On predicting the movie ratings

Tianxin Yu
Carnegie Mellon University
Human-Computer Interaction Institute
{tianxin1}@andrew.cmu.edu

ABSTRACT

In this paper, techniques from machine learning are used to predict IMDb movie ratings based on three sets of attributes: Movie Production Information, Content, and Social Media. This paper is a final report for a class on applied machine learning. It described the workflow to solve a classification problem including data preprocessing, feature space design, algorithm selection, error analysis and feature engineering, feature selection and model optimization. Naïve Bayes, logistic regression, J48 classifier, and SMO were applied and compared in this paper. A final model built on SMO showed an accuracy and Kappa statistics on final test data. The results may be of interest to the movie industry, cinemas, as well as online video platform as Netflix and Amazon to provide better user experience for customers.

Keywords

Move Ratings，Social Network，Classification

1. INTRODUCTION

Movies have become an indispensable part of our life. It is enjoyable and relaxing to see a great movie that you are longing for. However, it happens a lot that we expect a movie for a long time, and it turns out to be disappointing. In the past, there are limited ways to learn about the goodness of a movie before we see it in the theater. This is not surprising, since what shape a good movie are multi-facets [1]. There are just too many factors that affect the goodness of movies; the cast, director, budget, length, genre, etc.

Luckily, the Internet Movie Database (IMDb)[1], the biggest online movie database, provides us a new way to evaluate the goodness of a movie based on users' review, namely, the IMDb score. However, it is still hard to evaluate a movies' greatness from IMDb score for general audiences for two reasons. First, the IMDb score is based on users' review. It will not be reliable until a few months after the movies' release. Some movies can be under or over estimated before it gets popular, for example, *La La Land(2017),* a movie scored 8.3, has been scored around 7.6 during the first few weeks after it released. The second reason is that the IMDb score can be misinterpreted by the general audience, as other platforms such as Netflix provides ratings that are higher than IMDb and the score itself does not include any information regarding its percentile ranking.
As for the first problem, one potential solution is to predict the IMDb score based on information provided by IMDb as well as other platforms. The available attributes include factors directly related to natural of the movie, such as its genres, duration, budget, etc. Also, people involved in the movie are considered, such as its director and cast. In addition to the name of those they people, their popularity on social networking could also contribute to the prediction. The second problem is from the users' perspective. It can be addressed by turning a regression problem into a classification problem, with three class values that represent good, mediocre, and bad movies.

In this paper, a work using information from IMDB and social media to predict movies rating score is presented. The work would be valuable for movie lovers to decide whether they will watch the movie once it released for movie lovers. Also, for the movie industry, it could be used to evaluate what kind of movie will likely to be liked by the average audience.

2. Related Work

It has been over 20 years that researchers from Computer Science field started to predict movie ratings. Earlier work

[1] http://www.imdb.com/

has been done by Armstrong and Yoon in 1990s [2]. They used kernel regression and model trees to predict the IMDb score from information related to movie's production and content. They found that movie genres as "isDrama" and "isHorror" outweigh other features during the feature selection, indicating that the content of the movie was a better predictor the movie ratings than other features. However, they found that a high performance on accuracy was very hard to achieve given the natural that people's ratings are related to their subjective preferences and evaluations on the movies in addition to its genres.

In recent years, as social networking platforms get popular, more research focus on how to use information from online communities to predict movie ratings. Oghina and her colleagues use text post and comments from Tweeter.com and YouTube.com to predict IMDb scores [3]. They used both surface feature such as the number of likes and textural feature such as tweets in their prediction model. By comparing the likelihood of terms, their result showed that signals from different social networks are not isolated. Moreover, signals from social can be effective predictors for movie ratings. Further research conducted by Sitaram Asur and his colleague showed that the tweets sentiments collected two months before and after a movie's release can be used to predict the movie's box office performance [4]. They found that the tweets the amount of positive sentiment compared with negative ones (PNratio= Tweets with Positive Sentiment/ Tweets with Negative Sentiment) are more effect than the overall sentiment score.

Previous research showed how a movie's production information and signals from social media could be used to predict its performance regarding rating and box office. Although previous research used features from only one perspective in their prediction model, it shows great potential that combining features from social media and movie production will be effective in predicting its performance. Also, previous research also shows that features from different channels are not isolated, suggesting that there would be interactions between attributes. To sum up, there are two major limitations of previous research. First, none of these research combined features from social media and movie production to predict movie ratings. Second, some research used linear methods to predict movie's performance, the interaction between features may not be revealed by their models.

In this paper, I hypothesize that the combination of features from movie production and social media can be used to predict movie's rating score, thus provide users effective information on movie selections. Also, given the assumptions that the features in this data set are correlated, I hypothesize that classifier based on features' independence will not work well. Also, there might be interactions between the attributes, non-linear classifiers may outperform linear classifiers.

3. Dataset Description and Preparation

3.1 Dataset Description

The dataset comes from Kaggle.com[2]. It includes 5043+ movies scraped from IMDB website, spanning across 100 years. The initial dataset includes 28 variables, which can be classified into three categories:

- Information related to movies' production such as budget, color, country, cast, etc.
- Information related to movies' content such as genres.
- Information related to social media such as the number of likes for the movie and its director from Facebook.

I used WEKA *StratifiedRemoveFoldsFilter* twice to split data into development set, training set, and test set. After splitting, there are 1009 instances in development set and final test set, and 3027 instances in the training set.

As the first step for data cleaning, I cleared variables such as actors' name, directors' name since it will bring too many features that are in low frequency to the features space. For example, there were 2399 unique director names from 5043 instances. In development dataset, there were not of a single name that has appeared more than five times. Also, according to the source, using names could be inaccurate, since the same names could refer to different persons. As for languages and country, according to the source, there were around 50 unique values for each of the variables. Based on my observation on the development dataset, most of the values for variable "language" only appeared less than five times while English occurred in more than two-thirds of the instances. Therefore, I

[2] https://www.kaggle.com

transformed all non-English languages to value “others.” Similarly, I saved “USA” and “UK” as values for “country” and transformed all other countries into “others.”

3.2 Preparation

Further data cleaning was performed to correct system errors that came from data collection. More specifically, missing value and incorrect data.

Missing Value

In this dataset, missing values were represented as missing or zeroes. It could result from lacking information on IMDB or no response returned by scrappy HTTP request within a given timespan (<0.25 second). To deal with these records, a method introduced by Nick Armstrong and his colleagues was used [2]. All missing values were replaced with the mean value of the attribute over the training set.

Calibration

Some incorrect data also come from the ununiformed use of currency unit. For example, movies from Korean used both KRW (Korean Won) and US dollars for their budget value. It is hard to determine which currency unit was used for a movie from countries other than USA and UK. Since those countries occupied only less than 10% of the instances, I cleared their budget values and replaced them with the mean value of the attribute over the training set.

Feature Scaling

Many attributes were not following the Gaussian distribution in the dataset. Thus, I used unity-based normalization instead of Z score [5]. Feature scaling was used to transform all values from numeric variables into the range [0,1] with the following formula:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

3.3 Class Value

Another important issue to consider during the preprocessing is how to decide the class values. As I mentioned before, I turned the problem from a regression problem to a classification problem. The original dataset includes the IMDb scores, which is numeric attribute represents the mean value of all users’ rating for a movie. In this paper, I transformed this attribute into nominal and split the original values into three class values. Previous research used both Z-score and percentile to turn a numeric variable into a nominal variable. Since the criteria should be problem specific, I evaluate the three methods by considering:1) How meaningful will the classification results be for the audiences; 2) How will the method affect the classification result and the performance of algorithms? I tried three ways to classify the values:

1. **Classify the values by their Z-Score.** Values lower than Mean - 1 SD (Z-Score lower than -1) were regarded as “Low.” Values higher than Mean + 1 SD (Z-Score larger than 1) were regarded as “High.” Values between them were regarded as “Medium”. This method will screen out the top 16% movie for the audience, which approximately equals an IMDB score higher than 7.6. However, in this case, some great movies such as “Beauty and Beast” and “Legends of the Fall” were not even labeled as high. One of the reason could be that IMDB ratings are slightly less favor a certain genre such as romance. Regarding classifiers’ performance, since most instances go the “Medium” class, the accuracy will be good even half of the instances in other class values are classified as Medium. In fact, in my exploratory test, I found almost half of the movies from “High” class were classified into “Medium” using Logistic Regression, J48, and SMO while the accuracies were around 80%.

2. **Classify the values by percentile 25% and 75**%. Values lower than 25 percentiles were regarded as “Low.” Values higher than 75 percentiles were regarded as “High.” Values between them were regarded as “Medium.” This method will screen out the top 25% movie for the audience, which approximately equals an IMDB score higher than 7.2. Most movies that worth watching (e.g. movies from the top rated lists) are included.

3. **Classify the values by percentile 33% and 66%.** Values lower than 33 percentiles were regarded as “Low.” Values higher than 66 percentiles were regarded as “High.” Values between them were regarded as “Medium.” Also, since the movie rating score is following a Gaussian Distribution, the “Medium” class only covers a small range of movie score. The performance of the classifiers could be bad since the instances between different class value are not distinctive enough.

4. Data Exploration

4.1 Feature List

Table 1. Initial Feature Table

Feature Name	Type	Description
Genres (a selection from 21 different genres)	Text Feature	-
Movie production: • Gross • Budget • Year • Duration	Numeric	Normalized
Movie production: • Color • Language • Country	Nominal	-
Facebook (personnel): • Actor1_Facebook_likes • Actor2_Facebook_likes • Actor3_Facebook_likes • Cast_Facebook_likes • Director_Facebook_likes	Numeric	Normalized
Facebook (movie): • Movie_Facebool_likes	Numeric	Normalized
IMDb Forum(movie): • Number_critic_reviews • Number_user_reviews • Number_voted_users	Numeric	Normalized

4.2 Baseline Performance

For initial exploration, I choose four classifiers and applied them to the training data and tested on the development data to collect the baseline performance of them.

- Naïve Bayes was selected since it is simple and fast. However, given the assumption that the features are not independent, its performance may not be valid. I will discuss it later.
- Logistic Regression was selected since it is simple, effective, and robust to noise. If the features are roughly linear and the problems are linearly separable, this classifier could be a good choice.
- I also selected decision tree J48 as a non-linear classifier.
- Lastly, I selected SMO as another classifier. To test the baseline performance, I will use the default settings with a linear kernel; the performance should be close to Logistic Regression. However, during the tuning process, I will try different exponent values to test the performance of a non-linear model.

Since I have not turned text feature "genre" into a set of nominal features, LightSide with WEKA plugin were used to build the baseline model.

Table 2. Baseline Performance

Algorithm	Accuracy	Kappa Statistics
Naïve Bayes	0.55	0.25
Logistic Regression	0.62	0.30
J48	0.49	0.12
SMO	0.65	0.36

Baseline performance showed Logistic Regression performed a significantly higher accuracy compared with Naïve Bayes (t=4.29, p<0.01) and J48 (t=6.79, p<0.01). Also, SMO performed significantly better than Naïve Bayes (t=5.37, p<0.01) and J48 (t=8.08, p<0.01). Comparing with Logistic Regression, SMO performed significantly better with 0.65 accuracies compared with 0.62(t=2.3, P=0.019).

4.3 Initial Exploratory Data Analysis

4.3.1 Description Analysis and Correlation

Exploratory data analysis includes observation of data from the development dataset and statistic analysis on this dataset. Bivariate Correlation analysis showed that there were significant correlations between attributes, suggesting that the attributes are not independent (Table3). The result could explain why classifier Naïve Bayes is not suitable for this dataset since it requires independence between the variables [6].

Table 3. Correlation Table (part of the full table)

	A	B	C	D	E	F	G
A	1						
B	0.327**	1					
C	0.304**	0.102**	1				
D	0.186**	0.137**	0.102**	1			
E	0.036	0.109**	0.03	0.088*	1		
F	0.092*	0.093*	0.014	0.041	0.214	1	
G	0.261**	0.221**	0.044	0.119**	0.052	0.08	1
H	0.495**	0.519**	0.158**	0.283**	0.121**	0.134**	0.448**

A	score
B	num_critic_for_reviews
C	duration
D	director_facebook_likes
E	actor3_facebook_likes
F	actor1_facebook_likes
G	gross
H	num_voted_users

Also, descriptive analysis on Nominal Feature "genres" showed that some of the values are in low frequency, suggesting that feature selection might be applied in the tuning process to simplify the model and avoid over-fitting.

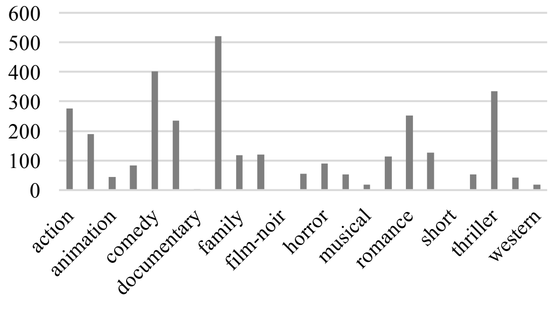
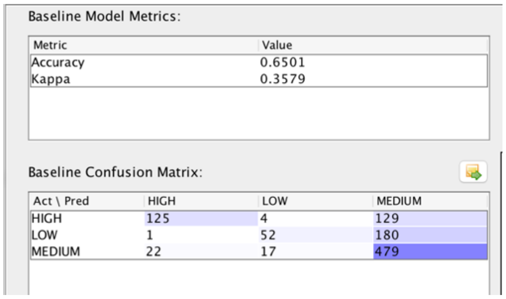


Figure 1. Frequency for each genre

4.3.2 Feature Engineering

In this process, I mainly focused on improving the performance of SMO, which was significantly better than other baseline algorithm. Feature Engineering and Exploratory Data Analysis are an iterative process. During this process, I trained models on the training set and test it on the development set. Then, I conducted error analysis over the test result and looked into development dataset to find out potential causes for the problematic features within the context. Looking into the confusion matrix, I found that major confusion appeared between "Medium"-"High" and "Medium"-"Low". 129 instances that scored "High" and 180 instances score "Low" have been classified as "Medium". Considering the class value "High" is more useful for the user given that the algorithm aims to screen out the best movies for them, feature engineering mainly focused on reducing false negative rate for instance in "High" score.



Baseline Model Metrics:

Metric	Value
Accuracy	0.6501
Kappa	0.3579

Baseline Confusion Matrix:

Act \ Pred	HIGH	LOW	MEDIUM
HIGH	125	4	129
LOW	1	52	180
MEDIUM	22	17	479

Figure 2. Confusion Matrix for SMO baseline

4.3.2.1 Movie Production Information

As for numeric features, I ranked features by feature weights and looked into confusion matrix with the absolute vertical and horizontal differences to identify problematic features. One of the problematic features is "budget.", which causing confusion between instance scored as "High" and "Medium".

Looking into the development dataset, I found that the title year of instances in this dataset spanned across 100 years. However, the movie budget and gross did not take the inflation rate into consideration. For example, the top rated movie *God Father (1972)* had a 6 million USD budget. The same crime and drama movie *The Shaw Shank Redemption (1994)* spent 25 million USD budget. These two movies are similar in ratings and genres, but the later has a four times higher budget. However, when we consider the inflation rate, the difference in the budget can be eliminated. Fig. 3 shows the cumulative inflation rate from 1910 to 2015. From the descriptive graph, we can see a discernable change across the years especially after the 1970s. One US dollars in 1970s worth almost 10 dollars now. Thus the comparison of budget across time is incorrect without adjustment.

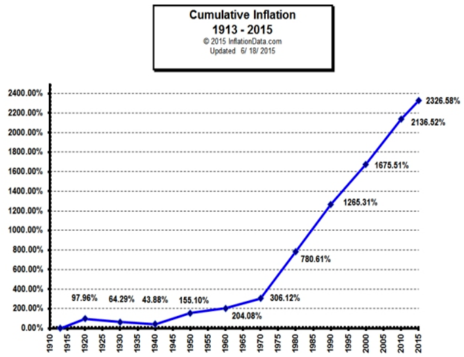


Figure 3. Cumulative Inflation[3]

Adjustments were made based on the cumulative inflation rate on attribute “budget” and “gross.” Firstly, I transformed movie title years into decades. Then, the I used the budget and gross value divided by cumulative inflation rate from the same decades to calculate adjusted budget and gross value. Lastly, I normalized the budget and gross value again. Slightly improvements on accuracy were observed after this error analysis. The accuracy improved from 0.650 to 0.657 with Kappa statistics improve from 0.37 to 0.38.

4.3.2.2 Social Networking

The second error analysis is related to social networking. I identified that some features related to social networking were problematic such as “movie Facebook like numbers.” In this dataset, the attributes related to social networking mainly focused on the popularity, for example, the number of likes and critic reviews from Facebook. One guess is that the popularity of social networking is time sensitive. Facebook was founded in 2004 and held its initial public offering (IPO) in February 2012. One guess is that the activity around a movies’ cast will be more active if the movie was released in recent years. In light of this, I created a new nominal attribute named “social-networking.” The attribute has three values. Instances with title years earlier than 2004 were assigned “NoFacebook.” Similarly, “EarlyFacebook” was assigned to movies released between 2004 and 2012, and “LateFacebook” for movies after 2012. After adding this feature, the model performance increased from 0.657 to 0.663 with Kappa Statistics improve from 0.38 to 0.387.

4.3.2.3 Movie Genres

A third error analysis was conducted on the movie content factors, the genres. Genres such as Action, Horror, Drama, and Animation were identified as problematic features. They have relatively high feature weight as well as high absolute horizontal differences or low absolute vertical differences. It makes sense since movies with the same genres can be totally different in quality.

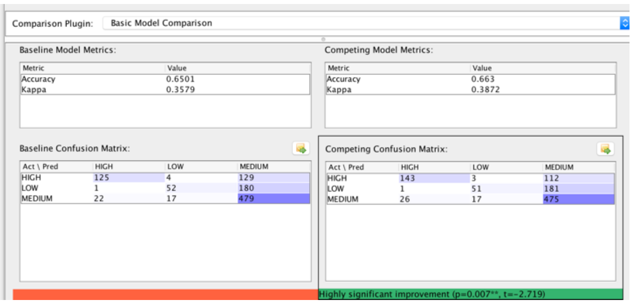
Previous research found that genre can be used to predict movies’ rating score. More specifically, drama and action can be used as predictors for higher ratings. However, our data shows a different story. Movies in the same genre can be scored either high or low. For example, the Batman movie *The Dark Knight (2008)* is scored 8.9 on IMDb and stayed on the top 5 list while another movie in the same genre *Catwoman (2004)* is scored 3.3. Why have movies in the same genre quite different ratings? Are there any moderators? Previous research find that the Star and movie budget have positive effect on movie box office [7]. One guess is that *Action*, *Horror*, and *Animations* more rely on its visual effect, which cost a lot of money on shooting and post production. This can be reflected by movie budget.

The interaction between attributes can be addressed by choosing the appropriate algorithm and introducing new features. For example, either changing the exponent value of SMO kernel or creating new features that multiply genres with movie budget could be effective. This section will

[3] https://inflationdata.com/Inflation/Inflation/Cumulative_Inflation_by_Decade.asp

focus on feature engineering, and algorithm optimization will be addressed in the following sessions. In this error analysis, I transformed the nominal feature "genres" into a set of binary features; each represents one attribute value. Then, I multiply feature "isAction," "isHorror," "isDrama," and "isAnimation" with the adjusted budget to create four new features that addressed the interaction between those genre types and budget. However, there was no discernable improvement on the model performance after this feature engineering.

After error analysis, three new features were introduced to the feature space. "AdjustedGross", "AdjustedBudget", and "SocialNetworking Index".


Comparison Plugin: Basic Model Comparison

Baseline Model Metrics:

Metric	Value
Accuracy	0.6501
Kappa	0.3579

Competing Model Metrics:

Metric	Value
Accuracy	0.663
Kappa	0.3872

Baseline Confusion Matrix:

Act \ Pred	HIGH	LOW	MEDIUM
HIGH	125	4	129
LOW	1	52	180
MEDIUM	22	17	479

Competing Confusion Matrix:

Act \ Pred	HIGH	LOW	MEDIUM
HIGH	143	3	112
LOW	1	51	181
MEDIUM	26	17	475

Highly significant improvement (p=0.007**, t=-2.719)

Figure 4. Improvement After Feature Engineering

SMO performance improve from 0.650 to 0.663, with Kappa index increased from 0.357 to 0.387. T Test showed that there was a significant improvement on the model performance (t=2.7, p=0.007). Confusion Matrix showed that the number of "High" scored instances that have been misclassified as "Medium" decreased from 129 to 112.

5. Optimization and Final Result

5.1 5.1 Tuning

During the tuning process, different selections of feature numbers as well as algorithm parameters were compared to find the best performance for SMO on the dataset.

As I mentioned in exploratory data analysis, some features (transformed from values of nominal parameter) might be reduced since they were in low frequency and could potentially result in over fitting. The *AttributeSelectedClassifier* in WEKA was used to select and rank the features. 10, 25, and the full set (40) were selected during the tuning process. This tuning procedure showed no statistically significant improvement by feature selection, indicating that it was not worth doing tuning on feature numbers. Thus, the full set was used in the final SMO model.

As for SMO parameter, there are two parameters that are critical for performance, the complexity parameter C and gamma parameter. Different settings for those parameters were tested during the tuning.

The value of C parameter controls the instances that are used to draw the linear separation boundary in the feature space. C value decides how soft the class margins are, namely. The higher value lead to more instances to build the boundary, which may result in a good performance on training data as well as the risk of over-fitting. The exponent parameter of Poly Kernel controls the linearity of the model when –E =1 represent the linear kernel. As I discussed above, there could be interactions between features. Thus, increasing the exponent value might result in a different performance.

After a few pilot experiments with WEKA *CVParameterSelection*, I selected 1(default) and 5 for C parameter (-C) and 1(default) and 2 for Kernel Exponent (-E). Thus, four different parameter selections were tested in total. The results were evaluated by a 5 fold cross validation first and the settings with significant improvement were trained with training data and test on development data. Comparison between baseline (-E=1, -C=2) and the best setting.

The test result showed that both (–E =2, -C=1) (accuracy = 0.681) and (-E=1, –C = 5) (accuracy = 0.66) will result in increment on performance, however, the effect from Parameter C was not significant. Although the condition (–E=2, -C=5) result in significant better performance compared with baseline, it was not significantly better than (–E =2, -C=1). Moreover, the (–E=2, -C=5) setting increased the complexity of the model thus the algorithm was much slower than other settings. After tuning process, (–E =2, -C=1) was selected for the final model with a 0.681 accuracy and a Kappa Statistic of 0.46 for estimated performance on unseen data.

5.2 5.2 Final Result

The final model was trained on the training data and test on the final test data, which had been unused. Using SMO with -C = 1 and -E=2, the model performance achieved

0.677 for a Kappa statistic of 0.457. The differences between the final result and the baseline models are statistically different. Table 4 showed the comparison between the final result and baselines. The result was significant better than the SMO baseline model (t=3.91, p<0.001).

Table 4. Model Comparison

Model	Accuracy	Kappa
1. Final Model SMO (-C=1, -E=2)	0.677	0.457
2. Baseline Naïve Bayes	0.55	0.25
3. Baseline Logistic Regression	0.62	0.30
4. Baseline J48	0.49	0.12
5. Baseline SMO (-C=1, -E=1)	0.65	0.37

6. Discussion and Future Direction

This paper showed that social media signals combined with movie content and production information can be used to predict movie ratings. The final performance was 0.677 accuracy with a Kappa Statistic of 0.457. Since most of previous research were working on regression problems, the result from this paper can be hardly comparable with other papers. However, the result showed great potential to use information from different channels to predict the goodness of a movie. As for my second hypothesis, both my analysis from the exploratory phase and optimization indicates that the attributes are not isolated thus a non-linear method could perform better since it can represent the potential interaction between attributes.

Although this paper discussed factors could affect a movie’s rating from three different perspectives, it is by no means a comprehensive analysis. There are a lot of things we can do to improve the depth as well as the scope of this paper.

First, the information from social media is limited. The dataset only considers the numeric features from social networks such as the number of review and likes. There are a lot of insights that textual feature can bring. For example, sentiment analysis on movie comment can be used to infer the goodness of movies [4, 8]. Textual features and sentiment analysis can also be less susceptive to error introduced by year of release since it focuses more on the quality of the movie rather than its popularity on social networking.

Second, there could be the systematic error introduced by IMDb users. Previous analysis showed that users on IMDb tend to assign the lower score to Romance than Crime, Documentary, or Action. One example is that *Titanic (1999),* one of the most famous movies and Oscar winner, scored only 7.7 on IMDb. There are some other platforms such as rotten tomato also provides reviews for movies. The next step on predicting movie rating might consider cross platforms validation while deciding the class value for each instance.

Lastly, the motivation for this paper is to predict whether a movie is worth watching for users. Although we use an objective standard to evaluate the movies, the goal itself is subjective. Users have different preferences for movies. Future work can focus on how to weight movies' content such as genres based on users' history data to provide customized services.

7. REFERENCES

[1] Oliver, Mary Beth, and Tilo Hartmann. "Exploring the role of meaningful experiences in users'appreciation of “good movies”." Projections 4.2 (2010): 128-150.

[2] Armstrong, Nick, and Kevin Yoon. Movie Rating Prediction. Technical Report, Carnegie Mellon University, 1995.

[3] Oghina, Andrei, et al. "Predicting imdb movie ratings using social media." European Conference on Information Retrieval. Springer Berlin Heidelberg, 2012.

[4] Asur, Sitaram, and Bernardo A. Huberman. "Predicting the future with social media." Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on. Vol. 1. IEEE, 2010.

[5] Tax, D. M., and R. P. Duin. "Feature scaling in support vector data descriptions." Learning from Imbalanced Datasets (2000): 25-30.

[6] Lewis, David D. "Naive (Bayes) at forty: The independence assumption in information retrieval." European conference on machine learning. Springer Berlin Heidelberg, 1998.

[7] Karniouchina, Ekaterina V. "Impact of star and movie buzz on motion picture distribution and box office revenue." International Journal of Research in Marketing 28.1 (2011): 62-74.

[8] Annett, Michelle, and Grzegorz Kondrak. "A comparison of sentiment analysis techniques: Polarizing movie blogs." Conference of the Canadian Society for Computational Studies of Intelligence. Springer Berlin Heidelberg, 2008.