Movie Analytics Report: Trends, Clusters, and Recommendations

Prepared By: David Le and Yaqin Albirawi

INFO-6150: Data Mining & Analysis

August 12, 2025

Objective

This report presents the data extraction and visualization of movie data to uncover insights into popularity, ratings, genres, and release dates. The analysis aims to identify meaningful patterns in movie characteristics, highlight trends over time, and demonstrate how the application of machine learning and statistical techniques can reveal deeper insights and generate movie recommendations.

Data

Data Collection

Movie data was collected from The Movie Database (TMDB) API v3. After creating an account and obtaining an API key, authenticated GET requests were made to retrieve results in JSON format. Since each API call returned only 20 movies ranked by popularity, a Python loop was used to iterate through all pages and compile the results into a Pandas DataFrame. **Please refer to the Appendix section for the Python script of data extraction.**

Data Description

Around 10,000 of the most popular movies were extracted as of August 8, 2025. The key movie information extracted and used includes:

- genre_ids: list of genre and sub-genre codes
- original_language: language code (e.g., 'en')
- title: movie name
- popularity: popularity score (TMDb's composite metric)
    - See References for more details on how this metric is calculated
- release_date: release date (YYYY-MM-DD)
- vote_average: average user rating (0-10 scale)
- vote_count: number of user ratings

Data Transformations

In the Python data extraction script, the genre_ids column was transformed into a new column containing genre names as text rather than numeric identifiers. The data was then prepared for analysis by creating separate columns for each genre and applying one-hot encoding, assigning a value of 1 to indicate the presence of each genre for a given movie.

Within the Power BI interface, the following data transformations were performed:

1. Removed columns that are irrelevant or not useful
2. Replaced empty one-hot encoded genre columns with 0
3. Created custom field that extracts the first genre (main genre) from list of genres
4. Filtered out movies with no release date and release dates after July 31, 2025
5. Changed column types to numeric, integer, or date where applicable
6. Filtered out records with no genre

Visualization for Decision-Making

The dashboard in the “Data Story Telling” tab contains interactive plots and information that can be used by decision-makers for high-level insights.

Information Available

The following highlights the type of information available in this dashboard:

- Total content count
- Table of movies ordered by popularity
- Popularity and vote/rating scores by month
- Table of movie counts by language
- Genre-specific popularity trends (scroll-enabled)
- Movie release counts by release year

Data Slicers

Useful data slicers are also available to filter the data by the features below.

- Genre: the main movie genre (scroll-enabled and multiselect-enabled)
- Year: movie release year range
    - *Note: can extend year range to 2025, but popularity plots will be skewed or distorted by recent movie releases that are trending*
- Language: the “Movie Count by Language” table can be used to filter for movies originally released in a specific language

Key Insights

Seasonal Popularity: “Average Popularity and Vote Scores by Month” plot shows that movies released from May to July tend to achieve the highest popularity scores. However, seasonal patterns vary by genre. For instance, documentaries peak in popularity around October, while some genres maintain relatively consistent performance throughout the year. These insights can help inform strategic decisions on optimal release timing for different types of films.

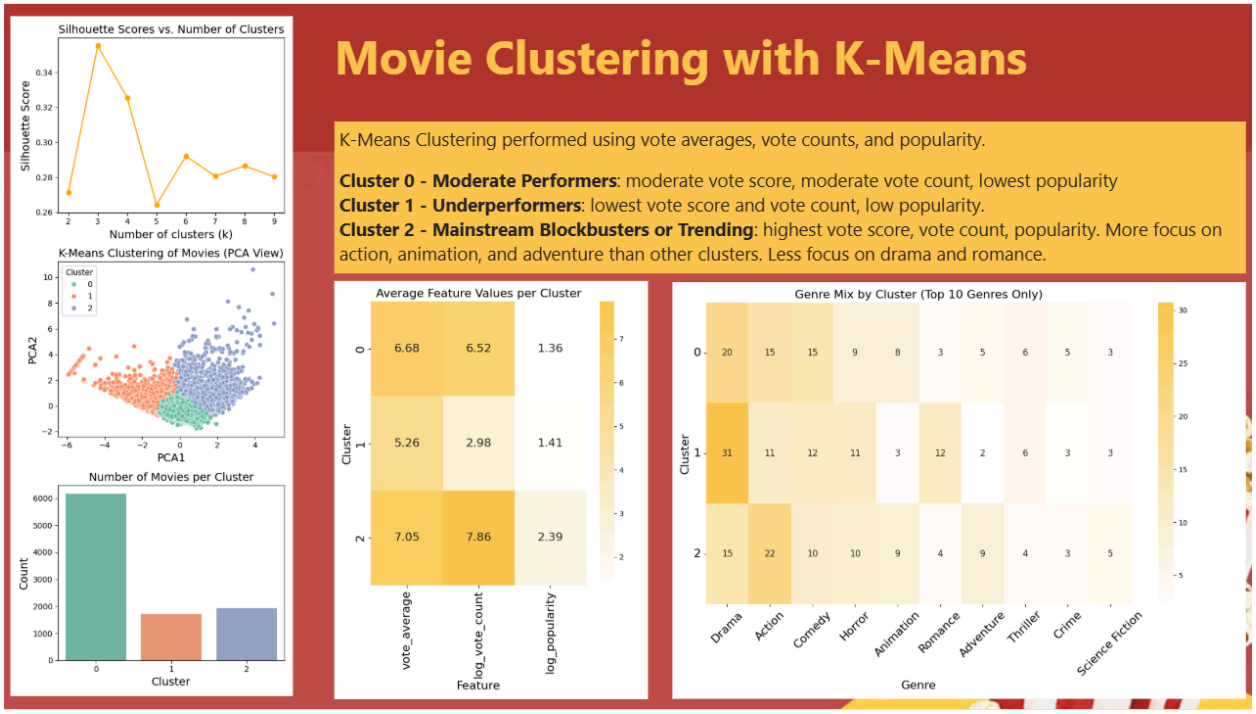
Film Industry Growth: “Movie Count by Release Year” plot shows a steady upward trend in the number of movies released annually, reflecting growth in film production over time (with the exception of a drop in 2020, which is likely due to the global disruptions caused by COVID-19).

Language Representation: English-language films dominate the dataset, followed by Japanese and Korean titles. However, when filtering by individual languages and examining release counts over time, Japanese films show a consistent year-over-year increase, while Korean film releases peaked in 2017. These trends may reflect the growing global influence of Japanese cinema, particularly animation, and point to potential opportunities for increased production or investment in those markets.

Genre Popularity Trends: The “Popularity Score by Release Year” plots for each genre can reveal which genre is steadily popular and which is trending upwards. This can help decide areas to invest time and money in. For example, Action and Adventure show sustained popularity over the years, Animation shows growing popularity, while Comedy remains steady without significant popularity growth.

Machine Learning: Clustering

Clustering was applied to the data to uncover natural groupings based on audience reception. Features included vote average, and log-transformed vote count and popularity to reduce skewness. The silhouette score indicated k = 3 as the optimal number of clusters as seen by the “Data Showcasing” dashboard below.



Cluster 0 contains the largest number of movies, followed by cluster 2 and cluster 1. The “Average Feature Values per Cluster” figure indicates that Cluster 0 falls within mid-range values for vote average, vote count, and popularity, suggesting it contains movies that are

moderate performers. Cluster 2 has the highest values across all three metrics, thus labeled as Mainstream Blockbusters or Trending. While Cluster 1 (Underperformers) records the lowest.

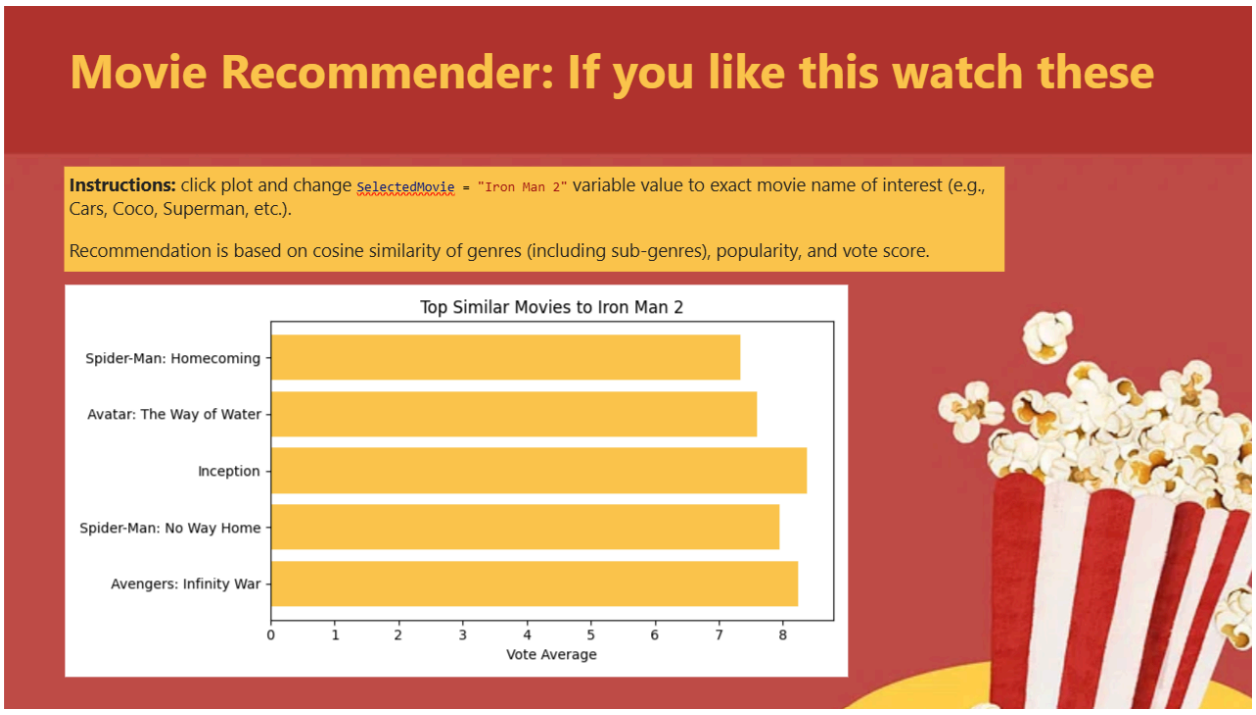
The “Genre Mix by Cluster” heatmap reveals that Cluster 2 (Blockbusters) tends to have a higher distribution of Action, Animation and Adventure compared to the other clusters. While Cluster 1 (Underperformers) are more heavily represented in Drama and Romance. Moderate performers (Cluster 0) exhibit a more balanced genre distribution.

The patterns suggest that mainstream audiences are more drawn to action-packed films, consistent with the success of major franchises such as Marvel, whereas drama and romance films tend to attract smaller audiences.

Machine Learning: Recommender System

A content-based recommender system was developed using cosine similarity applied to movie genres (including sub-genres), popularity, and vote score features. These features were selected to capture thematic similarity and audience reception, allowing the recommendations to reflect not only what a movie is about but also how well it is received.

The image below provides a sample output where the system recommends five movies similar to *Iron Man 2*. Three of the suggestions (*Spider-Man: Homecoming*, *Spider-Man: No Way Home*, and *Avengers: Infinity War*) are part of the Marvel franchise, while *Avatar: The Way of Water* and *Inception* may share comparable themes and/or audience appeal.

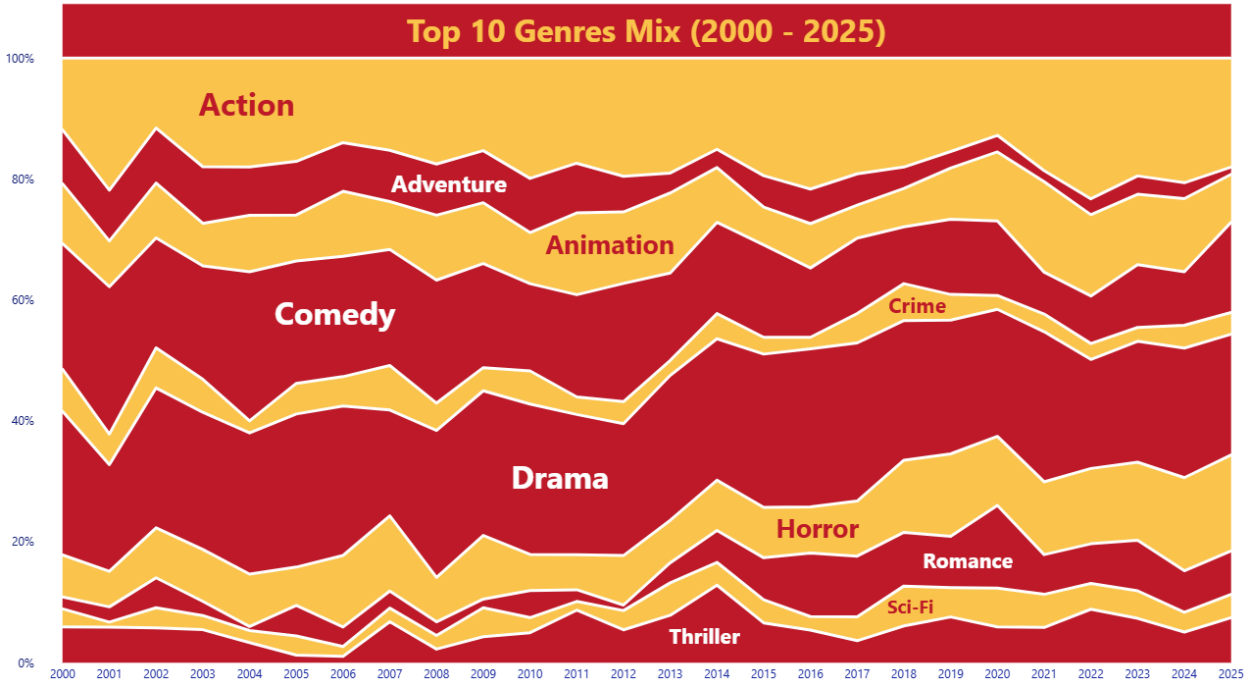


For data scientists and engineers, this system can serve as a prototype for a content-based recommendation system. The current implementation provides a concept of what could be integrated into streaming platforms or marketing analytics tools to enhance user engagement and retention.

Insights Through Data Art

The following visualization highlights the shifting composition of the ten most popular movie genres over the past 25 years. The flowing, stacked design provides an artistic view that emphasizes how genre preferences evolve, offering a clear, visually engaging view of trends.

For this stacked plot, looking at the width of each genre section and changes in width (e.g., thickens or narrows) helps reveal trends. That being said, Drama and Action have consistently dominated the genre landscape. Comedy maintains a strong but gradually declining presence, and Adventure's share has steadily decreased over the years. Animation, Horror, Romance, and Thrillers show intermittent growth, with notable spikes in certain years. Genres such as Crime and Sci-Fi remain smaller but stable contributors.



Conclusion

This project demonstrated that API-extracted movie data can be transformed and visualized to deliver actionable insights for strategic and technical use. The dashboard for decision-making highlighted trends and patterns in popularity, language, and genre. Clustering revealed distinct performance segments and movie profiles. Moreover, a recommender system demonstrated the potential for content suggestion. Finally, data art visualizations offered a creative view of long-term genre shifts. These methods presented the power of combining analytics, machine learning, and visualization to guide decisions in the film industry.

References

karmaecrivain94. (2018, October 25). [OC] The prevalence of movie genres between 1894 and 2025 [Post]. Reddit. r/DataIsBeautiful. Retrieved August 11, 2025, from https://www.reddit.com/r/dataisbeautiful/comments/8dyqfk/oc_the_prevalence_of_movie_genres_between_1894/

The Movie Database (n.d.). Getting started. TMDB Developer Documentation. Retrieved July 25, 2025, from https://developer.themoviedb.org/docs/getting-started

The Movie Database. (n.d.). Popularity & Trending. TMDB Developer Documentation. Retrieved August 11, 2025, from https://developer.themoviedb.org/docs/popularity-and-trending

Appendix: PowerBI Code for Data Extract

```
let
Source = Python.Execute("import requests#(lf)import pandas as pd#(lf)url =
""https://api.themoviedb.org/3/movie/popular?language=en-US""#(lf)#(lf)headers = {#(lf)
""accept"": ""application/json"",#(lf) ""Authorization"": ""Bearer
eyJhbGciOiJIUzI1NiJ9.eyJhdWQiOiJlZTlkMWM1MjE1NTZkMjNjMGMxNzAwNjkzM2RjNzU0O
SIsIm5iZiI6MTc1MzE0MTI4Mi41NzUsInN1YiI6IjY4N2VkMDIyNjU4NWJkNGVkYmQ5OTMzYiI
sInNjb3BlcyI6WyJhcGlfcmVhZCJdLCJ2ZXJzaW9uIjoxfQ.p0xkD4gOrrRxb2Y2Y4NVy9t8l-
wej31okWn2ejhGkvE""#(lf)}#(lf)#(lf)results = True#(lf)page = 1#(lf)popular_films =
[]#(lf)while results:#(lf) response = requests.get(url + f""&page={page}"",
headers=headers)#(lf) data = response.json()#(lf) results = data.get(""results"", [])#(lf)
popular_films.extend(results)#(lf) page += 1#(lf)#(lf)#(lf)#(lf)df =
pd.DataFrame(popular_films)#(lf)#(lf)genre_dict = {#(lf) 28: ""Action"",#(lf) 12:
""Adventure"",#(lf) 16: ""Animation"",#(lf) 35: ""Comedy"",#(lf) 80: ""Crime"",#(lf) 99:
""Documentary"",#(lf) 18: ""Drama"",#(lf) 10751: ""Family"",#(lf) 14: ""Fantasy"",#(lf)
36: ""History"",#(lf) 27: ""Horror"",#(lf) 10402: ""Music"",#(lf) 9648: ""Mystery"",#(lf)
10749: ""Romance"",#(lf) 878: ""Science Fiction"",#(lf) 10770: ""TV Movie"",#(lf) 53:
""Thriller"",#(lf) 10752: ""War"",#(lf) 37: ""Western""#(lf)}#(lf)#(lf)# now change genre_ids
to the text#(lf)df[""genre_ids""] = df[""genre_ids""].apply(lambda x: [genre_dict[i] for i in
x])#(lf)df2 = pd.DataFrame(columns=['Action', 'Adventure', 'Animation', 'Comedy', 'Crime',
'Documentary', 'Drama', 'Family', 'Fantasy', 'History', 'Horror', 'Music', 'Mystery', 'Romance',
'Science Fiction', 'TV Movie', 'Thriller', 'War', 'Western'])#(lf)#(lf)# one hot encode genre_ids
into df2#(lf)for index, row in df.iterrows():#(lf) for genre in row['genre_ids']:#(lf)
df2.loc[index, genre] = 1#(lf)df2.head()#(lf)#(lf)# combine df with df2#(lf)df = pd.concat([df,
df2], axis=1)#(lf)#(lf)df.head()"),
df1 = Source{[Name="df"]}[Value],
#"Removed Columns" = Table.RemoveColumns(df1,{"adult", "backdrop_path", "id",
"overview", "video", "poster_path", "original_title"})
in
#"Removed Columns"
```