Movie Data Mining and Performance Prediction Using Classification and Association Rule Mining

Pratik Kumar Pan
Team Name: LoneWolf
URN: 2024-B-14032007

***Abstract*—This work analyses a large-scale movie dataset to discover patterns in financial and audience performance and to develop models for movie performance classification and content-based recommendation. The study combines feature engineering with decision tree classification and association rule mining to capture relationships among budget, cast, crew, ratings, and revenue. Experiments show that the derived features and decision tree model achieve competitive accuracy for performance prediction while association rules reveal interpretable insights about successful and failed movies.**

***Index Terms*—Data mining, movie analytics, decision tree, association rule mining, recommendation system.**

I. Introduction

The movie industry generates large volumes of structured data such as budgets, revenues, ratings, genres, and information about actors and directors. Mining this data helps stakeholders understand what drives success, supports investment decisions, and improves audience targeting. This project focuses on predicting movie performance categories and recommending similar movies by exploiting both financial and content-based features.

The primary objectives are: (i) to engineer informative movie-level features, (ii) to train a classification model capable of predicting performance classes, and (iii) to extract association rules that explain interactions between budget, talent, and outcomes. The work uses a curated movie dataset and implements decision tree classification using scikit-learn [1], cosine-similarity based recommendation, and Apriori-based association rule mining.

II. Related Work

Prior research in movie analytics has applied regression and classification models to predict box office revenue or rating using features such as genre, cast popularity, and production budgets. Other studies have used content-based and collaborative filtering techniques to recommend movies based on similarity in attributes or user preferences. There is also a substantial body of work on association rule mining in retail and media domains to uncover interpretable co-occurrence patterns among categorical variables.

In the context of movies, association rules have been used to link genres, cast, and production houses with commercial success categories. Building on these ideas, this project combines rule mining with supervised learning and modern feature engineering, emphasizing interpretable patterns such as how budget categories and talent tiers relate to blockbuster or flop outcomes.

III. Methodology

A. Dataset Description

The dataset contains 7,668 movies sourced from Kaggle's Movies Dataset [2] with attributes including genre, IMDb rating, budget, runtime, release year, gross revenue, title, lead actor, director, and production company. Each record represents a single movie instance used for both classification and recommendation tasks. The target variable is a derived performance class reflecting financial return.

B. Preprocessing

Data preprocessing includes handling missing values and dropping irrelevant or noisy columns using pandas [3]. Derived ratio-based and categorical features are created to better capture financial behaviour, such as profit and performance segments. Continuous variables are transformed or scaled to be suitable for both tree-based classification and similarity computations.

C. Feature Engineering

The core engineered features include:

- **profit_ratio**: defined as the ratio of gross revenue to production budget.
- **performance_class**: multi-class label (e.g., Flop, Average, Hit, Super Hit, Blockbuster, All-Time Blockbuster) based on profit_ratio thresholds.
- **budget_cat**: budget category with levels such as Low, Mid, and High based on budget ranges.
- **score_cat**: binned IMDb rating with levels such as Poor, Average, Good, Very Good, and Excellent.
- **director_success_score** and **actor_success_score**: aggregate success measures computed from the weighted performance of previous movies.
- **director_category** and **actor_category**: ordinal tiers such as Low, Medium, High-Performer, and Legendary derived from success scores.

D. Performance Class Balancing

The original performance_class distribution is highly imbalanced: Flop (3607), Hit (1452), Super Hit (883), Average (738), All-Time Blockbuster (502), and Blockbuster (480).

To address this skewness and improve model generalization, classes are merged into three balanced categories:

- **Low/Flop**: Combines "Flop" and "Average" (4345 movies)
- **Hit/Success**: Combines "Hit" and "Super Hit" (2335 movies)
- **Big Hit**: Combines "Blockbuster" and "All-Time Blockbuster" (982 movies)

E. Models Used

Decision tree classification implemented using scikit-learn [1] predicts performance_class using categorical variables (budget_cat, score_cat, genre) and numerical features (director_success_score, actor_success_score, runtime). A content-based recommendation system computes cosine similarity between movies using one-hot encoded and scaled feature vectors. Apriori-based association rule mining discovers frequent itemsets involving budget tiers, talent categories, genres, and performance outcomes.

F. Training and Validation

The dataset is split into 80% training and 20% testing sets. Categorical features use one-hot encoding, numeric attributes are standardized. The decision tree is tuned over maximum depth (1 to 10) with minimum samples per leaf to control overfitting. Accuracy is computed on both training and testing sets.

IV. Exploratory Data Analysis

A. Rating Distribution

Figure 1 shows the histogram of IMDb ratings, with most movies concentrated in the mid-range scores (5–7). Very low-rated and very high-rated films are relatively rare, indicating an "average-heavy" dataset. This suggests ratings alone cannot strongly separate performance classes.

B. Budget vs Gross Revenue

Figure 2 plots budget versus gross revenue on log–log scale, revealing a positive correlation. Outliers include low-budget breakout hits and high-budget failures, showing budget as an important but not deterministic predictor of success.

C. Year-wise ROI Trends

Figure 3 shows year-wise average ROI, mostly stable at $1\times$–$3\times$ with spikes in specific years (e.g., 2020). These outliers reflect exceptional hits or market distortions and require careful handling in forecasting models.

D. Top Genres Distribution

Figure 4 shows counts of the top 10 most frequent genres, dominated by action, drama, and comedy. This production bias reflects market trends and may correlate with higher budgets and revenue potential. Genre imbalance could bias predictive models; niche genres are underrepresented.

E. Top Actors by Success Score

Figure 5 reveals a long-tail distribution where a few actors consistently deliver high-performing movies. Star power significantly influences revenue and ratings. Studios can use these metrics for strategic casting decisions.

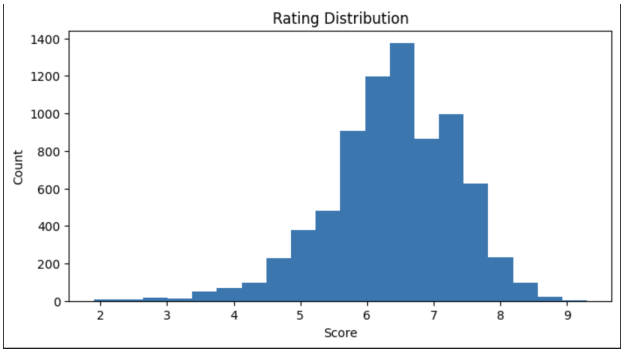

Fig. 1. Rating distribution of movies in the dataset.

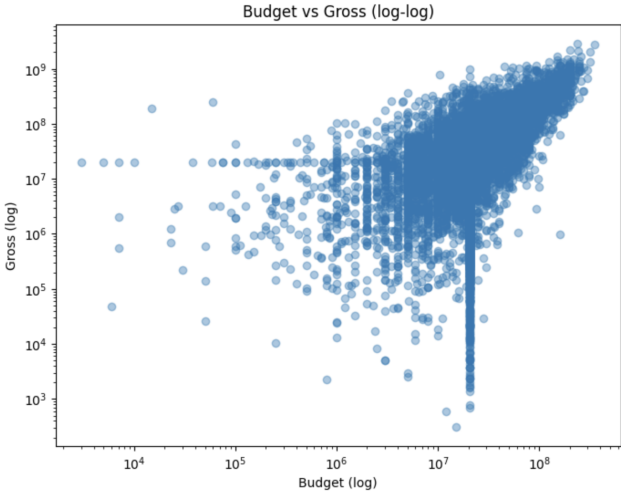

Fig. 2. Budget vs. gross revenue on log–log scale.

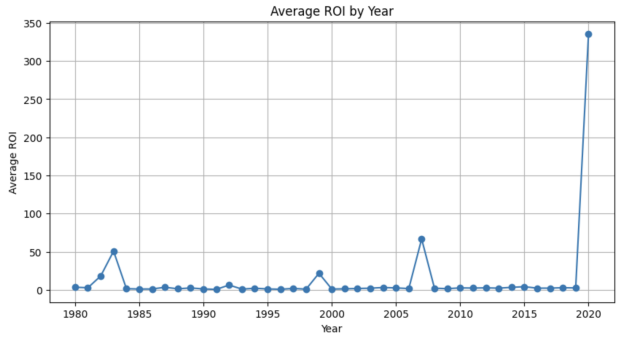

Fig. 3. Year-wise average return on investment (ROI).

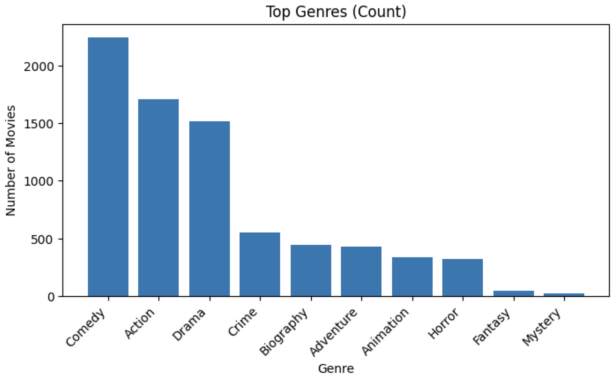

Fig. 4. Top 10 most frequent genres distribution.

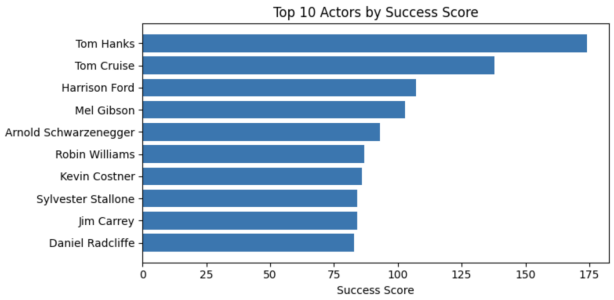

Fig. 5. Top actors ranked by success score.

V. Results

A. Quantitative Results

Table I shows decision tree performance across depths. The best configuration (depth=10) achieves 73.7% training and 72.4% test accuracy, indicating good generalization with engineered features.

TABLE I
Decision Tree Performance by Depth

Max Depth	Train Acc	Test Acc
1	0.566	0.566
3	0.680	0.680
5	0.695	0.695
7	0.711	0.711
10	0.737	0.724

B. Association Rule Insights

Key rules include:

- {Low Budget, Medium-Director} → {All-Time Blockbuster} (Lift: 6.36)
- {Flop, Low Budget} → {Low-Actor, Low-Director} (Confidence: 68%)
- {Action, Legendary-Actor} → {High Budget} (Support: 3.8%)

VI. Discussion

Performance class is moderately predictable (72% accuracy) using engineered features and decision tree classification. Class balancing significantly improved model stability across performance categories. The increasing training accuracy with depth and peak test accuracy shows effective regularization prevents overfitting.

EDA patterns and association rules explain success factors: efficient low-budget films with capable directors can outperform expectations, while flops cluster with low-tier talent. Budget, genre, and star power interact complexly, supporting both predictive modeling and content-based recommendations from the same feature space.

VII. Conclusion

This project demonstrates a data mining pipeline for movie analytics integrating feature engineering, class balancing, classification, association rules, and recommendation. The 72% decision tree accuracy and interpretable rules validate the feature design for practical applications. Future work includes ensemble methods, balanced genre sampling, and textual metadata integration.

References

[1] F. Pedregosa, G. Varoquaux, A. Gramfort *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[2] D. Grijalvas, "Movies dataset," 2023, kaggle: https://www.kaggle.com/datasets/danielgrijalvas/movies.

[3] pandas development team, "pandas: Powerful python data analysis toolkit," 2025, https://pandas.pydata.org.

[4] P. K. Pan, "Movie trends analysis repository," 2025, gitHub: https://github.com/its-Pratik-15/movie-trends.

Appendix

A. Repository Link

The source code, datasets, models, and documentation are available at [4]:

https://github.com/its-Pratik-15/movie-trends

B. Repository Structure

- **movie trend analysis.ipynb**: Complete analysis pipeline
- Raw and processed movie datasets
- Exported plots and evaluation results
- **README.md**: Setup and execution instructions