Review's take on Revenue

Nikhil Adhe, Deeksha Razdan

Abstract

In this project we predict the movie revenue (total revenue and per screen revenue) for the opening weekend. We try to focus on textual features that we can mine from the review rather than the metadata features. We used movie reviews and applied various Natural Language Processing techniques (vocabulary, stemming, lemmatizing) to extract the features from the movie reviews in the training set and then use them predict the revenue. This is done by training models on those features and mapping them to the revenue in the training set. We achieved a substantially better error margin than the baseline model without using metadata features.

Introduction

In this project, we present an approach for predicting gross revenue for movies. It is a problem that has been studied thoroughly in marketing, advertising, statistics etc. An estimate beforehand can help in creating better advertising strategies for Indie movies that aren't well embedded in the minds of the people. This is a complex problem, it may not always be a direct relationship between a review and the revenue, since some movies fare well even after bad reviews, and movies fail at the box office even after a splendid review.

We try to find the kind of influence a review by a critic has on movie goers, and hence the revenue.

We extracted features based on vocabulary of the training set, types of positive words, negative words, their parts of speech tags, and polarity of a review. The prediction is made by using various models (decision trees, ridge regression, neural networks) which are applied on those textual features. Among them, the best model- neural networks is chosen for final prediction.

Literature Survey

We have referred multiple papers, of which we provide a short review.

In this paper [Joshi et al] [1], used the reviews of film critics from several sources to predict opening weekend revenue. They created a dataset, where they collected movie reviews with metadata and revenue data. They concluded by showing that review text can substitute for metadata as a feature and can even improve over it for prediction. They used linear regression (elastic net) model over the textual and metadata features to predict the revenue.

In this paper [Ghose et al] [2], the metric polarity (of an opinion) is measured using the economic context in which the opinion is evaluated. In effect, the polarity is being

calculated this way: the polarity of each word in a text/review that has a low economic value (suggesting a negative impact) will have a negative score. It is called the economic value of text.

This paper [Nasukawa et al] [3] conducts a sentiment analysis approach to extract sentiments associated with polarities of positive or negative context for specific subjects from a document, instead of classifying the whole document into positive or negative. The essential issues in sentiment analysis are to identify how sentiments are expressed in texts and whether the expressions indicate positive (favorable) or negative (unfavorable) opinions toward the subject.

In this study [Poria et al] [4], the first deep learning method to extract 'aspects' in opinion mining was presented. Aspect extraction is a subtask of detecting the specific aspects of a product or service an opinion holder is either praising or complaining about. A 7-layer deep convolutional neural network was used to tag each word in opinionated sentences as either aspect or non-aspect word.

Here [Mishne et al] [5], sentiment analysis methods are applied to weblog data in the domain of movies since the extent of discussion about a movie or a product in weblogs has been shown to have a correlation with the movie's revenue or financial performance.

In this study [Sharda et al] [6], the use of neural networks in predicting the financial performance of a movie at the box-office before its theatrical release is explored. In their model, the regression problem is converted into a classification problem (ranging from a 'flop' to a 'blockbuster.') Their comparison of neural network to other models and statistical techniques show that neural networks are better predictors for such a dataset.

A new approach [Zufryden et al] [7] to assess the financial success of films resulting from advertising is explored in this paper. The model establishes a correlation among spending on advertisements, public awareness, the intention to watch the movie and expected ticket sales at the box office.

This study [Elberse et al] [8], the authors use data of a movie's stock price as it trades on the Hollywood Stock Exchange (a popular online market simulation), to study the impact of movie advertising on its revenue. They discover that advertising has a positive and significant influence on expected revenues, but that the influence varies strongly across movies of different quality.

In this paper [Eliashberg et al] [9], influencer perspective suggests that critics do influence box office results, and that positive reviews should encourage early rise in the revenue as opposed to negative reviews by the critics. Although, their findings showed a lack of relation between the influence and the revenue.

Here [Sawhney et al] [10], the authors find that the box-office revenue displays patterns. They find that there are only three classes of adoption patterns, represented by using a two-parameter - Exponential or Erlang-2 probability distribution, or a three parameter Generalized Gamma distribution. They also find that cumulative box-office revenues can be predicted with reasonable

accuracy often within 10% of the actual revenue using as little as two or three data points.

In this paper [Zhang et al] [11], the authors used the quantitative news data generated by Lydia, (system for large-scale news analysis), to help people to predict movie revenues. They used two different models (Regression and k-Nearest Neighbors). Furthermore, they claim that better results can be achieved by using the combination of IMDB data and news data.

Dataset Information

The data set has been taken from - http://www.cs.cmu.edu/~ark/movie$-data/ [1] which was formed by Joshi et al of the Carnegie Mellon University. This dataset consists of information for 1718 movies released in 2005 – 2009 and these are divided according to table no. 1. The training set consists of movies released between 2005 – 2007, the development set consists of movies released in 2008 and the test set consists of the movies released in 2009.

Each movie has metadata like name of the movie, its production house, genre, the scriptwriter, director(s), the country of origin, the primary actors and actresses starring in the movie, the release date, its MPAA rating, its running time and financial information.

The financial success of the movie has been taken in 2 forms: the opening weekend per screen revenue and the total gross revenue across all the screens.

Moreover, each movie has at least one review that has been scraped from one of the seven movie review sites- Austin Chronicle (www.austinchronicle.com), Boston Globe (www.boston.com), LA Times (www.calendarlive.com), Entertainment Weekly (www.ew.com),New York Times(www.nytimes.com), Variety (www.variety.com) and Village Voice (www.villagevoice.com). The dataset consists of only those reviews that appeared on or before the movie release date to ensure that the revenue information was not present in the review.

Dataset	Size
Train	1147 movies
Development	317 movies
Test	254 movies

Table 1: Dataset type and its size

Method

The aim of this project is construct a model that accurately predicts the revenue a movie might garner based on its reviews. This involves two steps:

1. Extracting features from the reviews for all the movies.
2. Applying suitable regression models that map these features to the two kinds of revenues for each movie – per screen revenue and the gross revenue for the opening weekend.

There are various ways to design suitable features from the reviews that might be suitable for predicting revenues. Some features are more powerful in their

predictive capability than others. We used four types of features:

1. Features based on the vocabulary of the concatenated reviews.
2. Features based on the number of positive or negative words in the reviews.
3. Features based on the polarity and the strength of 'likeness' or 'dis-likeness' of the movie.
4. Features based on the count of parts of speech tags given in a review.

After extracting features, applying a suitable regression model is also an important part of the process. It is possible that a model is more suited to map a feature to the per screen revenue or the total revenue than other types of features. We tried different linear models like Ridge regression, Lasso regression and non – linear models like decision trees and neural networks.

After obtaining the predicted revenues, the accuracy is judged based on the Mean Absolute Error (MAE) in US dollars. The model with the lowest MAE is chosen as the best model.

Different combinations of the types of features and the regression models were trained on the training set of 1,147 movies. The hyper-parameters for each regression model were adjusted by applying the trained models to the validation set of 317 movies. From all such experiments the regression model and the best type of feature with the lowest Mean Absolute Error (MAE) was chosen and applied to the test set of 254 movies.

Results

To establish a baseline, we concatenated all the reviews for a particular movie and constructed a feature vector for that movie based on the vocabulary of the concatenated reviews. We adopted three methods to refine the vocabulary of the training set –

1. Using vocabulary 'as is' *(F)*

2. Eliminating stop words (enumerated below) from the vocabulary. *(FS)*

3. Eliminating *stop words* and words having counts less than 5. *(FSI)*

The weights of the feature vector were the unigram counts of the words that appear in the three types of vocabulary as described earlier.

Stop word list (25 words) = a, an, and, are, as, at, be, by, for, from, has, he, in, is, it, its, of, on, that, the, to, was, were, will, with.

The modified size of the vocabulary can be seen in Table no 2. Hence the lengths of the feature vectors would be 186582, 186557, 41478 for the three types of vocabularies.

Vocabulary Type	**Size of every vocab set (unique words)**
F: Full vocab	186,582
FS: Full vocab - Stop words (25)	186,557
FSI: Full vocab - Stop words - Infrequent words	41,478

Table 2: Vocabulary type and its size

Once the feature vectors as above were obtained, we used them to train various models like Ridge regression, Lasso regression and decision trees in the regression task of mapping these feature vectors to the per screen and the total revenue in the training set. Once the models were trained, we used them to predict the per screen and the total revenue for the movies in the development set the results or which were obtained as below:

Vocabulary Type	Decision Tree (max depth- 10)
F	$ 9.30 M
FS	$ 9.04 M
FSI	$ 9.46 M

Table 3: MAE for total revenue using decision tree

Vocabulary Type	Ridge Regression
F	$ 12.40 M
FS	$ 12.15 M
FSI	$ 12.15 M

Table 4: MAE for total revenue using ridge regression

Vocabulary Type	Lasso Regression
F	$ 11.78 M
FS	$ 11.78 M
FSI	$ 13.20 M

Table 5: MAE for total revenue using lasso regression

Vocabulary Type	Decision Tree (max depth- 10)
F	$ 6,743
FS	$ 6,743
FSI	$ 7,431

Table 6: MAE for per screen revenue using lasso regression

Vocabulary Type	Ridge Regression
F	$ 12,286
FS	$ 12,178
FSI	$12,179

Table 7: MAE for per screen revenue using ridge regression

Vocabulary Type	Lasso Regression
F	$ 8,980
FS	$ 8,987
FSI	$ 11,071

Table 8: MAE for per screen revenue using lasso regression

As seen from tables 3 and 6, the Decision Tree regressor gave the lowest Mean Absolute Error (MAE) on both the per screen revenue ($6,743) and the total revenue ($ 9.04 million) applied on the feature vector obtained by eliminating the stop words of the vocabulary of all the reviews was chosen as the baseline model for our experiments.

A general observation was that by removing infrequent words (i.e. using the FSI vocabulary) resulted in an inferior performance. This could be because by removing those words, the size of the

feature vector reduces drastically from 186,557 to 41,478. It could mean that by removing so many words, some essential features are lost leading to inferior predictions.

Once the baseline model was established, we explored additional features which could be extracted from the movie reviews and that could serve as good predictors of the per screen and the total revenue. Three new features were extracted and the existing feature using vocabulary was refined. The features were:

1. The vocabulary obtained by concatenating all the movie reviews was refined by tokenizer in the NLTK library and the first feature vector was constructed. The decision tree model which was selected as the baseline earlier was applied to the new feature vector and the Mean Absolute Error on the development set for the per screen revenue and the total revenue were obtained as below.

Vocabulary Type	Decision Tree (max depth- 10)
F	$ 9.26 M
FS	$ 9.16 M
FSI	$ 8.91 M

Table 9: MAE for total revenue using decision tree

Vocabulary Type	Decision Tree (max depth- 10)
F	$ 7,692
FS	$ 7,774
FSI	$ 6,082

Table 10: MAE for per screen revenue using decision tree

To ensure that similar words are weighed appropriately in the feature vector, all the words were lemmatized using the lemmatizer in the NLTK library according to the part of speech it belonged to. This is necessary because the default lemmatizer lemmatizes words like 'loving' to 'loving' and not 'love' because it assumes that the word by default is a noun. Hence it is essential to pass the word along with its part of speech to the NLTK lemmatizer. The size of the vocabulary obtained was as follows:

Vocabulary Type	Size of every vocab set (unique words)
F: Full vocab	56,606
FS: Full vocab - Stop words (25)	56,468
FSI: Full vocab - Stop words - Infrequent words	24,861

Table 11: Vocabulary type and its size after lemmatization

On comparing tables 2 and 10, It is interesting to note that the size of the vocabulary has reduced substantially on lemmatizing all the words in the original raw vocabulary. For example, the size of the full vocabulary has reduced from 186,582 words to 56,606 words. Although the number of features has reduced substantially, it is expected to improve the performance of the predictors because words belonging to the same root would be weighed appropriately and hence the noise in the feature vectors would be reduced resulting in better predictive capability. On applying the decision tree model to this feature vector, the following results were obtained:

Vocabulary Type	Decision Tree (max depth- 10)
F	$ 8.71 M
FS	$ 8.46 M
FSI	$ 8.34 M

Table 12: MAE for total revenue using lemmatization

Vocabulary Type	Decision Tree (max depth- 10)
F	$ 7,869
FS	$ 6,037
FSI	$ 6,028

Table 13: MAE for per screen revenue using lemmatization

On comparing these results with the results obtained in tables 9 and 10 we observed substantial improvement in the performance of the predictors on using lemmatized feature vectors.

Neural networks were also used to map the feature vectors to the total revenue and the per screen revenue. A fully connected neural network with 5 hidden layers with 300 neurons in each hidden layer was employed with ReLU non – linearity after each layer. The ADAM optimization method was used with the learning rate set to 0.01 and the maximum number of epochs as 200 with the training being stopped when the loss was not improving over previous iterations. The following results were obtained for the per screen revenue and the total revenue on the development set:

Vocabulary Type	Neural Network (5 hidden layers)
F	$ 6.08 M
FS	$ 6.09 M
FSI	$ 6.10 M

Table 14: MAE for total revenue using lemmatization

Vocabulary Type	Neural Network (5 hidden layers)
F	$ 5,667
FS	$ 5,664
FSI	$ 5,635

Table 15: MAE for per screen revenue using lemmatization

2. The second type of feature that we considered was based on the number

of positive and negative words in the reviews for a movie. All reviews would have a set of both positive and negative words, but the favorable reviews would tend to have more positive words and the unfavorable reviews would tend to have more negative words. It is generally expected that movies with favorable reviews would perform well at the box office. This would make it more likely that feature vectors which are constructed based on the word count of the positive and the negative words in movie reviews would be able to incorporate the degree of favorability of a movie and map it to its per screen revenue and the total revenue as well.

Model	MAE
Ridge Regression	$ 9.70 M
Neural Network	$ 7.23 M

Table 16: MAE for total revenue using Feature 2

A list [13] of 2,040 positive and 4,817 negative words consisting of a total of 6,857 words was used. Since lemmatization had proved to be an effective technique earlier, the word list was lemmatized which gave a word list having a total count of 5,866 words. For each movie, all the available reviews were concatenated into a single review. Now every count of a positive word was assigned a weight of +1 and every count of a negative word was assigned a weight of -1 in the feature vector. If a particular word was present in the word list but absent in the review, the corresponding element in the feature vector was assigned a 0 value. Hence every movie had a feature vector that had a length of 5,866 elements. Once the feature vector was constructed, the following results were obtained on the Ridge regressor and the neural network with the previous configuration:

Model	MAE
Ridge Regression	$ 7,866
Neural Network	$ 5,663

Table 17: MAE for per screen revenue using Feature 2

3. The third type of the feature was based on the polarity of the review. As done previously, we stitched together all the reviews for a particular movie together into a single review and then constructed a feature vector for the concatenated review. For this we used NLTK sentiment analyzer package that takes the text as a whole and gives the extent to which the text (in our case the review) is positively, negatively inclined, or maybe it's neutral and assigns a score for each inclination accordingly. It also outputs a compound score, that

summarizes these three different scores. We used the compound score as our feature. The maximum score can be 1, which means that the text/review is highly positive, the minimum score can be -1, which suggests that the text/review is highly negative, or the score can be 0, which means that the nature of the text is not negative nor positive, it is neutral. The feature vector constructed in this way was mapped to the per screen revenue and the otal revenue using Ridge regression and the neural network and the following results were obtained:

Model	MAE
Ridge Regression	$ 6,433
Neural Network	$ 8,967

Table 19: MAE for per screen revenue using Feature 3

4. The fourth type of feature that was used was based on the Part of Speech tags of the words in the movie review. First all the reviews for every movie were concatenated into a single review. Next every word in this concatenated review was assigned its Part of Speech tag using the NLTK Part of Speech tagger. There are 36 possible tags which can be assigned. Now the feature vector was constructed based on the count of words having a particular tag. Hence every movie had a feature vector having a length of 36 elements, the weights of which were the count of Part of Speech tags of the words in the movie review. The intuition behind this feature vector is to see if a particular part of speech like adjectives (positive and negative) has more weightage in determining the revenue of a movie. This feature vector was used to train a Ridge Regression model and a neural network on the training set per screen revenue and the total revenue. The following results were obtained on the development set :

Model	MAE
Ridge Regression	$ 10.60 M
Neural Network	$ 12.80 M

Table 18: MAE for total revenue using Feature 3

Model	MAE
Ridge Regression	$ 8.40 M
Neural Network	$ 9.80 M

Table 20: MAE for per screen revenue using Feature 4

Model	MAE
Ridge Regression	$ 5,953
Neural Network	$ 5,724

Table 21: MAE for per screen revenue using Feature 4

It can be seen from the above experiments that Neural Networks was the best Regression models for predicting both the total revenue and the per screen revenue. Although neural networks seem to work for both, the type of feature vector that works the best is different for predicting the total revenue and the per screen revenue. For the total revenue the full vocabulary(F) with the NLTK tokenizer is the best feature.

Hence neural networks with the feature vector formed by using the full lemmatized vocabulary for predicting the total revenue and neural networks with the feature vector formed by removing the stop words and infrequent words from the full lemmatized vocabulary for predicting the per screen revenue were applied to the movies in the test set and the following were the results.

Model	MAE
Neural Network	$ 6,401

Table 22: MAE for per screen revenue on the test set

Model	MAE
Neural Network	$ 7.11 M

Table 23: MAE for total revenue on the test set

Discussion and Future Work

It can be observed that feature 1 (F) is the best for predicting total revenue while feature 1 (FSI) is the best for predicting per screen revenue. Neural networks are better at modeling complex features, such as feature type 1, where the feature vector lengths are 56606, 56468 and 24861. Neural networks vastly outperform linear models. Linear models are better for feature based on polarity due to their ability to capture the linear-ness in the feature. We conclude that text based features are better at predicting total revenue than meta data based features (proposed by [1]) Our text based features beat the results (Mahesh et al[1]) of text based features for total revenue.

Since in this dataset, we have multiple reviews for a given movie. Reviews from movie critics of different sources such as NY Times, The Chronicle, etcetera; it would be interesting to find out which source has the most influence, if at all. For this we need to gather a dataset in which we have a review from all the different sources for every movie. Also, models can be trained on hybrid feature vectors made by combining the different types of feature vectors extracted presently.

References

[1] Mahesh Joshi, Dipanjan Das Kevin Gimpel Noah A. Smith, 'Movie Reviews and Revenues: An Experiment in Text Regression'

[2] Ghose, P. G. Ipeirotis, and A. Sundararajan. 2007. Opinion mining using econometrics: A case study on reputation systems. In Proc. of ACL.

[3] Tetsuya Nasukawa, Jeonghee Yi, " Sentiment Analysis: Capturing Favorability Using Natural Language Processing," 2nd international conference on Knowledge capture 2003, pp 70-77

[4] Poria S., Cambria E., & Gelbukh A. (2016), "Aspect extraction for opinion mining with a deep convolutional neural network," Knowledge-Based Systems, 108, 42-49.

[5] Mishne and N. Glance. 2006. Predicting movie sales from blogger sentiment. In AAAI Spring Symposium on Computational Approaches to Analysing Weblogs.

[6] R. Sharda and D. Delen. 2006. Predicting box office success of motion pictures with neural networks. Expert Systems with Applications, 30(2):243–254.

[7] Zufryden, F. S. (1996). Linking advertising to box office performance of new film releases: A marketing planning model. Journal of Advertising Research, July–August. 29–41.

[8] Anita Elberse, Bharat Anand, "The effectiveness of pre-release advertising for motion pictures: An empirical investigation using a simulated market." Information Economics and Policy 19 (2007) 319–343

[9] Eliashberg, Jehoshua, Shugan, Steven M., 1997. Film critics: Influencers or predictors? Journal of Marketing 61 (April) 68–78.

[10] Sawhney, M.S., Eliashberg, J., 1996. A parsimonious model for forecasting gross box-office revenues of motion pictures. Marketing Science 15 (2), 113–131.

[11] W. Zhang and S. Skiena. 2009. Improving movie gross prediction through news analysis. In Proc. of Web Intelligence and Intelligent Agent Technology.

[12] S.-M. Kim and E. Hovy. 2004. Determining the sentiment of opinions. In *COLING 2004*, pages 1367–1373.

[13] William Gunn, "SciSentiment", positive-negative word list, github, 2011