# Data-Driven Analysis of the Cinema Industry: A Visualization Study on the TMDB 5000 Dataset

*YZV475E Data Visualization Term Project Report


1st Emir Arda Eker
*AI and Data Eng.*
*Istanbul Technical University*
ekere22@itu.edu.tr

2nd Yusuf Karamustafa
*AI and Data Eng.*
*Istanbul Technical University*
karamustafay21@itu.edu.tr

3rd Alper Düzgün
*AI and Data Eng.*
*Istanbul Technical University*
duzguna22@itu.edu.tr



***Abstract*—The objective of this study is to unveil the complex relationships between financial success, popularity, and artist networks in the cinema industry using the TMDB (The Movie Database) dataset. The study encompasses raw data cleaning, feature extraction, and a comprehensive Exploratory Data Analysis (EDA) process performed using Python. The findings are visualized through a three-page interactive dashboard created on the Power BI platform. The analysis reveals that while there is a strong correlation between budget and revenue, quality develops independently of the budget; specific clusters of actors create a "Star Power" effect on the box office; and the horror genre yields the highest Return on Investment (ROI).**



***Index Terms*—Data Visualization, Power BI, EDA, Social Network Analysis, TMDB, Film Industry.**


## I. Introduction

The film industry is a multi-billion dollar sector characterized by high risks and high reward potential. For producers and investors, predicting the box office success of a movie is of critical importance. This project analyzes the TMDB dataset, covering approximately 5000 films, to seek data-driven answers to the question: "What are the factors that make a film successful?"

The study consists of two main phases: a data processing and statistical analysis (EDA) phase using Python, followed by a storytelling and visualization phase using Power BI. The subsequent sections of this report detail the data preparation, statistical findings, and visualization strategy.

## II. Data Preparation and Processing

Two main data files, *tmdb_5000_movies.csv* and *tmdb_5000_credits.csv*, were used in this project. Data processing steps were carried out using Python (Pandas library).

### A. Data Merging and Cleaning

The two datasets were merged using the 'movie_id' key. Records with 'Revenue' and 'Budget' values of 0 were not excluded from the analysis; instead, a flag column named 'is_financial_valid' was created to allow filtering during financial analyses.

### B. JSON Parsing

The 'Genres', 'Keywords', 'Cast', and 'Crew' columns in the dataset contained nested JSON structures. These structures were parsed (using the 'explode' operation) and flattened to produce 6 different CSV files suitable for the Power BI relational model (Star Schema):

- *cleaned_movies.csv*: Main movie data.
- *movie_genres.csv*: Movie-Genre mappings.
- *actor_network.csv*: Actor relationships (Graph data).
- *correlation_matrix.csv*: Correlation data for the heatmap.

## III. Exploratory Data Analysis (EDA)

In-depth statistical analyses performed on the dataset provided critical insights that guided the dashboard design.

### A. Financial Correlations

As expected, a strong positive correlation ($r = 0.73$) was observed between Budget and Revenue. However, the correlation between 'vote_average', representing film quality, and 'budget' was remarkably low ($r = 0.09$). This demonstrates that a high budget does not guarantee a high-quality production (Fig. 1).

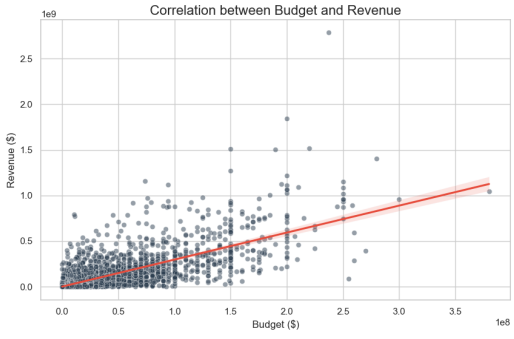


Fig. 1. Correlation matrix between variables. While the relationship between Budget and Revenue is distinct, Score and Budget are uncorrelated.

### B. Outliers and ROI

Upon analyzing Return on Investment (ROI), low-budget horror films such as "Paranormal Activity" and "The Blair Witch Project" were found to provide profitability exceeding 10,000%. These outliers indicate that the horror genre has a high financial leverage effect.

### C. Popularity Distribution

The popularity metric exhibits a right-skewed distribution. While 95% of the data falls below 50 points, extreme values such as "Minions" (875) exist. Therefore, it was decided to use a Logarithmic Scale in visualizations.

### D. Network Analysis

An examination of cast lists revealed distinct "Clustering" within the sector. For instance, actors in Marvel movies or Wes Anderson's team form closed-loop networks with intense internal ties, isolated from other groups.

## IV. Visualization Strategy

To meet project requirements and present the data as a cohesive story, a 3-page Power BI Dashboard architecture was constructed.

### A. Page 1: Executive Summary

This page summarizes the general state of the industry for the user.

- **Visuals Used:** Geospatial Map (Country-based production distribution), Area Chart (Annual Revenue Trend).
- **Features:** A "Tooltip" feature on the map displays the total number of films produced in a country when hovered over.

### B. Page 2: Financial Deep Dive

Investigates the relationship between financial success and quality.

- **Visuals Used:** Scatter Plot (Budget vs Revenue on Logarithmic Axis), Correlation Matrix Heatmap.
- **Technical Detail:** A "Decomposition Tree" visual allows for interactive breakdowns of revenue by director and genre.

### C. Page 3: Talent Ecosystem

This is the most innovative section of the project, analyzing actor networks and director performances.

- **Visuals Used:** Network Navigator (Graph Visualization), Top 10 Directors Bar Chart.
- **Data Refinement:** To prevent the "Hairball" (clutter) effect in the network graph, only actor pairs who have co-starred in more than one film were included (Weight ¿ 1).

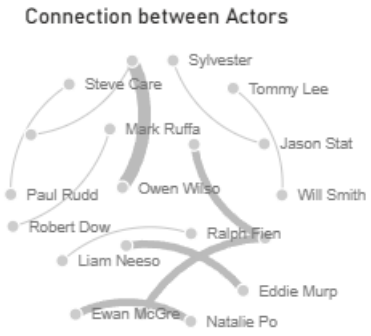


Fig. 2. Network Graph showing the collaboration network between actors.

## V. Conclusion

This project has demonstrated that there is no single formula for success in the cinema industry. The key takeaways are:

1) **Star Power:** Network analysis confirms that specific clusters of actors dominate box office success.
2) **Genre Efficiency:** While Drama is dominant in terms of volume, Horror and Science Fiction are more advantageous in terms of Profitability (ROI).
3) **Director Factor:** It was observed that the director factor is more determinant than the budget on quality (IMDB Score).

The developed interactive dashboard serves as a tool that can assist producers in making more accurate budget and casting decisions based on historical data.

## References


[1] Kaggle, "TMDB 5000 Movie Dataset," [Online]. Available: https://www.kaggle.com/tmdb/tmdb-movie-metadata.

[2] C. Knaflic, *Storytelling with Data: A Data Visualization Guide for Business Professionals*. Wiley, 2015.

[3] Microsoft, "Power BI Documentation," [Online]. Available: https://docs.microsoft.com/en-us/power-bi/.


## Appendix A: Data Preparation Script (Python)

The following Python code was used to transform the raw data into a format suitable for Power BI.

```python
import pandas as pd
import numpy as np
import json
from itertools import combinations
from collections import Counter
import os

# Ensure directories exist
os.makedirs('data/processed', exist_ok=True)

# LOAD DATA
# Make sure your raw files are in 'data/raw/' or adjust the path below
try:
    credits_df = pd.read_csv('data/raw/tmdb_5000_credits.csv')
    movies_df = pd.read_csv('data/raw/tmdb_5000_movies.csv')
except FileNotFoundError:
    print("Error: Could not find raw files. Please check paths 'data/raw/tmdb_5000_credits.csv'")
```

```
    # Fallback for flat structure
    credits_df = pd.read_csv('tmdb_5000_credits.csv')
    movies_df = pd.read_csv('tmdb_5000_movies.csv')

# MERGE DATASETS
movies_df = movies_df.rename(columns={'id': 'movie_id'})
credits_df_clean = credits_df.drop(columns=['title'])
merged_df = movies_df.merge(credits_df_clean, on='movie_id')

# PARSE JSON COLUMNS
def parse_json_col(df, col_name, key_name='name'):
    try:
        return df[col_name].apply(lambda x: [i[key_name] for i in json.loads(x)] if isinstance(x, str) else [])
    except Exception as e:
        return df[col_name]

json_cols = ['genres', 'keywords', 'production_companies', 'production_countries', 'spoken_languages']
for col in json_cols:
    merged_df[col] = parse_json_col(merged_df, col)

# SAVE FILES

output_path = 'data/processed/'

cleaned_movies.to_csv(f'{output_path}cleaned_movies.csv', index=False)
movie_genres.to_csv(f'{output_path}movie_genres.csv', index=False)
movie_countries.to_csv(f'{output_path}movie_countries.csv', index=False)
movie_cast.to_csv(f'{output_path}movie_cast.csv', index=False)
network_df.to_csv(f'{output_path}actor_network.csv', index=False)
corr_melted.to_csv(f'{output_path}correlation_matrix.csv', index=False)

print("SUCCESS: 6 CSV files generated in 'data/processed/'")
```

Listing 1. Data Pre-processing Script