From Storyline to Box Office: A New Approach for Green-Lighting Movie Scripts

Jehoshua Eliashberg, Sam K. Hui, and Z. John Zhang[1]

The Wharton School

University of Pennsylvania

Philadelphia, PA 19096

1st Submission: Dec 21, 2005

Revised: August 18, 2006

[1] We thank the AE and the three anonymous reviewers for their valuable and insightful comments on our manuscript. We are, of course, responsible for the contents of this paper.

From Storyline to Box Office: A New Approach for Green-Lighting Movie Scripts

Abstract

Movie studios often have to choose among thousands of scripts to decide which ones to turn into movies. Despite the huge amount of money at stake, this process, known as "green-lighting" in the movie industry, is largely a guesswork based on experts' experience and intuitions. In this paper, we propose a new approach to help studios evaluate scripts which will then lead to more profitable green-lighting decisions. Our approach combines screenwriting domain knowledge, natural language processing techniques, and statistical learning methods to forecast a movie's return-on-investment based only on textual information available in movie scripts. We test our model in a holdout decision task to show that our model is able to improve a studio's gross return-on-investment significantly.

Key Words: Entertainment Industry, New Product Development, Forecasting, Contingency Data Analysis

1. Introduction

The motion picture industry is a very important industry worldwide. Many new products are developed and launched in this industry. More than 4,000 movies are produced worldwide each year (MPAA 2004). In the United States alone, around $9 billion is spent on theatrical tickets in 2004 (Eliashberg et al. 2005). While many of the movies are financed and produced by Hollywood major studios, recently, "a new wave of outsiders rushing to finance movies are to some extent changing the way films are produced" (Mehta 2006). The new players include wealthy financiers, private equity firms, hedge funds, and other institutions that invest in the early stage of movies' production. Their metric is return on investment (ROI).

Despite the market size and investment interests, new movie production is a risky venture. Profitability varies greatly across movies. While producers sometimes make large amount of profit from blockbusters, they also lose millions of dollars in movies that end up in oblivion. For example, the movie *Gigli* cost approximately $54 million to produce, but its box office revenue was only around $6 million. Considering that studios generally receive a share of around 55% of the gross box office revenue for their production (Vogel 2004; also at www.factbook.net/wbglobal_rev.htm), Gigli generated a ROI[2] of -96.7% for the studio. On the other end of the spectrum, although the movie *In the Bedroom* cost only $1.7 million to produce, it generated more than $35 million in box office revenues and thus a ROI of +667%. Across a sample of 281 movies produced between 2001 and 2004, the studio's ROI ranges from -96.7% to over 677%, with a median of -27.2%. As a result of the huge variance in ROI, the selection of which movies to produce is critical to the profitability of a movie studio.[3]

However, deciding which scripts to produce is a dauntingly difficult task, as the number of submissions always greatly exceeds the number of movies that can be made. Each year it has been estimated that more than 15,000 screenplays are registered with the Writers Guild of America, while only

[2] Defined as (0.55 * box office revenue – production budget) / production budget

[3] Of course, there are other ancillary sources (e.g. DVDs) from which studio's can generate revenue to finance movie production. However, in this paper we will focus on studio's return of investment as measured with respect to domestic (U.S.) box office revenue only. The approach we develop in this paper, however, can be readily extended to other markets.

around 700 movies are made in the U.S. (Eliashberg et al. 2005). Thus, studios need a reliable approach to guide the green-lighting process.

Currently, major studios still employ an age-old, labor intensive methodology: they hire "readers" to assist them in evaluating screenplays. Typically, three to four readers are assigned to read each script. After a reader reads a script, he / she writes a synopsis of the storyline and makes an initial recommendation on whether the screenplay should be produced into a movie and the changes, if any, that are needed before actual production. These recommendations are made mostly subjectively and by experience. This means that the success of a movie production depends on the quality of the available readers and their acumen in picking out promising scripts. This approach becomes especially problematic when disagreements among readers occur. Indeed, according to some industry insiders we talk to, on top of the disagreements among readers, studio executives, as well as managers at different levels in the development process, frequently also disagree for many reasons to make green-lighting an uncertain or sometimes even arbitrary process. Not surprisingly, the result of this process is highly unpredictable. Even the scripts for highly successful movies, such as *Star Wars* and *Raiders of the Lost Ark*, were initially bounced around at several studios before Twentieth Century Fox and Paramount, respectively, agreed to green-light them (Vogel 2004). For that reason, studios and movies financiers can potentially benefit from a more objective tool to aid their green-lighting processes and to provide a reliable "second opinion" about the potential success or failure of adopting a script.

No such tools, as far as we know, are currently available to aid screenplay screening. The main obstacle in developing such a tool has been the lack of reliable predictors for the financial success of a movie at the green-lighting stage: there are simply too few tangible determinants for the success of a movie before it is produced. In this paper, we propose a new and rigorous approach that can potentially help studios and movie investors screen scripts and make more profitable production decisions. To insure that our approach can help green-lighting decisions, we exclude factors such as promotion and advertising, number of screens, competitors, etc., even though they play a pivotal role in the success or failure of a movie, since this information becomes available only at a later stage. Our tool forecasts ROI based only

on the storyline. We extract textual information from them using domain knowledge from screenwriting and the bag-of-word model developed in Natural Language Processing. Once calibrated, these types of textual information are then used to predict the return-on-investment of a movie using Bag-CART (Bootstrap Aggregated Classification and Regression Tree) methodology developed in statistics (Breiman 1996; Breiman et al. 1984).

The rationale for our approach is simple. As industry insiders acknowledge, a good storyline is the foundation for a successful movie production. As Sir Ridley Scott, a famous director of motion pictures, once pointed out, "any great film is always driven by script, script, script" (Silver-Lasky 2003). Of course, what are in the script are the stories. Peter Gruber, the producer of Batman, also suggested the same: "At the end of the day when you get done with all the fancy production design…what's up on the screen is the script. Plain old-fashion words. It all starts there and it all ends there" (Silver-Lasky 2003). To the extent that the success of a movie depends on the stories told in a script, a sophisticated textual analysis of the scripts, or their proxies, that are already made into movies will help us identify the hidden "structures" in the texts, which essentially capture what the story is, how it is told, etc,. Then, by relating those structures with the subsequent financial return from the movie, we can learn what kinds of stories may resonate with audience and what elements in a story will drive ROI performance. Once those hidden "structures" or determinants are identified, we can then analyze in the same way a new script and predict its financial return, once it is made into a movie.

Our approach is developed with movie scripts in mind. Ideally, we would like to implement our approach with movie scripts in electronic form. However, as most movie shooting scripts are not publicly available in electronic form and we cannot collect a sufficient number of them,[4] we restrict our attention to "spoilers" in implementing our proposed approach. A spoiler is an extensive summary of the storyline of a movie written by movie viewers after they watch a movie. Each spoiler is typically around 4-20 pages long. It is essentially a blow-by-blow description of a movie so that its readers do not have to go to

[4] We have located only 52 electronic scripts for our sample of 281 movies. This number is too small to implement our approach.

a movie theatre to know the story told in the movie, hence the name "spoilers." Examples of spoilers can be found at www.themoviespoiler.com. When we develop our prediction tool, we view spoilers as a proxy for the actual shooting script for three reasons.

First, by limiting ourselves to the texts that contain less information, we essentially stack the deck against ourselves in predicting movie successes. Even so, we find that the performance of our approach is very encouraging, showing a great deal of promise for its practical applications. Indeed, with actual scripts in the digital form available to studios that contain the descriptive information in spoilers, the performance of our approach is expected to improve.[5] Second, spoilers in digital form are easily available to us on the Internet, while scripts are not. It would be truly a monumental work if we were to produce digital scripts for most of the movies in our sample. To the extent that spoilers tell the stories in actual scripts in a descriptive, detailed way, using spoilers is not only expedient, but also quite reasonable. Indeed, when we test the similarity between spoilers and scripts (when both are available) along the semantic and textual variables we considered, we are able to show positive and significant correlation between a spoiler and a script in all the variables we extract.[6] Third, the main purpose of our paper is to illustrate and expound a new approach. As it would soon become clear, our approach can be easily extended to handle actual scripts if they become available. There is no variable extracted from a spoiler that we cannot extract from a script.

For the current study, we implement our approach using available spoilers for 281 movies, which are all released during the period 2001-2004.[7] Using a simple random sampling scheme, we divide our data into a training sample of 200 movies, which we use to fit our model, and a testing sample of 81 movies, which is used in a holdout prediction test to assess the predictive performance of our approach.

---

[5] Nevertheless, it is also plausible that viewers, when writing spoilers, may use words and semantics that are different for good movies than for bad movies such that spoilers can a better predictive validity than original scripts. What mitigates this possibility is the fact that spoilers, focusing on the storyline, are generally written in a descriptive and matter-of-factly fashion.

[6] Detailed results are available upon request.

[7] The 281 movies are selected based on their time period (1/1/2001 to 12/31/2004) and the availability of the movie spoiler on www.themoviespoiler.com, where we downloaded all our spoilers. In the same time period, the 1378 of movies are released.

The results we obtained are very encouraging, even though our current second-best implementation cannot fully exhaust the potential of the approach we propose here. Based on a holdout prediction test, we find that our model is able to capture signals from stories contained in spoilers and use them to differentiate between profitable and unprofitable stories for movies. Compared to randomly choosing movie scripts, making production decision based on our forecast will significantly increase ROI, which we show in Section 4. This improvement is especially significant considering the fact that the stories in scripts represent only one of the many factors that can affect a movie's ROI.

In the next section, we outline the background and rationale of our approach by reviewing the relevant literature in forecasting a motion picture's box office performance, natural language processing, screenwriting, and modern statistical learning. In Section 3, we describe the textual information we extract from scripts or spoilers. In Section 4, we calibrate our prediction model and present our empirical results. In Section 5, we conclude with suggestions for future research.

2. Relevant Literature and Our Approach

The success of a movie has been commonly measured by its box office performance and many researchers have built models to forecast such performance. Several researchers forecast box office performance based on box office performance in the early weeks (e.g., Sawhey and Eliashberg 1996); others have made forecast based only on pre-release information (e.g., Neelamegham and Chintagunta 1999; Eliashberg et al. 2000; Shugan and Swait 2000). Both streams of research provide promising results in predicting box office revenues and have been proven to aid distributors' planning.

However, the above-mentioned research focuses on predicting box office performance after a movie is already produced. We address the problem of forecasting the movie's performance prior to its production. This is an important distinction for two reasons. First, after a movie is made, the costs of making the movie are incurred and sunk and hence the relevant performance metric is the total box office. However, before a movie is made, the costs are avoidable and studios are making an investment decision

assessing the possible return on investment (ROI). In this latter case, ROI is the relevant metric for performance. Second, before a movie is made, the relevant piece of information available in making the investment decision is the story in a movie script, while after a movie is made much more information is available about the movie's cast, marketing budgets, etc.. This makes forecasting the success of a movie pre-production so much harder. While many studio managers and critics argue that the "storyline" is the most important predictor of a movie's box office success (Hauge 1991), there has been no rigorous way for them to incorporate the storyline into their forecasting. As a result, the common view among industry experts is that picking a successful movie from scripts is like picking a horse long before a race: it is difficult, if not impossible (Litman and Ahn 1998).

Likewise, academic researchers have not thus far addressed this issue, despite its huge economic importance. This lack of attention can partly be explained by the textual nature of the "data" – the stories. Textual data are extremely high dimensional: a text article contains thousands of different words of different frequency and in different order. If we take each word, its frequency, and its order of appearance as a distinct dimension, the dimensionality for even a short news article would exceed millions and even trillions. Therefore, it is infeasible to use any standard econometric approach to perform the task at hand, at least not before we can reduce the number of dimensions.

Fortunately, we can apply the "Bag-of-Words" model, developed recently in Natural Language Processing, to represent the data and cut down the number of dimensions, as the first step towards developing our forecast model.

Natural Language Processing

Natural Language processing, a sub-field in computer science, has a long history of analyzing textual information. It has found use in many areas such as authorship attribution (Holmes 1994; Holme and Forsyth 1995), text categorization (Sebastiani 2002), and even automatic essay grading (Larkey 1998). The primary way computer scientists represent a document is by using the "bag-of-word" model: a document is represented entirely by the words that it contains and how many times the word appears,

neglecting the order in which the words appear. This representation, while rather simplistic, delivers surprisingly good results in performing a diverse array of tasks. For instance, commercial software has been developed to assess the complexity and difficulty of a text in order to determine the suitability of the text for an age group (see www.lexile.com). Similarly, Landauer et al. (1997) developed a computer program capable of grading student essay. They found very high inter-rater reliability between expert's judgment and the grade reported by the computer program. Together with the bag-of-word representation, some researchers also add semantic summaries such as "number of paragraph", "number of words", "average sentence length" etc., to represent the general semantic texture of a document.

Specific to our task at hand, the bag-of-word approach will help us pick up the themes, scenes, and emotions in a script. For instance, the frequent appearance of words such as "guns," "blood," "fight," "car crashes," and "police" may indicate that the script contains a crime story with action sequences. When this information is coupled with known box office receipts for the movies already made in the recent past, we would know if the movies of this type tend to sell well or not in theatres.

Admittedly, while we expect the bag-of-word representation to be useful in picking out some important predictors, its usefulness in performing our task is limited. This is because movie viewing is a hedonic consumption experience: the enjoyment that a person gets from a movie is "an outcome of the dynamic interaction between stable individual difference factors, temporary moods, and the emotional content of the experience" (Eliashberg and Sawhney 1994). It is clearly infeasible to leave out "word order" from a movie script while attempting to capture the storyline in a movie script. To take a simple example, the plot "the villain kills Superman" and "Superman kills the villain" will clearly trigger different emotional response from the audience, even though both sentences contain exactly the same words with the same frequency. Therefore, specific to the analysis of movie stories, we need to incorporate domain knowledge from screenwriting experts in evaluating a movie script to exhaust the potential of using scripts to predict movie successes.

Domain Knowledge in Screenwriting

Much has been written about how a story should be told and what kind of stories would resonate with audience (e.g. Blacker 1988; Field 1994; Field 1998; Hauge 1991). The experts describe many specific criteria a good movie story has to possess. We have summarized these aspects and include them in the Appendix. Presumably, if a script scores high on all those criteria, its likelihood of success at box office is higher.

Since a computer cannot understand storylines, it is impossible for any automated textual analysis to pick up this genre and content information from a script. Thus, we hire human judges to read movie spoilers and answer a pre-determined set of 22 questions as shown in the Appendix. The answers to those questions can then supplement the bag-of-word representation and feed into our textual analysis to calibrate our model for predicting the financial success of a movie production. Of course, all these different kinds of textual information have to be integrated in a meaningful way in the specific context of movie production.

Statistical Learning Techniques

Once we have extracted information from movie spoilers, we have a dataset containing a large number of predictors. Many different procedures have been developed in the last 5-10 years to tackle the problem of prediction when there are a large number of predictors. These procedures include neural network, support vector machine, nearest-neighbor, trees, and many other variable selection techniques used in conjunction with regression (Hastie et al. 2001). However, most of these approaches are not suitable for our problem. This is because many factors in a movie script can interact with each other in a very complex, nonlinear fashion. For instance, in a drama, complex emotions with multiple scenes plus a surprising ending may be essential for a good movie. However, for an action movie, a large number of car chase sequences with a logic ending may resonate better with audience. Thus, the analytical tool we propose must be flexible enough to accommodate all possible interactions among different components in a script and especially those interactions that audience appreciates through their revealed preferences at the box office. At the same time, since our goal is to develop a decision aid for movie producers, the

methodology we choose must be easy to interpret and must lead to intuitive insights.[8] With this goal in mind, we choose to use Bag-CART (Bootstrap Aggregated Classification and Regression Tree) procedure (Breiman 1996; Breiman et al. 1984), a technique that is uniquely suitable for uncovering complex interactions between predictors which may be difficult or impossible to uncover using traditional methods (Lewis 2000).[9]

Bag-CART is an extension of CART (Classification and Regression Tree), a procedure developed by Breiman *et al* (1984). Using trees as a data analysis tool has a long history in statistics. Morgan and Sonquist (1963) first proposed a simple method to fit binary trees to data, called "automatic interaction detection" (AID). Kass (1980) extended the AID methodology to categorical data, resulting in a new method named "chi-square AID" (CHAID). AID and CHAID use significance test to determine tree size, and are usually recommended to be used in large dataset and for exploratory purpose. Breiman et al. (1984) further developed the above techniques into CART by providing formal grounding of trees in probability theory and also proposed to use cross-validation to determine optimal tree size, making the technique more suitable for prediction. In the last ten years, CART has been used extensively in many different disciplines as a prediction methodology. For instance, it has been used in meteorology to predict UV Radiation on the ground (Burrows 1996), in engineering to predict the quality of glass coating (Li et al. 2003), in economics to predict views in welfare policy (Keely and Tan 2005), in neurology to predict the recovery of memory after brain injury (Stuss et al. 2000), in computer science to predict storage device performance (Wang et al. 2004), and very recently in medical science to predict the occurrence of prostate cancer (Garzotto et al. 2005).

A CART model is basically a form of binary recursive partitioning. Formally, we start with data $(\mathbf{x_i}, y_i)$ for i = 1,2…,N, with $\mathbf{x_i} = (x_{i1}, x_{i2}\ldots x_{ip})$, where **x** are the predictors variables and y is the response

[8] We thank an anonymous reviewer who points out this issue to our attention.
[9] We have also implemented other statistical methodologies including linear models (with and without variable selection), neural network, and Bayesian treed models. We find that the Bag-CART procedure outperforms all of the alternative approaches we tested. Detailed results are available upon request.

variable. Starting with all the data, we search to find splitting variable j and split point s, to form two exclusive partitions defined as:

$$R_1(j,s) = \{X \mid X_j \le s\}, R_2(j,s) = \{X \mid X_j > s\}.$$

The variable j and the split point s are chosen so as to minimize the within-partition sum-of-squared error:

$$\sum_{x_i \in R_1(j,s)} (y_i - \bar{y}_{R_1})^2 + \sum_{x_i \in R_2(j,s)} (y_i - \bar{y}_{R_2})^2 ,$$

where $\bar{y}_{R_1}$ and $\bar{y}_{R_2}$ denote the sample average of the response variable y in the partition R_1 and R_2 respectively (Hastie et al. 2001). Then, we repeat the splitting process on each of the two regions and on all subsequent regions as we repeat the procedure, until a pre-defined "stopping criterion" is reached. A stopping criterion can be defined in terms of maximum tree depth, minimum node size, or using some external statistics (e.g. Cp statistics[10]). Once a stopping criterion is chosen, the specific stopping value is usually found using cross-validation. In this paper, we use maximum tree depth M as the stopping criterion, since it is most straightforward to implement when we incorporate the "bagging" procedure later.

While CART is a very powerful technique by itself, it suffers from the problem of estimation instability. Since the fitting procedure is based on an algorithm of finding a series of optimal split variables and split points, a slight change in the data can change the whole structure of the resulting regression tree (Breiman 1996). Further, since the procedure does not "look-ahead" when deciding a split, it may lead to suboptimal trees.[11] This problem is particularly serious when the sample size is small. Breiman (1996) approached this instability problem by "Bootstrap Aggregation", or "Bagging" in short. Bagging is an application of the bootstrap technique (Efron and Tibshirani 1994) that is used to improve unstable or weak predictors. In a Bagging procedure, the original data is repeatedly sampled with replacement, creating bootstrapped data sets. Using CART procedure, a different regression tree is fitted

[10] Cp statistics is similar to AIC statistics, which is often used for model selection. (Akaike 1974)
[11] We thank an anonymous review for pointing out this issue.

for each sample. The final prediction is then made by averaging over the predictions from each regression trees (Breiman 1996). Specifically, the bagging estimate is defined by:

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^{*b}(x)$$

where $\hat{f}^{*b}(x)$ denotes the prediction of the CART tree fitted using b-th bootstrap dataset as its training sample. As in the original CART model, each tree is grown until a maximum tree depth M is reached. Optimally, M should match the dominant level of interaction among the predictor variables. Hastie et al. (2001) suggest choosing M between 4 and 8. In our procedure, we pick M to be 4, but we also tested other values for M and found that the results are insensitive to the particular choice made.[12]

Bagging only works on estimators that are nonlinear functions of the data (Hastie et al. 2001). In particular, for an unstable procedure like CART, bagging can dramatically reduce estimation variance and thus lead to improved prediction. Hastie et al. (2001) gives a simple argument, based on bias-variance tradeoff, showing why bagging is effective. Their central claim is that using multiple bootrapped averages will reduce variance while maintaining the same bias, and thus reducing the overall mean square error of prediction. Apart from delivering superior prediction results, Bagged-CART has the added advantage of generating the "relative relevance" of each predictor, when compared to other black-box machine learning algorithm, by looking at the relative appearance frequencies in the bootstrap runs (Efron and Tibshirani 1998). In addition, one can potentially develop intuitive insights by examining the various trees that results from each bootstrapped sample.

3. Extracting Textual Information

We extract information from a spoiler on a number of levels: 1) Semantics; 2) Bag-of-Words; 3) Genre and Content Analysis. The first two levels summarizes the lower-level content of a script (i.e. information available without an understanding of the storyline) and is done automatically by a computer,

[12] Another way to do this is to find the "optimal" M using cross-validation. In our context, M does not have a significant effect on prediction performance. We fit the model with different values of M from 4 to 10 and the prediction performance is very similar.

while the latter two levels focuses on the higher-level aspects of a script with the help of three independent raters with cinema training. Together they extract a large amount of information from a movie spoiler in a systematic and rigorous way.

(i) Semantics

Using the "Spelling and Grammar" function on MS Word, we extract a number of semantics information for each spoiler. An example of semantic information extracted from the movie "A Walk to Remember" is shown in Table 1.

[Insert Table 1 about here]

(ii) Bag-of-Words

The next level of information extraction comes from the frequency of each individual word in the spoiler. Consistent with the "bag of words" representation of a document in natural language processing, we wrote a program to break each document into the words that appears in the document and their frequencies of occurrence. Of course, not every word has a distinct substantive meaning. We used a "stemmer" to group the different forms of the same word together. For example, go / went / gone would be treated as the same word "go". To further isolate more important words to feed into the analysis, we created an "importance" index for each word. The importance index for the i-th word is defined as follows:

$$I_i = \left(1 - \frac{d_i}{D}\right) \times N_i$$

where d_i denotes the number of documents containing the i-th word, D denotes the total number of documents, and N_i is the total frequency of occurrence of the i-th word across all documents.

This importance index is defined so that words that appear in almost every documents (e.g the, they, he / she, are) will be screened out and words that only appear very few times (e.g. catheter, Yiddish,

demilitarized) will be screened out as well. After calculating the importance index for each word, we retain the 100 most important words for our analysis.

To reduce dimensionality, a process that is frequently necessary for this kind of analysis, we run a principal-component analysis on the document-word frequency data set. That analysis shows that the "elbow" happens between the second and third principle component. Thus, we decide to retain the first two principal components only. We ran a principal-component based factor analysis and calculated the factor score on each factor for each of the spoilers. When we analyze the factor loadings on each word on each of the two factors, we find that factor 1 captures "dialogue" words related to characters communicating with each other (e.g. say, talk, tell, ask) while factor 2 captures words related to a "violent scene" (e.g. open, door, shoot, die)[13].

(iii) Genre and Content Analysis

Genre and the content of the storyline are much higher-level aspects of a spoiler. One would have to understand the meaning of the storyline of a spoiler in order to extract higher-level information. Thus, the task cannot be done by a computer and has to rely on external judges. We hired three independent judges with training in cinema to classify each movie into different genres (more than one genre can be selected) and asked them to rate each spoiler on a list of 22 questions and provide yes/ no answers to each question. Their answers are then aggregated. Thus, each genre/content variable can take values 0, 1, 2, or 3 depending on the number of raters who answered "yes" to the relevant question. We generate these questions from four books about screenplay writing (Blacker 1988; Field 1994; Field 1998; Hauge 1991), which represent the expert's opinion on how a successful script should have and be written. The questions are listed in the Appendix. It is important to note here that these questions are not about a movie's box

[13] These two factors account for 10.6% and 6.3% of variances respectively. The first set of words have a factor loading of > 0.4 on the first factor and also have a low factor loading (<0.1) on the second factor. The second set of words have a factor loading of > 0.25 on the second factor and also have a low factor loading (<0.1) on the first factor.

office performance, but about how a storyline is told. Of course, studios have some leeway in generating this part of data by using more specific questions.

4. Predicting ROI based on Textual Information

Using the procedure we discussed in Section 3, we extracted predictors from the textual data. The summary statistics of each predictor is shown in Table 2.

[Insert Table 2 about here]

As shown in the left panel of Figure 1, the distribution of ROI is highly skewed. If ROI is used as the response variable, the estimates in the CART model will be dominated by a few observations. Since ROI can range from -1.0 to $+\infty$, we log-transform our response variable to log(ROI + 1) to allow for a more symmetric shape with thinner tails, as shown in the right panel of Figure 1. Summary statistics of ROI and log(ROI+1) is listed in Table 3.

[Insert Table 3 about here]

[Insert Figure 1 about here]

With these extracted textual data, we then fit a Bagged-CART model with log(ROI + 1) as the response variable and textual information we collected in previous section as predictor variables. We use the library *rpart* in R to fit CART tree, and the bagging part is fitted by a program we coded in R. A CART model fitted with all the 200 training observations is shown in Figure 2.

From the result, we see that this specific tree first splits based on whether the movie is an action movie. Action movies are classified to the left node while non-action movies are classified into the right node. Then it is interesting to see that different variables are relevant for Action and Non-Action movies. For instance, the most important predictor for Action movies is whether it is logical, while for non-action

movies it is not desirable to have a surprise ending. In this fashion, CART model takes into account the interaction effects among variables, which is vital in our application because we expect that the different components in a movie will interact with each other in a rather complex way.

[Insert Figure 2 about here]

As mentioned before, the main shortcoming of CART models is instability. A slight alteration of the data will change the whole structure of the tree, making predictions unstable (Hastie et al. 2001). A solution suggested by Breiman (1996) is to draw multiple bootstrap samples of the data and then create a CART tree for each of these bootstrap data sets. The predictions made from each bootstrapped CART model is then averaged to give a final prediction. We created 1000 Bootstrap trees. Two of the bootstrapped CART trees are shown in Figure 3 and 4.

[Insert Figure 3 about here]

[Insert Figure 4 about here]

It is interesting to note that each of the above tree seems to focus on a different aspect of the data. The first tree starts by considering whether the story exposes the viewers to the setting early ("Early Exposition") while the second tree starts by considering whether the premise in the story is clear, which is in agreement with one of the criteria of what expert readers would first consider when they evaluate a screenplay. In the language of the statistics literature, these different trees act as different "experts" taught by past box office performances, who would then "vote" at the end to give a final prediction.

We can interpret the frequency of the occurrence of each predictor in the bootstrap CART tree as the "relative relevance" of the predictor in predicting movie ROI. The more times the predictor appears in CART trees, the more relevant it is. We define relevance index as the percentage of times (out of 1000 trees) a variable is included in the trees. The relevance indices of the fifteen most "relevant" variables,

based on 1000 bootstrap samples, are shown in Figure 5. Passive, violent scenes, and the number of sentences required to describe the storyline are the three most relevant predictors. It is important to note here that the relevance index does not say anything about the relative economic significance of a factor in determining the performance of a movie, as in a conventional regression analysis. Of course, one should not expect such an interpretation either, as factors in a script interact with each other in a highly nonlinear way so that an attempt to quantify the "marginal contribution" of a factor is doomed to failure. In other words, for our task at hand, the predictive accuracy dictates that we give up on the idea of quantifying the marginal effect of a variable. The only thing we can say is that a factor is relevant to our predictions to a varying degree.

[Insert Figure 5 about here]

After we applied the Bag-CART procedure to our dataset, we can assess the model "fit" by looking at the in-sample R^2 value. Although our model is a non-linear model, we can still define the in-sample R^2 as in the case of linear model:

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}$$

This measure of R^2 would, similar to the case of linear regression, indicates the proportion of variance in the in-sample y's that is explained by our model fit. Here, N denotes the number of movies in the training sample, which in our case is equal to 200.

The in-sample R^2 value of our model is 0.481. This compares favorably to the in-sample linear regression-based R^2 value of 0.233 obtained by Ravid (1999) and 0.413 obtained by Litman and Ahn (1998), even if they use more predictors that are available much later in the production process (e.g., date of release, support from a major distributor) or even predictors that are only available after a movie is released, e.g., number of screens, competitive forces, whether a movie wins Academy awards.

Although this comparison appears impressive considering that we only put in very limited information in our model, it should be taken with caution. The additive tree model we used is inherently a more flexible class of models than linear regression models that the above researchers used. Therefore, in-sample R^2 values may not be directly compared to conclude that our model fits better than its alternatives. A more relevant performance measure is the out-of-sample predictive accuracy of our Bagged-CART model.

To assess the predictive validity of our tool, we perform a holdout prediction test on a holdout sample of 81 movies. We compute the Mean Squared Error of our Bagged-CART in predicting log(ROI+1) to the Mean Squared Error using the mean log(ROI+1) of the training set. The MSE of our model for the holdout sample is 0.752 while that of the training mean is 0.830. A comparison using Mean Absolute Error gives similar results. Our Bagged-CART model has an MAE of 0.673 while using the training mean for prediction results in a MAE of 0.720.

To further assess the predictive ability of our approach, we classify the 81 holdout movies into two groups, "above median" group and "below median" group, based on our predicted ROI for each movie. The classification procedure is done as follows. First, we calculate the median ROI for the 200 movies in the training sample[14]. The median ROI, -27.2%, represents the performance we expect of an "average" movie. Then, if a movie's predicted ROI is higher (lower) than the median ROI, it will be classified into the "above median" ("below median") group. This classification based on predicted ROI is then compared with the classification based on the actual movie performance. The results are shown in Table 4. The diagonal entries in Table 4 represent agreement between predicted classification and actual classification. If we add up the numbers on the diagonal and divide it by 81 (the number of movies in the test set), we find that the percentage of correct classification using our prediction model is 61.7%. This compares favorably to the benchmark accuracy of 51.3% and 50.0% using "maximum chance criterion" and "proportional chance criterion" (Morrison 1969) respectively. Although a correct classification rate of

[14] Using the median is more preferable than using the mean because of the existence of outliers and the skewness of the distribution of ROI among the movies, as depicted in Figure 1.

61.7% is not very high in an absolute sense, this is rather expected, as we do not use many other factors that are known to affect the final return after a movie production, including advertising and promotion effort, seasonal effects, screen numbers, competition, etc. Indeed, considering the fact that our data is sparser than the ideal one that studios can put together, this is a rather remarkable accomplishment, capturing a good deal of contributions to the financial success from the movie scripts alone.

[Insert Table 4 about here]

We can go one step further to ascertain how each of the three types of textual variables helps to explain the variations in the dependent variable. We ran six separate analyses by only including a subset of the original variables: (i) content analysis and genre, (ii) word frequencies, and (iii) semantics as predictors. The resulting prediction errors, as measured by mean square error, mean absolute error, and hit rate, are shown in Table 5. From the table, we see that the three types of variables appear equally important in predictive power. Thus, including all three types of information in our model would allow us to optimize the predictive performance.

[Insert Table 5 about here]

Of course, a more relevant question is whether the signals captured by our predictive model are of any economic significance in terms of increased profitability for the movie studio. To assess the economic significance of our approach, we set up the following scenario to simulate the studio's production decisions. Since studios' production decisions are made in slates per year, we attempt to identify a production portfolio of X movies among the 81 movies to produce, with $X \leq 81$, and we will then compare our return on investment to the some benchmark returns on investment if we had chosen the same number of movies from the same set of scripts without the aid of our approach.

For instance, to choose a set of 30 movies, we perform textual analysis on each of the 81 spoilers and predict their return on investment based on our model. Then, these spoilers are ranked based on predicted ROI and the 30 movies that give the highest predicted return on investment are selected. Our portfolio uses a total budget of $1044.5 million, and generates a gross box office revenue of $1996.8 million, resulting in a studio's net revenue ("rentals") of $1098.2 million and hence a return of 5.1% on investment. We compare the performance of our portfolio against two benchmarks. In the first benchmark, 30 movies are randomly chosen to form a portfolio. Each of the 81 movies in the holdout sample has equal probability of being selected in the portfolio. The ROI of this portfolio is recorded, and this procedure is repeated 1000 times. Then, the mean ROI based on random selection is calculated by averaging the ROI that results in 1000 portfolios. We find that the mean ROI in this case is -18.6%. In the second benchmark, we try to replicate more closely the way studios select movies by MPAA ratings. Studios in general make roughly 60% R-rated movies and 40% non-R (G / PG / PG-13) rated movies (MPAA 2004). Thus, we replicate this rule of thumb by fixing the number of R-rated movies to be made in the 30-movie portfolio case to be 18 and the number of non-R rated movies to be 12. Then we select 18 R-rated movies randomly from the pool of R-rated movies in the 81 holdout movies, and likewise for non-R rated movies. The mean ROI of a portfolio selected using this "MPAA-based" selection method is -24.4%. Our method outperforms both benchmarks by a significant margin. Indeed, as shown in Figure 6, our approach always produces a significant economic gain no matter how many movies are selected for the portfolio, suggesting that our model is able to capture determinants from the textual information in movie scripts and hence significantly improve the studio's profitability.

[Insert Figure 6 about here]

5. Conclusions

The "green lighting process" – deciding which among scripts from a large number of screenplays to turn into a movie – is one of the most important financial decisions movie studios and independent

firms have to make almost every day. Despite its financial importance, the decision is still made in practice largely relying on age-old tradition of judgments and intuitions. As a result, this decision process is subject to many random influences, generating highly variable, unpredictable outcomes.

In this paper, we demonstrate for the first time that it is possible to improve the green-lighting process using an objective and rigorous approach that only relies on information on the storyline derived from movie scripts. Our proposed approach uses a combination of screenwriting domain knowledge, natural language processing techniques, and modern statistical learning methods to relate the stories in a script to potential customer responses to them. This approach is systematic and rigorous enough to inject a large dose of objectivity into the green-lighting process. It is also versatile and flexible enough to detect highly nonlinear, implicit interactions among a large number of factors that ultimately make a movie "click" or not "click" with movie-goers. Such an approach is uniquely suitable for forecasting a movie's return on investment pre-production. We demonstrate, using spoilers as proxies for scripts, that our approach is able to differentiate between the scripts that would be successful from those that would result in a financial failure. The economic gain from using this approach is quite substantial and a studio can make more profitable production decisions using the approach we develop here.

One may argue that the premise underlying our approach is formulaic script writing, which in turn may lead to a potential narrowing of the new product development process, leading to unmet demand. We would like to point out that rather than coming out with a set of rigid rules to follow, our approach will only suggest the structural regularities that a successful script generally possesses. We believe that there is room for creativity within the structural regularities. In a sense, our approach is similar to Goldenberg et al. (1999), who identified a set of templates for successful advertisement. Goldenberg et al. (1999) showed that people's creativity generally improves after learning about the "creativity templates" that they identified. Similarly, our research may assist studios in their green-lighting process by providing a set of structural templates for successful scripts.

Notwithstanding these promising initial results, we believe that our approach can be improved in four ways, with sufficient resources. First, studios have easy access to shooting scripts in electronic form.

This would eliminate the need to use spoilers as proxies. Then, the same procedures and analysis can be performed on the scripts to come up with most likely better and more comprehensive determinants to improve prediction. We do expect that our approach, with all its built-in flexibilities, will perform better with scripts, although such improvements may not be drastic. Second, studios are not as constrained by economic resources and they can hire more and perhaps better "experts" to generate higher-level textual information from scripts. Therefore, they can make a one-time investment to "train" a model based on our approach with more expert input. Better expert input can only improve predictive accuracy. Third, the approach proposed here can also be extended to include other aspects of the creative process (e.g., actors, directors, and filming locations). Finally, a significant share of a studio's revenues from a movie comes from auxiliary markets, such as cable, DVD, international markets, etc. The approach we have developed here can be trained on the returns from any of the markets or on the returns from any combinations of the markets. By thus forecasting the revenues from the auxiliary markets, a studio can better gauge the total profitability of a movie production even before it is produced.

Reference

Akaike, Hirotugu (1974). A New Look at the Statistical Model Identification. *IEEE Transactions on Automatic Control*, 19 (6), 717-723.

Blacker, Irwin R. (1998). *The Elements of Screenwriting*. Macmilan Publishing, New York, NY.

Breiman, Leo (1996). Bagging Predictors. Machine Learning, 24 (2), 123-140.

Breiman, Leo, Jerome H. Friedman, Richard A. Olshen and Charles J. Stone (1984). *Classification and Regression Trees*. Wadsworth Publishing, Belmont, CA.

Burrows, William R. (1996). CART Regression Models for Predicting UV Radiation at the Ground in the Presence of Cloud and Other Environmental Factors. Journal of Applied Meteorology, 36(5), 531-544.

Efron, B. and Tibshirani, R. (1994). *An Introduction to the Bootstrap*. Chapman and Hall, New York, NY.

Efron, B. and Tibshirani, R. (1998). The Problem of Regions. Annals of Statistics, 26, 1687-1718.

Eliashberg, Jehoshua, Anita Elberse, Mark A.A.M Leenders (2005). The Motion Picture Industry: Critical Issues in Practice, Current Research and New Research Directions, Marketing Science, forthcoming.

Eliashberg, J., & Sawhney, M.S. (1994). Modeling Goes to Hollywood: Prediciting Individual Differences in Movie Enjoyment. Management Science, 40(9), 1151-1173.

Eliashberg, Jehoshua, Jedid-Jah Jonker, Mohanbir S. Sawhney, and Berend Wierenga (2000). MOVIEMOD: An Implementable Decision Support System for Pre-Release Market Evaluation of Motion Pictures. Marketing Science, 19(3), 226-243.

Field, Syd (1994). *Screenplay: The Foundations of Screenwriting*, 3d ed. Dell Publishing, New York, NY.

Field, Syd (1998). *The Screenwriter's Problem Solver*. Dell Publishing, New York, NY.

Garzotto, M. T.M. Beer, R.G. Hudson, Y.C Hsieh, E. Barrera, T. Klein, and M. Mori (2005). Improved Detection of Prostate Cancer Using Classification and Regression Tree Analysis. Journal of Clinical Oncology, 23(19), 4322-4239.

Goldenberg, J., D. Mazursky, and S. Solomon (1999). Creativity Templates: Towards Identifying the Fundamental Schemes of Quality Advertisements. Marketing Science, 18(3), 333-51.

Hastie, Trevor, Robert Tibshirani, Jerome Friedman (2001). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York, NY.

Hauge (1991). *Writing Screenplays that Sell*. HarperCollins Publishers, New York, NY.

Holmes D. (1994). Authorship Attribution. Computers and the Humanities, 28, 87-106.

Holmes D. and R. Forsyth (1995). The Federalist Revisited: New Direction in Authorship Attribution. Literary and Linguistic Computing, 10 (2), 111-127.

Kass, G.V. (1980). An Exploratory Technique for Investigating Large Quantities of Categorical Data. Applied Statistics, 29, 119-127.

Keely, Louise C., and Chih Ming Tan (2005). Understanding Divergent Views on Redistribution Policy in the United States. Working Paper.

Landauer, T. K., Laham, D., Rehder, B. & Schreiner, M. E., (1997). How well can passage meaning be derived without using word order? A comparison of Latent Semantic Analysis and humans. Proceedings of the 19th annual meeting of the Cognitive Science Society. 412-417. Mahwah, NJ : Erbaum.

Larkey, Leah S. (1998). Automatic Essay Grading Using Text Categorization Techniques. Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval.

Lewis (2000). An Introduction to Classification and Regression Tree (CART) Analysis, presented at the 2000 Annual Meeting of the Society of Academic Emergency Medicine in San Francisco, CA.

Li, Mingkun, Shuo Feng, Ishwar K. Sethi, Jason Luciow, Keith Wagner (2003). Mining Production Data with Neural Network & CART. Third IEEE International Conference on Data Mining, 2003, 731.

Litman, B.R., & Ahn, H. (1998). Predicting Financial Success of Motion Pictures. B. R. Litman The Motion Picture Mega-Industry. Needham Heights, MA : Allyn & Bacon.

Mehta, Stephanie N. (2006). Money Men. *FORTUNE*, May 23, 2006.

Morgan, J.N. and Sonquist, J.A. (1963). Problems in the Analysis of Survey Data, and a Proposal. Journal of the American Statistical Association, 58, 415-434.

Morrison, D. G. (1969). On the Interpretation of Discriminant Analysis. *Journal of Marketing Research*, 6, 156-163.

MPAA (2004). MPAA Economic Review. [www.mpaa.org].

Neelamegham, Ramya and Chintaugunta, Pradeep (1999). A Bayesian Model to Forecast New Product Performance in Domestic and International Markets. Marketing Science, 18 (2), 115-136.

Ravid, S. A. (1999). Information, Blockbusters, and Stars: A Study of the Film Industry. Journal of Business, 72 (4), 463-492.

Sawhney, Mohanbir S., & Jehoshua Eliashberg (1996). A Parsimonious Model for Forecasting Gross Box-Office Revenues of Motion Pictures. Marketing Science, 15 (2), 113-131.

Sebastiani, Fabrizio (2002). Machine Learning in Automated Text Categorization. ACM Computing Surveys, 34 (1), 1-47.

Shugan, Steven M. and Joffre Swait (2000). Enabling Movie Design and Cumulative Box Office Predictions. ARF Conference Proceedings.

Silver-Lasky, Pat (2003), *Screenwriting in the 21st Century.* Sterling Publishing Company Inc.

Stuss, D.T, F.G. Carruth, B. Levine, C.F. Brandys, R.J. Moulton, W.G. Snow, and M.L. Schwartz (2000). Prediction of Recovery of Continuous Memory after Traumatic Brain Injury. Neurology 2000, 54, 1337-1344.

Vogel, Harold L. (2004). *Entertainment Industry Economics: A Guide for Financial Analysis*. Cambridge, UK : Cambridge University Press.

Wang, Mengzhi, Kinman Au, Anastassia Ailamaki, Anthony Brockwell, Christos Faloutsos, and Gregory R. Ganger (2004). Storage Device Performance Prediction with CART models. SIGMETRICS Performance Evaluation Review, 32(1), 412-413.

Appendix

List of 22 questions

1) *Clear Premise (CLRPREM)*: The story has a clear premise that is important to audiences.
2) *Familiar Setting (FAMSET)*: The setting of the story is familiar to you.
3) *Early Exposition (EAREXP)*: Information about characters comes very early in the story.
4) *Coincidence Avoidance (COAVOID)*: Story follows a logical, causal relationship. Coincidences are avoided.
5) *Inter-Connected (INTCON)*: Each scene description advances the plot and is closely connected to the central conflict.
6) *Surprise (SURP)*: The story contains elements of surprise, but is logical within context and within its own rules.
7) *Anticipation (ANTICI)*: Keep readers trying to anticipate what would happen next.
8) *Flashback Avoidance (FLHAVOID)*: The story does not contain flashback sequences.
9) *Linear Timeline (LINTIME):* The story unfolds in chronological order.
10) *Clear Motivation (CLRMOT)*: The hero of the story has a clear outer motivation (what he/she wants to achieve by the end of the movie).
11) *Multi-dimensional Hero (MULDIM)*: Many dimensions of the hero are explored.
12) *Strong Nemesis (STRNEM)*: There is a strong nemesis in the story.
13) *Sympathetic Hero (SYMHERO)*: Hero attracts your sympathy because he/she exhibits courage AND belongs to one of the followings: -good/nice, funny, good at what he does OR has power.
14) *Logical Characters (LOGIC)*: Actions of main characters are logical considering their characteristics. They sometimes hold surprises but are believable.
15) *Character Growth (CHARGROW)*: Conflict is important enough to change the hero**.**
16) *Important Conflict (IMP)* : The story has a very clear conflict, which involves high emotional stakes
17) *Multi-Dimensional Conflict (MULCONF)*: The central conflict is explained in many different points of view.
18) *Conflict Build-up (BUILD)*: The hero faces a series of hurdles. Each successive hurdle is greater and more provocative than the previous ones.
19) *Conflict Lock-in (LOCKIN)*: The hero is locked into the conflict very early in the movie.
20) *Unambiguous Resolution (RESOLUT)*: Conflicts is unambiguously resolved through confrontation between the hero and nemesis at the end.
21) *Logical Ending (LOGICEND)*: The ending is logical and believable.
22) *Surprise Ending (SURPEND)*: The ending carries surprise and is unexpected.

Semantics Information Extracted for a Spoiler

Title	*"A Walk to Remember"*
Number of Characters (NCHAR)	3269
Number of Words (NWORD)	740
Number of Sentences (NSENT)	43
% of Passive Sentences (PASSIVE)	2%
Characters Per Word (CHARPERWD)	4.2

Table 1. Semantics information extracted for a spoiler.

Variable	Mean	Std. Dev.	Median	Min	Max
Comedy	1.155	1.345	0.000	0.000	3.000
Action	1.310	1.200	1.000	0.000	3.000
Drama	0.785	1.046	0.000	0.000	3.000
Thriller	0.495	0.902	0.000	0.000	3.000
Horror	0.265	0.753	0.000	0.000	3.000
CLRPREM	2.270	0.950	3.000	0.000	3.000
FAMSET	2.475	0.844	3.000	0.000	3.000
EAREXP	2.785	0.469	3.000	1.000	3.000
COAVOID	2.085	0.755	2.000	0.000	3.000
INTCON	2.450	0.742	3.000	0.000	3.000
SURP	2.625	0.506	3.000	1.000	3.000
ANTICI	1.890	0.867	2.000	0.000	3.000
FLHAVOID	2.465	0.907	3.000	0.000	3.000
LINTIME	2.725	0.634	3.000	0.000	3.000
CLRMOT	2.690	0.613	3.000	0.000	3.000
MULDIM	1.920	0.759	2.000	0.000	3.000
STRNEM	1.970	1.173	2.000	0.000	3.000
SYMHERO	2.705	0.565	3.000	0.000	3.000
LOGIC	2.840	0.394	3.000	1.000	3.000
CHARGROW	1.845	0.784	2.000	0.000	3.000
IMP	2.680	0.632	3.000	1.000	3.000
MULCONF	1.670	0.796	2.000	0.000	3.000
BUILD	2.565	0.669	3.000	0.000	3.000
LOCKIN	2.530	0.633	3.000	0.000	3.000
RESOLUT	2.195	0.794	2.000	0.000	3.000
LOGICEND	1.470	0.782	1.000	0.000	3.000
SURPEND	1.760	0.973	2.000	0.000	3.000
Word-Factor 1 Score	0.000	0.925	-0.156	-2.028	2.668
Word- Factor 2 Score	0.000	0.900	-0.019	-2.791	3.210
NCHAR	7261.690	4670.0	5765.000	1606.000	27391.000
NWORD	1642.610	1036.0	1294.500	359.000	5815.000
NSENT	101.400	98.9	78.500	5.000	1171.000
PASSIVE	0.099	0.064	0.090	0.000	0.390
CHARPERWD	4.243	0.161	4.200	3.800	4.800

Table 2. Summary Statistics of Predictor Variables.

Variable	Mean	Std Dev	Median	Min	Max
ROI	-0.072	0.916	-0.272	-0.967	6.773
Log(ROI + 1)	-0.424	0.872	-0.318	-3.411	2.051

Table 3. Summary Statistics of Response Variables.

		Predicted	
		< Median ROI	> Median ROI
Actual	< Median ROI	34	6
	> Median ROI	25	16

Table 4. Comparison of classification based on predicted ROI and classification based on actual ROI.

	MSE	MAE	Hit Rate
NONE	0.830	0.720	N/A
Content Only	0.797	0.700	55.6%
Word Only	0.817	0.708	58.0%
Semantics Only	0.799	0.707	54.3%
Content X Word	0.771	0.677	56.8%
Content X Semantics	0.774	0.688	54.3%
Word X Semantics	0.764	0.678	60.5%
All 3 Types	0.752	0.673	61.7%

Table 5. Comparing the model fit by using a subset of all the predictors.

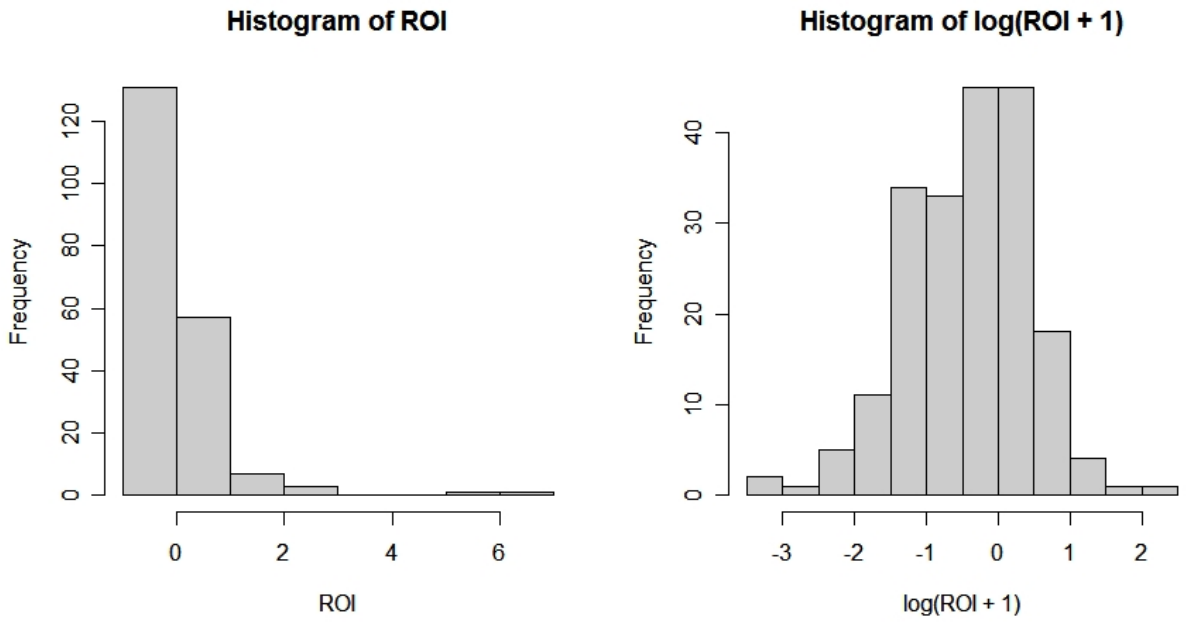

Figure 1. Histogram of ROI and log(ROI + 1).

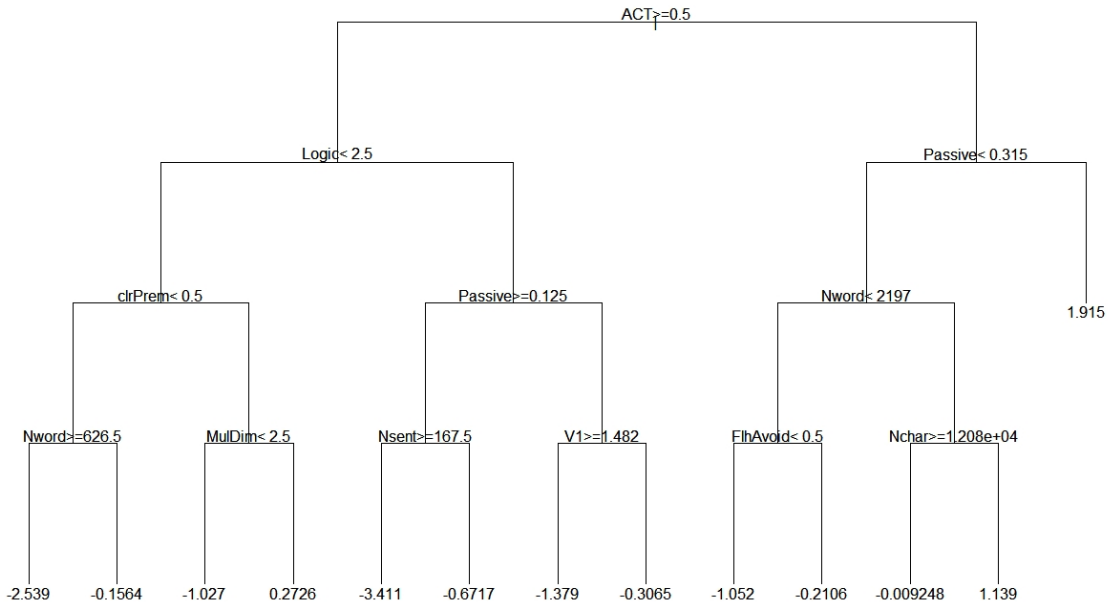

Figure 2. CART model fitted with all 200 training observations. Fitted values are predicted values of log(ROI + 1).

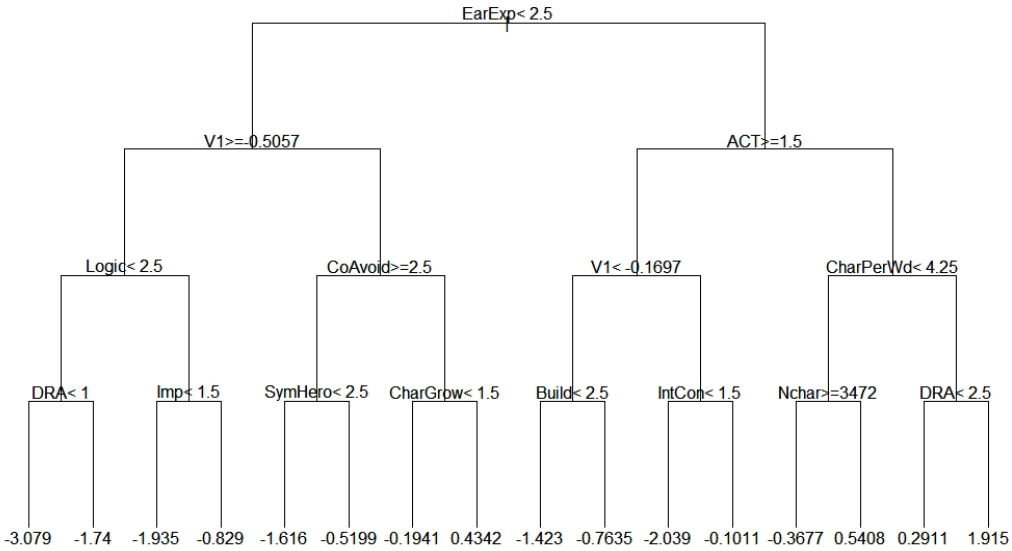

Figure 3. An example of CART model built from bootstrapped sample.

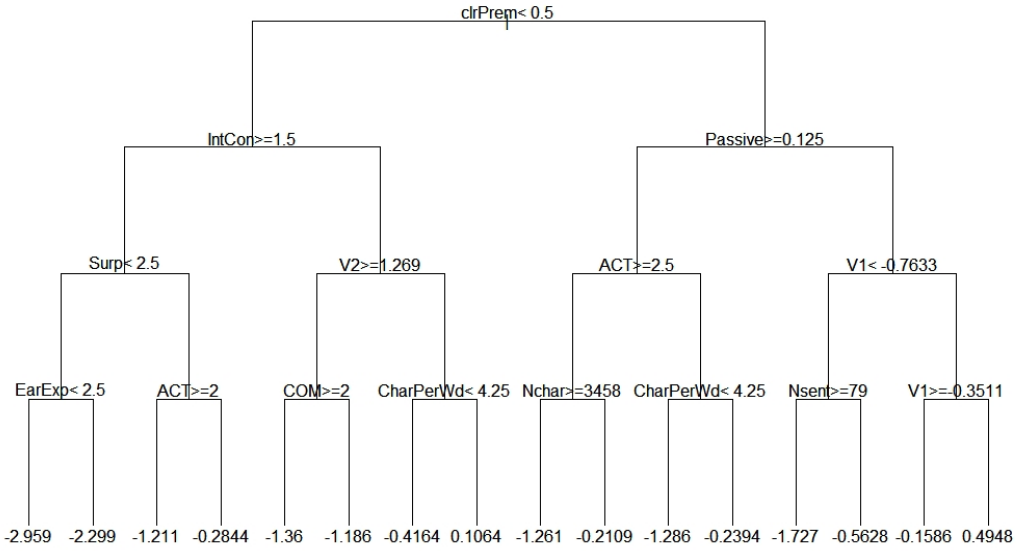

Figure 4. Another example of CART model built from bootstrapped sample.

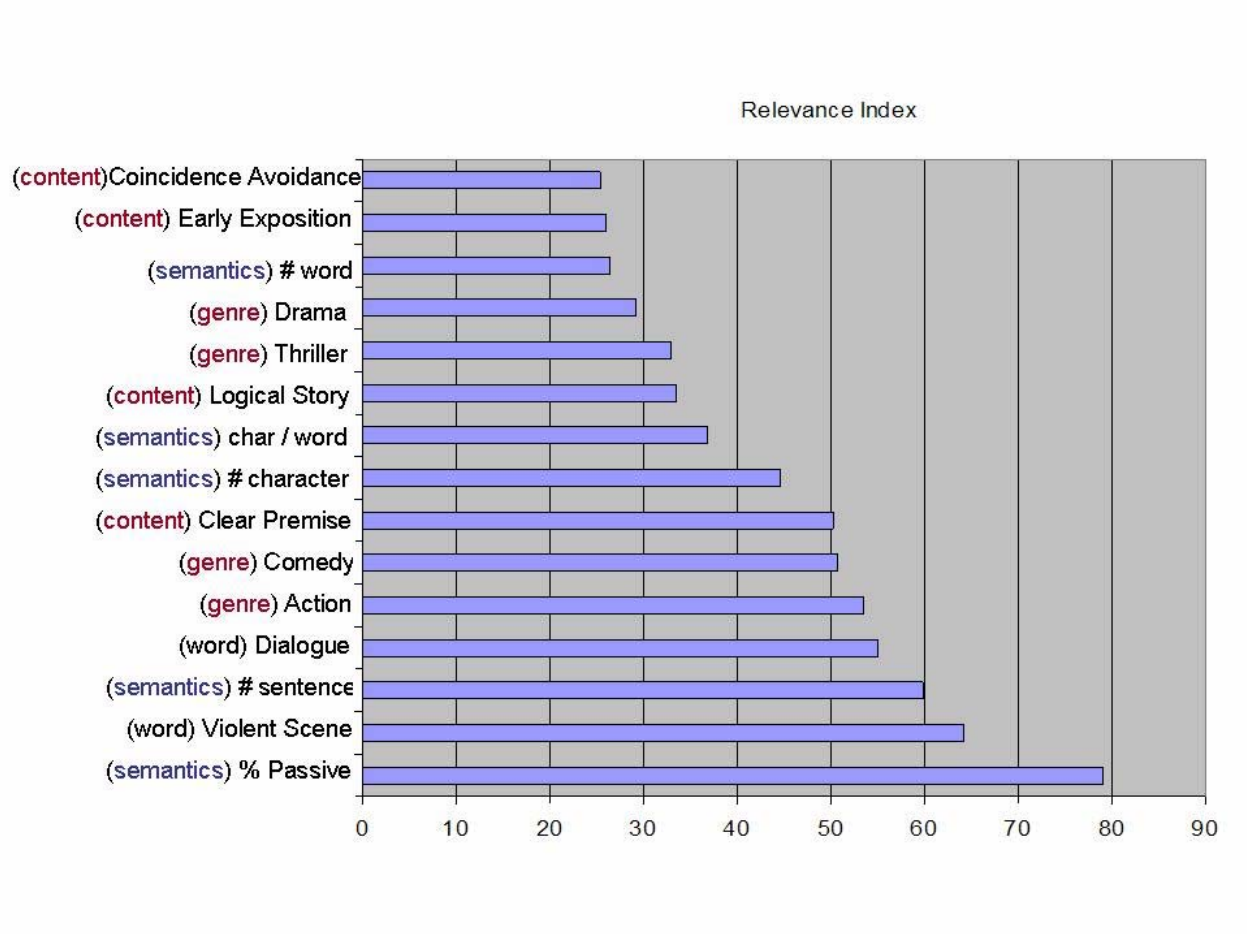

Figure 5. Relevance Index of the 15 most “relevant” predictors.

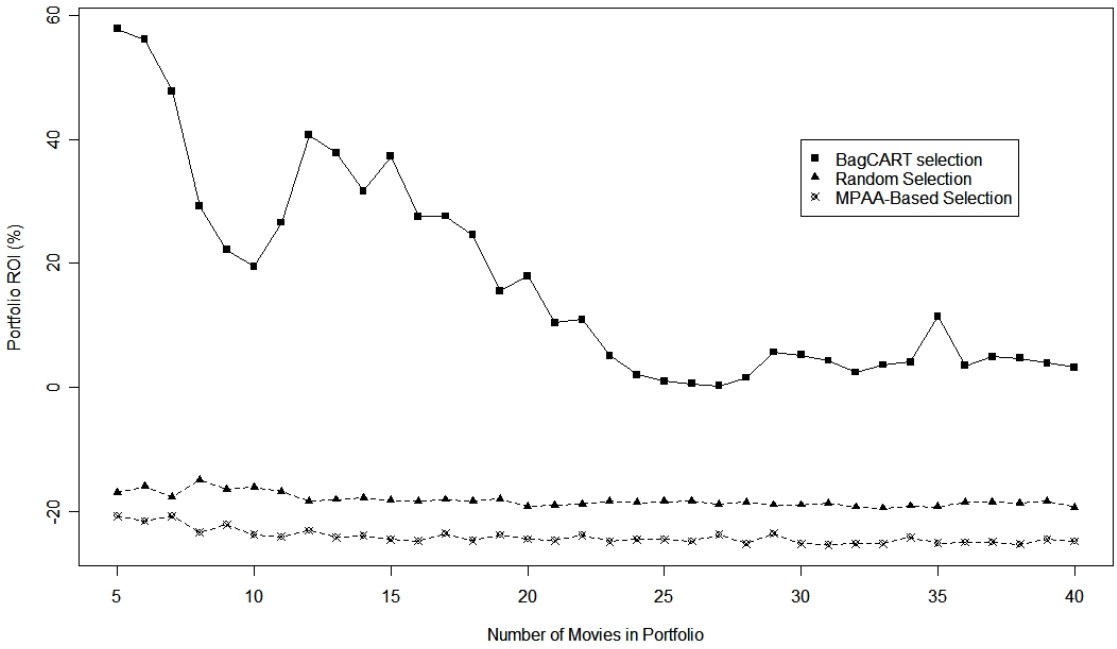

Figure 6. Comparison of ROI of a randomly-selected portfolio with a portfolio selected using Bag-CART model.