Behind the TV Shows:
Top-Rated Series Characterization and Audience Rating Prediction

Team 256: Yushu Chai, Yiwen Xu, Zihui Liu

1. Introduction

The television production industry has maintained its vitality over the past few years. TV broadcasters make profits by selling time to advertising agencies based on the projected audience sizes. Failure to predict a TV series' popularity to audience could result in substantial losses to either broadcasters or advertisers. Therefore, both TV broadcasters and advertisers are under pressure to make wise investment decisions about TV series prior to their release. Although a large number of research papers are focused on exploring the factors affecting movie ratings, very few studies have looked into TV series. While movies and TV series have similarities, TV series are different from movies in many aspects that worth investigating. The Internet Movie Database (IMDb) is a comprehensive online database that has a high degree of interactions with users, making it a fertile source of machine learning problems [2]. In this project, our team endeavor to build several supervised learning models to predict the popularity of TV series using viewer ratings on IMDb as an indicator, and then used unsupervised learning to investigate the key features of the TV series. These key features obtained by unsupervised learning steps were then be used to improve our current prediction models.

2. Related Work

Previous relevant research is mostly focused on the prediction of movie ratings and avenues. Given the similarity between movie and TV series, we employed some of the machine learning algorithms from these studies and made adjustments based on the unique problems we explored. There are two major approaches in movie ratings and revenue prediction: one utilizes sentiment analysis, and the other runs linear regression and K-means clustering on the database.

In the first approach, some studies [3][4][6] applied sentiment analysis and classification using the text data posted by viewers. Comments on social media are information that directly shows viewers' preferences. These algorithms could do a better job than linear regression. However, pre-processing the data is quite time consuming. According to these studies, the authors spent a great deal of time correcting grammar errors and formatting the texts. In the second approach, Apala [1] and Joshi [6] used K-means clustering and regression models. Although both algorithms are easy to implement and interpret, the error rate could be relative high (with R^2 around 0.5).

3. Dataset and Features

The TV series data recorded on IMDb websites were extracted and provided by Andrej Krevl from the Stanford SNAP Research Group. We filtered and cleaned the data by excluding the TV series with very small numbers of ratings, say less than 10 viewer ratings, and those with missing feature information for the accuracy of prediction. Features of the data set include the genre (vector of Booleans), release year (numeric), number of seasons (numeric), number of episodes for each season (numeric), number of ratings (numeric), number of critics (numeric), number of reviewers (numeric), runtime (numeric), aspect ratio (categorical), and color (Boolean). We randomly selected 450 complete samples from the data set for testing and training purposes.

4. Methods

To predict the TV show ratings, our first intuition was to use regression models. The very basic model is thus the multiple linear regression, with all features, whether numeric or categorical, included in the hypothesis. To improve the regression model, we have also fitted a locally weighted linear model and a reduced linear model by backward search and principal component analysis. Another idea is running a classification model, where we segmented the ratings into several groups and utilized logistic regression for classification. The segmentation was then revised by K-means clustering.

4.1 Supervised Learning

4.1.1. Model I: Multiple Linear Regression

The hypothesis of the linear regression used here is

$$rating = h(x) = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n = \sum_{i=1}^{n} \theta_i x_i,$$

where θ_i are the coefficients that we are seeking, and $h(x)$ is the output used to calculate MSE. We used stochastic gradient descent to minimize the cost function and obtained the θ_i's.

4.1.2. Model II: Locally Weighted Linear Regression

To improve the accuracy of the prediction model and make the choice of features less influential, we performed a locally weighted linear regression model to minimize

$$\sum_{i=1}^{m} w_i \, (y_i - \sum_{j=1}^{n} \theta_j x_j)^2, \text{ where } w_i = \exp\left(-\frac{(x_i - x)^2}{2\tau^2}\right).$$

The w_i's are non-negative valued weights. A large w_i means that θ will be picked to make $(y_i - \sum_{j=1}^{n} \theta_j x_j)^2$ small. The bandwidth parameter τ, which controls how quickly the weight falls off with distance of its x_i from x, was chosen to minimize the test error.

The locally weighted linear regression is a non-parametric method which, unlike the unweighted linear regression, cannot reduce the training set as it proceeds, because every time a query is obtained at a new location, the entire training set is needed to determine what the local neighborhood is.

4.1.3. Model III: Linear Regression with Selected Features and Interaction Effects between Original Features

We suspected that some of the features in the original linear model were irrelevant to the learning task, and overfitting might thus increase the estimate of generalization error. Therefore, as another way of improving the linear model and reduce the dimension of the feature matrix, backward search was performed starting with the full sets including all possible interaction terms. For each iteration of the backward search, we did cross validation to evaluate the model, where the measurement is the Akaike Information Criterion (AIC), which is

$$AIC = 2k - 2\log(L),$$

where k is the number of estimated features in this model, and L is the maximum of the likelihood function. At each step, we deleted a feature whose elimination gave the greatest reduction in AIC value.

4.1.4. Model IV: Classification Using Logistic Regression

By using logistic regression, we predicted the probability of the rating being 10 (the highest possible rating on IMDb), and label this observation as Fair (rating below 8), Good (rating between 8 and 9), and Very Popular (rating above 9). The cost function of logistic regression is

$$h(\theta) = \frac{1}{1 + e^{-\theta^T x}}$$

4.2. Further Improvement I: Principal Component Analysis (PCA)

PCA projects the normalized data onto a space where the decrease of variance is the maximum. For the input x, we would like to find a unit-length vector u to maximize:

$$u^T \left(\frac{1}{m} x_i x_i^T\right) u$$

the u_k's are the principle components (PC). We then used the PC's to revise our linear regression.

4.3. Further Improvement II: K-means clustering

To find a better and more reasonable segmentation of the ratings, we performed K-means clustering on the training set with K = 3, 4, and 5. After randomly initializing the K centroids randomly, the K-means clustering algorithm iterates on assigning each point to the closest centroid and letting the new centroid be the mean of the points in this cluster, until there is a convergence. We utilized the clustering result for the classification model.

5. Results and Discussion

To predict the ratings and evaluation our regression models, we used 5-fold cross validation. We divided the entire dataset in 5 equal-length subsets. Each time we held one of the five as the test set, and trained on the other four as a whole. For each training set, we built a multiple linear regression model, and made predictions on the test set to obtain the mean squared error (MSE):

$$MSE = \frac{1}{m-n-1}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2,$$

where $\boldsymbol{m}$ is the number of observations in each training set, and $\boldsymbol{n}$ is the total number of predictors in the regression model; y_i is the ratings of the test data; $\hat{y}_i$ is the predicted rating, i.e., output value. The average of the MSE for each step of the 5-fold cross validation is our estimate of the generalization model.

The logistic regression model is evaluated by the hold-out cross validation approach with ½ of the dataset for training, and another ½ for test. The misclassification rate is used as a measurement.

$$Misclassification\ Rate = \frac{\#Data\ points\ incorrectly\ classified}{Total\ \#\ data\ points}$$

The full linear regression model gave an error estimate of 0.3736. It is still necessary to diagnose the model:

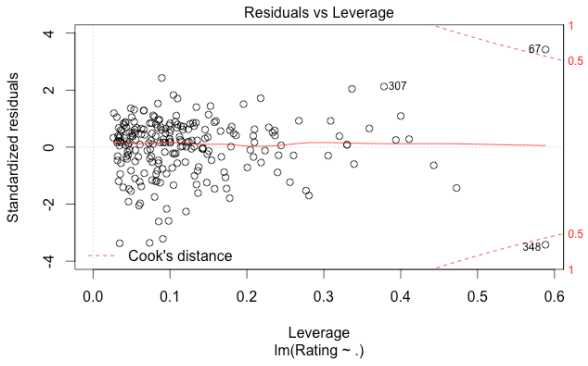

Figure 1. Residual vs. Leverage.

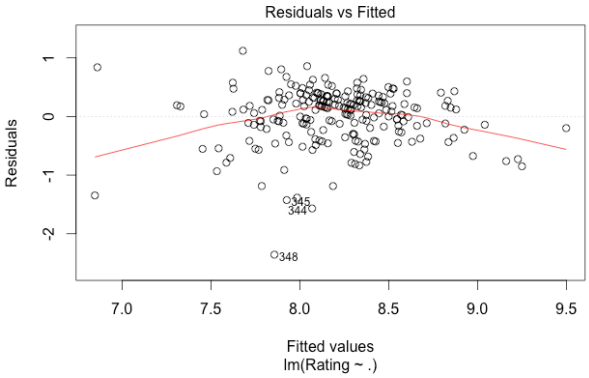

Figure 2. Residual vs. Fitted Value.

While Figure 1 shows that there are only two points of high leverage, which is acceptable, the pattern in the Figure 2 shows that the assumption of constant variance of error terms is not perfectly satisfied, so it is necessary to improve the linear model by locally weighted regression.

The estimate of generalization error given by the locally weighted regression is 0.3398, which is slightly lower than the error of the linear model. This means that the problem of non-constant variance is, to some extent, addressed by this approach. But since the problem of the variance of error terms is not significant, as shown by Figure 2, this locally weighted regression model does not significantly improve the linear model.

Considering the tedious input features $x_i's$ used in our previous two models, we attempted to perform dimension reduction by backward search feature selection, using PCA as a comparison. The following plot (Figure 3) keeps track of the AIC's of the process of backward search. The model at the last step gives the lowest AIC, and also the most reduction in estimated generalization error, i.e., 0.3321.

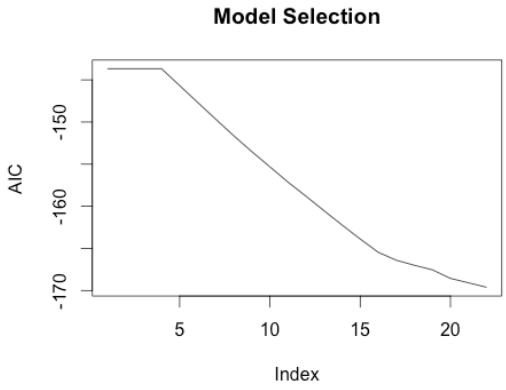

Figure 3. Feature Selection Process.

An alternative method we used for dimension reduction is PCA. The plots (Figure 4) are projections of the data onto the first three principle components, i.e., the scores for the first three principal components. The observations of different subgroups lie near each other in the low-dimensional space.

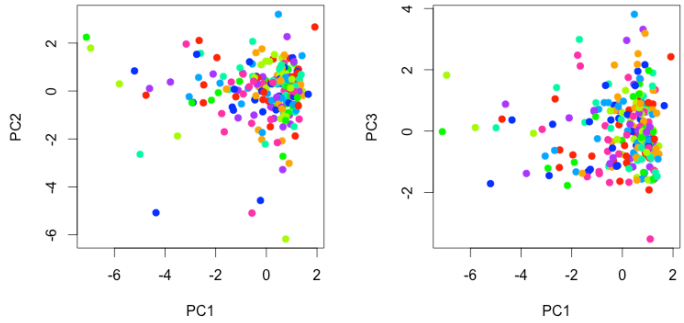

Figure 4. 3-D PC projection.

We then used the first k (k = 2, 3, …, 10) PC's to fit linear models, and the estimated generalization error of different number of PC's is plotted below in Figure 5.

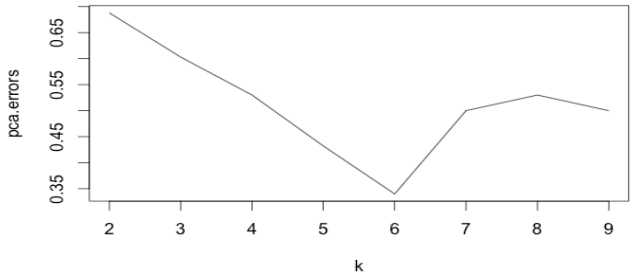

Figure 5. Estimated Errors at Different # PC's.

When k = 6, the lowest estimated error was 0.3399, which was lower than the full linear model, and close to the error by the feature selection method.

Table 1. Comparison Between Regression Models

Linear regression models	**Features included***	**Estimate of generalization error**
Linear regression	All	0.3456
Locally weighted linear regression	All	0.3398
Linear Regression with Selected Features and Interaction Effects between Original Features	Without 'Run Time', 'Aspect Ratio' and some of the genre features. **	0.3321
Linear regression using the first six PC's	The first six PC's	0.3399

Although the methods above reduce the estimated error of the original linear regression model, the improvement is not very significant. Thus, we also considered classification model, which gave a rough range into which the predicted rating may fall, and could be easier to interpret and use by TV show producers or investors.

Based on the current dataset, we segment the ratings into several categories: Fair (rating below 8), Good (8-9), and Very Popular (9-10), to fit a classification model using L1-regularized logistic regression. When predicting the ratings of an upcoming TV show, this classification error can give a category as reference for the TV show producers.

Table 2. Confusion Matrix for Classification Model

Original Rating	Class	Counts	# correctly assigned to this group	# misclassified from/to other groups	Total # misclassified	Test misclassification rate
< 8	Fair	65	50	15	36	0.16
8-9	Good	148	130	18		
9-10	Very Popular	12	9	3		

The confusion matrix above gives very good misclassification rate, and since the classification model is more straightforward and makes more practical sense, we recommend that it is used if TV show producers and investors would like to predict their product's popularity.

Furthermore, since the segmentation in the original logistic regression model is based on our intuition, we also use the K-means clustering result (K = 3, 4, and 5) to divide them into subgroups. Figure 6 is an example of K = 3.

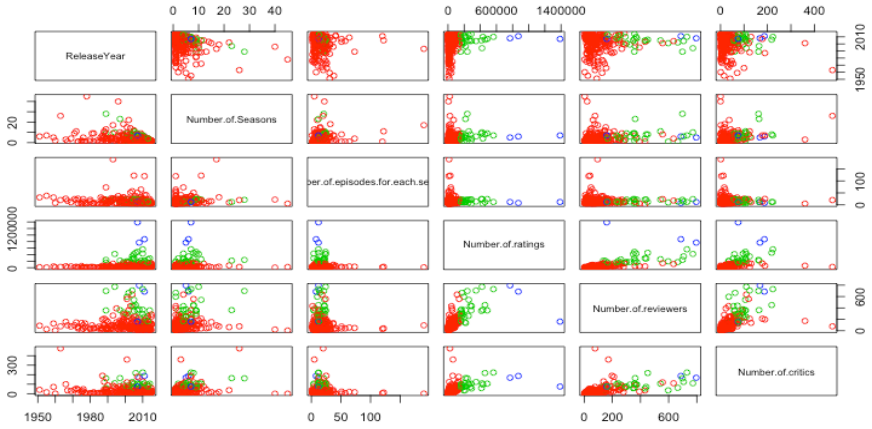

Figure 6. K-means clustering when K = 3.

For each value of K, we calculate the lowest and the highest ratings for the K clusters, and use them as the cutoffs of the classes in the logistic regression. When K = 3, the test misclassification rate is the lowest, which is about 0.18. The ranges are: below 0.75, 0.75 – 0.88, and above 0.88, as each range represents the lowest and highest ratings of that cluster. However, the logistic regression model after K-means clustering does not give a better misclassification rate.

6. Conclusions and Future Work

This project mainly utilized regression models to predict viewer's ratings of TV series based on the existing IMDb database. In particular, the classification model with ratings divided into three subgroups provides the best outcome and is recommended for prediction, although the outcome falls in a relatively wide range. Nevertheless, linear regression using selected features, either by using backward search or PCA, provides improved results compared to linear regression with all available features. If given more time, we will try using more sophisticated models to run on the data, for example, neuro network and kernel; we will also analyze the TV series ratings based on different time spans to see how the importance of features changes over time.

7. References

[1] Apala, Krushikanth R., et al. "Prediction of movies box office performance using social media." *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*. IEEE, 2013.

[2] Augustine, Achal, and Manas Pathak. *User rating prediction for movies*. Technical report, University of Texas at Austin, 2008.

[3] Brendan O'Connor, et al. "From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series." *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 2011

[4] Jain, Vasu. "Prediction of Movie Success using Sentiment Analysis of Tweets." *The International Journal of Soft Computing and Software Engineering* 3.3 (2013): 308-313.

[5] M. Joshi, D. Das, K. Gimpel, and N. A. Smith. "Movie reviews and revenues: An experiment in text regression". *NAACL-HLT*, 2010.

[6] Oghina, Andrei, et al. "Predicting imdb movie ratings using social media." *Advances in information retrieval*. Springer Berlin Heidelberg, 2012. 503-507.

[7] The Internet Movie Database. Retrieved from: http://imdb.com