Predicting movie ratings

Carolyn Au *auc@stanford.edu*, Justin Cunningham *jcnnghm@stanford.edu*,
and Weixiong Zheng *zhengwx@stanford.edu*

***Abstract*—The project aims to explore machine learning methods to predict a movie's critical success prior to it's release, i.e. based solely on metadata available about the movie. The code for this project can be found at `https://github.com/jcnnghm/cs221-project`**

I. Introduction

The movie industry generates multi-billion dollars in revenue and most movies cost millions of dollars to create. However, not all movies are successes. With such high risk and large amounts of money involved, it would be useful to be able to predict the success of a movie before it is released. In this project, we try to predict the critical success of a movie, as shown by user ratings on IMDb, based solely on metadata about a movie available prior to it's release.

II. Related Work

Henning-Thurau, Houston and Walsh [1] performed an empirical study to distinguish direct and indirect relationships among different determinants of movie success and found that star and director power does not guarantee success. However, cultural familiarity (e.g. sequels to a successful movie), release dates (e.g. during the summer), budget and awards were a huge predictor of success. Relatedly, Deniz and Hasbrouck [2] performed statistical analysis on the top 150 grossing movies of 2010 and found that genre, MPAA rating, budget, star power, adaptation from another medium, sequels and remakes are significant predictors of box office revenue.

There are also other student projects that attempted to predict user ratings. One was done for the Machine Learning class (CS229) [3], where they used Naive Bayes and Support Vector Machines to predict IMDb user ratings and profitability. Their system performed moderately well on their test data. Another similar project [4] used regression (Support Vector Regression, Boosted Decision Trees, Gradient Boosting Regression and Random Forest Regression) over a different dataset (The Sagel Index of the top and worst 1000 films), predicting audience ratings on Rotten Tomatoes. Their system resulted in error rates of roughly 10%.

III. Methodology

A. Dataset

Our dataset is the list of all movies from IMDb that fulfill the following properties

- Released in the US
- Generated gross earnings in the US
- Has at least 1,000 user votes to rate the movie

After pruning the database of 3 million entries, we are left with 9888 movies which is a reasonable number for our algorithm sot run on. Limiting the data to movies with a reasonable number of user votes also ensures that the rating data is not too noisy. We save 20% of the data for testing, and use the rest for development.

B. Concrete Example of Input and Output

An abbreviated concrete example input to our algorithm for the movie *Winnie the Pooh* is:

```
”3014657” : {
    ”rating” : 7.3,
    ”features” : {
        ”based − on − children′s − book” : 1,
        ”disney − animated − feature” : 1,
        ”owl” : 1,
        ”Lasseter, John_producer” : 1,
        ”Luckey, Bud_actor” : 1,
        ”character − name − in − title” : 1,
        ”Klein, Sebastian_actor” : 1,
        ”sequel” : 1,
        ”genere_Family” : 1,
        ”Cummings, Jim_actor” : 1,
        ”rabbit” : 1,
        ”genere_Comedy” : 1,
        ”friendship” : 1,
        ”genere_Animation” : 1,
        ”tiger” : 1,
        ”Mitchell, Nicole_writer” : 1,
    },
}
```

The output is simply the predicted rating, in this case 7.324, after training with the features from the input.

C. Implementation

We started by importing the IMDb dataset [5] into MySQL to allow us to quickly explore the dataset and to speed up development. We built a framework to make it relatively easy to build feature extractors, whose output could be merged together and fed as input into our algorithm. We also cross referenced our data set with critic and audience ratings from

Rotten Tomatoes using their API [6].

We then created feature extractors for the metadata available in the dataset. In order to reduce the feature vector size, we filter out

- actors with at least 2 movie appearances
- keywords used to describe at least 10 movies

Additionally, since our features have wildly different ranges (some are counts while others are dollar amounts), we perform feature normalization so that each feature contributes equally to the final result.

For both the baseline and oracle, we used Stochastic Gradient Descent using Linear Regression to predict the user rating of a movie on IMDb (on a scale of 0 - 10) with 100 iterations and η of 0.001.

D. Evaluation Metric

We used the standard deviation of the predicted values from the actual values, calculated as

$$\sqrt{\frac{\sum_{i=1}^{n}(\text{Predicted}_i - \text{Actual}_i)^2}{n}}$$

IV. Preliminary results

A. Baseline

For our baseline, we included the following features:

- Complete cast members, which includes actors, directors, producers, etc.
- Movie genre
- Keywords describing the movie

This results in a sample standard deviation of 0.06 on the training data, and 1.78 on the test data.

B. Oracle

For our oracle, we added additional features which are movie data available *after* a movie is released. These are:

- Gross earnings
- IMDb user ratings
- Number of votes for the ratings

This results in a sample standard deviation of of 0.03 on the training data, and 0.34 on the test data.

We had initially added critic and audience ratings from Rotten Tomatoes, but it produced unexpected results, namely critic scores were negative weights and negative audience ratings had the opposite effect (movies rated *Spilled* had larger weights than *Upright*)

V. Next Steps

A. Improve Features

We will extract more features from IMDB database such as budget, release month, etc. We will also try to construct more advanced features from current features such as combinations of actor and actress, combinations of director and screenwriter, etc. This step will involve a lot of feature tuning and experimentation.

B. Improve Models

We will try different Machine Learning models to identify the best model for this problem. Models we plan to experiment with include Multiclass Classification, Neural Network, Support Vector Machine, etc. This step will involve a lot of error analysis and tuning on models and algorithms.

VI. Challenges

It would be interesting to see if different markets behave differently. Our dataset only includes movies released in the US; how would our system change if data from other markets were included in both the training and test detasets? Our dataset is also relatively small; would adding more data improve our error rates? Finally, how can we model cultural familiarity which is known to be a major predictor of success [1]?

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] B. Deniz and R. B. Hasbrouck, *WHEN TO GREENLIGHT: Examining the Pre-release Factors that Determine Future Box Office Success of a Movie in the United States* International Journal of Economics and Management Sciences, 2012.

[3] D. Cocuzzo and S. Wu, *Hit or Flop: Box Office Prediction for Feature Films* Stanford CS 229 project, Dec 2013.

[4] S. Mevawala and S. Phadke, *BoxOffice: Machine Learning Methods for Predicting Audience Film Ratings* The Cooper Union for the Advancement of Science and Art.

[5] `http://www.imdb.com/interfaces`

[6] `http://developer.rottentomatoes.com`