TMDB-Movie Analysis
Project Documentation

Supervised by

Eng. Marwan Mokhtar

Implemented
by

Ahmed Ali Ali
Ahmed Alaa eldin
Abdelrahman Mahmoud saad
Mohamed Moawed
Mohamed Darwesh

Contents

Project Overview 3

Data Requirements 3

Data Sources: 3

Data Transformations: 3

Data Modeling 4

Data Model: 4

Explanation of the Relationships: 4

Calculations and Formulas 5

Custom Calculations: 5

DAX Functions Used: 6

Conclusion 6

Project Overview

The TMDB Data Analysis project leverages the TMDB movie database to analyze various aspects of film performance, focusing on profitability, crew and cast roles, and viewer ratings. The primary goal is to gain deeper insights into movie success factors by integrating, cleaning, and enriching the data, and then analyzing it through a Power BI dashboard. This project includes the extraction and normalization of data, the development of custom measures to calculate weighted ratings and profit margins, and the inclusion of dynamic visualizations for interactive analysis.
The analysis helps to highlight trends related to movie budgets, revenues, and ratings while considering factors like crew roles and cast performance. Additionally, the project examines how release dates and other variables affect a film's popularity and profitability, providing key insights to the entertainment industry. Advanced DAX calculations and Python scripts to update incomplete data enhance the reliability of the analysis.

Data Requirements

Data Sources:

The data for the TMDB Data Analysis project is sourced in two primary ways. The first method utilizes the TMDB API to retrieve live, up-to-date information on movies, including budget, revenue, and cast/crew details. This allows for real-time data acquisition and ensures that the most current movie details are captured. Additionally, a custom Python script was developed to fill in missing budget and revenue information for movies that had incomplete data in the original dataset.

The second method involves using a pre-existing dataset from Kaggle, which provides a rich collection of movie metadata. This dataset includes key information such as movie titles, genres, release dates, and popularity metrics. The combination of these two data acquisition methods—using both the TMDB API and the Kaggle dataset—ensures a comprehensive and well-rounded dataset for analysis. The Kaggle dataset can be accessed at Kaggle TMDB Movie Metadata.

Data Transformations:

1- Normalization of JSON Data

The credits data had JSON-like structures for cast and crew fields. These fields were flattened and extracted into separate columns and tables to make the data easier to work with in Power BI.

2- Data Cleaning

During the data cleaning process, several steps were taken to ensure completeness of the dataset:

- Removed rows with missing or invalid budget or revenue values, such as movies with a budget of less than $200,000 or revenue less than $100,000.

- Identified movies with missing budget or revenue information, where the data existed on the TMDB website but was absent in the dataset. To address this, I obtained an API key from the TMDB website and developed a Python script to retrieve the missing information and update the dataset accordingly.

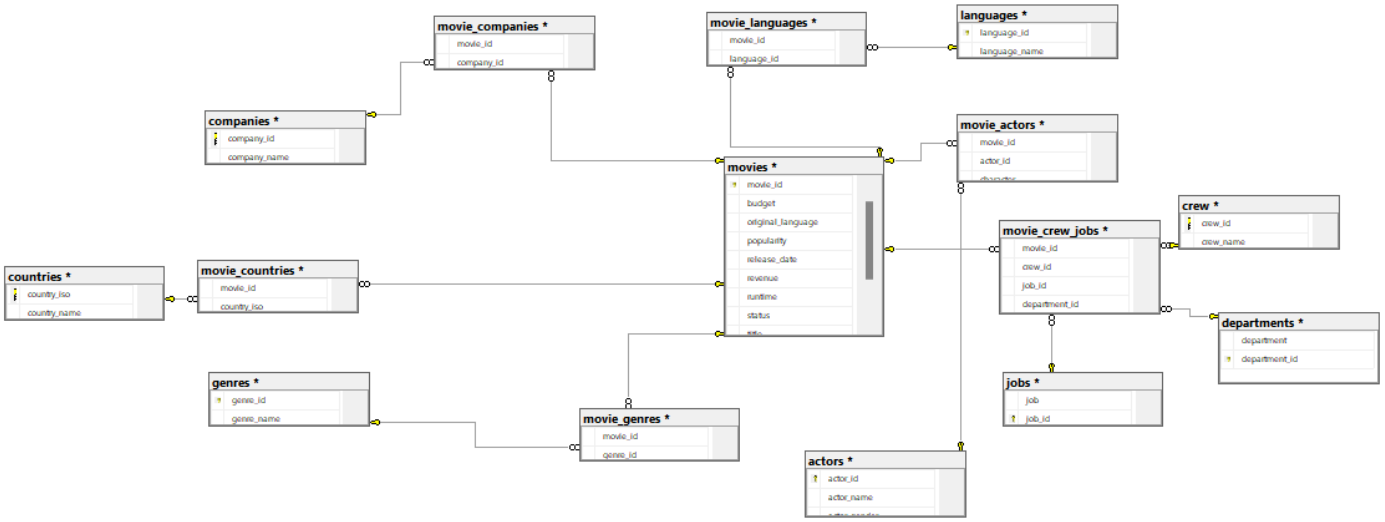
- Filtered out movies with unrealistic or outlier values in terms of revenue or budget.

3- New Columns:

- Profit Margin: Calculated as (revenue - budget) / revenue * 100.
- Weighted Rating: A custom measure was created to calculate the weighted rating of movies based on their vote_average and vote_count. The formula considered both the average rating and the number of votes a movie received.
- Actor Weighted Rating: Calculated by averaging the weighted ratings of all movies an actor has participated in, factoring in how many characters the actor played in each movie.
- Job Count per Movie: Aggregated the number of distinct jobs assigned to crew members for each movie.

Data Modeling

Data Model:



The data model for the TMDB Data Analysis project was structured to provide both flexibility and efficiency in querying and reporting. The movie data was normalized across several key tables, including movies, actors, crew, jobs, movie_actors, and movie_crew_jobs, creating a relational structure that minimizes redundancy and ensures data integrity. Each table was connected via relationships, enabling cross-referencing between movies, cast members, crew, and their respective roles.

Explanation of the Relationships:

- **Movies and Actors (many-to-many)**:
  - A movie can have multiple actors, and an actor can act in multiple movies.
  - This many-to-many relationship is handled via the movie_actors bridge table, which connects movie_id from the movies table to actor_id from the actors table.

- **Movies and Crew (many-to-many):**
  - A movie can have multiple crew members, and a crew member can work on multiple movies.
  - This many-to-many relationship is managed via the movie_crew_jobs bridge table, which links movie_id from the movies table to crew_id from the crew table, with additional details from the jobs table specifying the crew's role.
- **Crew and Jobs (many-to-one):**
  - Each crew member has one specific job in a movie, but each job can be associated with multiple crew members.
  - This is a many-to-one relationship between jobs (many) and crew (one).
- **Crew and Departments (many-to-one):**
  - Each crew member belongs to one department, but a department can have many crew members.
  - This is a many-to-one relationship between departments (many) and crew (one).
- **Movies and Genres (many-to-many):**
  - A movie can belong to multiple genres, and a genre can apply to multiple movies.
  - This many-to-many relationship is handled by the movie_genres table, which links movie_id to genre_id.
- **Movies and Companies (many-to-many):**
  - A movie can have multiple production companies, and a company can produce multiple movies..
  - The movie_companies table handles this many-to-many relationship, linking movie_id to company_id.
- **Movies and Languages (many-to-one):**
  - Each movie is primarily associated with one language, but a language can be associated with many movies.
  - This is a many-to-one relationship, where the movies table has a foreign key to the languages table.
- **Movies and Countries (many-to-many):**
  - A movie can be associated with multiple countries (filmed or released), and a country can have multiple movies.
  - This many-to-many relationship is managed via the movie_countries table, linking movie_id to country_iso.

Calculations and Formulas

Custom Calculations:

1- Weighted Ratings:

Formula: Weighted Rating = $(R \times v) + (C \times m) / (v + m)$

Where:

R = Average rating of the movie
v = Number of votes
C = Mean vote across all movies

m = Minimum number of votes required

2- Profit Margin:

Formula:
Profit Margin = (Revenue - Budget) / Revenue

3- Year-to-Year Growth:

Formula:
Year-to-Year Growth = (Total Revenue in Year n - Total Revenue in Year (n-1)) / Total Revenue in Year (n-1) × 100

DAX Functions Used:

CALCULATE(): Modifies filter context for calculations.
SUM(): Sums up revenue and budget figures.
YEAR(): Extracts year from release date.
PREVIOUSYEAR(): Calculates previous year's metrics for growth.
DIVIDE(): Safely performs division.

Conclusion

The TMDB Data Analysis project successfully demonstrates the integration of comprehensive data from multiple sources to generate valuable insights into the film industry. Through meticulous data cleaning and normalization, we enhanced the accuracy of budget and revenue figures by leveraging the TMDB API to fill in missing information.

The data model established clear relationships between various entities, allowing for detailed analysis of movie performance, crew contributions, and audience engagement metrics. Dynamic visualizations, including charts and graphs, effectively communicate the findings and highlight key trends within the dataset.

Custom calculations, such as year-to-year growth and average ratings by job roles, provide a deeper understanding of the factors influencing movie success. User interactivity features enhance the user experience by allowing stakeholders to filter and explore the data according to their specific interests.

Overall, this project not only meets its objective of providing a robust analysis of TMDB data but also serves as a foundation for future research and exploration in the realm of film analytics. As the film industry continues to evolve, the insights gained from this analysis will contribute to informed decision-making and strategic planning.