BoxOffice: Machine Learning Methods for Predicting Audience Film Ratings

Shivam Mevawala
The Cooper Union for the
Advancement of Science and Art
41 Cooper Sq, New York, NY 10003
shivmevawala@gmail.com

Sharang Phadke
The Cooper Union for the
Advancement of Science and Art
41 Cooper Sq, New York, NY 10003
phadke@cooper.edu

***Abstract*—This project explores machine learning methods for predicting the audience ratings of movies. The process of selecting and extracting features for each movie in a corpus of samples is discussed. In particular, methods for converting symbolic film features, such as cast members and directors, into numerical values are presented. The results of several regression algorithms are discussed. This approach can be used to predict the success of movies, as characterized by audience ratings, based solely on primitive features, such as the choice of director, genre, and cast with an average error of approximately 10%.**

***Keywords*—audience ratings, films, motion pictures, box-office, regression**

I. Introduction

The film industry is a multi-billion dollar industry whose success depends on producers' abilities to choose winning combinations of stories, directors, actors, and crew-members. Thousands of feature films are produced each year, yet only three out of ten movies break even at the box-office [1]. By studying the factors that influence the popular success of movies, we hope to better inform the choices of producers.

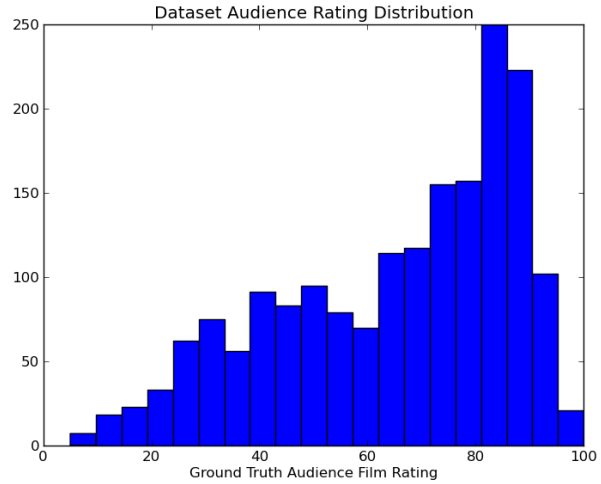
The success of a film can be characterized in many ways, including box-office revenue, critic ratings, and audience ratings; as well as numerous other valuable but unquantifiable measures. The body of economic research pertaining to the success of motion pictures identifies three major influences: movie characteristics, studio marketing, and non-studio factors [1]. Movie characteristics include the reputations of directors and actors, as well as the genre, story, and cultural influence of the story. Studio marketing refers to the production and proliferation of on-line, print, and television advertisements. Research shows a clear positive correlation between studio marketing budgets and box-office revenue. However, marketing expenses do not necessarily correlate to ratings. Finally, non-studio factors, such as reviews, awards, and other unpredictable parameters, can also have a major impact on the success of a film.

A popular approach to predicting box-office success was recently developed by Google [2]. This approach utilizes Google's vast corpus of search data to predict box-office performance using query volume with over 90% accuracy. Although this method is robust, it accounts for 70% of the variance in box-office performance only one week prior to the release of the film. As a result, it is of no use to producers and other film professionals when making early-stage decisions.

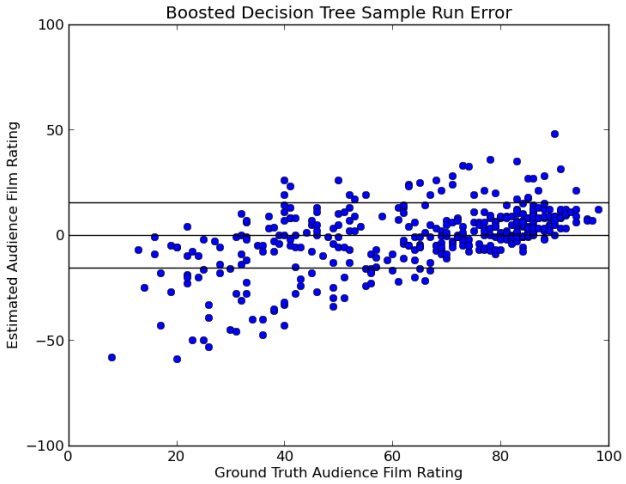
For this project, we chose to use movie characteristics exclusively, as these features can be considered by producers prior to beginning work on a movie. In addition, we chose the audience rating as our criterion variable, because this metric has a particularly broad impact, and is influenced by millions of viewers around the world.

II. Feature Selection

The first major challenge of this project was selecting features by which to characterize movies such that a regression algorithm could be applied to predict the audience ratings of movies. Based on the available sources of data and the decision to use exclusively predetermined movie characteristics, the director, actors, and genres of movies were chosen as our major features. Release date, movie length, and production company were also considered, but were excluded in this set of simulations. Many of these features have been shown to have significant correlation with box-office success in [3], but basic movie characteristics such are not commonly used in machine learning based regression approaches. Much



(a) Example Audience Rating Ground Truth



(b) Sample Run Error for 1 Iteration

Fig. 1. Sample Audience Film Ratings

of the reason was the challenge of converting symbolic features into numerical values.

Previous approaches to the challenge of converting symbolic features into numerical values have used search engine popularity and social media presence to transform names of actors and directors into numerical data. However, popularity is not directly correlated to a movie professional's talent and consequently may not be correlated with the audience ratings movies. In addition, it is difficult to obtain historical search engine popularity, a constraint that would limit our analysis to current films. In 2011, Xue and Chen from Stanford University attempted to classify the box-office success of movies into bins ranging from $< \$100\text{K}$ to $> \$100\text{M}$ [4]. Since they used the number of search engine results, they could only train and test on movies within the last two years. On top of that, movie professionals are often popular for completely unrelated reasons. Xue and Chen's approach had a misclassification rate of approximately 35%.

This project used a simple statistical approach to overcome the challenge of symbolic features. For the most prominent director and the four most prominent cast members of each film, the scores of all other films those professionals have been a part of were aggregated. Characteristics of each of these collections of movie scores, such as estimates of the mean and variance, were used to characterize each professional. Similar scores were chosen as features for each genre, computed over all films in our database with each genre.

III. Feature Extraction

After exploring available data sources and selecting a series of features, two popular film databases, The Movie DB (TMDB) and Rotten Tomatoes, were used to extract features. Python was chosen as a scripting language to easily access the APIs of each of these databases, and SQLITE was chosen as a local database engine to store our data.

A. Data Sources

The first step in feature extraction was finding a list of films with varying audience ratings. The Sagel Index of the top 1000 and worst 1000 films was chosen as an initial dataset, as these lists provided a set of films with a wide array of ratings [5], [6]. In addition, the IMDB 200 Worst Flops of All Time were added to the data set because many of the films in the Sagel worst 1000 list were not found in the on-line databases [7]. A histogram of the audience rating of the the entire data set can be seen in Figure 1a, which clearly demonstrates the bi-modality of the data.

The names of movies from these lists were extracted and used to query the TMDB API for the metadata of each movie [8]. Finally, for all movies in the database, the audience rating of each movie was found by querying the Rotten Tomatoes API [9]. Rotten Tomatoes was chosen because it is a popular source for audience and critic ratings for Hollywood films, especially among young people.

B. Numerical Director and Cast Member Features

Numerical feature values for directors and cast members were obtained by performing queries to the TMDB API to obtain lists of other films that each film professional had participated in. In order to maintain a manageable level of dimensionality, only the four most significant cast members, as determined by TMDB, were stored for each movie in the training and test sets. Finally, the RottenTomatoes API was used to find the audience ratings of all of these movies. All of these values were stored in our SQLITE database.

C. Numerical Genre Score Features

We applied a similar method to numerically rate each of the genres in the data set. All movies in the initial data set were grouped by genre, and the sample mean and variance of the audience ratings of each of these collections was found using maximum likelihood estimation. These distribution parameters were used as genre features.

D. Features for Regression

In the simulations performed in this project, the collections of film professional and genres were distilled into regression features by estimating means and variances for the set of audience ratings for each collection. Then for each film in the training and test sets, the cast ratings and variances were averaged to normalize the importance of all cast members relative to each other. This, simplified method for converting non-numerical features into numerical values led to decent regression results, but more complex models could have been used. For example, the method of simply estimating means and variances does not account for the improvement in film professionals' careers over time.

IV. Regression Methods

Four different regression techniques were attempted in this project: Support Vector Regression, AdaBoosted Decision Tree Regression, Gradient Boosting Regression, and Random Forest Regression. A diverse set of techniques was chosen because of the non-linearity of the feature space - because each of these techniques varies greatly in performance on non-linear data, each of them was simulated. Each method yielded acceptable results on average, but varied heavily in certain cases. The Python library `scikit-learn`, which implements each of these methods, was used to simulate the regression models [10]. The results of a sample run over a single iteration can be seen in Figure 1b, which shows sample predictions as a function of ground truth film ratings, as well as the sample variance of the test set.

TABLE I. Regression Algorithm Error Metrics Over 100 Iterations

Regression Algorithm	Mean Error	Median Error
SVR	11.36%	8.27%
Boosted Decision Trees	10.98%	7.97%
Gradient Boosting Regression	10.94%	8.25%
Random Forest Regression	11.08%	8.39%

V. Results

Despite the skewed nature of the data set, each of the four regression methods produced reasonable results. Several simulations were conducted for each of the methods, and parameters of each method were adjusted to achieve optimal performance. However, most parameters were found to have little to no effect on the overall performance. Eventually, each of the regression methods were tested in further detail after their most significant parameters had been established through manual test runs. As can be seen in Figure 2, both the average regression error, as well as the standard deviation in the error varied significantly with the ground truth film rating. For lower ground truth values, the regression error was significantly more negative (and larger in magnitude), while the error for highly rated films was significantly lower. The overall error metrics for each of the regression algorithms can be seen in Table I.

VI. Discussion

For each of the regression methods, the median regression error was found to be significantly lower than the mean. This indicates that a small number of predictions had very high errors, while the bulk of predictions fell within a smaller variance. This can be observed in a sample run in Figure 1b. In fact, the median error is more indicative of the performance of each method, because the median is relatively uninfluenced by such outliers. The best performing regression method was found to be boosted decision trees, with a median error of 7.97%. Decision trees are particularly powerful for this data set because they can discriminate on features based on their relative entropy, while methods traditionally used

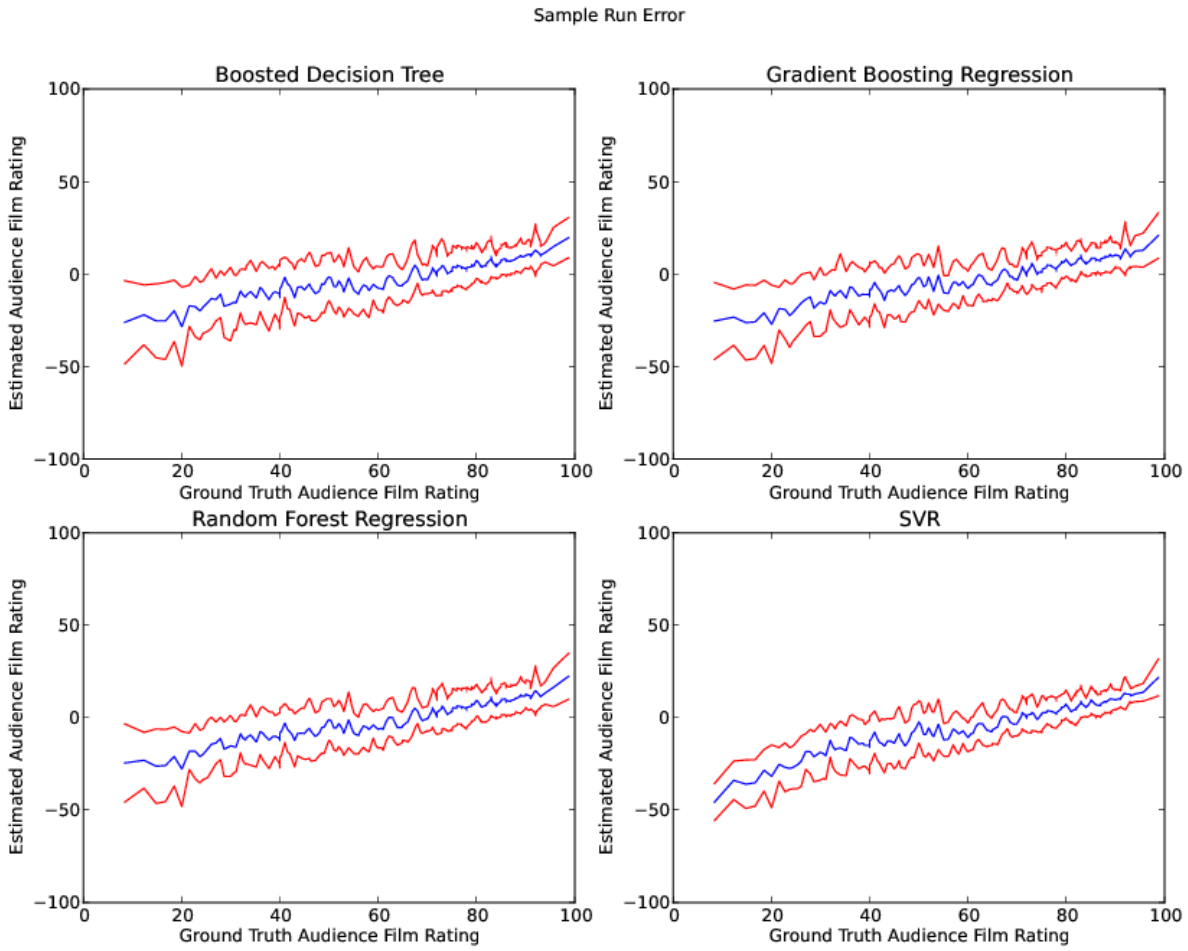


Fig. 2. Smoothed Average Regression Error Over 100 iterations

for classification such as SVMs consider each feature equally.

A. *Quality of Data*

A major challenge with this project was amassing a data set of primitive film features using on-line APIs. The results indicate that films with low audience ratings may have had little-known actors or directors, and thus had significantly noisier predictions. Another possibility is that because the data set included films that were made before Rotten Tomatoes was established, those films may not have amassed a significant amount of ratings. In general, a higher quality data set may have allowed for more robust simulations and performance.

B. *Future Work*

In addition to performing simulations on a more representative data set, the results of this project can be improved by developing more robust methods of converting symbolic features into numerical values. For example, adding the feature of release date of a film, and accounting for the time variance of film professionals' ratings may have significantly reduced the regression error. Many film professionals, especially directors, take part in numerous low rated films before becoming successful, and such a model would account for that time variance.

VII. Conclusion

This project used regression methods to predict audience film ratings based on results of simple techniques

for converting symbolic film features into numerical values, achieving a median regression error under 8%. The findings of this work may help producers make better choices before investing in the filming process, and may improve the average quality of films.

ACKNOWLEDGMENTS

The authors would like to thank Professor Sam Keene and fellow students of the Cooper Union EE class of 2014 for their support and advice on this project.

REFERENCES

[1] H-T Thorsten, H. Mark, W. Gianfranco, "Determinants of Motion Picture Box Office and Profitability: An Interrelationship Approach," *Review of Managerial Science*, 2006.

[2] P. Reggie, C. Andrea, "Quantifying Movie Magic with Google Search," Google Whitepaper - Industry Perspectives + User Insights.

[3] L. Barry, "Predicting Success of Theatrical Movies: An Empirical Study," *Journal of Popular Culture*, 2004.

[4] X Rui, C. Yanlin, "Box Office Prediction for Upcoming Films," 2011, Stanford University Final Projects.

[5] The Sagal Index of The 1,000 Best Films of All Time. [Online]. Available: http://www.sagal.com/bestfilmsindex.htm

[6] The Sagal Index of The 1,000 Worst Films of All Time. [Online]. Available: http://www.sagal.com/worstfilmsindex.htm

[7] the Biggest Box Office Flops in Movie History! [Online]. Available: http://www.sagal.com/bestfilmsindex.htm

[8] The Movie Database API. [Online]. Available: http://docs.themoviedb.apiary.io/

[9] Rotten Tomatoes API Documentation. [Online]. Available: http://developer.rottentomatoes.com/docs

[10] scikit-learn User Guide. [Online]. Available: http://scikit-learn.org/stable/user_guide.html