Forecasting Hollywood: Can Movie Revenues Be Predicted?

Muntasir Chowdhury
School of Computer Science
McGill University
260558036
Email: muntasir.chodhury@mail.mcgill.ca

Kashif Javed
School of Computer Science
McGill University
260535684
Email: kashif.javed@mail.mcgill.ca

Faiz Khan
School of Computer Science
McGill University
260435930
Email: faiz.khan@mail.mcgill.ca

We explored the potential of predicting movie gross revenues using the average ratings of the main cast/crew involved, budget size and the season it was released in. We created a data set by parsing and combining movie data from The-Numbers and IMDB. We created maps to take the cast and crew to a real value indicative of the average rating that they achieved over all their movies. Using this we created regression predictors that yield the revenue of the movie given our mappings. We tested four different types of regression: standard regression, gradient descent, ridge regression and lasso regression. We determined an optimal step size for gradient descent as being 0.00001 for 100,000 iterations and a k = 10 number of folds for validation. The standard regression had the best performance being able to predict movie budges with nearly 90% accuracy.

Our code and data is available at :

- https://github.com/dridon/aml1/
- https://github.com/dridon/aml1/tree/master/src/data/features

I. Introduction

A. Problem Description

Modern day movie budgets are now reaching near the half billion dollar mark. Estimates for the budget of James Cameron's Avatar reach up to $425,000,000[1]. As movie budgets grow, so do their required revenue to yield an acceptable profit for the investment size presenting an interesting problem. It is important to know the potential success of a movie in order to evaluate its investment potential. However, movies are complex products shaped by various commercial and artistic constraints, and require the involvement of a diverse group of people (cast and crew). In addition to these factors a movie's success is affected by marketing, public perception and critical ratings. This makes predicting the potential revenue for a potential movie a challenge. We describe this more formally as follows:

"*Produce a predictor using some feature set that allows us to accurately estimate the gross revenue for a potential movie.*"

In this document we will describe our attempt at tackling this challenge through several regression models. We apply our feature set constructed from parsing two major online movie databases.

II. Related Work

A. The Numbers Bankability Index

The "Hollywood Creative Graph" represents films using information about 80,000 people from the movie industry and attempts to measure each individual's influence on that graph by assigning a "bankability" value to them [2]. This "bankability index" is essentially aimed at understanding how much an individual should be paid for working on a film using box-office data by estimating how much value they bring in to the industry each year. One may be able to generate the potential revenues by using this data. However, this index is commercial and their methodology private.

B. Word Of Mouth

Dellarocas et al[4] leveraged data about a film's initial commercial performance, marketing campaign, and early public reaction (word of mouth) to forecast revenue. They used a data set of 80 movies with reviews from over 1,000 critics and nearly 35,000 individuals and fit it to a modified Bass equation. They divide the movie into two sets, fit on one and test on the other. They saw a remarkable success with an error of less than 3%.

In a similar vein Rui et al.[5] and Asur et al.[6] have tried to gauge the influence of "online word of mouth" on movie sales by analysing Twitter data. Rui et al collected 4,000,000 tweets for 63 movies between June 2009 and February 2010. They took in to consideration the authors and the number of followers and the intention of the tweets. However, they do no cross validation or estimation of true error and directly move on to inference from their fit.

Asur et al. used 24 movies with 2.89 million tweets and 1.2 million users. Their fit had an R^2 value of 0.973. However, they also provide no measure of true error.

In all cases of *Word Of Mouth*, their methodology is restricted to released or soon-to-be released movies (otherwise no reviews or tweets will be present) and can not be used for movies being considered, planned or in production. Granted that their data sets are large, none of the above considered movies greater than 80 in number. This is expected because the data collection tasks would be enormous and Twitter data is not available for a large number of movies that pre-date the

service. Nonetheless, considering that there is a large range of movies, testing over a small set raises concerns that the fits only performed well in local regions.

C. Neural Networks

Zhang et al[7] employ a BP neural-network to predict revenues. They bin movies in to 6 ordered-categories for classification targets. Each representing a different ranges of movie revenues. They had a maximum of 40 movies per category. Several variables were trained, some examples being nationality, advertising, content and showing time. They used 6-fold cross validation and were able to classify a movie correctly with 68.1% accuracy and within one class difference with 97.1% accuracy.

Their data is also limited by the number of movies they use to train. Also, their results represent the Chinese market and includes a mixture of Hollywood and Chinese movies. This data may not be fully reflective of Hollywood earnings.

D. Other

Dellarocas [4] describes the forecasting of motion picture revenues to be discernable in to two approaches. The first using econometric factors that predict motion picture revenues such as The Numbers bankability index. The second using factors that influence an individual's decision to watch a movie such as the Word Of Mouth approaches. There has been a large amount of work done in the prediction of movie revenues and a comprehensive review is out of the scope for this document. We refer interested readers to [8], [9], [10] and [11].

Most of these approaches use budget and revenue of a film as items in the dataset, none factor in the historical critical and audience ratings from aggregate movie siteThis suggests our space is linear when trying to predict revenue.

III. Dataset Description

The data set we use to train should be viewed in two parts. The first is the raw data set that is generated from mining, parsing, filtering and concatenation that is discussed further below. The second is the formatted movie set along the actors, directors, producers and screenwriter dictionaries.

The formatted movie set contains a data point for each movie in the raw set. For each movie we present a boolean indicating if they were released during a "hot season". We also present a measure of ratings by critics and audiences for the directors, producers, screen writers and main actors of the movie. Lastly, we have the budget of the movie and the target value, gross revenue for a movie. The dictionaries for the cast and crew show their average ratings over all their movies and are used for prediction.

See the appendix for more details.

IV. Methods

Our methodology is described in three main steps. First, we mine our data from two major movie sites and filter it. Then we apply four different regression algorithms with cross validation to get initial estimates on errors. Then we apply a set of experiments varying parameters in order to draw insights from our data.

A. Libraries

TABLE I
Python Libraries Used In Study

Library	Usage
Beautiful Soup 4[15]	parsing HTML pages
unicodecsv[16]	drop-in replacement for csv with unicode support
lxml[17]	xml parser for poorly formatted pages
numpy[18]	matrix operations
scipy[19]	extended operations such as geometric mean
scikit-learn[20]	implementations of ridge and lasso regression

A number of libraries were used in our implementations. We describe them briefly in Table I.

B. Data Collection

We extracted our data from two major movie sites: The Numbers [3] and IMDB[13]. The first is a collection of movie information with budget, revenue, cast, crew with all entries indexed. However, for a large number of movies the information is incomplete. The second one is a large site and community with extensive data but it has no simple movie index to parse data from. A crucial element of our raw data is taken from a third movie aggregation site called Rotten Tomatoes. Instead of parsing the relevant data from the site itself we collected it from The Numbers site.

We first used The Numbers to extract a list of movies, $\tilde{2}$0,000 entries, and filter the ones that have values for Budgets, Release Dates and Revenues, Critic Label/Rating and Audience Label/Rating (both ratings are from Rotten Tomatoes[14]). This gives us around 4,000 movies. We then use the movie names collected from The Numbers to search on IMDB for the main cast and crew of the movie alongside the IMDB audience rating. We then take the union of all the movies with data points for all the columns from The Numbers with the IMDB data to get our raw feature set. This raw set has a size of $\tilde{3}$,300 movies.

C. Feature Selection

Release Dates and Cast/Crew names are poor features for only 3,300 examples. Considering there are thousands of individuals amongst all considered people, we would end up having to estimate thousands of values with very little data if we go with something along the lines of Naive Bayes.

We thus create a rating function for all personnel involved in a movie. For each director, producer, screen writer and actor, we create a mapping for the person to the geometric mean of the ratings of all the movies that they had significant involvement with. Then for every movie we take the set of actors and take the average of the mean ratings for all actors involved in the movie. This gives us a measure on the rating of the movie for the set of actors involved. We repeat this for the directors, producers and screenwriters of the movie.

We produce a rating per person for the ”Critical” rating from Rotten Tomatoes, the ”Audience” rating from Rotten Tomatoes and the IMDB Rating. Thus, we have, for each movie, 3 measures for each: the set of directors, set of actors, set of producers and the set of screen writers giving us a total of 12 features.

We then label the summer months between May and August (inclusive) and the holiday season of November and December together to be the 6 months giving us the ”hot season”. We then convert the Release Dates to indicate a 1 for being released during the hot season and 0 otherwise.

Lastly, since movie budgets and revenues can range from thousands to hundreds of millions, we take the base 10 logarithms of each to compress the possible inputs and outputs. For some cases we had a few cases (less than 50) where there were $0 revenues. For these we used a revenue of $1, a small increment considering the rest are many orders of magnitudes higher, that helps smoothen out the transformation.

We are thus left with 14 features and a target value all in real numbers making regression an optimal choice.

D. Algorithms

We investigated four regression algorithms in our study: Standard Linear Regression, Gradient Descent, Ridge Regression and Lasso Regression.

1) Standard Linear Regression: We apply the standard form of linear regression with the least-squared error in closed form. We initially investigate a 10-fold validation and present the results for each of the test sets along with the mean squared error. We then range k from 10 to 33, bounding the size of the test set below by atleast 100 movies, and present the mean squared error for each k.

2) Gradient Descent: We apply gradient descent using the least squared error. We first vary alpha and use cross validation error to obtain an estimate for an optimal alpha with 100,000 iterations. The alphas vary from 10^{-5} to 10^{-8} with one order of magnitude increments. We then use that alpha to determine an optimal iteration. After that we use the optimal alpha and iteration values in the general analysis of our predictability. We vary iterations from 10^{1} to 10^{5} increments, simply because execution time becomes unfeasible. We also use these values to train the predictor for the full feature set. We use 10,000 iterations for varying the k parameter because the processing time was not available.

3) Ridge and Lasso: We apply the ridge and lasso from the scikit-learn library. We repeat the experiments done for standard regression on ridge and lasso and present their results.

4) Cross Validation: In order to test on a prediction we need a method to map a given set of actors, directors or producers to their average rating. We cannot create this prior to cross validation because we will be using the test set and potentially corrupting our data. Thus we have to generate these mappings from an individual to rating each time for during each fold in cross validation and essentially regenerate the entire feature set. In the case that an actor, director, producer or screen writer is not present in a mapping, we simply assign them a 50% rating for Rotten Tomatoes and 5.0 for IMDB. That is we do not assume any information for missing cases.

The cross validation scheme is done as follows:

1) The raw feature set, R, is loaded as a list of movie features for each movie
2) The movies are shuffled around in R to get R’
3) R’ is then divided in to k folds
4) For each fold f in the k fold we merge all but fold f in to f’
5) We train on f’, to get predictor P and generate mappings from an individual to their average rating call M
6) We use M to convert the f fold into input parameters and test the predictions against the actual values
7) Then we take the mean of all the errors from the k runs to estimate the true error

V. Results

We present the results of our experiments in this section and expand on them in the next. It should be noted that all errors are mean square errors of the log10 of the revenue. This means that a mean square error of one means that there is a one order of magnitude difference between the true estimate of revenue and our predicted value.

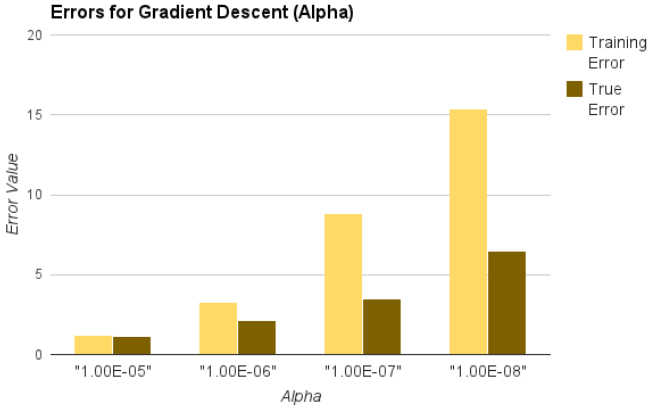

Fig. 1.

We first present the results for the determination of an optimal alpha for gradient descent in Figure **??**. The training error and true error estimate are shown with the variation of the step size. Similar results are in Figure **??** for the variation of the iteration size for gradient descent.

We determine the optimal value for alpha to be around 0.0001 with an iteration count of 100,000. We then use this to estimate gradient descent weights along with the other regressions under consideration. Table **??** shows the prediction weights using the entire data set for training. It should be noted, the weights in the cross validation would be slightly different.

The error for each k for each regression is shown in Figure **??**. This figure shows how the error changes as we increase

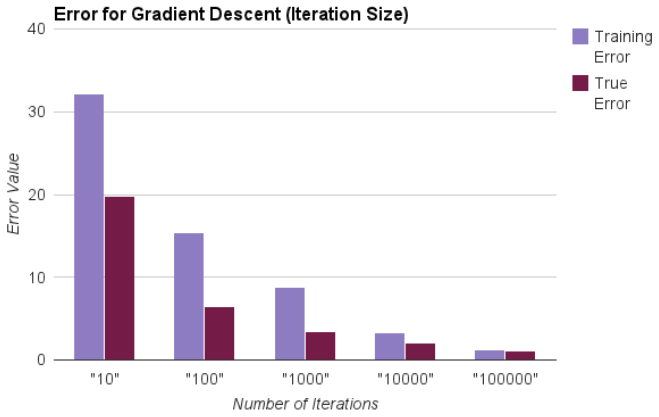

Fig. 2.

TABLE II
REGRESSION WEIGHTS FOR DIFFERENT PREDICTORS

	SR	GD	Ridge	Lasso
Hot Season	0.015	0.634	0.016	0.0000
Budget	1.081	0.393	1.080	0.0000
Director RT Critical Average	-0.005	-0.006	-0.005	0.0000
Director RT Public Average	0.006	0.004	0.006	0.0032
Director IMDB Average	0.043	0.147	0.043	0.0000
Producer RT Critical Average	-0.004	-0.008	-0.004	-0.0074
Producer RT Public Average	0.005	0.004	0.005	0.0010
Producer IMDB Average	0.129	0.184	0.128	0.0000
Screenwriter RT Critical Average	0.003	-0.001	0.003	0.0000
Screenwriter RT Public Average	0.010	0.002	0.010	0.0031
Screenwriter IMDB Average	-0.155	0.103	-0.154	0.0000
Actors RT Critical Average	0.003	-0.009	0.003	0.0000
Actors RT Public Average	-0.002	-0.006	-0.002	0.0000
Actors IMDB Average	-0.145	-0.102	-0.145	0.0003
Intercept	-0.178	-0.132	-0.178	0.0003

SR and GD stand for standard regression and gradient descent, respectively.

the value of k. Standard and Ridge regression were essentially the same thus their curves overlap on the figure. Noticing the k value does not seem to affect the error estimate after k = 10, we simply use it as our cross validation number. The results for true and training error for 10-fold validation are then presented in the histogram in Figure **??**.

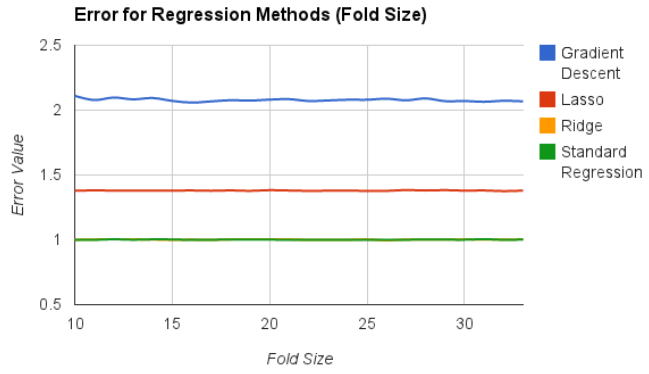

Fig. 3.

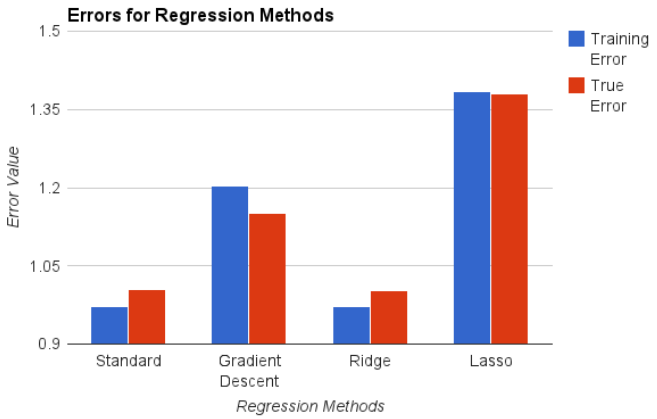

Fig. 4.

VI. DISCUSSION

Although our results seem to show that our predictor is fairly successful, there are some cases such as Gradient Descent that are alarming and raise questions. To discuss these further we now proceed to analyze the results presented in the previous section.

A. Analysis

In our analysis for gradient descent from Figure **??**, we notice that for an iteration count of 100,000 the error increases after 0.00001. This means that more iterations are needed to improve the error after this step size. However, iteration sizes of 100,000 and higher did not terminate in a feasible amount of time for us to perform experiments on, thus 0.00001 was optimal for our needs.

The iteration count however showed the opposite trend in Figure **??**. The higher the iteration count the better the error estimate proved to be. However, again, our resources were limited and we chose 100,000 for our experiments apart from the variation of the size of folds on which we used 10,000 as our iteration size.

Gradient descent proved problematic because it constantly showed a better test error over a training error in Figure **??**, Figure **??** and Figure **??**. We were not able to isolate the reason as to why this seems to be the case in only this regression. The data is always randomly shuffled before a validation and this phenomenon repeated multiple times thus we don't believe that the test error is performing better simply because of a local distribution. Other predictors did not show this issue thus we are still able to draw important insights from them.

The weights shown in Table **??** vary significantly depending on the regression task. Standard and Ridge Regression both produce the same results. Figure **??** also shows these to be the best estimators of our data with error values of around 1, which results in around a 90% success rate since its a log10 of the revenue. As one would expect the highest weight seems to be on the budget. Taking in to account the IMDB ratings being out of 10 and RottenTomato ratings are out of 100, it seems that the IMDB ratings still reflect the revenue a bit more

strongly then the RottenTomato ratings. Gradient Descent on the other hand places the most weight on the season the movie is released on. Lasso seems to suggest that the majority of the weights are small enough to be ignored.

It is also reflected that our k-fold validation did not change with the k parameter as shown in Figure **??**. This could be due to the size of our small data set. A fold size of k = 10 will produce a test set of size around 330 and k = 33 will produce a size around 100. This means the training set was, the same for the most part. We were unable to explore smaller sizes due to time constraints.

Overall the best performer is standard regression with an error of around 1 and the worst performer is Lasso with an error of around 1.4. Thus our best predictor shows an accuracy around 90%. Since our results seem to be successful over the majority of our experiments we also feel that the movie prediction space is linear in nature.

B. Applications

The most immediate application would be to implement the predictor as a tool that allows an user to choose the different parameters for a movie (budget, actors, directors) and get an estimate of the revenue. Training the tool on a larger database consisting of a more complete list of movies such as the entire IMDB website. Using this financial decisions may be made that include the initiation of a movie production, cancellation or budget change.

Even if a studio does not use the tool for the initial planning of a film it might turn out useful when it's trying to determine the final parameters of the film. For example, a scenario where all the cast and crew of a film have been hired and the lead actor still needs to be chosen. The studio may be able to generate a list of likely candidates using this predictor to determine the most optimal choices. The tool would go through the entire database of actors and list the ones whose involvement would result in the highest revenue.

C. Future Perspectives

Our initial data set of 20,000 films had to be drastically reduced when we aimed to discard incomplete entries. As a first step, a larger dataset of complete data points would give more insights about the model. More ratings for existing entries could be added from another major aggregate site called Metacritic. This could lead to more accurate results or give us an idea of which site's rating model is better.

Finally, another interesting prediction question would be to calculate a movie's potential critical rating or reception. Tweaking parameters for high critical success could be one application. But a more interesting one would be to have a model which tries to optimize both critical and commercial success simultaneously.

VII. Conclusion

We explored the potential of predicting movie gross revenues using the average ratings of the main cast/crew involved, budget size and the season it was released in. We tested four different types of regression: standard regression, gradient descent, ridge regression and lasso regression. We determined an optimal step size for gradient descent as being 0.00001 for 100,000 iterations and a k = 10 number of folds for validation. The largest discrepancy was with gradient descent that consistently showed a higher training error as compared to a test error. However, none of the other regressions presented this issue and results showed that standard regression had the best performance, being able to predict movie budges with nearly 90% accuracy.

Statement Of Original Work

We hereby state that all the work presented in this report is that of the authors.

References

[1] "Movie Budget and Financial Performance Records" (The-Numbers) [online], http://www.the-numbers.com/movie/budgets/ (Accessed:15 September 2014)

[2] "The Numbers Bankability Index" (The-Numbers) [online], http://www.the-numbers.com/bankability/ (Accessed:15 September 2014)

[3] "The-Numbers" (The-Numbers) [online], http://www.the-numbers.com/ (Accessed: 15 September 2014)

[4] C. Dellarocas, N. Awad and X. Zhang, 2004, "Exploring the Value of Online Reviews to Organizations: Implications for Revenue Forecasting and Planning", ICIS 2004 Proceedings

[5] H. Rui, Y. Liu and A. Whinston, 2013, "Whose and what chatter matters? The effect of tweets on movie sales", Decision Support Systems 55.4 (2013): 863-870

[6] S. Asur and B. Huberman, 2010, "Predicting the Future with Social Media", Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on. Vol. 1, pp 492 - 499

[7] L Zhang, J Luo and S Yang, 2009, "Forecasting box office revenue of movies with BP neural network." Expert Systems with Applications 36.3 (2009): 6580-6587

[8] B. Litman and L. Kohl, 1989, "Predicting financial success of motion pictures: The'80s experience." Journal of Media Economics 2.2 (1989), pp 35-50

[9] M Sawhney and J. Eliashberg, 2002, "The drivers of motion picture performance: the need to consider dynamics, endogeneity and simultaneity." proceedings of the Business and Economic Scholars Workshop in Motion Picture Industry Studies. Florida Atlantic University

[10] M Sawhney and J. Eliashberg, 1996, A Parsimonious Model for Forecasting Gross Box-Office Revenues of Motion Pictures, Marketing Science 15.2, pp 113-131.

[11] S. Sochay, "Predicting the performance of motion pictures." Journal of Media Economics 7.4 (1994), pp 1-20

[12] F. Zufryden, 1996, "Linking advertising to box office performance of new film releases: A marketing planning model." Journal of Advertising Research 36 (1996), pp 29-42

[13] "IMDb - Movies, TV and Celebrities" (IMDB) [online], http://www.imdb.com/ (Accessed: 15 September 2014)

[14] "Rotten Tomatoes" (Rotten Tomatoes) [online], http://www.rottentomatoes.com/ (Accessed: 15 September 2014)

[15] "Beautiful Soup" (Crummy) [online], http://www.crummy.com/software/BeautifulSoup/ (Accessed: 15 September 2014)

[16] "unicodecsv 0.9.4" (Python) [online], https://pypi.python.org/pypi/unicodecsv/0.9.4 (Accessed: 15 September 2014)

[17] "lxml - Processing XML and HTML with Python" (lxml) [online], http://lxml.de/ (Accessed: 15 September 2014)

[18] "NumPy" (NumPy) [online], http://www.numpy.org/ (Accessed: 15 September 2014)

[19] "SciPy" (SciPy) [online], http://www.scipy.org/ (Accessed: 15 September 2014)

[20] "scikit-learn: machine learning in Python" (scikit-learn) [online], http://scikit-learn.org/stable/ (Accessed: 15 September 2014)

VIII. Appendix

A. Data Dictionary

Initially we downloaded the indexed movie pages from The Numbers. After filtering out the entries that were missing too many relevant fields we queried the IMDB search engine for the remaining movies. We downloaded these pages and used the information there to add the IMDB score and fill in whatever fields were empty in the remaining movie list. The "full_raw_features.csv" contains the data parsed from The Numbers movie pages and the IMDB page. Each entry is a movie and contains the following information:

Movie Name, Release Date, Genre, Budget, Gross, RT Critic Label, RT Critic Rating, RT Audience Label, RT Audience Rating, IMDB Rating, Directors, Producers, Sreenwriters, and Actors.

RT Critic Rating - This is the "Critical Rating" given to a movie on Rotten Tomatoes. Rotten Tomatoes aggregates movie reviews from major critics and publications and gives them a quantitative value. It then takes the average of those values to produce the Critical Rating.

RT Audience Rating - This is the average of the ratings given to a movie by registered users on the Rotten Tomatoes website.

RT Critic Label and RT Audience Label - We do not end up using these in the final feature set.

IMDB Rating - This is the average of the ratings given to a movie by registered users on the IMDB website.

Director - The name of the director(s) of the movie.

Producers - Names of the studio that produced the movie.

Screenwriters - Names of all the screenwriters of the movie.

Actors - The first 3 names from the actor lists on The Numbers and the IMDB movie pages. These lists are in order of importance of the actor in the movie.

From this file we calculated the three kinds of ratings for every unique director, producer, screenwriter and actor. These are listed in four separate data files for each type of the four mentioned types.

Person's RTC Mean - The average of the Critical Ratings for all the movies the person is listed in. If a person has been in the role of both a director and an actor then her values for each role are independent and mutually exclusive.

Person's RTA Mean - The average of the Audience Ratings for all the movies the person is listed in.

Person's IMDB Mean - The average of the Critical Ratings for all the movies the person is listed in.

From these we form our final feature set, a version of which is outputted in "feature.csv". The format is outlined below.

The **Hot Season** is set to 1 if a film was released in the months of either May to August OR November-December. It is set to 0 otherwise. May to August is generally referred to as the summer blockbuster season when movie revenues are higher. This is due to schools being closed during that time and release of higher budget movies being concentrated around that period. November and December coincide with the holiday season and sees the release of award-season movies and critical favorites.

A crucial point to note is that this feature set is derived from the training data we have. If we change the number of data points for the training data then the values in the ratings file for director, producer, screenwriter and actor will change. As a result so will the average people ratings for movies in "full_raw_features.csv".

TABLE III
Feature Set

Feature	Description
Hot Season	This is 0 or 1 depending on the release date
Budget	The log of the movie's budget in dollars
Director RT Critical Average	Average of the RTC Mean for all directors involved
Director RT Public Average	Average of the RTA Mean for all directors involved
Director IMDB Average	Average of the IMDB Mean for all directors involved
Producer RT Critical Average	Average of the RTC Mean for all studios involved
Producer RT Public Average	Average of the RTA Mean for all studios involved
Producer IMDB Average	Average of the IMDB Mean for all studios involved
Screenwriter RT Critical Average	Average of the RTC Mean for all screenwriters involved
Screenwriter RT Public Average	Average of the RTA Mean for all screenwriters involved
Screenwriter IMDB Average	Average of the IMDB mean for all screenwriters involved (DT)
Actors RT Critical Average	Average of the RTC Mean for all actors involved
Actors RT Public Average	Average of the RTA Mean for all actors involved
Actors IMDB Average	Average of the IMDB Mean for all actors involved

TABLE IV
Output Set

Output	Description
Gross	The log of the movie's revenue in dollars