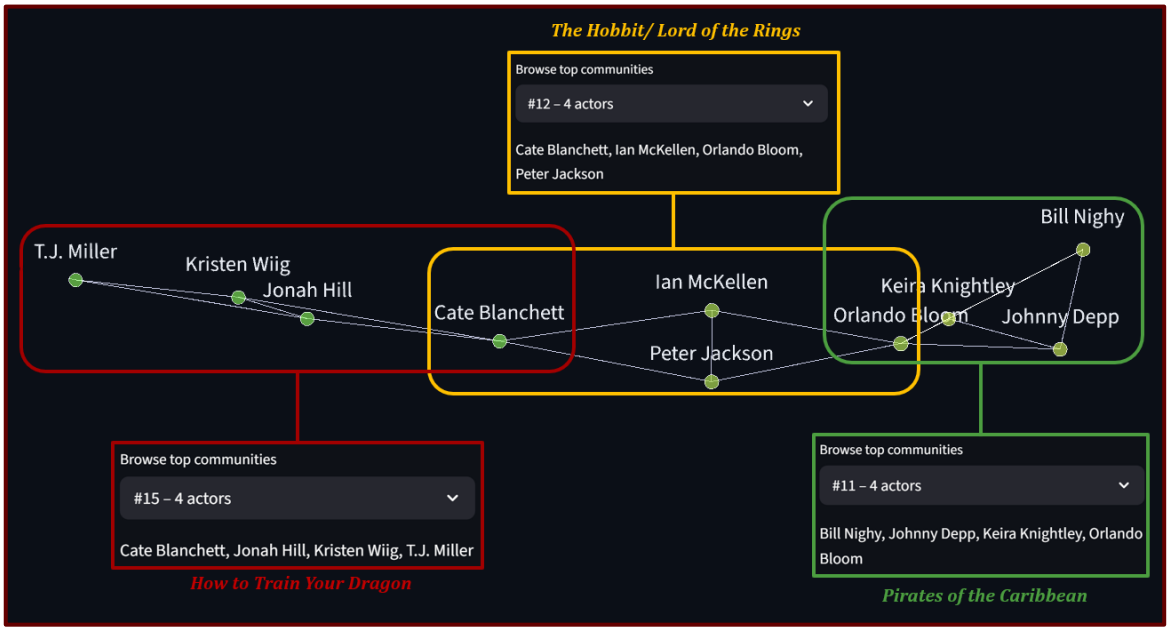

Figure 3.2 - *Refined Communities Using Clique-Percolation*

Fast and Clustered – Using the dendrogram option, we identified additional connections and highlighted movies such as "Fast and Furious" and "Deadpool". The dendrogram also provided a high-level view of connections within Hollywood.

Figure 4 – *Communities Found Using the Dendrogram*

Impediments

While searching in this direction we faced some issues:

1. Dataset size: Running all the collected data through the algorithms was overwhelming and hurt responsiveness. To address this, we defined filtering criteria for which movies to include, as implemented in `organize_data.py`.
2. Too many regions: Since many regions dominated our data, we limited the dataset to movies released in the US, aiming to approximate the Hollywood industry as closely as possible.

Curtain call, please

When the audience is ready to say goodbye, but studios are not

Our Solution

To study when franchises risk brand fatigue, we analyzed sequels in order of release and asked whether each one matched or surpassed the best performance seen so far. We defined success as being at least as strong as the previous peak, then estimated the probability of success at each sequel position (2nd, 3rd, 4th, etc.). Success was measured along three dimensions: critic reviews, audience reviews and return on investment (ROI).

We applied Bayesian shrinkage to stabilize sequel success probabilities. Unlike naive estimates that become noisy when dealing with franchises with few sequels. Shrinkage handles extremes by combining local rates with the global average, yielding more reliable and interpretable trends.

Evaluation

Evaluation Criteria

We define success as the ability of our method to uncover clear and interpretable trends in franchise performance across successive releases. In particular, we expect evidence of brand fatigue, shown by a declining probability of success in later installments. Any consistent pattern is considered a valid outcome. To guard against chance findings, we evaluated results across multiple franchises rather than relying on single examples.

Setup

We constructed a dataset of movie franchise by querying the Wikidata database for films and their associated franchise IDs. This was enriched with revenues, budgets, and review scores from Rotten Tomatoes and TMDB. After cleaning the data and removing incomplete entries, we focused our analysis on consistent, well-defined franchises.

To avoid overfitting to isolated cases, we limited comparisons to the first six installments of each franchise, since few extended further – hence confidence level dropped. Finally, we explored visualization methods designed to reveal whether clear trends of brand fatigue emerged.

Results & Visualization

We began by examining how many movies appear at each sequel index (1st film, 2nd, 3rd, etc.):

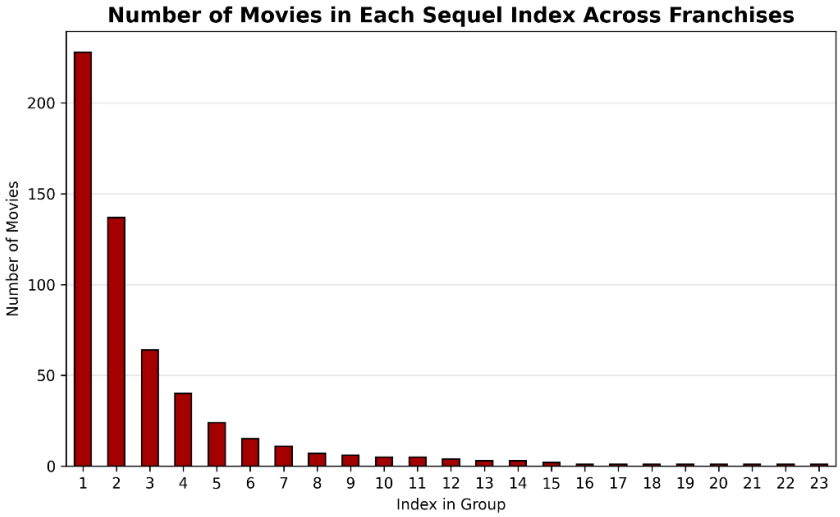

Figure 5 – *Number of Movies in Each Sequels Index Across Franchises*

From this distribution, we tested success probabilities up to the 6th or 7th sequel. Since confidence levels dropped noticeably beyond the 6th, we chose to cap our analysis at six installments.

We defined success as an ability to create a trend that will put light on whether brand fatigue is a real issue. In order to do so, we wanted to create a visualization that will clearly present a change in probability. Here are the results according to different success metrics:

Note: Since we examine the probability of a sequel matching or surpassing the best performance so far, it made no sense to include the first movie in the franchise. Hence, all the plots below start at index 2, meaning the second movie in the franchise onwards.

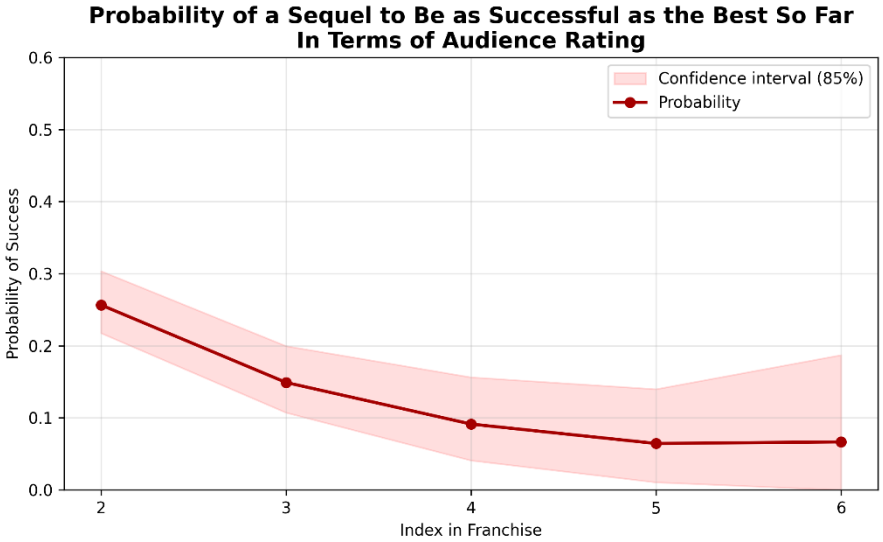

Figure 6 – *Probability of Success with Audience Rating Metric*

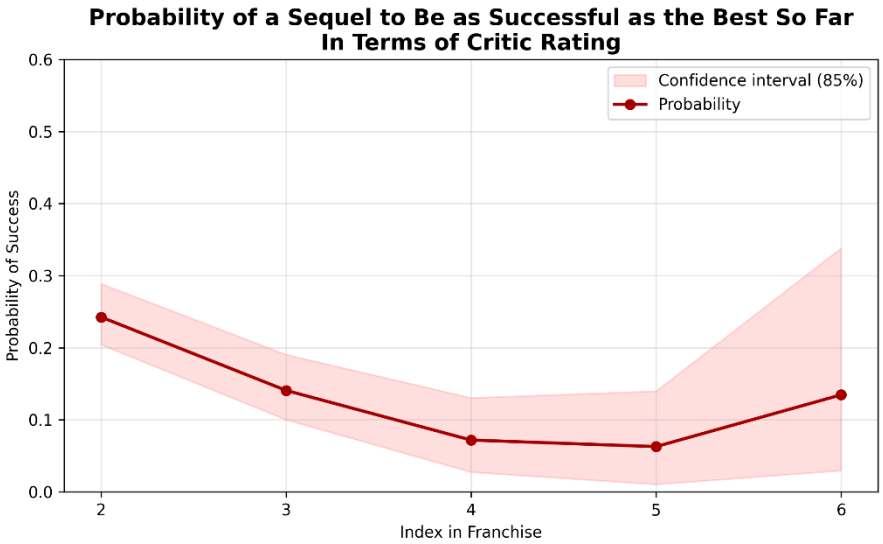

Figure 7 *– Probability of Success with Critic Rating Metric*

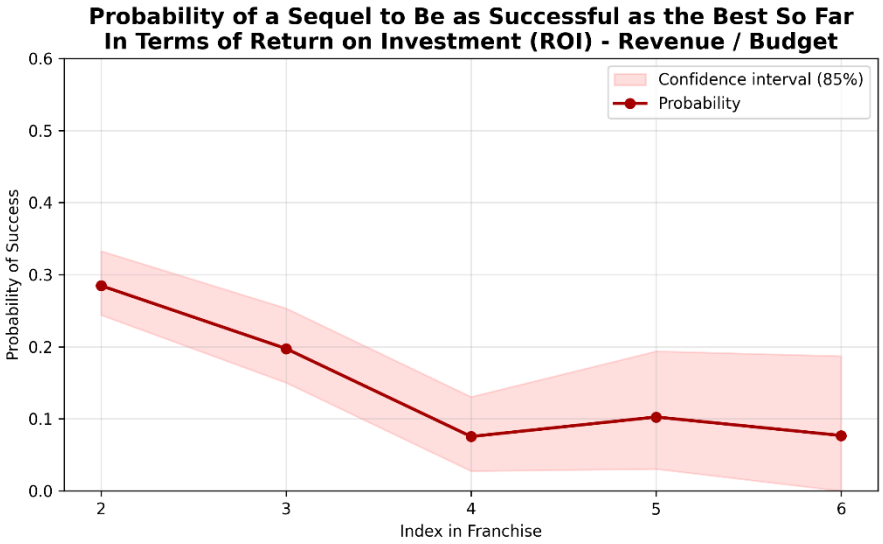

Figure 8 *- Probability of Success with ROI Metric*

Across all three metrics, we observe a clear downward trend in the probability that sequels will surpass the franchise's best performance so far. From the second through the fourth installment, the likelihood of success steadily declines, illustrating the onset of brand fatigue. Beyond this point, in the fifth and sixth entries, probabilities rise slightly – but these increases coincide with wider confidence intervals and remain well below the levels seen for second and third films.

Notably, the probability that a second film matches or exceeds the first hovers just under 0.3 across metrics, aligning with our initial expectation that most sequels struggle to replicate the impact of the original.

We chose to present 85% confidence intervals rather than the conventional 95%. The reason is that our dataset contains relatively few franchises with a large number of sequels, which causes confidence intervals to widen at higher sequel indices.

Impediments

Our first challenge was identifying franchises, as the raw data did not explicitly group films together. Textual matching risked errors, so we queried Wikidata to retrieve franchises.

Once the dataset was cleaned and enriched with revenue, budget, and review data, we faced design choices such as whether to treat the first film as a baseline and whether to analyze trends within individual franchises or across the entire set.

Finally, estimating probabilities for later sequels proved difficult, since only a few franchises reached those numbers, leading to noisy results that required smoothing.

Reel hits meet real hits

Do ticket sales dance to the movie's track?

Our Solution

We set out to explore the link between a movie's success and the success of its soundtrack on **Spotify**. First, we defined movie success using two metrics: revenue and rating. We then matched each movie with its corresponding soundtrack, enabling us to test for correlations between cinematic performance and soundtrack popularity. Soundtrack success was measured using Spotify's album *popularity* – a relative, undisclosed metric that reflects how an album performs compared to others on the platform.

Evaluation

Evaluation Criteria

We define success as uncovering statistically meaningful and interpretable links between movie success metrics (revenue, ratings) and soundtrack features (album popularity, track count and album length). Our method's performance is evaluated by its ability to discover strong, statistically significant and interpretable correlations between cinematic success and soundtrack popularity.

Setup

The key challenge in this phase was enriching our movie dataset (which already included attributes such as ratings and revenue) with reliable soundtrack information. Since soundtrack data was not directly available (due to Spotify's API terms of use), we turned to external sources and APIs to extract relevant details based on movie titles and metadata. This process involved careful matching to ensure that each soundtrack correctly corresponds to its movie, minimizing mismatches and bias. The final step was merging the soundtrack data with the movie dataset, creating a unified resource that enabled meaningful correlation analysis between cinematic success and soundtrack popularity.

Results & Visualization

To ensure our analysis reflected credibility, we gave greater weight to movies with a higher number of user ratings, since their evaluations are more reliable. We then explored the relationships between movie performance metrics (revenue, vote average) and soundtrack characteristics (album popularity, number of tracks, album length). We chose to ignore movies in which the soundtrack popularity was 0, or the revenue of the movie was at the bottom 5% of our data, because extreme blockbuster flops can dominate the scale and distort the correlation values, while we are looking for general trends rather than special cases.

To capture different types of relationships, we generated two correlation heatmaps:

1. Pearson correlation, which highlights linear relationships between variables.
2. Spearman correlation, which captures monotonic but potentially non-linear relationships.

Heatmaps visualization allowed us to identify patterns and dependencies between movie success metrics and soundtrack features:

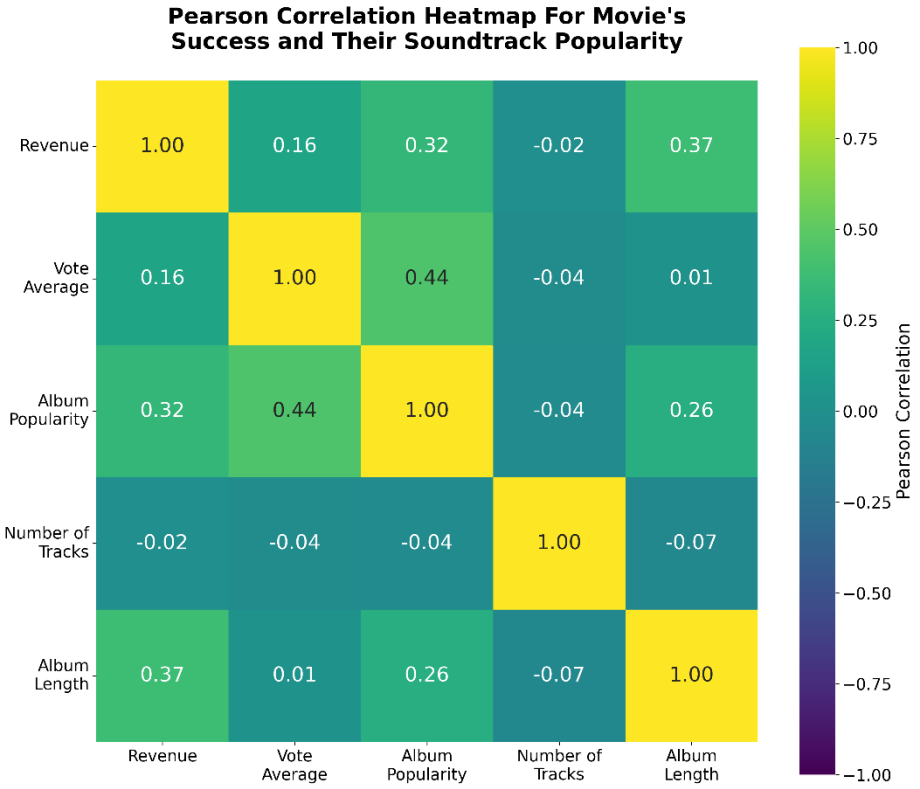

Figure 9 – *Pearson Correlation Heatmap: Movie Success & Spotify Soundtrack Popularity*

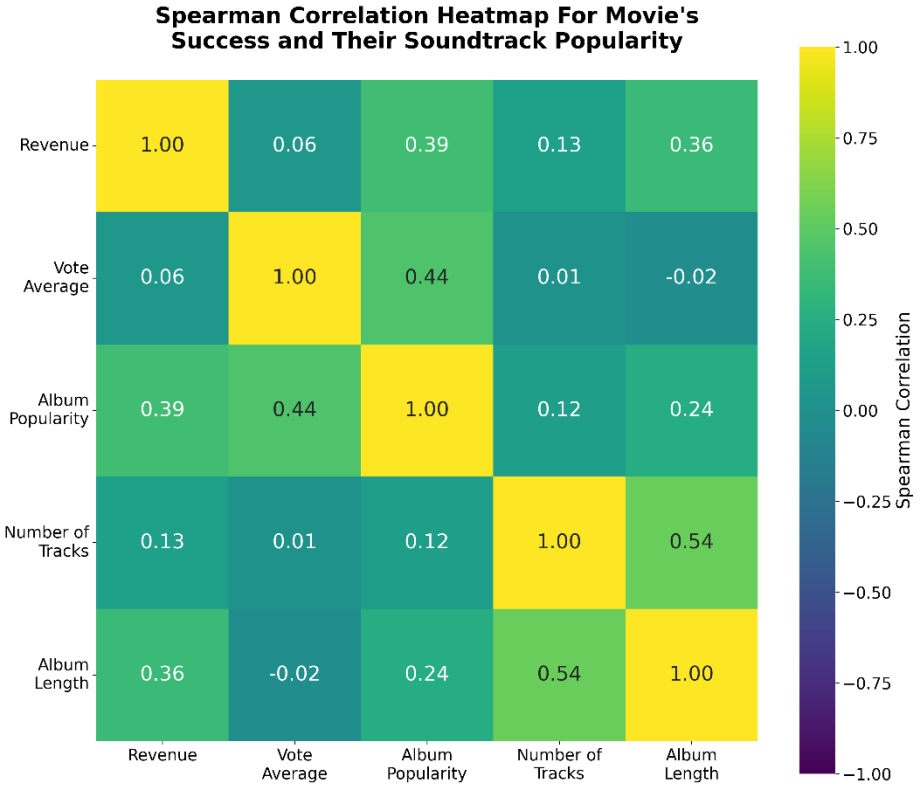

Figure 10 – *Spearman Correlation Heatmap: Movie Success & Spotify Soundtrack Popularity*

(*) All reported correlations are statistically significant at $p < 0.03$.

Following that, we wanted to dive deeper into a relationship that stood out as particularly interesting (highest meaningful correlation value). We created a plot of the relationship between album popularity and vote average, to further enrich our understanding:

Note: Ratings range from 0–10, but in practice the movies we present are rated 4–9, so the y-axis starts at 4 for clarity.

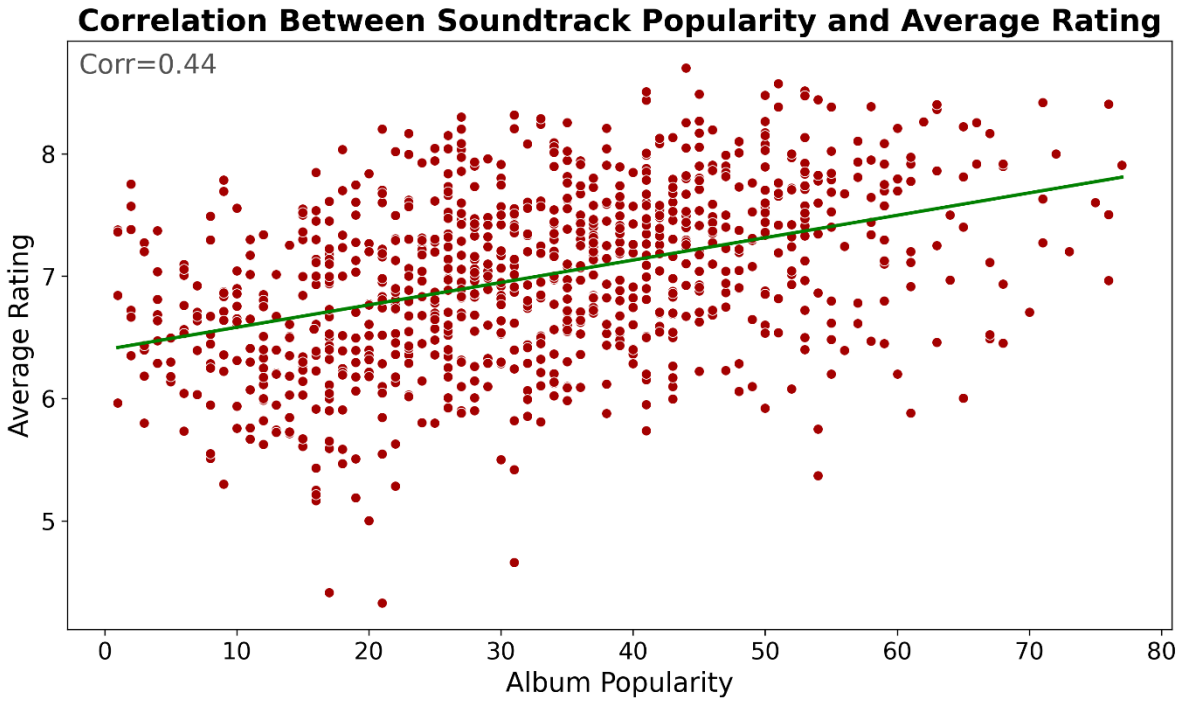

Figure 11 – *Scatter Plot of the Relationship Between Soundtrack Popularity and Average Rating*

Scatter plot is an effective way for showing how soundtrack popularity relates to movie ratings because it displays individual data points and trends. In this case, it visualizes the correlation we observed in the heatmap.

The plot supports our initial hypothesis regarding the link between these two success metrics. However, it is important to stress that correlation does not imply causation, and each case must be considered individually. Furthermore, while higher soundtrack popularity tends to align with higher ratings, the broad dispersion of data points and the modest correlation value highlight that many additional factors beyond music influence audience evaluation.

Impediments

Linking movies to their soundtracks required searching for additional datasets and devising a method to reliably retrieve soundtrack information based on a movie's title and other attributes.

This proved challenging, even with access to Spotify's API. Spotify do not offer an option to tag albums as 'movie related', so we had to design careful queries to capture the correct soundtracks, and then manually verify that the albums returned by `spotipy` matched the intended movies rather than unrelated releases.

Manual verification, combined with Spotify's daily request limits, restricted our work to a subset of 1,000 movies instead of the full dataset, replacing films without official albums. The combination of query design, manual verification, and API restrictions made data collection the most time-consuming and resource-intensive stage of this part of the project.

Reel Patterns: Conclusion

As the journey comes to an end

Future Work

First, the most straightforward expansion would be to explore the same directions we defined in this project on TV shows as well. Moreover, we have several ideas for enjoyable and interesting expansions to our project according to each chapter:

1. **What can I say, we cliqued:**
   We can look for association rules over co-casting "baskets." Treat each movie as a basket of contributors and mine frequent itemsets/rules (e.g., {Director=X, Actor=Y} ⟹ {Actor=Z}) to discover casting motifs that transcend communities. Then compare rules inside vs. across communities. Another fun idea could be extending the actor graph to a multilayer network (actors-directors-composers) and rank nodes with PageRank to surface role-specific influence.
2. **Curtain call, please:**
   Create a prediction model that based on the gathered data indicates the studios whether they should create another sequel to their franchise.
3. **Reel hits meet real hits:**
   Due to data limitations, this chapter was examined on a subset of 1,000 movies – we can expand this subset and potentially cover the entire movie database with more information, to see if the trends we found stay consistent across the entire data. Another intriguing idea is creating a movie↔artist bipartite graph and run PageRank-style propagation to measure artist "cinematic impact", then test how these scores relate to revenue/ratings.

Conclusion

Reel Patterns explored how films succeed and connect from three angles. Co-casting networks revealed clear, interpretable communities and bridge performers – showing that collaboration is patterned, not random. Franchise dynamics showed diminishing success: higher-index sequels less often match or beat a brand's best, as we expected. Soundtrack popularity correlates only modestly with revenue and ratings – music can amplify a film but can't replace core quality, with memorable outliers in both directions.

On the process side, for some of us it was our first experience working with big data. We wrangled messy metadata, limit tested Spotify's API and debated (politely) chapter names. Turning this chaos into stories was not only rewarding – it was genuinely fun.

P.S.

Yes, we really do have a section in the report called "Guardians of the Clique".

No, we do not regret it :)