Predicting Movie Success through IMDb Ratings and Box Office Earnings

A Data Mining Approach

Nathan Harris
College of Engineering and Applied Science
University of Colorado Boulder
Boulder CO, USA
nharris@colorado.edu

Mayumi Shimobe
College of Engineering and Applied Science
University of Colorado Boulder
Boulder CO, USA
mayumi.shimobe@colorado.edu

1. Problem Statement and Motivation

Our project aims to analyze trends and patterns in the film industry by leveraging comprehensive IMDb datasets to predict movie success—focusing on IMDb ratings as a proxy for success and incorporating box office earnings where possible. By applying data mining techniques to publicly available datasets, we hope to uncover hidden relationships among movie attributes such as runtime, release year, genre, and cast, and ultimately develop predictive models that can aid decision-makers in the industry.

2. Motivation and Literature Survey

Recommendation engines appear to be the most popular use for Movies and TV data sets. Eighty percent of Netflix streaming time originates from a user clicking on a recommended title [3]. These recommendation engines typically work as content-based, collaborative recommendation, or a hybrid approach [5, 8]. The content-based approach focuses on similarities between the attributes of a movie or television show whereas collaborative approach focuses on the similarities between the behavior of the user and a class of users [7]. IMDb utilizes its own recommendation engine which can be observed when scrolling to the bottom of a given title under "More like this" which leads us to believe they are focused on a content-based approach [4]. Behavior can be analyzed as explicit or implicit, explicit meaning when a user gives a positive rating to a title while implicit is when they binge watch it in one weekend [3, 9]. As far as predicting the success of film at the box office or with average reviews, it seems that there is no consensus on the indicators for predicting a film's success. Some indicators have been combinations of characters, plot complexity, star power, budget, or "buzz," social media chatter on a particular film [1, 2]. While other indicators are external from a movie's data set, such as search engine results, social media content, and ratings on expert websites [10]. Evaluating previous movie titles is often used to support these more external data indicators.

For our project we will be focused on a content-based approach and utilize ratings as a measurement of success. Utilizing a collaborative approach would involve obtaining demographic user data in addition to implicit/explicit behavior. While we found one data set that includes this, it is far below the threshold for minimum data objects.

3. Proposed Work

3.1 Data Collection and Integration

We will use datasets from The Internet Movie Database, IMDb, and Box Office Earnings Data. Details are depicted in Section 4.

3.2 Data Preprocessing and Cleaning

In this section, we describe the steps we will take to transform the raw IMDb and box office data into a clean, integrated dataset suitable for predictive modeling.

Data loading: We will use MySQL's bulk-loading method (`LOAD DATA LOCAL INFILE`) to import the IMDb datasets (e.g., title.basics.tsv and title.ratings.tsv) and the box office data from Kaggle.

Any issues related to file access will be resolved by verifying working directory settings and ensuring the `local_infile` parameter is enabled on both the client and server sides.

Data cleaning: We will handle missing values by replacing IMDb's special marker (\N) with SQL NULLs, removing duplicate records, and standardizing data types, ensuring that numeric fields such as runtime, startYear, averageRating, and numVotes are stored as integers or decimals. Outlier detection will be performed using statistical methods introduced in the class to identify extreme values in IMDb ratings and box office earnings. Where necessary, outliers will be either transformed or excluded from the training set: removing outliers where the rating is only provided by a small number of reviews, i.e. two people gave 5 stars versus two thousand people who gave 4.5 stars Socioeconomic factor like covid affecting movie theater attendance around 2020 will also be considered for outliers.

Data Integration and Transformation: The core IMDb tables—title_basics and title_ratings, along with any related attributes from chosen datasets—will be joined on the primary key (tconst). We will then integrate box office data from Kaggle to enrich our analysis. Categorical attributes such as genres will be preprocessed by extracting the primary genre and converting multi-valued strings into dummy variables, thereby facilitating their use in regression models. This step will also include a preliminary analysis of multicollinearity to guide subsequent feature selection.

3.3 Predictive Modeling

With the preprocessed dataset, we will pursue a dual modeling strategy to both predict continuous outcomes (IMDb ratings and box office earnings) and classify movies as successful or not. We will consider two possible modeling approaches.

Logistic Regression: To classify movies as "successful" (e.g., an IMDb rating above a threshold, such as 7.0 or top 20% of the total vote), we will build a logistic regression model. Evaluation metrics may include accuracy and confusion matrices. This classification approach will complement the continuous prediction by providing actionable insights into categorical success.

Additional Machine Learning Algorithms: Depending on preliminary results, we may also experiment with non-linear models such as Random Forests or Decision Tree [11] to capture complex interactions among variables. These methods will help us validate whether advanced algorithms offer improvements over traditional regression techniques.

Our evaluation will employ k-fold cross-validation to ensure robust performance estimates [11]. We will generate scatter plots (for regression) and residual and diagnostic plots to visualize and interpret model performance. Finally, model outcomes will be compared against external benchmarks (industry reports and expert analyses) to validate our findings described in section 5 Evaluation Methods.

3.4 Distinction from Previous Work

Various research has been done with the content-based approach in the past to predict success in the movie industry. Nithin V.R. et al [12], for instance, explores logistic regression, SVM regression, and linear regression on IMDb and Rotten Tomatoes to predict both gross revenue and IMDb ratings. They discussed handling duplicates and missing data, yet outliers were not handling. In this study we intend to work on outliers to see the prediction is improved. We chose IMDb and Box Office Earnings to have the data availability, completeness, and consistency among data. In addition, we would make comparison with results from Kristianto et al of Random Forest and XGBoost [13] to see further investigation of machine learning is worthwhile.

4. Data Set Details

Data Source and URL:
IMDb Datasets: Publicly available at https://datasets.imdbws.com/
Box Office Data: Obtained from https://www.kaggle.com/datasets/harios/box-office-data-1984-to-2024-from-boxofficemojo.
Description and Dimensions:
From IMDb Datasets: two data files, title.basics and title.ratings, are considered. Preliminary cleaning process of other files listed on Part 1, such as

title.principles, implies too many outliers and empty datasets to have fair number of datasets for predictive analysis.

title.basics.tsv.gz: Approximately 25.3 million records with 9 attributes:

- tconst: Nominal (string) – Unique identifier.
- titleType: Nominal (string).
- primaryTitle / originalTitle: Nominal (string).
- isAdult: Nominal (binary integer).
- startYear / endYear: Interval (integer).
- runtimeMinutes: Ratio (integer).
- genres: Nominal (string, multi-valued).

title.ratings.tsv.gz: Over 15 millions records with 3 attributes:

- tconst: Nominal (string). Ordinal (integer)
- averageRating: Ratio (decimal).
- numVotes: Ratio (integer).

From Box Office Earnings Dataset:
Year: Date (Date)
Title: Nominal (string)
Gross: Ratio (Integer)

5. Evaluation Methods

Once study of the predictive models is conducted, the result would be evaluated by its accuracy and confusion matrices. Any visual plots (predicted vs. actual values) would be provided for better analysis. In addition, the results will be compared with literature. Nithin VR et al found accuracy of roughly 51% for linear regression, 42% for logistic regression, and 39% for SVM [12]. We will investigate the effect of outlier handling as well as necessity for the machine learning algorithms by comparing with accuracy of ~84% reported by Kristianto et al using Random Forest [13].

6. Tools

Our project will employ the following tools:

Programming Languages: Python (with Pandas, NumPy, scikit-learn, and StatsModels for analysis and modeling).

Database: MySQL on AWS RDS for data storage, integration, and querying.

Visualization: Matplotlib and Tableau for generating both static and interactive visualizations.

Version Control: GitHub for collaborative coding and version management.

7. Milestones

We have set up AWS RDS instance, created database/tables, loaded IMDb and Box Office datasets and conducted initial data cleaning and preprocessing for some of datasets.

Original plan: by Mar 7, 2025, we perform exploratory data analysis using SQL and Python and begin feature engineering, including encoding categorical variables and assessing multicollinearity.

Status: this has been completed for most of IMDb rating datasets and results are depicted in Section 8. Non-numerical features like genres are to be considered

Original plan: by March 14, 2025, we develop and test predictive models (linear regression, logistic regression, and possibly additional machine learning algorithms such as Random Forests). Evaluate model performance and refine feature selection based on analysis and correlation matrices.

Status: the linear and logistic regressions have been conducted and analyzed. Current results so far have been shown in Section 8. To improve prediction, we plan to work on additional classification modeling such as Random Forests and XGBoost by April 6, 2025. The additional classification modelings have been suggested in the feedback from Part 2 by professor.

By April 11, 2025: Finalize visualizations and interpretability analyses. Compare model results with external benchmarks and expert analyses.

By April 18, 2025: Prepare the final report and project presentation, incorporating feedback and ensuring that all aspects of the evaluation framework are clearly documented. We have set our weekly Agile meeting every Thursday to meet deadlines for each project part and final submission.

8. Results So Far

8.1 Exploratory Data Analysis

Our analysis began by integrating the IMDb datasets into a single view (named imdb_integrated) that joins the title_basics and title_ratings tables on the movie identifier (tconst). This integrated view allowed us to

work with key attributes such as primaryTitle, titleType, isAdult, startYear, runtimeMinutes, genres, averageRating, and numVotes.

After loading the integrated dataset into a pandas DataFrame, we performed a descriptive statistical analysis using the describe() method. The summary statistics are shown in Table 1. These statistics revealed that:

- *Only about 2.6% of titles are flagged as adult content, so the isAdult attribute is not considered a key predictor.*
- The startYear values have a median around 1997, suggesting that the dataset spans a long period and that very old or future-dated entries (e.g., the year 2025) might need further review.
- The runtimeMinutes attribute displays a wide range, with many titles being relatively short (possibly TV episodes or short films) but with a few entries having extremely high values (up to 35,791 minutes), which calls for additional investigation.
- The numVotes attribute is highly skewed; with a mean of about 1,450 votes, with a high standard deviation (~13253). The distribution is highly skewed (long tail) with 25% of movies having ≤15 votes, 50% having ≤36 votes, and 75% having ≤135 votes.
- The averageRating is bounded between 1 and 10 and exhibits a roughly unimodal distribution that peaks around 6.5–7. A computed skewness (using SciPy's skew function) confirms a long-tail distribution—indicating that while most movies receive moderate votes, a small number of highly popular titles receive extremely high vote counts.

	isAdult	startYear	runtimeMinutes	numVotes
count	632363.000000	632285.000000	484862.000000	6.323630e+05
mean	0.025851	1990.463242	60.493019	1.449651e+03
std	0.158690	23.719229	73.295754	2.325252e+04
min	0.000000	1878.000000	0.000000	5.000000e+00
25%	0.000000	1978.000000	26.000000	1.500000e+01
50%	0.000000	1997.000000	53.000000	3.600000e+01
75%	0.000000	2006.000000	90.000000	1.350000e+02
max	1.000000	2025.000000	35791.000000	3.000635e+06

Table 1. Result of Statistical Analysis (Descriptive Statistics): This table includes metrics such as count, mean, standard deviation, min, 25th percentile, median, 75th percentile, and max for the key numeric attributes.

Figure 1 presents a histogram of IMDb average ratings. The plot clearly shows that most ratings are concentrated in the mid-range (approximately between 5 and 8), with relatively fewer movies rated at the extreme ends of the scale. This bounded distribution suggests that, unlike other numeric attributes, the ratings do not suffer from extreme outliers.

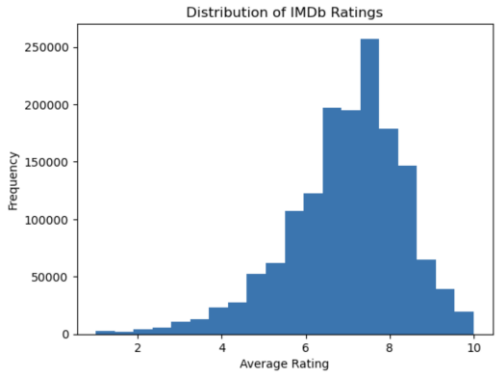

Figure 1. Distribution of IMDb Ratings.

Next, we conducted a multicollinearity analysis by encoding the categorical genres attribute using one-hot encoding. We then computed and visualized the correlation matrix for selected numeric attributes (e.g., startYear, runtimeMinutes, numVotes, averageRating) along with the most frequently occurring genre columns. Figure 2 shows the resulting heatmap.

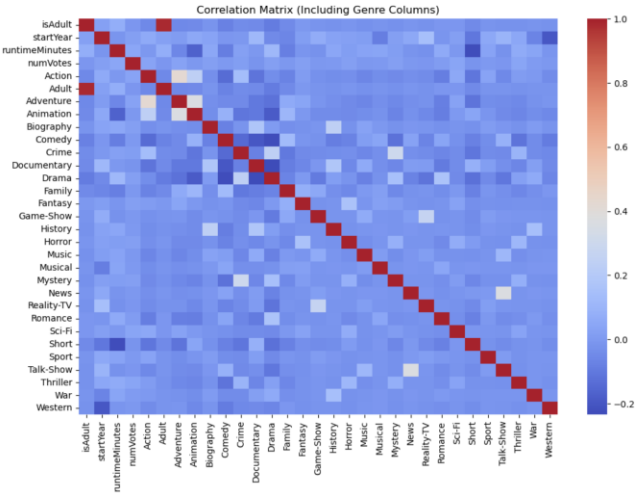

Figure 2. Heatmap for multicollinearity analysis of attributes

The heatmap indicates that most pairwise correlations are close to zero—implying that no strong linear relationships exist among individual numeric features or between these features and the genre indicators. This lack of strong correlations suggests that while simple pairwise linear dependencies may not be evident, more complex interactions (such as nonlinear or interaction effects) could still exist and be important for predictive modeling.

8.2 Data Preprocessing for Modeling

For predictive modeling, we performed several key preprocessing steps:

- Skewness Adjustment: Given the highly skewed distribution of numVotes, we applied a log transformation (using the log1p function) to stabilize variance [11].
- Missing Value Treatment: Missing values in our dataset were imputed using the column mean to ensure that no NaN values interfered with model training.
- Feature Scaling: Numeric features (including the log-transformed vote counts) were standardized using a standard scaler so that each feature contributes equally to the model.
- Target Variable Creation: For regression, the continuous target variable is averageRating. For classification, we created a new binary variable (success) by marking movies in the top 20% of ratings as successful.

The processed data were then split into training and testing sets (80/20 split) to allow for robust evaluation of our models.

8.3 Regression Modeling

8.3.1 Linear Regression Model

We implemented a linear regression model using scikit-learn. The model was trained on the preprocessed numeric features (including the log-transformed vote counts and encoded genres) to predict the continuous IMDb rating. We evaluated the model using standard metrics:

- Root Mean Squared Error (RMSE): Our linear regression model achieved an RMSE of approximately 0.946, indicating that on average, our model's predictions differ from the actual ratings by about 0.95 units on the 1–10 scale.
- R-squared (R^2): The R^2 value was approximately 0.1055, meaning that about 10.5% of the variance in movie ratings is explained by the model.

These results are modest compared to the findings reported by Nithin et al., where linear regression accuracy was reported at approximately 51%. Possible reasons for our much lower performance include the limited set of features used, potential nonlinear relationships not captured by linear regression, and differences in data preprocessing and outlier handling.

8.3.2 Logistic Regression Model

To further investigate the predictive power of our dataset, we implemented a logistic regression model using scikit-learn to classify movies as "successful" or "unsuccessful." For this binary classification task, success was defined as movies whose IMDb ratings fall within the top 20% of the distribution. Our target variable "success" was thus set to 1 for these top-rated movies and 0 otherwise. The model was trained on the same set of preprocessed features used in the regression analysis, which includes the scaled numeric attributes (such as startYear, runtimeMinutes, and the log-transformed vote counts) along with one-hot encoded genre indicators.

After splitting the data into training and testing sets (80%/20%), we fitted the logistic regression model within a pipeline that also handled missing values using a mean imputation strategy. The model's performance was then evaluated using standard classification metrics, including accuracy, precision, recall, and F1-score.

The resulting classification report is summarized below:

- Overall Accuracy: The model achieved an overall accuracy of approximately 79% on the test set.
- Class 0 (Unsuccessful Movies): For movies classified as unsuccessful, the precision was 0.79, and recall was 1.00, resulting in an F1-score of 0.88. This indicates that the model is highly effective at identifying movies in the majority class.
- Class 1 (Successful Movies): However, for movies in the top 20% (successful), the model's precision dropped to 0.50 and recall to 0.00, yielding an F1-score of 0.01. In effect, the model failed to correctly identify any successful movies.
- Macro-Average: The macro-average F1-score was 0.45, while the weighted average F1-score was 0.70, reflecting the imbalance between the classes.

These findings suggest that although the logistic regression model demonstrates good overall accuracy—largely driven by the dominant unsuccessful class—it performs very poorly in detecting successful movies. The near-zero recall for the positive class indicates a significant class imbalance issue, where the majority of titles fall into the unsuccessful category. This could be due to the top 20% threshold leading to relatively few examples of successful movies in the training data, or because logistic regression may be insufficiently flexible to capture the complex, potentially nonlinear relationships in the data.

Comparing these results with the findings reported by Nithin et al. (where logistic regression yielded around 42% accuracy), our model's performance appears suboptimal for identifying the minority (successful) class. Future improvements may involve rebalancing the dataset using oversampling or alternative algorithms (e.g., ensemble methods) that are more robust to class imbalance.

ACKNOWLEDGMENTS

We greatly appreciate feedback and comments from professor and fellow students in Spring 2025 semester of CSPB4502 Data Mining class in University of Colorado Boulder.

REFERENCES

[1] Coming to a Screen Near You: How Data Science Is Revolutionizing the Film Industry. Retrieved February 1st 2025 from: https://datacolumn.iaa.ncsu.edu/blog/2023/01/30/coming-to-a-screen-near-you-how-data-science-is-revolutionizing-the-film-industry/

[2] Warner Bros Will Now Use AI to Help Decide Which Movies to Make. Retrieved February 1st 2025 from: https://www.ign.com/articles/2020/01/08/warner-bros-will-now-use-ai-to-help-decide-which-movies-to-make

[3] Netflix Recommendations and Page Rank. Retrieved February 1st 2025 from: https://blogs.cornell.edu/info2040/2020/11/08/netflix-recommendations-and-page-rank/

[4] Deep Dive into Netflix's Recommender System. Retrieved February 1st 2025 from: https://towardsdatascience.com/deep-dive-into-netflixs-recommender-system-341806ae3b48

[5] Building a Collaborative Filtering Recommendation Engine. Retrieved February 1st 2025 from: https://datacolumn.iaa.ncsu.edu/blog/2020/02/10/building-a-collaborative-filtering-recommendation-engine/

[6] MovieLens 100k Dataset. Retrieved February 1st 2025 from: https://grouplens.org/datasets/movielens/100k/

[7] Netflix Research Archive. Available at: https://research.netflix.com/archive

[8] Data Science in the Film Industry. Retrieved February 1st 2025 from: https://www.kdnuggets.com/2019/07/data-science-film-industry.html

[9] How Data Analytics Help Movie Success Prediction. Retrieved February 1st from: https://datavisitor.com/how-data-analytics-help-movie-success-prediction/

[10] Building Recommender Systems. Retrieved February 1st 2025 from: https://www.kdnuggets.com/2019/04/building-recommender-system.html

[11] Jiawei Han, Micheline Kamber, Jian Pei. Data Mining: Concepts and Techniques, 3rd Edition. Morgan Kaufmann, 2011.

[12] Nithin V R, Sarath Babu P, Pranav M, Lijiya A, "Predicting Movie Success Based on IMDb Data," International Journal for Research in Applied Science & Engineering Technology (IJRASET), vol. 5, issue X, pp. 503–507, Oct. 2017.

[13] Michael Kristianto, Deastri Anggie Shanovera, Janette Mirna Putri Trijono, Ican Sebastian Edbert, Derwin Suhartono, "Movie Success Prediction Using Machine Learning Models." International Conference on Technology Innovation and Its Applications, 2024