Case Study on Rotten Tomato Movie Review

Sharvil Vilas Turbadkar, Yueyuan He, and Jian Jian

Dash App Link-https://movie-feedback-analysis.herokuapp.com/

IST 707 Project Group 2,iSchool Syracuse University

Introduction

Rotten Tomatoes staff first collect online reviews from writers who are certified members of various writing guilds or film critic-associations. To be accepted as a critic on the website, a critic's original reviews must garner a specific number of "likes" from users. Those classified as "Top Critics" generally write for major newspapers. The critics upload their reviews to the movie page on the website and need to mark their review "fresh" if it's generally favorable or "rotten" otherwise. It is necessary for the critic to do so as some reviews are qualitative and do not grant a numeric score, making it impossible for the system to be automatic.

The website keeps track of all the reviews counted for each film and calculates the percentage of positive reviews. Major recently released films can attract more than 400 reviews. If the positive reviews make up 60% or more, the film is considered "fresh", in that a supermajority of the reviewers approve of the film. If the positive reviews are less than 60%, the film is considered "rotten". An average score on a 0 to 10 scale is also calculated. Their opinions are also included in the general rating.

When there are sufficient reviews, the staff creates and posts a consensus statement to express the general reasons for the collective opinion of the film.

When a film or TV show reaches the requirements for the "Certified Fresh", it is not automatically granted the seal, but is instead flagged for the staff's consideration. Once the team assesses the reviews and response to the film or TV show and decide that it is unlikely that the score will fall below the minimum requirements in the future, they will then mark it as "Certified Fresh".

The Dataset

Data has been scraped from the publicly available Kaggle Website.

datasets. Rotten tomatoes is a review-aggregation website for movies. From best to worst, the rating consists of ‘certified fresh’, ‘fresh’ and ‘rotten’. Audiences use this website to express their opinions about movies. The reason for us choosing this dataset is it captures all essential attributes we need to build a robust model that predicts how good a model is. It captures the audience as well as the critic score which is essential for capturing the collinearity between the two opinions and make a balanced judgment on how good a movie will be on the basis of various attributes. So, this dataset can

reflect the attitude of the market towards movies.

Business Questions

In this report, we will use different analytic approaches to answer below business questions:

Which attributes will affect the likelihood of watching a movie the most?

Which class of movies has the strongest polarity?

Do critic and audience views match?

Which writers, directors and studios have the strongest polarity?

Quantifying polarity across time to gauge market effects.

Do directors and actors digress from their genre to star in the different labels of movies to change polarity?

Objective

Our objective is to take these data points and try to build a model that is capable of correctly predicting whether a movie would be successful given many attributes like the director who directed it, the writer who wrote the story ,the studio that published the movie .We plan on using many. Our goal is to correctly predict with an accuracy of 90% as in the movie industry domain predicting a movie's success is relatively easier and dependent on intrinsic factors related to the domain than in other domain where factors are more extrinsic and unpredictable like weather prediction

Performance measures would be computed across both the datasets to estimate how good our model will perform. Incorrectly predicting a movie to be rotten is similarly disastrous than incorrectly predicting a movie to be fresh as both will result into financial losses thus highlighting the need of focusing on all three parameters being precision, recall and accuracy.

Any statistical model results that are confusing to be represented can be represented using descriptive statistics. Descriptive statistics will be used to convey stories through visual insights thus systematizing the raw dataset in detail

Previous Work

Many different researches have been carried for the prediction of movies by using different approaches like document mining using news, articles, blogs, articles and social media. but quite a few have explored the domain and addressed the problems in detail along with solving them to give a broad-based perspective of underlying problems. Furthermore, we have deployed market basket analysis to capture factors with highest support for a given antecedent. Sentimental analysis was also carried using VADER lexicon to capture raw polarity of critics the idea of mining movie dataset was diversified to consolidate and cover many hidden insights and communicate stories through use of rich plots. Another similar work has been presented in [1] where social media including twitter and YouTube's comments are used for same purpose. [2] Presents prediction of popularity of a

movie by the articles on Wikipedia. The research shows that these articles can be used to get some future outcomes. It also uses financial data of movies from box office mojo by using Pearson's correlation coefficient and linear regression. There is a research that predicts the opening weekend revenue. It takes the movie information like actors, director, genre and released date etc. from Metacritic and financial data like budget, opening week gross revenue from the numbers. The pre-released articles about the movies are collected from seven different articles sources. [4] Predicts the gross income of movies using forecasting techniques to map how a movie would gross [5] Uses IMDb data and data from 'boxofficemojo'. They applied PART and C4.5 and used correlation coefficient as a measure, they have created 2 dataset of Pre-release movies and post release movies and applied experimentation on it. [3] Uses IMDb data, rotten tomatoes and Wikipedia data about the movie and machine learning algorithms are applied on it like linear regression, SVM regression and logistics regression.

Exploratory Data Analysis

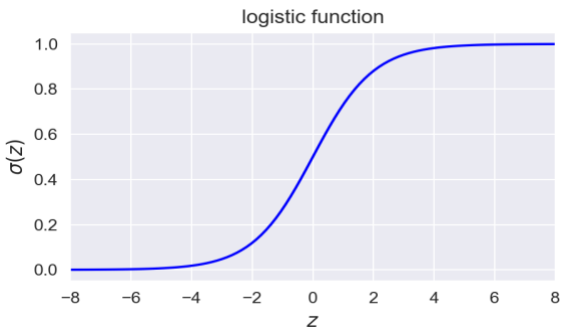
To model categorical variables, we use one-hot encoding. Since we have 3 tomatometer status, we create 3 'dummy' variables that are set to 0 or 1. Every dummy column is assigned one of the 3 categories and is given the value '1' for rows of that category, and '0' otherwise. This is very efficient when we want numerical columns for modelling purposes that only accept quantitative values. The correlation heatmap generated during EDA shows high correlation amongst many columns within the data frame that introduce the problem of multicollinearity. We tried to tackle this problem by standardizing the data. StandardScalar(). The output of the model $y=\sigma(z)$ can be interpreted as a probability y that input z belongs to one class (t=1), or probability $1-y$ that z belongs to the other class (t=0) in a two class classification problem. The output of the model $y=\sigma(z)$ can be interpreted as a probability y that input z belongs to one class (t=1), or probability $1-y$ that z belongs to the other class (t=0) in a two class classification problem.

This centers the data around zero and reduces the impact of high values like runtime in minutes audience count and reduces the effect of other imperative attributes like tomatometer status and other dummy variables by standardizing them from 0 to 1.We imputed the missing values by using median captures the value of central tendency without being prone to the effect of outliers. We also had to winsorize columns like movie runtime as they exceeded a certain threshold of 773 minutes which was the

runtime for the longest running movie ever made.

Model Evaluation

Our aim in this work was to examine and hyper tune the values for algorithms to predict the success of the movie. We made use of Logistic regression that makes use of a binary cross entropy function to predict two values by computing the log of odds ratio. The philosophies behind these three algorithms are quite different, but each has been shown to be effective in finding. The goal is to predict the target class t from an input z. The probability P(t=1|z) that input z is classified as class t=1 is represented by the output y of the logistic function computed as y=σ(z) The logistic function σ is defined as:σ(z)=1/1+e−z.The output of the model y=σ(z) can be interpreted as a probability y that input z belongs to one class (t=Movie is critically successful), or probability 1−y that z belongs to the other class (t=Movie is Rotten) in a two class classification problem.



Ensemble Models

Ensemble modeling is a process where multiple diverse models are created to predict an outcome, either by using many different modeling algorithms or using different training data sets. The ensemble model then aggregates the prediction of each base model and results in once final prediction for the unseen data. The motivation for using ensemble models is to reduce the generalization error of the prediction. Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction the low correlation between models is the key. Just like how investments with low correlations come together to form a portfolio that is greater than the sum of its parts, uncorrelated models can produce ensemble predictions that are more accurate than any of the individual predictions. Our purpose of using random forest is to emphasizes feature selection and find which attributes contribute the most towards estimating the value of the target variable. We can determine how well our model predicts movie success at every step by using criterions like gini.

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

The root node shows the gini index of the whole data set, prior to branching. The lower the gini score, the purer the data is. The worst-mixed data would give a gini index of 0.5.

To refresh, there are 144 rotten reviews and 111 critically fresh reviews in our data. The Gini Index for this would be 0.492 which means it is very mixed. But don't worry, the tree will lower the gini indices as new branches and nodes are formed.

Gini-Index=$1-[(144/255)_2)+(111/255)_2]=$ 0.4916

Dull Studio	
YES	NO
144	111

Since audience_rating is the most important feature according to feature importance we will put it at top and test every combination until we get gini as low as possible. We continue this process using cross validation where we iterate over a combination of dataset chunks till we get the highest accuracy.

The basic idea of the gradient boosting decision tree is combining a series of weak base classifiers into a strong one. Different from the traditional boosting methods that weight positive and negative samples, GBDT makes global convergence of algorithm by following the direction of the negative gradient

Let $\{x_i,y_i\}_{ni=1}$denote the dataset. Softmax is the loss function. Gradient descent algorithm is used to ensure the convergence of the GBDT. The basic learner is h(x), where $x_i=(x_{1i},x_{2i},\ldots,x_{pi})$p is the number of the predicted variables. $y_i$is the predicted label. The steps of GBDT are presented as follows:

Step 1:The initial constant value of the model β is given:

$$F0(x)=\text{argmin}\beta\sum i=1NL(yi,\beta)$$

Step 2:For the number of iterations m=1:M(M is the times of iteration), the gradient direction of residuals are calculated.

$$y_{i*}=-[\partial L(y_i,F(x_i))/\partial F(x_i)]_{F(x)-Fm-1(x)},i=\{1,.,n\}$$

Step 3:The basic classifiers are used to fit sample data and get the initial model. According to the least square approach, parameter a_m of the model is obtained and the model $h(x_i;am)$ is fitted which in this scenario is our training tomatometer model

Step 4: Loss function is minimized. According to Eq. (4), a new step size of the model which is the learning rat of the model for which we have tuned the hyperparameters in our app the current model weight, is calculated

$$\beta_m=\text{argmin}_{\alpha,\beta}\sum_{i=1N}L(y_i,F_{m-1}(x)+\beta h(x_i;a))$$

Step 5:the model is updated as follows

$$F_m(x)=F_{m-1}(x)+\beta_m h(x_i;a)$$

However, limited to the dimension and size of the sample data, information gain of feature branch points is needed to be calculated multiple times when raw data is input into GBDT to be analyzed. It leads to an increase of the iteration number and slows the speed of convergence and update. In this paper, we propose to optimize initial data which is input into GBDT by using ABC.

For Information gain if we consider this example

Dull Studio	
YES	NO
144	111

Information gain:-
(144/111)*log((144/111),2)-
(111/144)*log((111/144),2)
=>-(0.3755)-(37551)
=>**0.75 is the entropy loss**
We negate this by the entropy value from root node to get information gain

KNearest Neighbors

k nearest neighbor (KNN) method is a popular classification method in data mining and statistics because of its simple implementation and significant classification performance. However, it is impractical for traditional KNN methods to assign a fixed k value (even though set by experts) to all test samples. Previous solutions assign different k values to different test samples by the cross-validation method but are usually time-consuming. This paper proposes a KTree method to learn different optimal k values for different test/new samples, by involving a training stage in the KNN classification .KNN uses Distance Metrics to know the input data pattern in order to make any Data Based decision. KNN Model helps in improving the performance of Classification and Clustering of target variable which is the tomatometer status in this scenario. In K is the number of nearest neighbors of a test data point. These K data points then will be used to decide the class for test data pontin K-means, we select the number of centroids that define the number of clusters. Each data point will then be assigned to its nearest centroid using distance metric (Euclidean or, Manhattan or any other distance metric suitable for the dataset).

We use Manhattan Distance if we need to calculate the distance between two data points in a grid-like path. As mentioned above, we use the Minkowski distance formula to find Manhattan distance by setting *p's* value as *1*.

$$d = \sum_{i=1}^{n} |x_i - y_i|$$

Euclidean distance is one of the most used distance metrics. It is calculated using the Minkowski Distance formula by

setting p's value to 2. This will update the distance 'd' formula as below :

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Association Rule Mining

Association rules are used for generating candidate item sets which are the most frequently occurring consequences and antecedents to generate key insights from the data .We are using the concept of apriori to analyze key parameters that lead to a movie being rotten or critically fresh. Frequent item sets are the ones which occur at least a minimum number of times in the transactions. Technically, these are the item sets for which support value (fraction of transactions containing the itemset) is above a minimum threshold .

Apriori principle helps in making this search efficient**.** It states that all subsets of a frequent itemset must also be frequent. This is equivalent to saying that the number of transactions containing items {Famous_Genre,FreshMovie} is greater than or equal to the number of transactions containing{Famous_Genre,FreshMovie, Famous_Studio}

This is called the anti-monotone property of support. Just like the anti-monotone property of support, confidence of rules generated from the same itemset also follows an anti-monotone property. It is anti-monotone with respect to the number of elements in consequent. This means that confidence of (A,B,C→ D) ≥ (B,C → A,D) ≥ (C → A,B,D). To remind, Confidence for {X→Y} = support of {X,Y}/support of {X}.

Rules are formed by the binary partition of each itemset. If {Famous_Genre,, Famous_Studio,FreshMovie} is the frequent itemset, candidate rules will look like:

(Famous_Genre,Famous_Studio}→(FreshMovie),

(Famous_Genre→ FreshMovie,Famous_Studio).

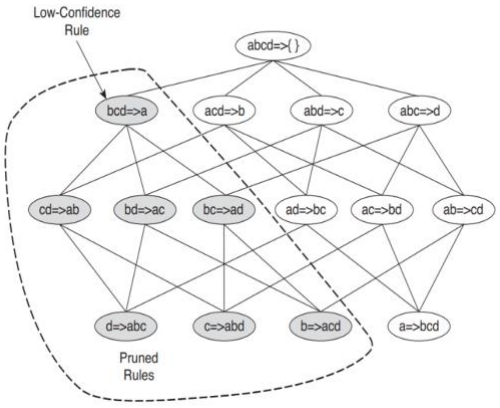
Consider F1 and F2:

F1=Fraction of transactions having(FreshMovie) also having all the item sets

F2 = fraction of transactions having (Famous_Genre,Famous_Studio) also having all the itemsets

So it will be observed that F1 < F2.

Using this property of confidence, pruning is done in a similar way as was done while looking for frequent item sets.



With these two steps, we have identified a set of association rules which satisfy both the minimum support and minimum confidence condition. The number of such rules obtained will vary with the values of *minsup* and *minconf*. Now, this subset of rules thus generated can be searched for highest values of lift

$$Lift(X \rightarrow Y) = \frac{support(X \cup Y)}{support(X).support(Y)}$$

Sentimental Analysis

So, what exactly is a sentiment? Sentiment relates to the meaning of a word or sequence of words and is usually associated with an opinion or emotion. And analysis? Well, this is the process of looking at data and making decisions. For analyzing critic reviews it was imperative on our part to perform sentiment analysis to retrieve the polarity of emotion generated from the corpus of data. VADER has been found to be quite successful when dealing with social media texts, NY Times editorials, movie reviews, and product reviews. This is because VADER not only tells about the Positivity and Negativity score but also tells us about how positive or negative a sentiment is. VADER has a lot of advantages over traditional methods of Sentiment Analysis, including:

It works exceedingly well on social media type text, yet readily generalizes to multiple domains. After applying the vader lexic onto our corpus of critic consensus data we found our corpus to be 37% positive ,62% neutral and 0% negative which add up to one .The Compound score is a metric that calculates the sum of all the lexicon ratings which have been normalized between -1(most extreme negative) and +1 (most extreme positive). \Using upper case letters to emphasize a sentiment-relevant word in the presence of other non-capitalized words, increases the magnitude of the sentiment intensity. For example, “The movie is a remarkable feat of achievement” conveys more intensity than “The movie is a REMARKABLE feat of achievement”

Vader also has the capability of handling slangs, emoticons and conjunctions like but that reduce the intensity of the polarity

For example: The movie was a knockout for all the action films of this year has a very high strong polarity.

Results

The results generated by the models are very intuitive and have a string accuracy across all models for both the training and test datasets in comparison to that of baseline models

As mentioned due to the problem of multicollinearity the model would tend to

overfit. However in order to avoid this winsorized the values.

(R1,p): array function which returns a column range which is the Winsorized version of R1 replacing the lowest and highest 100*p*/2% of the data values. Standardization also reduced the impact of heavy values and tried tackling the issue of multicollinearity.

Model	Precision	Recall	Accuracy	ROC-AUC
KNN	0.9038	0.91903	0.903204	0.958111
Logistic	0.875505	0.839535	0.839341	0.92258
Random_Forest	0.901374	0.883168	0.880062	0.839695
Gradient_Boosting	0.900162	0.875393	0.874499	0.949661

As we can see from the model Logistic Regression tends to perform much better at predicting whether the movie will be a rotten or a certified fresh movie as the binary cross entropy function denoted below will classify the positive and negative points using this function

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

L1 penalty equal to the absolute value of the magnitude of coefficients

L2 penalty equal to the square of the magnitude of coefficients.

After tuning for other factors we like solver and maximum iterations we tuned grid searched our model to get the best fit .Similarly using feature selection we understood the attributes that gave the best split and accordingly computed base learners and boosted these individual learners using entropy and gini criteria that decide the best split and for which threshold. In the scenario of numeric variables we split by criteria's like movie runtime greater than 150 minutes and being shorter than 150 minutes. We also used max iterations till it converged to a minimum loss.

Similarly in random forest and gradient boosting we tuned multiple hyperparameters like maximum number of features to be considered while making the split using information gain which can be computed from entropy loss at every step

$$E = -i\sum Cpi \log 2 pi$$

Information Entropy can be thought of as how unpredictable a dataset is.

A set of only one class (say, blue) is extremely predictable: anything in it is blue. This would have low entropy.

A set of many mixed classes is unpredictable: a given element could be any color! This would have high entropy.

We also assigned a high max depth for the tree to expand as further down as possible. Similarly we also took into account different number of base estimators that will be used for boosting the model to minimize Out of Bag error. For KNN Model we computed the error graph for every neighbor from 1 to 35 by computing Euclidean and Manhattan distance

$$d(x,x') = \sqrt{(x1 - x'1)2 + \ldots + (xn - x'n)2}$$

and finding the right number for k. After computing this we can fit and transform on the training dataset

References

[1] Nithin VR, Pranav M, Sarath Babu PB, Lijiya "A Predicting movie success based on IMDB data" International journal of data mining and techniques, Volume 03, June 2014, pages 365-368

[2] Andrei Oghina, Mathias Breuss, Manos Tsagkias, and Maarten de Rijke , “Predicting IMDB Movie Ratings Using Social Media”, Advances in Information Retrieval , Volume 7224, 2012, pp 503-507

[3] Mestya´n M, Yasseri T, Kerte´sz J (2013): “Early Prediction of Movie Box Office Success Based on Wikipedia Activity Big Data”. PLoS ONE 8(8): e71226.doi:10.1371/journal.pone.0071226

[4] Wenbin Zhang ,Steven Skiena :”Improving Movie Gross Prediction Through News Analysis”, Department of computer science stony brook university, 2009 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology – Workshops, Pages 301-304

[5] Khalid Ibnal Asad , Tanvir Ahmed , Md. Saiedur Rahman: “Movie Popularity Classification based on Inherent Movie Attributes using C4.5,PART and Correlation Coefficient”, IEEE/OSA/IAPR International Conference on Informatics, Electronics & Vision, Pages 747 - 752