ckucewicz / movie_analysis_project

Code Issues Pull requests Actions Projects Wiki Security Insights Se

1 star 0 forks 1 watching 1 Branch 0 Tags Activity

Public repository

1 Branch 0 Tags Go to file t Go to file Add file Code

ckucewicz Finalized notebook c23e1e0 · 2 minutes ago

Visualizations	Delete Visualizations/roi_by_profe...	13 hours ago
presentationDeliverables	Finalized notebook	2 minutes ago
zippedData	Adding data files	2 weeks ago
.gitignore	Initial Commit	2 weeks ago
Notebook.ipynb	Finalized notebook	2 minutes ago
README.md	Update README.md	12 hours ago
miscellaneous_notebook.ipy...	Data Preparation update	3 days ago

README

Movie Analysis Project

Author: Chris Kucewicz

Business Understanding

Background

My company is looking to get into movie creation using their newly created movie studio.

Business Goals

The primary focus of this data science project is to analyze and assess which features of a movie are the most cost efficient. The movie's return on investment will be used to measure cost efficiency in order to make an informed decision regarding what features of movie creation my company should invest in.

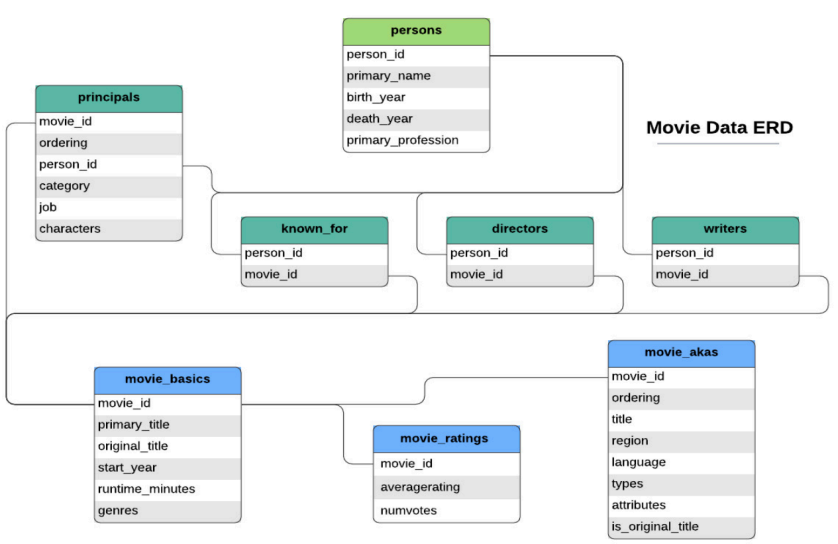
Business Success Criteria

The success of this project will be measured by providing three well-supported recommendations on the most cost efficient movie features (actors, directots, genre, marketing cost, movie rating (G, PG, PG-13, etc.)) to invest in. For this project, the most "cost efficient" features are measured by their return on investment which is defined as 100% times the total revenue divided by the initial investment of the film.

Data Understanding

Data on movies is collected by a variety of different sources. For this project, I used data from the following sources:

- The Numbers' `budgets` dataset
  - This dataset includes 6 features and 5,782 observations. Each entry in the dataset represents a different movie. For each entry, information is included about the movie's release data, production budget, domestic gross box office, and worldwide gross box office.
- IMDB's film database
  - This database includes 8 tables. Its entity-related diagram (ERD) is shown below.
  - From this database, I used the following tables: `movie_basics` , `persons` , and `principals`
  - `movie_basics` includes **6 features** with **146,144 observations**. Each entry in this dataset represents a different movie, where `movie_id` is its unique ID (primary key). Additional information is included about each movie such as `original_title` , `runtime_minutes` , and `genres` .
  - The `persons` table includes **5 features** with **606,648 entries**. Each entry represents a person who took part in a movie, where each person has a unique identifier (`person_id`). This table also includes information about each person such as their `primary_name` , `birth_year` , and `primary_professions`
  - The `principals` table contains **6 features** and **1,028,186** entries, where each entry represents a person who worked in a movie. This table contains two foreign keys (`movie_id` and `person_id`). Additional information includes the `character` the person played and their role on the film (`category`).



Data Preparation

During the data preparation stage, I focused on cleaning four datasets: `budgets`, `movie_basics`, `persons`, and `principals`.

The data cleaning process began by converting columns to their appropriate Python data types. To facilitate this, I created a function called `get_info()` to check each table's `.info()`, which allowed me to verify the data types and identify `NaN` values in each feature. For instance, I converted the `production_budget`, `domestic_gross`, and `worldwide_gross` columns in the budgets dataframe from objects to floats, as they were originally stored incorrectly.

I filtered out outliers and included only movies released before 2024. Irrelevant columns, such as `id` in `budgets`, were removed. `NaN` values were addressed by filtering out rows in movie_basics where both `runtime_minutes` and `genres` were `NaN`.

In the `principals` table, I removed the `job`, `characters`, and `ordering` columns due to redundancy or irrelevance, and in the `persons` table, I removed `birth_year`, `death_year`, and `primary_profession` for the same reasons.

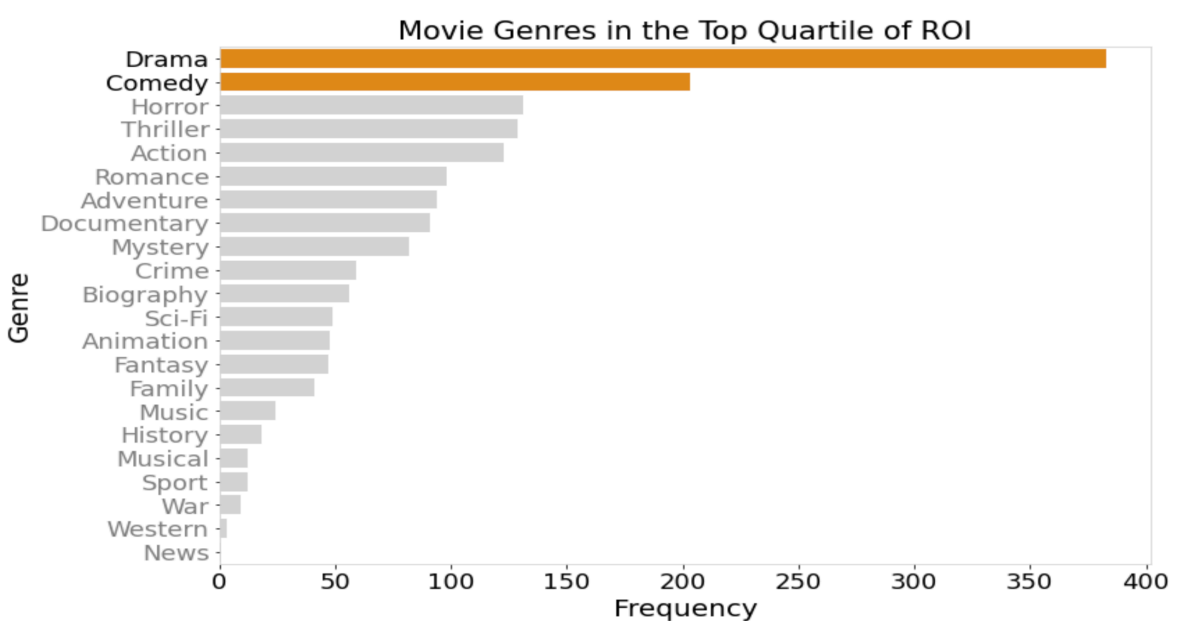
After cleaning and processing all four dataframes, I used filtering and join operations to create three new dataframes: `top_people_budgets`, `top_roi_movie_basics`, and `budgets_no_outliers`. These dataframes contain information about movies in the top 25% of ROIs, which I used for further analysis.

- `top_people_budgets` includes information about individuals who worked on movies in the top 25% of ROIs, such as their names, movie titles, and job professions.
- `top_roi_movie_basics` contains details about movies with the highest 25% ROI, including titles, genres, runtimes, and budget information.
- `budgets_no_outliers` provides budget information about all movies from the cleaned dataset.

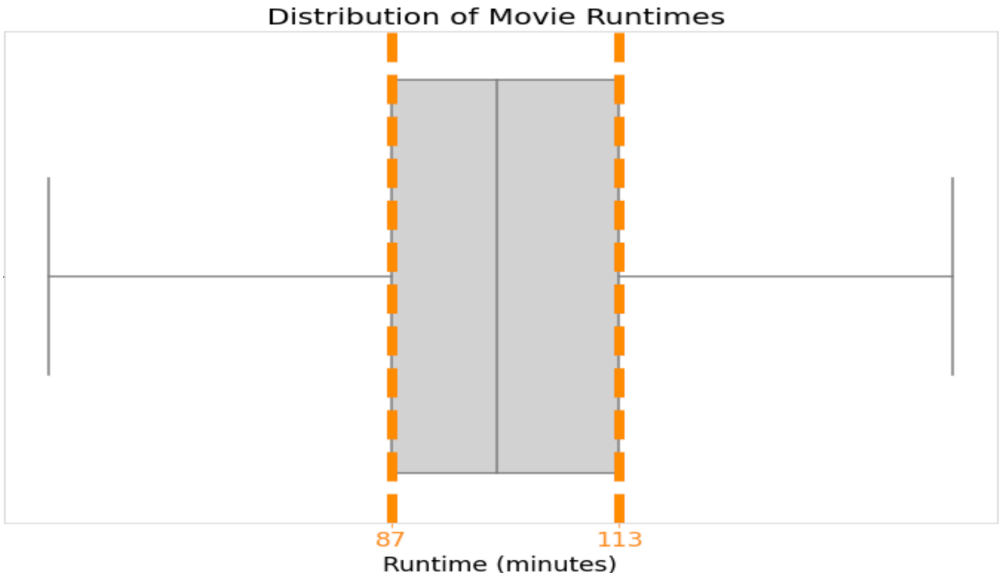
Exploratory Data Analysis

The following are findings from this analysis:

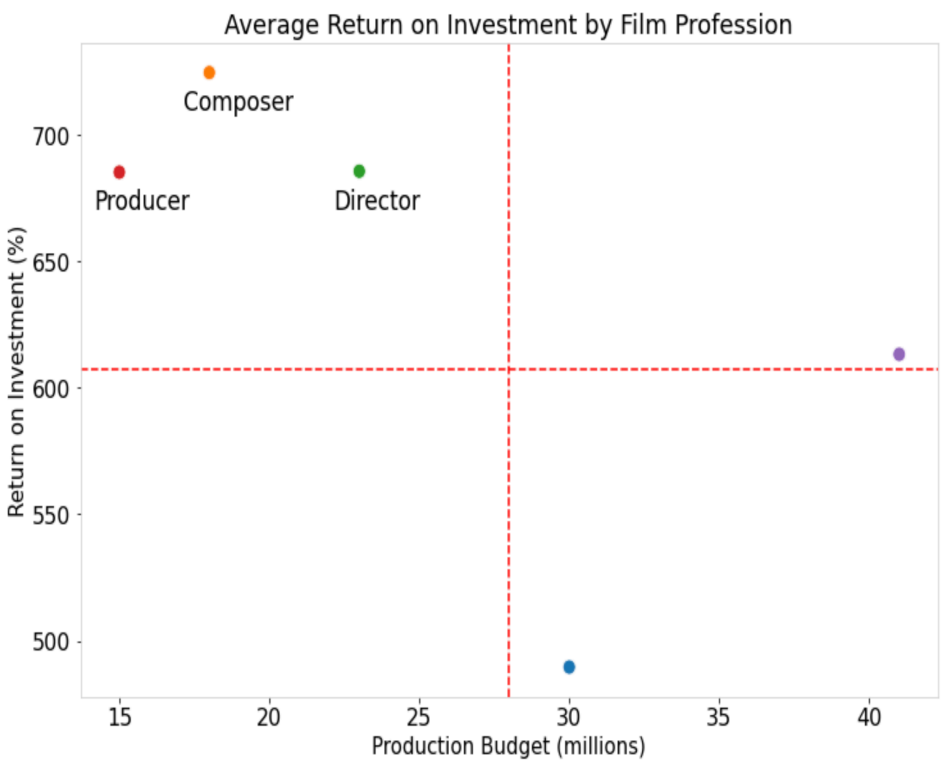
- Of the 5,636 movies with functional budget data, **37%** did ***NOT*** achieve a positive ROI.
- The typical movie had an estimated **16 million dollar production budget**, generated an estimated **26 million dollars in worldwide gross revenue**, and produced an estimated **66% return on investment**.



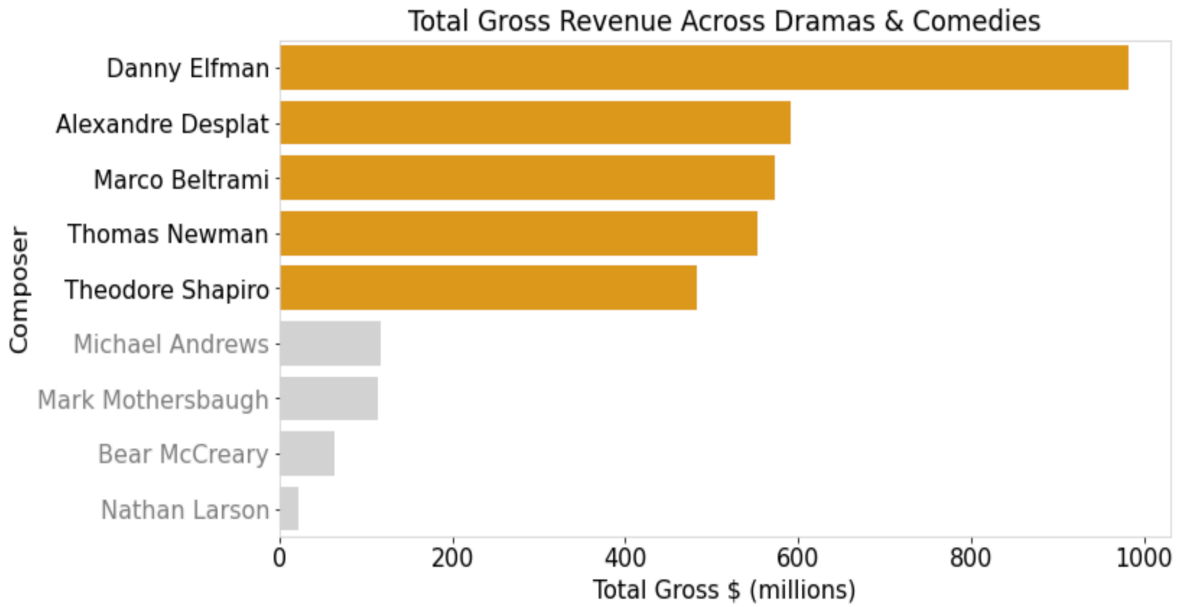
- **Dramas** and **comedies** were the two **most common genres** for the movies in the top 25% of ROI.



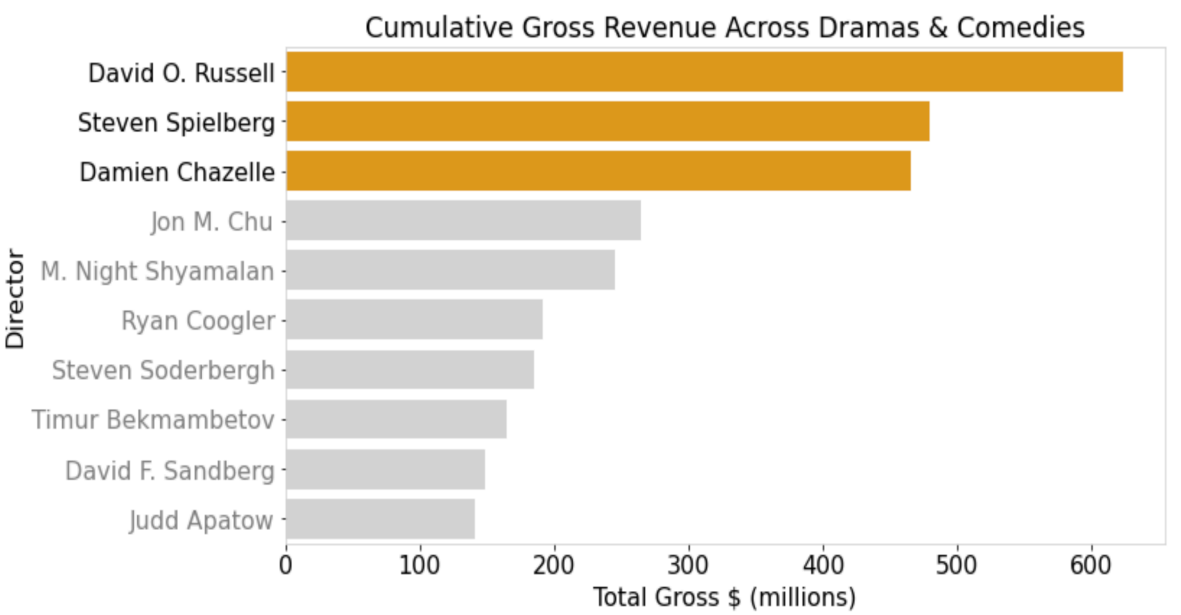
- The middle 50% of the movies with the highest ROI had **runtimes between 87 and 113 minutes**.



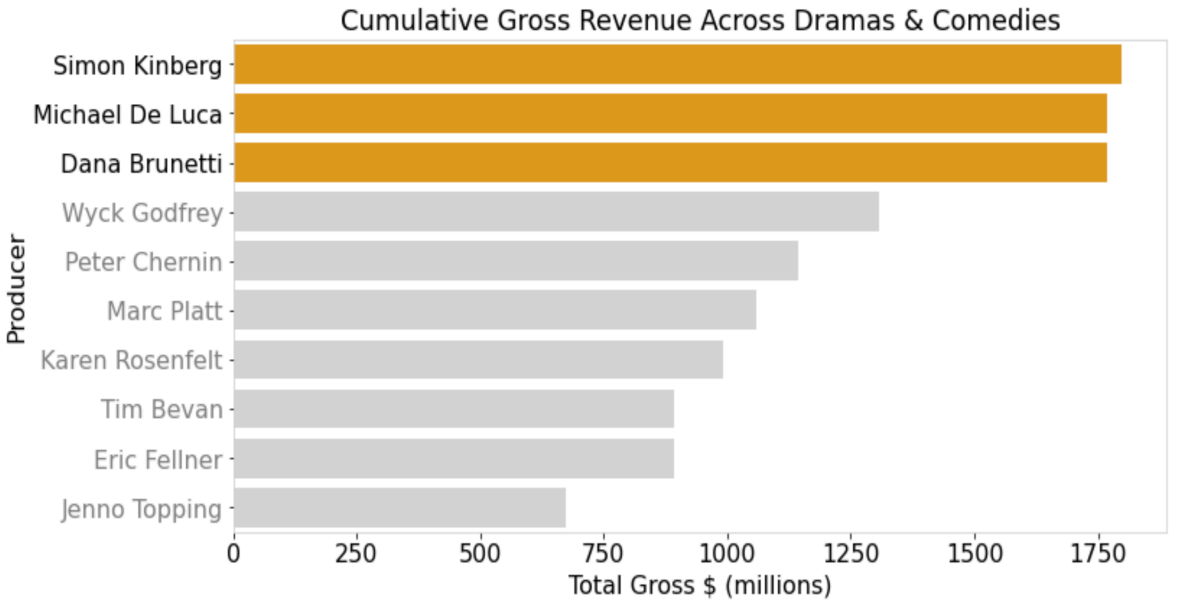
- The three film **professions that generate the highest ROI** for dramas and comedies are: **composers**, **directors**, & **producers**.



- The 5 highest grossing drama & comedy ***composers*** are: Danny Elfman, Alexandre Desplat, Marco Beltrami, Thomas Newman, Theodore Shapiro



- The 3 highest grossing drama & comedy ***directors*** are: David O. Russell, Steven Spielberg, Damien Chazelle



- The 3 highest grossing drama & comedy ***producers*** are: Simon Kinberg, Michael De Luca, Dana Brunetti

Conclusion

Limitations

While the datasets and tables provided a variety of data, there was a notable limitation in the availability of budget information for movies. This led to a significant discrepancy between the number of entries in the budgets table (~5,000) and other tables within the IMDB database, one of which contained over 1 million entries. Consequently, the analysis was restricted by the limited amount of budget data, reducing the number of movies that could be analyzed. A more comprehensive dataset that includes budget information for a wider range of movies would enable a more thorough analysis and yield more informed recommendations about the factors that influence a movie's return on investment.

Recommendations

This analysis leads to three recommendations for movie creation:

1. Focus on creating movies within the ***drama or comedy genres***.

> - Over **one-third** of the movies with the highest ROI were classified as dramas and/or comedies.

2. Create movies with ***runtimes between 87 and 113 minutes***.

> - Half of all movies with the highest ROI had runtimes between 87 and 113 minutes.

3. Focus on hiring high-quality ***composers, directors, and producers*** who specialize in comedy & drama, as these three film professions had the highest ROI out of all.

> - **Recommended drama & comedy composers** (top five highest cumulative grossing): Danny Elfman, Alexandre Desplat, Marco Beltrami, Thomas Newman, Theodore Shapiro

> - **Recommended drama & comedy directors** (top three highest cumulative grossing): David O. Russell, Steven Spielberg, Damien Chazelle

> - **Recommended drama & comedy producers** (top three highest cumulative grossing): Simon Kinberg, Michael De Luca, Dana Brunetti

Next Steps

With these recommendations in mind, I am interested the following next steps:

- Gather more budget data on a wider number of movies
- Perform regression analysis to answer the question: Which factors most strongly correlate with a movie's ROI?

Additional Information

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Languages

- **Jupyter Notebook** 100.0%