DS3001 Final Project:
Predicting IMDb Ratings of New Films

Jackson Haiz Quinn Connor Teagan Ryan

Abstract

Movies are a major part of global popular culture, and understanding what drives audience ratings is valuable for filmmakers, streaming platforms, actors, and investors alike. Being able to estimate a movie's rating before it is released can help guide decisions about marketing, budgeting, and production, and can provide stakeholders with an early sense of how a film might be received. This project develops a predictive model for IMDb movie ratings, using information published on their website. Our goal is to answer the question: to what extent can movie metadata—such as genre, runtime, release year, and creator history—predict IMDb audience ratings prior to a film's release?

We begin by merging several IMDb databases and engineering additional features to capture the experience and track the record of creators, including the average rating of a director's prior films and the number of previous works associated with each writer. This resulted in a dataset containing 300,000 movies and TV shows, dating back to 1894. However, because we are analyzing only movies, we filtered the dataset accordingly during cleaning. This resulted in an approximately 43,000-movie working dataset, which will serve as the basis for data splitting. Because our purpose is to simulate real-world forecasting, we split the training and testing set chronologically – training on movies released from 2000 to 2019 and testing on movies released from 2020 onwards. Two predictive models are trained and then evaluated: Ridge Regression and Random Forest Regression. Hyperparameters are selected using time-series cross-validation to avoid information leakage and better match the historical structure of movie releases.

Although Ridge Regression achieves the strongest cross-validation performance, the Random Forest model performs best on unseen movies. On the test set, the Random Forest achieves an RMSE of 1.095, an MAE of 0.775, and an R^2 of 0.358, outperforming the Ridge model across all test metrics. Feature importance results reveal that vote counts, runtime, and director history play an important role in predicting ratings, while genre indicators also make meaningful contributions.

Residual analysis shows a clear regression-to-the-mean pattern, with the model underpredicting highly rated films and overpredicting poorly rated ones. This limitation reflects the concentration of IMDb ratings, the human tendency to avoid assigning extreme ratings, and the challenges of relying solely on metadata. Overall, our findings suggest that pre-release data can reliably predict mid-range ratings, while accurately modeling extremes may require further contextual information in future work.

1. Data Description

IMDb provides publicly available, non-commercial datasets that contain extensive information about films, TV shows, and other media. These datasets include details such as release date, runtime, genres, cast and crew, and audience rating. For this project, the primary goal is to develop a predictive model that estimates a movie's IMDb rating at the time of release using only features known beforehand – including genre, runtime, release year, and the historical success of its directors and writers. The key variable of interest is the average IMDb rating, which represents the weighted average of user-submitted scores. Our dataset reflects the version available as of September 26, 2025.

1.1. Variables of Interest

A description of the variables from IMDb.

Identifier / Helper Variables

- `tconst` (string) – alphanumeric unique identifier of the title
- `primaryTitle` (string) – the main title used by the filmmakers on promotional materials at the point of release

- `numVotes` (integer) – number of votes the title has received

Target Variable (y)

- `averageRating` – weighted average of all individual user ratings

Predictor Variables (x)

- `startYear` (YYYY) – release year of the title
- `runtimeMinutes` – primary runtime of the title in minutes
- `genres` (string array) – up to three genres associated with the title
- `directors` (array of nconsts) – director(s) of the title
- `writers` (array of nconsts) – writer(s) of the title

1.2. Cleaning the Data

Several steps were required to merge and clean the raw IMDb data files. To prepare the data, we first combined the datasets `title.basics.tsv.gz`, `title.crew.tsv.gz`, and `title.ratings.tsv.gz` using `tconst` as a primary key. Next, we dropped the unneeded columns in `title.basics`, specifically `originalTitle`, `isAdult`, and `endYear`. From there, we filtered the data to only keep rows where `titleType` equals `"movie"`, excluding TV shows and other types of media. All `\N` values were converted to `NaN` values to ensure that these data points were recognized by Python as missing data rather than literal strings. Additionally, due to the heavy right skew in the vote count, we applied a log-transformation (`log1p`) to the `numVotes` variable to normalize the distribution. To ensure reliable ratings and reduce the possibility of outliers driven by insufficient ratings, we removed movies with fewer than 500 votes to mitigate noise from obscure titles. Because IMDb lists each movie's genres as a comma-separated string, our group had to write a function to parse this string into an accurate list of the movie's genres and subgenres. This action was also necessary for movies with multiple writers and directors, which required the same transformation into a list format. We then restricted the dataset to movies released after the year 2000, focusing the scope of our model on films from the modern technological era. After filtering, we were left with 43,816 rows. To understand the underlying distribution of audience ratings, we also plotted a histogram of `averageRating` (Figure 1), which shows a strong concentration around a rating of 7 with relatively few extreme values.

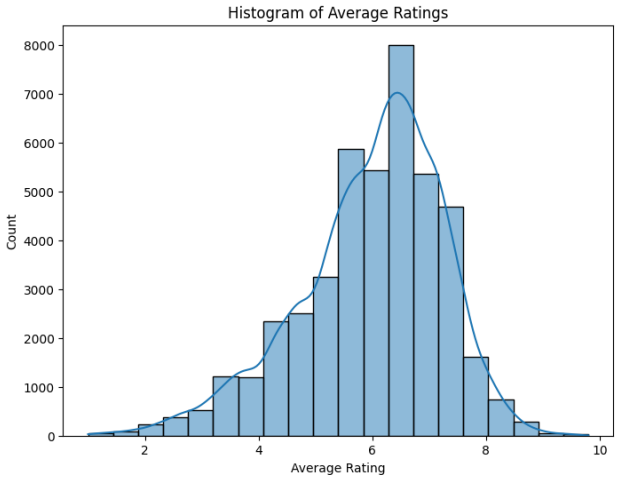

Figure 1. Distribution of the average number of votes

1.3. Exploratory Data Analysis

To understand each of our key numerical variables, our group engaged in exploratory data analysis. The findings from our exploratory data analysis will work to help reinforce and explain our findings from both our Ridge Regression and Random Forest models, and provide context for the outlook of movies we are analyzing.

We first analyzed the underlying distribution of audience ratings by plotting a histogram of average ratings. This distribution is strongly centered around a rating of 7 with relatively few extreme ratings (Figure 1). Additionally, to identify the means and variation in our data, we created a summary statistics table (Table 1) for three of our numeric variables. This enables us to examine the numerical properties of each variable more closely, which will help us better interpret the model results later. For `averageRating`, this table indicates that movies are scored at 6/10. Additionally, the findings show that movie ratings are relatively consistent, with a standard deviation of ±1.3 points from the mean. Despite this, the rating range exemplifies extremes in our data, with movies scored as low as 1 and as high as 9.8. The total average runtime of a movie is 1 hour and 45 minutes (105 minutes). A majority of films fall within 23 minutes of the average length, ranging from 36 minutes to 808 minutes. The average number of votes for our movie is 19,642. The standard deviation of 83,247.5 indicates a substantial spread in the number of votes, with some movies receiving a few thousand votes while blockbusters receive millions. The range of the film reinforces this notion, for movies in our dataset can have as few as 500 votes to as many as 3 million votes.

To understand the types of films we are analyzing, we examined the composition of film genres. A majority of the

Metric	Mean	Std Dev	Range
averageRating	5.9822	1.2749	1.0, 9.8
runtimeMinutes	105.2076	22.9290	36, 808
numVotes	19641.9793	83247.5574	500, 3077253

Table 1. Summary Statistics for Filtered Dataset

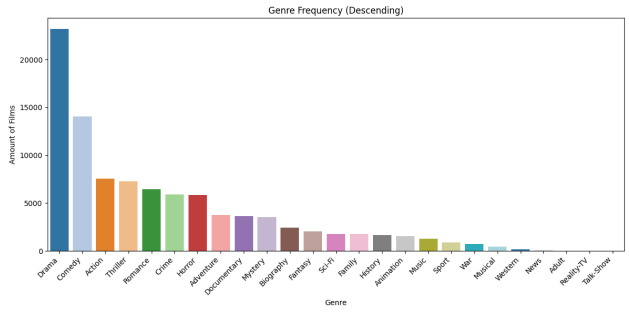

Figure 2. Distribution of Genres

movies we analyze are dramas and comedies. The total number of films declines with each subsequent genre, with Drama and Comedy followed by Action, Thriller, and Romance. This indicates that, although our dataset covers a wide range of movies, the genres are dominated by a few particular groups. This can be observed in Figure 2.

1.4. Variable Engineering

To capture the historical prestige of the directors and writers, we engineered two features based on past performance. First, we created the feature `prior_writer_count`, which counts the total number of films a writer has worked on prior to the release year of the target movie. Second, we created `prior_director_mean`, a target-encoding feature that records the average IMDb rating of all previous films directed by that specific director. This prevents data leakage by ensuring that a director's future success does not influence the prediction of their earlier work. Lastly, because movies can be associated with up to three genres, we created dummy variables for each genre category to allow films to have multi-genre representation.

2. Method Overview

2.1. Problem and Modeling Strategy

The goal of this project is to predict a movie's IMDb rating using only information available prior to its release, using such factors as genre, runtime, release year, and the historical success of its directors and writers. Because our target variable is continuous, we frame the issue as a case for regression. We also understand that interpretability is an important factor, so we also present categorical quality tiers (ex., bad, average, good) for a classification style-evaluation alongside our regression predictions.

We begin by establishing baseline performance using a naive model that predicts the mean rating for all films. We then evaluate two predictive models that reflect different assumptions about how metadata relates to audience rating. Ridge Regression is used for its interpretability and its ability to handle multicollinearity among predictors through L2 regularization. This model provides clear insight into how pre-release features relate to expected ratings. In contrast, Random Forest regression is chosen for its ability to capture non-linear patterns and interactions that linear models may miss, offering a more flexible framework. This combination of models employs us to evaluate both how explainable the signal is and how much additional accuracy can be gained by incorporating model complexity. The data was split chronologically to reflect a realistic forecasting scenario in which older films are used to predict future releases. Model performance will be assessed on metrics like RMSE, MAE, and R^2. Additional classification-style evaluation using binned predictions and a confusion matrix helps evaluate how well the model distinguishes broad rating categories.

2.2. Models and Hyperparameter Tuning

Two models were chosen to balance the interpretability of results and predictive complexity.

Ridge Regression (with L2 regularization): Ridge regression serves as the model for our prediction that reduces overfitting and handles multicollinearity among features such as genre indicators, runtime, vote count, and historical production success. This is important as we are aiming to generalize our predictions to new movies, and have some columns that might be related (like `numVotes` and `genre`). Next, we tuned the regularization strength alpha using a logarithmic grid of values ranging from 0.001 to 1000. Cross-validation on the training set was used to select the alpha value that minimized average RMSE. Importantly, this model's simplicity provides a clear interpretation of how each predictor affects the rating.

Random Forest Regression: Random Forest regression is a tree-based ensemble method that captures nonlinear relationships and interaction effects. In addition to strong predictive flexibility, random forests provide feature importance scores, which offer insight into which attributes most strongly influence the predicted rating. To tune the model effectively, we performed a grid search over a range of hyperparameters, including the number of trees (`n_estimators` = [100, 200, 300, 500]), tree depth (`max_depth` = [None, 40, 60, 80]), the number of features considered at each split (`max_features` = [`"sqrt"`, `"log2"`]), and the minimum number of samples required at a leaf node (`min_samples_leaf` = [1,

2, 4]). This search allowed us to identify the configuration that best balanced model complexity and predictive performance.

2.3. Model Training

The training process began by splitting the dataset into training and testing subsets using an 80/20 chronological split. To better reflect real-world forecasting conditions, the training set consisted of the earliest 80% of titles, while the most recent 20% were reserved for testing. This setup prevents information leakage from future data and avoids unrealistic scenarios, such as having a sequel appear in the training set while its predecessor is placed in the test set. Hyperparameter tuning for both models was carried out using 5-fold cross-validation with a time-series split on the training data, allowing us to evaluate model performance while reducing the risk of overfitting. For the Ridge Regression model, we built a scikit-learn pipeline that combined a `SimpleImputer` (using mean imputation for missing numeric values), a `StandardScaler` to standardize predictors, and the final Ridge estimator. This pipeline ensured that imputation and scaling were fit only on the training data and then applied consistently to the test set.

2.4. Implementation Details and Baseline Definition

The models were trained using a feature matrix and a target vector. The feature matrix included both numeric variables and one-hot encoded genre indicators. The numeric predictors consisted of `runtimeMinutes`, `startYear`, `log1p_numVotes`, `prior_director_mean`, and `prior_writer_count`. The one-hot encoded genre columns (e.g., `genre__Action`, `genre__Comedy`, ...) transformed the multi-label genre field into a set of binary features representing the presence or absence of each genre. Together, these components formed the feature matrix `X`, while the corresponding IMDb ratings served as the target vector `y`.

To substantiate the validity of both models, we will compare their initial results with a benchmark, baseline model that predicts the mean training-set rating. This baseline yielded an $RMSE$ of 1.709, an MAE of 1.033, and a R^2 of -.001. We expect both the Random Forest model and the Ridge Regression Model to outperform this baseline, thus authenticating their use as models fit to accomplish our goal.

2.5. Model Validation

Model performance was evaluated using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R^2. These metrics provide complementary views of predictive accuracy, including overall error magnitude, average absolute deviation, and variance explained. Although the task is a regression problem, many practical uses require broad quality categories rather than a precise decimal rating. To address this, we converted continuous ratings into categorical bins using a custom `bin_rating` function. Ratings were assigned to 1-point intervals (0–1, 1–2, ..., 9–10) using `np.digitize`, and each movie's actual and predicted rating was mapped to an interval label. From this, we constructed a confusion matrix by applying `bin_rating` to both `y_test` and the Random Forest predictions and calling scikit-learn's `confusion_matrix` with the ordered bin labels as category levels. This allowed us to identify where the model systematically over- or under-predicted.

3. Results

3.1. Model Performance Comparison

Hyperparameter tuning using time-series cross-validation resulted in different strengths for the two models. Ridge Regression achieved the highest cross-validation score, suggesting strong stability on the training distribution, while the Random Forest model achieved superior predictive performance on the test data. Although the Random Forest performs better overall, both models explain only a modest portion of the variance, which is expected given the subjectivity and noise inherent in IMDb ratings.

Parameter	Value
`alpha`	100.0
Best CV Score	0.939

Table 2. Ridge Regression Hyperparameters

Parameter	Value
`max_depth`	40
`max_features`	"sqrt"
`min_samples_leaf`	2
`n_estimators`	500
Best CV Score	0.867

Table 3. Random Forest Regression Hyperparameters

Metric	Ridge Regression	Random Forest
RMSE	1.183	1.095
MAE	0.816	0.775
R^2	0.307	0.358

Table 4. Model Test Performance Comparison

Both models perform substantially better than the baseline model trained on the target vector mean, as highlighted in section 2.4. This was expected, as the benchmark is essentially an Ordinary Least Squares (OLS) model designed to set a lower bound for improvement. The ridge regression handles multicollinearity, reduces fitting, and leverages all

predictors, whereas the Random Forest model captures nonlinearities and is robust to outliers. These are just a few of the reasons why each model should and did perform superior to the simple benchmark. These results indicate a meaningful contrast as ridge regression generalizes more smoothly during cross-validation, likely because it is a highly constrained (regularized) linear model. Also, random forest captures nonlinear patterns that ridge regression can not, which leads to better real-world predictive performance on the test data, despite its lower CV score. This discrepancy also suggests mild overfitting in Random Forest, but not enough to outweigh its improved predictive accuracy. The relationship between predicted and actual ratings is shown in Figure 3. Most predictions fall within the central range, reflecting the overall rating distribution, while deviations from the diagonal highlight differences in model accuracy.

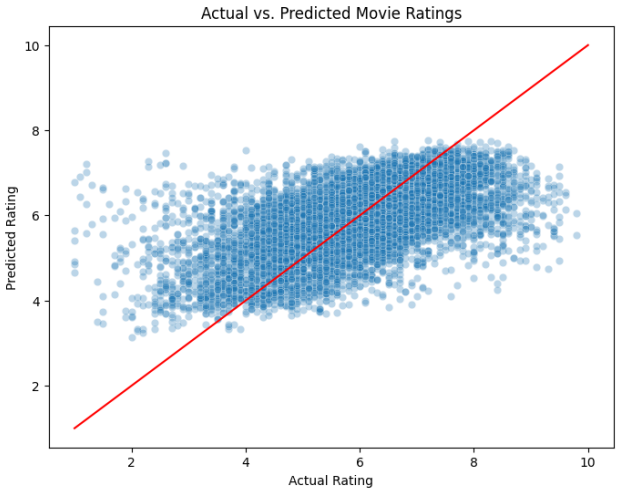

Figure 3. Comparison of actual vs. predicted movie ratings

3.2. Feature Importance

Figure 4 illustrates the top feature importances in the Random Forest model. The most influential predictors were `log1p_numVotes`, `runtimeMinutes`, `prior_director_mean`, the genre indicator variables, `prior_writer_count`, and `startYear`. These results indicate that both movie-level characteristics and creator history contribute meaningfully to rating prediction. However, metadata alone cannot fully explain rating outcomes, as many influential factors—such as narrative quality, marketing, audience expectations, and franchise reputation—are not represented in these features.

3.3. Prediction Fit and Error Behavior

To assess model reliability across the rating spectrum, we analyzed several forms of prediction error. A consistent

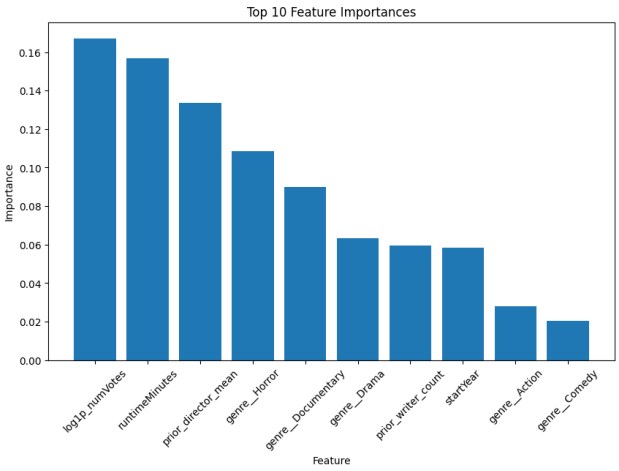

Figure 4. Top predictors for Random Forest

pattern emerges as the Random Forest tends to over-predict low-rated movies and under-predict high-rated ones. This behavior, visible in the error distance plot, reflects a form of regression towards the mean, which is common in datasets where extreme values are relatively rare. Most movies in our dataset fall between ratings of 5 and 8, and the model naturally learns to center predictions near this range. Despite this limitation at the extremes, the overall distribution of residuals is approximately symmetric and centered around zero, which may suggest that the model does not systematically favor over- or under-prediction across the whole dataset. This supports the idea that while Random Forest captures the central tendency of IMDb ratings alright, it struggles to differentiate either highly acclaimed or widely disliked films from more typical ones. Figure 4 plots the distribution of residuals. The residuals are approximately symmetrically distributed around zero, suggesting that although extremes are difficult to capture, the model is not systematically biased.

3.4. Binned Classification Performance

Looking further into the practical utility of the model by constructing a confusion matrix, this categorical evaluation highlights where the Random Forest succeeds and where it struggles. The model performs strongest in the most common intervals, particularly the 6-7 and 7-8 bins, where the density of training examples is the highest. However, its ability to correctly classify movies with very low or very high ratings is considerably weaker, with predictions frequently shifting towards the middle bins. This reinforces the earlier findings that metadata-driven models have difficulty identifying extreme films. The confusion matrix, therefore, serves as supplemental evidence that the model is reliable for forecasting "typical" films, but is limited when predict-

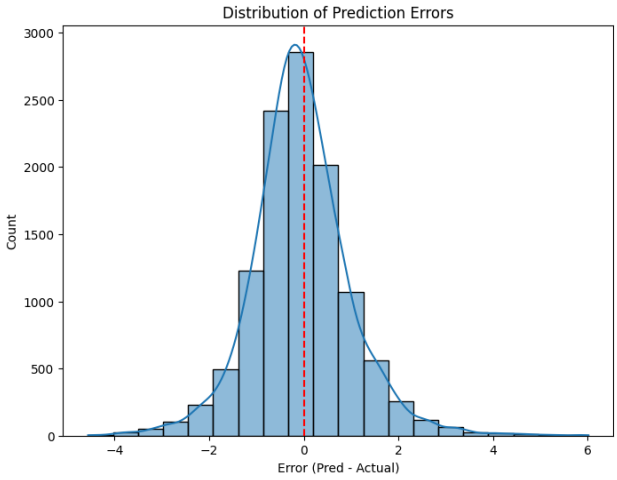

Figure 5. Distribution of Residual Errors

ing outliers. Below, Table 5 is the confusion matrix for the binned rating predictions – this pattern reinforces that the model captures general trends but lacks the information needed to accurately identify exceptional films.

	0–1	1–2	2–3	3–4	4–5	5–6	6–7	7–8	8–9	9–10
0–1	0	0	0	0	3	2	1	0	0	0
1–2	0	0	0	9	9	14	10	2	0	0
2–3	0	0	0	45	98	54	26	7	0	0
3–4	0	0	0	56	300	207	79	3	0	0
4–5	0	0	0	54	507	676	275	10	0	0
5–6	0	0	0	17	344	1403	1034	55	0	0
6–7	0	0	0	4	101	1096	2364	290	0	0
7–8	0	0	0	0	19	242	1180	467	0	0
8–9	0	0	0	0	10	93	214	115	0	0
9–10	0	0	0	0	3	18	30	1	0	0

Table 5. Confusion matrix comparing actual (columns) and predicted (rows) rating bins.

4. Conclusion

The goal of this project was to determine how accurately IMDb movie ratings can be predicted using only pre-release data. By combining multiple IMDb datasets, creating new engineered variables, and implementing both Ridge Regression and Random Forest models, we explored the extent to which audience score can be anticipated before a movie is released to the public. The Random Forest model ultimately achieved the best performance on unseen data, though the overall accuracy of both models revealed clear limitations. The highest R^2 value obtained was 0.358, indicating that a large portion of the variance in the dataset remained unexplained by the variables included in our models. This highlights fundamental limitations in a metadata-only approach, emphasizing the complexities of audience reception.

Several patterns in the results highlight why this is the case, as both models struggled consistently with extreme ratings, tending to over-predict poorly reviewed films and under-predict highly rated ones. This regression towards the mean reflects the concentrated nature of IMDb ratings as well as the absence of more informative contextual features that influence audience opinion, such as plot quality, cast population, marketing visibility, and brand reputation. Even with our feature engineering, metadata cannot capture these more nuanced drivers of audience opinion. The confusion matrix further illustrates that the models perform best within densely populated mid-range categories, while films in less frequently rated bins are misclassified more often.

Another key challenge in predicting IMDb ratings lies in the nature of the ratings themselves. While IMDb scores reflect viewership opinions on the used metadata (director, length, genre, etc.), these opinions shift as movie-goer demographics, global cultural expectations, and franchise loyalty inevitably change. Our earliest movies in the dataset, from 2000, were watched and reviewed in a completely different context than movies today. Moreover, IMDb ratings are especially prone to self-selection bias, as usually, viewers who choose to rate a movie have a strong positive or negative opinion on the film, which tends to differ from the general consensus. Since our models rely strictly on metadata and are unable to capture rater bias and shifting cultural norms, it makes sense that our model has a lower R^2 value.

Although these limitations restrict predictive accuracy, the project does demonstrate that certain measurable characteristics like runtime, historical director success, and genre provide meaningful indicators about future audience ratings. The models offer a baseline for what can be achieved using structured pre-release information and they highlight where additional data sources are needed. Future work should incorporate details like plot summaries, cast-level indicators, or production budget to better capture the narrative and creative elements as a whole which strongly influence film reception. This might look like including plot summary embeddings from an NLP model, implementing genre-specific models to separate estimators, and adding a temporal aspect to account for evolving audience preferences across decades.

Overall, this project demonstrates that metadata alone offers only a partial view of audience response. While mid-range predictions are reasonably stable, accurately forecasting outliers remains challenging. These findings emphasize both the value of the limitations of machine learning approaches in predicting inherently subjective outcomes like movie ratings.

5. References

IMDb Datasets (Accessed: Sept 26, 2025). https://datasets.imdbws.com/