Predicting a Movie's Revenue Before its Release

IMDBot (bfoulon, xzhan199, yhua12, yzhou153)

Goal

Movie revenue matters a lot for investors. Is it possible to predict a movie's revenue *a priori* based on known info (such as genre, runtime, cast choices, director popularity) to reduce the investment risk? IMDBot aims to predict a movie's revenue before it's released based solely on data from IMDB and Twitter.

Data

We used IMDB's dataset to get the title, release date, genres, director, writer, actors, and runtime of movies released from 2009-2019. IMDB's dataset does not include revenues or countries, so we got these values by web scraping IMDB's website. Movies without revenues listed were not included; this left a dataset of ~24,000 movies.

Tweets mentioning the director and writer of each movie were collected using Twitter's API via Python's Tweepy package. Sentiment analysis was performed on those tweets using Python's TextBlob package. We recorded (1) the number of tweets and (2) average sentiment analysis for each writer and director into our database.

Some promising features were unfortunately not useful. Budget, for example, was left blank most of the time, rendering it unusable as a feature. In general, for each numerical column, we created a column showing if its value was null or not. For category columns, we applied Sklearn's CountVectorizer to treat the *N* most frequent strings as *N* binary presence-or-absence variables for each movie. (Note: we refer to *N* as "max features" in later sections).

Model and Evaluation Setup

For training and testing, the standard 80/20 train/test split was used; 20% is enough to ensure a large enough test set without taking too much data away from training. To forego the need for a validation set and to increase model robustness, 10-fold cross validation was used.

Our revenues span 9 orders of magnitude ($13 to $2.8 billion). Given that scale, it is unsurprising that mean absolute error (MAE) and mean squared error (MSE) were huge (though their relative values varied). R^2, being normalized, is not affected by the scale of the data, making it a good measure of model effectiveness.

As a baseline, we used Sklearn's DummyRegressor (strategy = mean). As expected, it yielded a bad test R^2 (-0.031), along with an MAE of $17.1 million.

Table 1 compares machine learning (ML) models that were tried out. Random Forest (RF) and Gradient Boosting (GB) performed the best. RF was selected as the ML approach for our model.

Model Type	R^2	MAE (Millions USD)
Random Forest (RF)	0.554	14.60
Gradient Boosting (GB)	0.552	15.81
Ridge Regression	0.361	25.29
Decision Tree	0.107	17.77
Dummy Regressor (baseline)	-0.031	17.13

Table 1: Models and their performance.

RF and GB are well-suited for complex data. RF outputs the average prediction of an ensemble of decision trees, making it robust and less prone to overfitting. GB is another ensemble method that gives bad predictive models high weights and good ones low weights at the end of each iteration; this lets the model focus on the bad ones in order to improve overall performance. The fact that L_2-regulated (Ridge) linear regression does badly but the more complex RF and GB models do well suggests that correlations between the features and revenues are non-linear.

Results and Analysis

***Claim #1*: Using up to 30 features for vectorizing gives the best balance between R^2 and MAE**

We tried out different max feature values (N) for vectorizing category columns. As Figure 1 shows, increasing max features lowers MAE. However, R^2 decreases after 20 max features. Therefore, we chose 30 for our model, which balances the R^2 and MAE.

Max Features	R^2	MAE (Millions USD)
10	0.542	14.97
20	0.555	14.67
30	***0.554***	***14.60***
40	0.550	14.50
50	0.548	14.49
100	0.544	14.38

Table 2: The model's performance based on the max number of features used in CountVectorizer for treating categorical variables. All category columns were used.

***Claim #2*: Genres play an important role in movie revenues.**

We trained the model with different combinations of categories while controlling other variables. As Figure 1 shows, genres (G) is the most important individual category in R^2 terms. Including all writers, directors, country, actors and genres as categories gives us the best R^2 value.

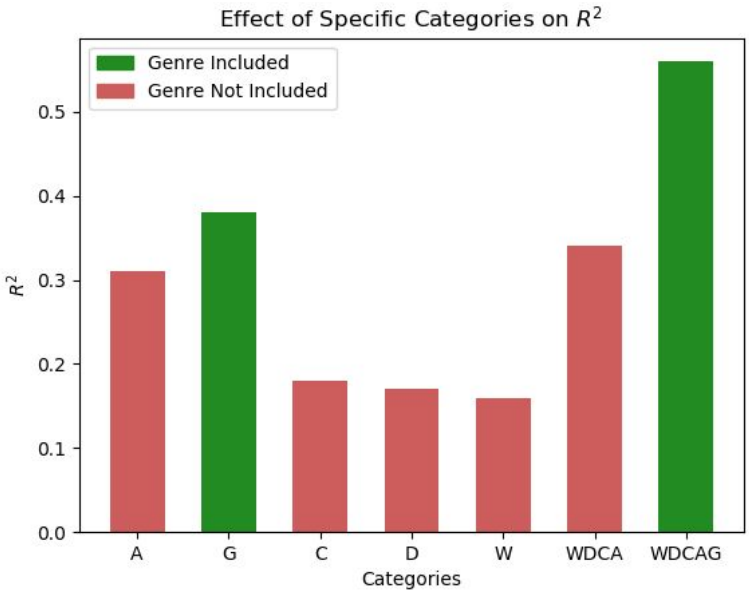

Figure 1: R^2 values based on category variable inclusion. Category labels refer to Writers (W), Directors (D), Countries (C), Actors (A), and Genres (G).

<u>Claim #3</u>: Model works better with adventure, sci-fi, action, animation and comedy genres

To investigate specific genres, we trained models focused on one genre while keeping other settings constant. As Table 3 shows, adventure-, action-, comedy-, animation-, and sci-fi-specific models had the highest R^2 values, suggesting that those genres are most predictive of revenue. Part of this may be due to low-count genres having small sample spaces in our data. However, this is not always the case: Drama, Romance, and Documentary have high counts but low R^2 values, while Sci-Fi has a low count but a high R^2.

Genre	Count	R^2	MAE (Millions USD)
Adventure	1772	0.646	66.7
Sci-Fi	579	0.632	64.4
Action	3034	0.614	40.9
Animation	1125	0.559	39.5
Comedy	7727	0.553	14.9
Fantasy	864	0.461	49.9
Mystery	1069	0.341	17.9
Thriller	2544	0.278	19.0
Horror	1582	0.216	15.3
Crime	2051	0.210	15.4
Romance	2993	0.195	11.0
Drama	11397	0.137	11.8
Documentary	2820	-1.711	16.6
Musical	206	-21.467	16.4

Table 3: Selected genres and their effects on the model.

<u>Model Prediction Analysis</u>: When revenue increases, our model's prediction absolute error also increases.

Figure 2a shows the absolute error on the model's predicted revenue vs actual revenue for each movie. In general, our model's predicted absolute error tends to be larger for movies with larger revenues, but the effect is not as pronounced for revenues <$500 million (i.e., the majority of movies). Moreover, the MAE and individual errors relative to revenues is typically small enough that our model can estimate pretty well which movies will be hits or flops.

Figure 2b illustrates this point: the overlap between predicted and actual revenue is substantial, and there are very few qualitative misses. For example, movie 310 is estimated to make ~$550 million instead of $1.2 billion; this is a big difference for bookkeeping, but qualitatively the model correctly predicts that the movie will be a hit.

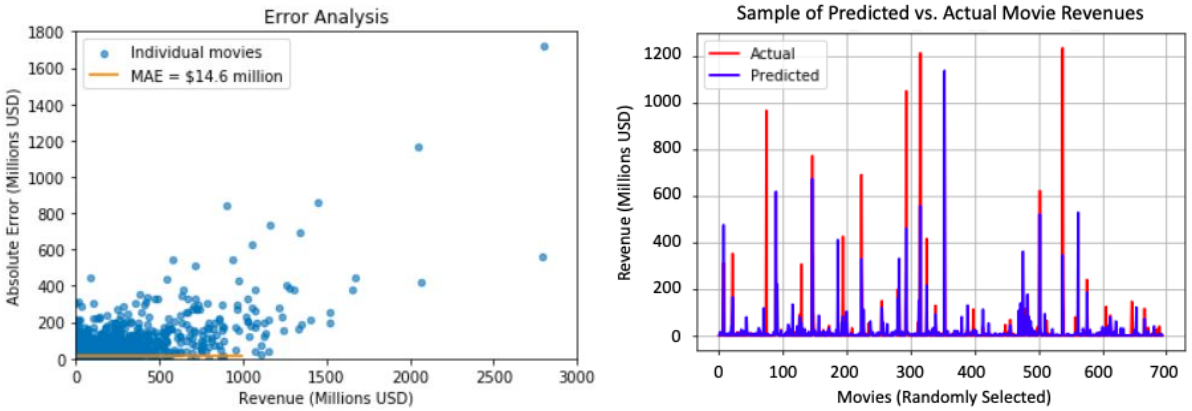

Figure 2: Left (a): Comparing the model's prediction absolute error to the movie's actual revenue for each movie. Right (b): Comparing predicted and actual movie revenues for randomly selected movies in our dataset.