LIGHTS, CAMERA, RANDOM FOREST! PREDICTING MOVIE RATINGS USING IMDB DATA

CONNOR MCBRIDE, KURT HEALY, KADEN PARKER

Abstract. What factors affect a movie's rating? In this paper, we use the non-commercial IMDb dataset [IMD25] to analyze which data can be used to predict a movie's rating. We consider questions such as "Is it possible to accurately predict a movie's rating using data such as director, cast, and genre?", "Can a director's prior ratings be used to predict their next rating?", and "Which factors influence a movie's rating?" We use linear regression, random forest regression, and XGBoost to create models, which we then use to help us answer these questions. It is time for Lights, Camera, Random Forest!

1. Research question and overview of the data

We seek to analyze what data are needed to predict a movie's rating. It often seems as if having an accomplished director or popular lead actor is enough to ensure the success of a movie. It also seems like some genres are more likely to produce a highly rated title. We will report the results of using linear regression, random forests and XGBoost to try and predict the rating of a movie from other data about that movie. Additionally, we analyze whether a director's rating is correlated with their prior ratings, which we call the director's "momentum."

Carnegie Mellon does in-depth work analyzing the IMDb movie data [Car22]. They measure how factors such as the movie's rating (G, PG, R, etc.) or release season impact the movie's rating on IMDb through basic statistical analysis and visualizations. We similarly measure how relevant features impact a movie's rating, but rather train various regression models to predict the movie's rating. The relevant features used in each dataset are shown in Table 1.

dataset Name	Useful Features in dataset
akas	title, language of title, whether title is original title
basics	title, year of release, genre and runtime
crew	writers and directors
ratings	rating for each title (this is the target variable)

Table 1. IMDB data used in this analysis.

Date: 24 November 2025.

2. Data Cleaning / Feature Engineering

The raw IMDb datasets are very clean to begin with. The format of their values were very consistent and any null values were represented as `\\N`. However, each of the questions we were trying to answer required their own feature engineering. Answering our research questions with different models and engineered features required us to format the data slightly differently. Therefore, we discuss the data cleaning and feature engineering for each question individually.

2.1. **Random Forest Regressor.** A large challenge with our data and using random forests is the need for numerical data to train the regressor on. Normally, one-hot encoding is a potential solution for this, but with some of the columns of our dataset containing over three million unique strings, this is not feasible. Thus, for an initial model, we took what we believed to be the possibly most informative numerical features and joined datasets to capture them.

We one-hot encoded the genre of movie, as there were a small enough number of them. We dropped text data from our joined dataframe such as, `originalTitle`, `primaryTitle`, `titleType`, and `tconst`. The dataset didn't contain `Nan` values, but did contain `\\N`, which represented `Nan`, so we replaced these with `Nan` values. After a first iteration, we made a second model where we dropped all non-genre columns.

2.2. **Linear Regression.** To analyze the "momentum" of a director, we needed to get all of the ratings for each director into the same dataset. We use the basics, crew, and ratings datasets for this.

First, we manipulated the crew dataset to create a 1-1 map between movie title and director. We want to do this so that we can union this dataset with the other datasets. To do this we removed all rows that did not have a director. Then, since some movies have multiple directors, split each row with multiple directors into separate rows so that each row contains only one director.

Second, we took the ratings dataset and applied a mask to only keep rows that had more than 1000 votes. We did this because most titles had few votes and we wanted the ratings to be statistically significant. Then we merged these two datasets on title. This created a dataset that had title, director and average rating.

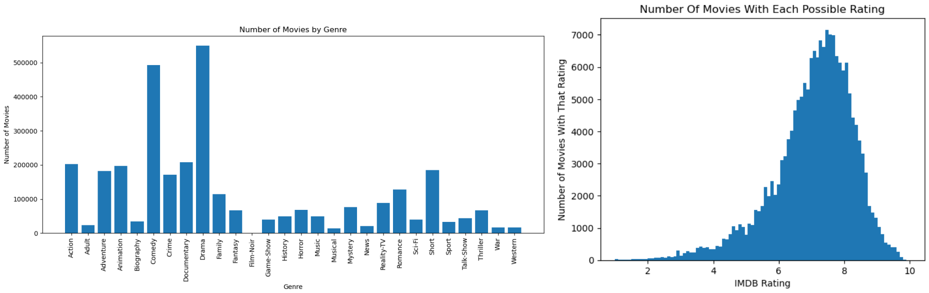
Third, we took the basics dataset and decided to only keep movies that were released in the year 2000 or later. [Car22] analyzed the distributions of movies by rating per decade. We saw that the distributions of ratings from the 1980s and 1990s were quite different than the distribution of ratings after that so decided to only analyze movies from the 2000s onward.

We then dropped all columns except for title and year the movie was released. Then we joined all of the manipulated data on title. Finally, for each director we ordered the ratings for each director into a sequence of data points. We will use these sequences for regression.

2.3. **XGBoost Regressor.** To train an XGBoost regression model, we needed to ensure that strings were not used in the columns, as XGBoost cannot take strings. We used one-hot encoding to create binary indicator columns for the genres column, as it was given as a list of strings. We also created numeric features for the model to use. We attempted to record how "popular" certain actors and directors were by counting the number of movies they appeared in or directed, as more prominent actors and directors may contribute to a higher-rated movie. Because certain datasets had movie titles that had ratings while others contained every movie title, this created `Nan` values in some of the features for actor and director popularity when merging the datasets. We replaced the `Nan` values with 0 as the trees handle 0 as missing values. Finally, we also included the director momentum from the previous model as a feature.

3. Data Visualization and Basic Analysis

We found that the average movie rating by genre did not provide any useful insights. All of the average movie ratings were clustered together by genre around 6.6 - 7 on IMDb. So while there are many more counts of some movies by genre than others, their means are all clustered together. This lets us assume that the IMDb movie reviews are similarly distributed by genre regardless of count.



Counts of Movies by Genre

Number of Ratings per Rating

Figure 1. Comparison of movie genres and ratings.

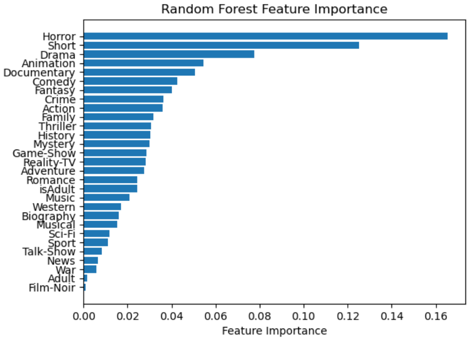
4. Learning Algorithms and In-depth Analysis

4.1. **Random Forest Regressor.** We trained our initial random forest regressor on test data with features of genres, and other numerical features. Each model was trained using an 80-20 train-test split. This gave us a mean squared error of 1.81373, a mean absolute error of 1.01438, and an R^2 value of 0.08272.

Table 2. Model Score Summary

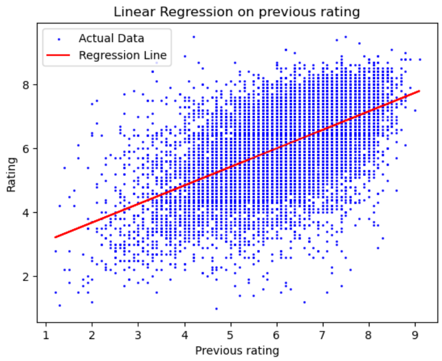
Model	MSE	MAE	R^2
Baseline (guess sample mean)	2.002	1.0573	0
Random Forest	1.81373	1.01438	0.08272
Linear Reg. (1 Prior Rating)	0.8567	0.7129	0.3245
Linear Reg. (5 Prior Ratings)	0.6124	0.6062	0.4693
XGBoost	1.20123	0.82773	0.36629

We trained an additional random forest regressor trained on test data with just genres as features. This gave us a mean squared error of 1.79779, a mean absolute error of 1.01558, and an R^2 value of 0.0907. Both of these beat guessing the expected value of the ratings. This answers that we can accurately predict a movie's rating within a score of 1 on average based on the features. We also answer the question of which movie genres are most predictive.



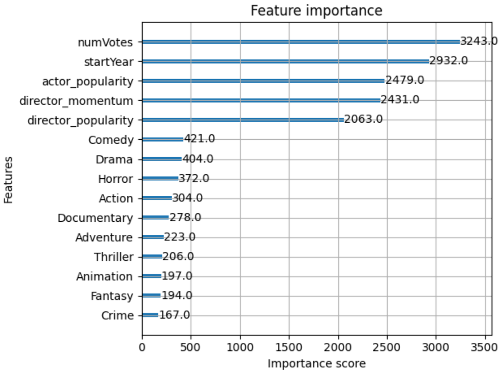
Random Forest regression model trained on IMDB movie genre data's most important features.

4.2. **Linear Regression.** To see if the "momentum" of a director could be used to predict their next movie rating we decided to use linear regression. We took the prepared dataset of chronological sequences of movie ratings for each movie director and used a function that sliced out sequences of length `n`. For example, if [1,2,3,4,5] was a sequence of movie ratings and we wanted sequences of length 3 then the function would create a new dataframe with rows [1,2,3], [2,3,4], and [3,4,5]. We use the last column of this new dataset as the target variable y and the rest of the columns as our independent variables X. Training scikit-learn's linear regression function on this dataset produced a mean absolute error of 0.7129 when we only considered the previous rating for a director. Interestingly, when we consider longer sequences, we minimize the mean absolute error at a value of 0.6062 when we consider the previous 5 movies. After 5 the mean absolute error begins to rise. This means that given only the previous rating we can predict with 0.7129 of the actual score on average, and on average we can predict within 0.6062 given the last 5 ratings.



Linear Regression Model Regression Line

4.3. **XGBoost Regressor.** We similarly sought to predict the average movie rating on IMDb based on basic information about the movie, such as the year released, popularity of the actors and directors who worked on it, and genre of the movie. We also incorporated the engineered feature of director's momentum with a rolling window of 5 movies. The model achieved a MSE of 1.2012, MAE of 0.8277, and R^2 value of 0.36629. The complete code used for model creation and data visualization can be found at the repository [McB25].



15 Most Important Features in XGBoost Model

5. Ethical Implications and Conclusions

Our project does not post many immediate ethical concerns. One potential consequence is that movie companies could determine the most important features used by a website like IMDb to rank and recommend movies, and try to artificially increase their values to be recommended by the algorithm more.

References

[Car22] Carnegie Mellon University, Department of Statistics & Data Science. Imdb movie analysis. https://www.stat.cmu.edu/capstoneresearch/fall2022/315files_f22/team24.html, 2022.

[IMD25] IMDb. Imdb non-commercial datasets. https://developer.imdb.com/non-commercial-datasets/#titleprincipalstsvgz, 2025.

[McB25] Connor McBride. movie-review-analysis. https://github.com/connor-mcbride/movie-review-analysis, 2025. GitHub repository.