Predicting Movie Box Office Profitability: A Neural Network Approach

Conference Paper · December 2016

DOI: 10.1109/ICMLA.2016.0117

2 authors:

Travis Rhee
Queen's University
1 PUBLICATION **12** CITATIONS
SEE PROFILE

Farhana H. Zulkernine
Queen's University
58 PUBLICATIONS **577** CITATIONS
SEE PROFILE

Predicting Movie Box Office Profitability

A Neural Network Approach

Abstract – **In this research, we have developed a model for predicting the profitability class of a movie namely "Profit" and "Loss" based on the data about movies released between the years 2010 and 2015. Our methodology considers both historical data as well as data extracted from the social media. This data is normalized and then given a weight using standard normalization techniques. The cleaned and normalized dataset is then used to train a back-propagation cross entropy validated neural network. Results show that our strategy of identifying the class of success is highly effective and accurate when compared to the results from using a support machine vector on the data.**

Keywords — Artificial Neural Network; Backpropagation; Social Media; Data Analytics; Movie; Box Office; Predict; Profitability

I. Introduction

The film industry has grown immensely over the past few decades generating billions of dollars of revenue for the stakeholders [7]. Now people can watch movies online and offline on a variety of mobile devices during leisure or travel through Netflix, Youtube and downloads [14]. A prediction system to assess the box office success of new movies can help the movie producers and directors make informed decisions when making the movie in order to increase the chance of profitability and box office success.

New social media tools are constantly appearing which are enabling people to gather information on films and post comments about movies. These comments can influence the initial prediction about the box office success of a movie which some of the existing research [22] do not take into account. Critic reviews often come out a few days before the film is released and may, therefore, help in prediction and at the same time influence the box office revenue.

Artificial Neural Networks (ANN) have shown great promises in prediction systems in various application domains such as stock markets [7], medical imagery [1], social media analytics [9] and also box office predictions for movies [1][9][16]. However, many different neural network models exist and the design strategies and the variables used in the analytic process greatly affect the outcome and accuracy of the prediction [1][16][22]. Many statistical machine learning approaches are also applied to similar problems [1][22].

In this paper, we propose a back-propagation neural network model for predicting the box office success of movies using data from movie databases [17] available on the web and the data from multiple social media networks [11][18][14]. We designed a three layer back-propagation artificial neural network and extracted data from the top 100 revenue grossing American movies released between 2010 and 2015. After the data cleaning and preprocessing stage, we retained and used 375 movies for analysis. We selected 14 movie features to be used as inputs for the network and used these to predict and classify the success of a movie either as a "bomb" or "success" based on its revenue and cost. We validate our approach using cross-entropy validation and the final results show 91% accuracy in the prediction.

The rest of the paper is organized as follows. Section two describes the previous work from the literature study. Section three describes the data and the various sources the data are collected from. Data pre-processing and the steps taken to clean and normalize the data are explained in section four. It also describes the input data we used in our analytic model. Section five describes our approach, experiments, validation process and the results. The future work is discussed in section six. Finally, section seven presents a summary and concludes the paper.

II. Previous Research

Previous research done in the area of predicting box office success have applied different techniques such as neural networks [6][13][9] and statistical Baysian [1] and linear regression modeling techniques [16][22]. The most recent work in the literature in box office prediction using neural networks is by Kaur, and Nidhi [9]. They use a multilayer perceptron with a Levenberg-Marquant learning algorithm to classify Bollywood films into 3 categories based on the box office performance using 111 instances of observations and 8 attributes. It resulted in a classification accuracy of 93.1%.

A multilayer perception is used in the work done by Li, Jianhua, and Suying, [13]. They use the data of 241 Chinese/American films and attempt to predict each film into 6 classes based on box office success ranging from a box office blob to bomb or mega success. based on 11 weighted continuous variables. The model used a multilayer backpropagation network consisting of 30 nodes in the first hidden layer and 10 nodes in the second hidden layer, which was then standardized and validated through a 6-fold cross validation to measure performance. The results were not very accurate at classifying movies in their actual class but had high relative accuracy (97.1%) indicating that the predicted class was one class away from the actual class.

A similar study is done by Delen and Shard [6]. They use 834 movies from 1998-2002 and run it through a multilayer perceptron network. They try to classify each movie into one of 9 classes based on its box office revenue using 7 continuous variables and a 10-fold cross validation. Accuracy is measured by the percentage correct classification rate and the 1-away classification rate which resulted in a 36.9% accuracy rate and a 75.1% accuracy rate respectively. The work compares the

neural network approach to other statistical analysis strategies such as the discriminant analysis, multiple logistic regression and decision trees. However, the neural network approach proved to be the most accurate.

Chang and Lee use a Bayesian Belief Network to determine the causal relationships between 18 variables in predicting box office success for Korean movies [1]. Sensitivity analysis is used to determine the important attributes. They use the number of movie goers as their metric for success and divide them into two groups. The first group is further split into inferior and superior categories based on the median and the second group is split into bad, standard and excellent categories. Movies in the superior and excellent category are the most successful in the two groups. When compared to artificial neural networks and decision tree approaches, the Bayesian Belief Network proves to be the most accurate.

None of the above papers account for the social media data which has become an important indicator and influencing factor of a movie being successful. Skiena and Zhang use ratings from the IMDB, a popular database of information about movies, and sentiment analysis on news data gathered by "Lydia" (www.textmap.com) to determine the gross revenue of a movie. They use linear regression and K nearest neighbour models and demonstrate that the linear regression model is better at predicting low gross turnover movies while the K-nearest network is good in predicting high gross turnover movies. Accuracy of the approach increases when the ratings from IMDB are combined with the news analysis [22].

Breuss, de Rijke, Oghina, and Tsagkias[3] explore the idea that user activity on social media may have an impact on box office performance of a movie. They look for a correlation between the movie information extracted from the IMDB and people's remarks regarding the extracted movies on social media websites such as the Twitter [20] and Youtube [21]. More specifically the authors look into the qualitative data (textual) which represent the comments of people about the quality of a movie, and the quantitative (surface) data which represent the number of people commenting about a movie. These social media data is used in a linear regression model and the results indicate that the Youtube like to dislike ratio and the textual analysis of Twitter data are the best in predicating the IMDB rating of a movie [9].

Asur and Huberman research the effect of Twitter on box office revenues [2]. Extracting data from the Twitter Search API, they identify the week before the movie is released as the critical period. They measure the rate of tweets and compare it to box office revenue using linear regression. These two variables show a strong positive correlation. When compared to the "Hollywood Stock Exchange" which uses prices for "movie stocks" to predict the revenue, the Twitter model surpasses it in accuracy. It also shows better accuracy than the news/IMDB model of Skiena and Zhang [22]. Sentiment analysis further improves the model by a significant margin.

Most of the ANN predictive models do not use data from the social media. Considering the success of ANN as predictive models and the importance of the data from the social media networks in predicting the success of movies in the box office, in this work we explore the performance of an ANN model with the data from social media in predicting box office success of the movies. Therefore, our work extends the work of Delen et al. [6], Li et al. [13], and Chang et al. [1] while including the social media data analysis from Breuss et al. [16].

III. Data Description

We retrieve movie data from the OpusData website [17], which hosts a large amount of data about movies and is used in many other research works. We extract the top 100 movies based on the total gross turnover of the box office for the years for 2015, 2014, 2013, 2012 and 2011. So our initial database consists of the data of 500 movies from OpusData which we refer to as the *Opus data*. The movie attributes of the Opus data include the following:

• Budget • Domestic revenue • International revenue • Total box office revenue	• Release date • Genre • Top actors • Directors • DVD sales

We also extract related movie data and user reviews of three different websites namely the Internet Movie Database (IMDB) [11], Rotten Tomatoes [18] and Metacritic [14]. We extract the following IMDB information and refer to it as the *IMDB data*. Some of this information is also included in the Opus data.

• Producer • Release date • User ratings • Number of votes	• Budget • Actors • Director

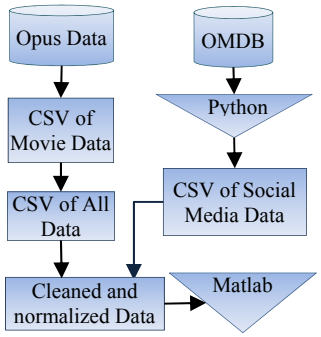

Fig. 1: Data pre-processing workflow.

The python library of OMDB (Online Movie Database) API [16] is used in a Python program to extract the IMDB, Rotten Tomatoes and Metacritic data. We use the movie titles in our Opus data to search and extract the corresponding user ratings and the number of user-reviews used to generate these ratings. We refer to this data as the User Reviews or *UR data*.

- Metascore
- Rotten Tomatoes Meter
- Rotten Tomatoes User Rating
- Rotten Tomatoes User Meter
- Number of Rotten Tomatoes User Votes

IV. Data Preprocessing

The data collected from the various online data sources are cleaned to remove incomplete and missing data.

A. Data Cleaning and Integration

The following steps are used to clean the extracted movie and user review data and integrate the various extracted data sets namely the Opus data and the IMDB and UR data. We wrote custom Python programs as needed for the various processing steps.

1. We remove the records which do not have any budget amount and very little information beyond box office revenue. Many of these movies are Chinese, Russian or Indian movies and hence the titles cannot be translated properly to link and extract the corresponding user reviews and ratings. We remove all such movies from the data set.
2. The 'DVD sales' data item is removed for all movies as it is sparsely filled.
3. The Opus data only list one actor name per movie while the IMDB data contains the names of 3 to 4 actors per movie. Therefore, we use movie titles to link the Opus data and the IMDB data and include all the actor names available in the two data sets in the integrated data set.
4. The integrated data set contains the data items in Table 1.

At this stage we have a list of 375 movies in our final data set containing the following variables.

TABLE I. LIST OF VARIABLES IN THE CLEANED DATA

Movie Title	Run Time	IMDB Votes
Total Box Office	Rating	Metascore
Dom. Box Office	Release date	*RT User Reviews
Intl. Box Office	Actors	RT User Votes
Budget	Directors	RT Meter
Sequel	IMDB Rating	RT User Rating

*RT = Rotten Tomatoes

B. Normalizing the Data

The non-numerical variables such as actors, directors and release dates are first converted to a numerical value as discussed in the next section. The numerical values are then normalized so that the values lie between 0 and 1 to avoid large variations in the values. We use feature scaling as shown in Eq. (1) for normalization where X is the actual value of a variable, X_{min} and X_{max} are respectively the minimum and maximum of all the values of this variable in the complete data set. X' represents the normalized values in the range of [0, 1]. The data pre-processing workflow is shown in Fig. 1.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad \text{.....(1)}$$

C. Preparing the Input Data Items

We use 14 variables out of the 18 as listed in Table I as the inputs to our neural network model as described below.

1) Average Actor Star-power

Using a custom application, we compute each actor's 'Actor Power Score (APS)' by counting the number of times this actor appears in our final data set. Then for each movie in our data set, we add the power scores of all the actors in the movie which we call the 'Total Actor Star-power (TAS)' of the movie. We divide this by the number of actors (NAct) in the movie to calculate the 'Average Actor Star-power (AvAS)' and add it to the final data set against each movie. The computations are shown in Eq. (2).

We normalize the AvAS by using feature scaling as stated in Eq. 1. Only the normalized AvAS is used in the model.

$$APS_{actor} = count_{actor}(\text{all movies})$$

$$TAS_{movie} = \sum_{(\text{actor in movie})} APS_{actor}$$

$$AvAS_{movie} = TAS_{movie}/NAct_{movie} \quad \text{.....(2)}$$

2) Director Star-power

For directors, we follow the same strategy as for the actors to compute a 'Director Power Score (DPS)'. Since there is only one director per movie in our final data set, we do not need to compute the total or average director star-power in this case. The DPS of the respective director of a movie is assigned to that movie as its 'Director Star-power (DS)' in the final data set. We normalize the DS using feature scaling of Eq. 1. The normalized DS is used in the model.

$$DS_{movie} = APS_{director} = count_{director}(\text{all movies})$$

3) Competition Factor

We compute the count of all movies which were released within 2 weeks, before or after, and call it the 'Competition Score (CS)'. Then a 'Competition Factor (CF)' is calculated from the inverse of CS which implies that the higher the competition, the lower the score.

$$CF_{movie} = 1/CS \quad \text{.....(3)}$$

4) Significance of the month of the Release Date

We also check for each movie if its release date is in the month of a holiday. We only consider the following significant Canadian holiday and festival days in this work: Valentine's Day in February, St. Patrick's Day in March, Victoria Day in May, Independence Day in July, Halloween in October, Thanksgiving in November, and Christmas in December. If the movie is released in a holiday month it is given a score of 1, otherwise it is given a score of 0.

5) MPAA Movie Rating

Motion Picture Association of America (MPAA) movie rating is a categorical data that can have four values as G, PG, PG-13 and R. We converted these four values to numerical values of 0 to 3 respectively such that G=0, PG=1, PG-13=2 and R=3.

6) Sequel

The sequel data item identifies whether or not a movie is a sequel or continuation of an earlier release. It is a binary value where 0 indicates not a sequel and 1 indicates a sequel.

7) Budget

A budget represents how much money was spent to produce a movie in US dollar (USD). We normalize it using feature scaling as stated in Eq. 1.

8) IMDB Rating

The IMDB rating represents the average of the IMDB user ratings of a movie that varies from 0 to 10. This rating is

normalized using Eq. 1 with $X_{min} = 0$ and $X_{max} = 1$.

9) IMDB Votes

This is the number of IMDB users who cast votes to rate a movie. It reflects how many users watched a movie. Therefore, this data item is normalized using Eq. 1.

10) Metascore

This data item represents the user rating from the Metacritic website which varies from 0 to 100. It is also normalized using Eq. 1 with $X_{min} = 0$ and $X_{max} = 100$.

11) Rotten Tomatoes (RT) User Rating

RT user rating varies from 0 to 10 and represents the average rating of the RT users for a movie. The data item is normalized using Eq. 1 with $X_{min} = 0$ and $X_{max} = 10$.

12) Rotten Tomatoes (RT) Meter

RT Meter represents the percentage of critics who think a movie is good or bad and the value varies from 0 to 100. The values are normalized using Eq. 1 with $X_{min} = 0$ and $X_{max} = 100$.

13) Rotten Tomatoes (RT) User Reviews

RT user reviews is an average of all RT user review-scores for a movie. It varies from 0 to 100 and we normalize it using Eq. 1 with $X_{min} = 0$ and $X_{max} = 10$.

14) Rotten Tomatoes (RT) User Votes

Number of RT user votes represents the number of users who voted for each film. It is normalized using Eq. 1.

V. Our Approach: The Neural Network Model

We define an artificial neural network model that is trained using our pre-processed data set. The model is then validated and tested to predict success or failure of a movie given a set of input data items. In this section we explain our neural network model and how it classifies movies into 'success' or 'failure'.

A. Class Labels

For this initial version we classify the movies into two categories, success and failure. Taking into account the classification of success and failure used in some of the previous work, we define an acceptable profit amount as shown in Eq. 4 as a measure of the success of a movie. It is a widely used measure which has heuristically produced more accurate results than just revenue minus cost. The total box office revenue is divided by two in order to factor in marketing costs and other costs during distribution which are not publicly available [22].

$$\text{Profit} = (½ * \text{Revenue}) - \text{budget} \quad \text{.....(4)}$$

For our normalized final data set we compute the profit for each movie and add another data field to indicate if the movie is a success or failure. If the profit is positive, the movie is labelled as a 'success'=1, and otherwise it is labelled as a 'flop'=0.

B. Neural NetworkModel

For our network we design a multi-layer backpropagation neural network [6][13] as shown in Fig. 2. In this work we use a single hidden layer with 25 nodes that was determined experimentally to provide the best result. We use 14 input nodes for the 14 input data items. We use 2 output nodes at the output layer where one indicates success and the other failure.

C. Implementation Tool and Prototype

We use Matlab's Neural Network toolbox specifically the neural pattern recognition tool to design, train and validate our neural network model for classifying our movie data set into two classes namely success and failure. We use a scaled conjugate gradient backpropagation algorithm. Performance is evaluated using a cross-entropy method in which the training stops when the percentage of error in the output cannot be improved any more. We split our data set into three parts for each of these steps: 70% for training, 15% for validation and 15% for testing.

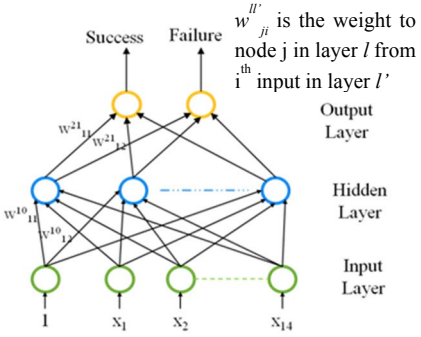

Fig. 2: Neural network with 14 input, 25 hidden and 2 output nodes.

For the input data, we use our normalized final data set. The first step is to select the input data source which contains our normalized data set of 14 input data items. We also need to select the target data source which contains two columns for success or failure for each movie with the correct column containing 1 and the other containing 0 depending on if the movie is a success or failure. We define these files as CSV files. We generate the final program code automatically using the tool and modify it to generate the predicted output classes for unlabelled data.

D. Results, Validation and Discussion

1) Results

The performance of the training, validation and testing runs of the model is shown in Fig. 3 using four different confusion matrices. The red blocks show the wrong classifications and the green ones show the number of movies that were correctly classified. The blue block at the bottom rightmost corner shows the overall accuracy. Output class 1 in the figure denotes successful movies and class 2 denotes flop movies. The blue block in the training confusion matrix shows that we achieve 90.9%

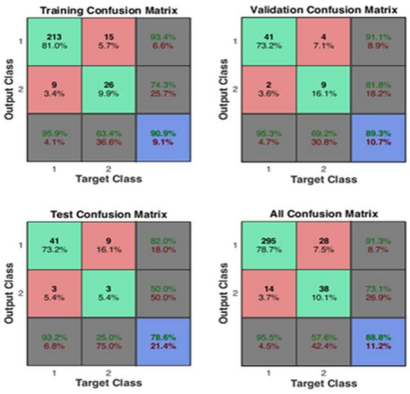

Fig.3: Confusion matrices for training, testing, validation and total data.

accuracy in the training phase. The test confusion matrix shows an accuracy of only 78.6%. Out of 56 test movies, 15 were incorrectly classified (red blocks show false positive and false negatives). The validation matrix shows better accuracy of 89.3% than the test matrix. The all confusion matrix combines all the confusion matrices and shows 88.8% accuracy.

The confusion matrices also have higher false positives than false negatives. This effect may be due to our selection of data from the top 100 movies every year. Out of the 375 movies only 66 movies are labelled as flops. This is possibly contributing to the high false positive, i.e., flop class 2 movies being classified as success or class 1 movie. Therefore, Inclusion of movies from further down on the list may increase the classification accuracy which we would like to explore in our future work.

Fig. 4 shows the performance of the training, test, and validation data over each epoch. The best accuracy is observed after 22 epochs for all the different types of runs. After the 22nd iteration, performance ceases to improve. An epoch is a single run of the neural network with the complete input data set. Multiple epochs are executed repeatedly until the desired accuracy is achieved.

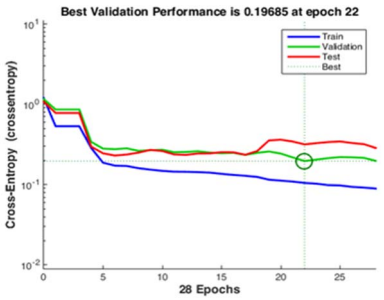

Fig. 4: Performance of training, test, and validation sets.

The receiver operating characteristic (ROC) curves as shown in Fig. 5 illustrate that the results are fairly accurate at correctly classifying both classes with the exception of the test data being less accurate compared to the training and validation ROC. There are more true positives than false positives as shown by the concave nature of the curves.

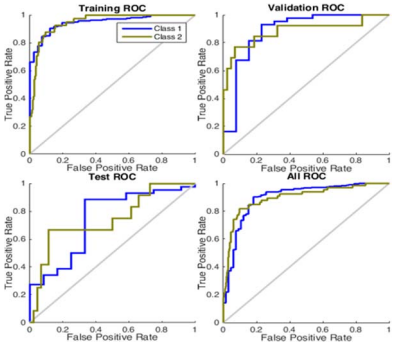

Fig. 5: ROC curves for training data, test data, validation data and total data.

2) Validation Compared with SVM Approach

In order to compare the results with another classification technique, we use the commonly used statistical approach to classification, the linear support vector machine, to classify our data set. We use the Matlab tool's statistical analysis package which has the SVM algorithm built-in. The SVM approach is commonly used for two-class problems, such as the one presented in this paper and hence we use it to compare the performance of our neural network classification approach.

Fig. 6 shows the confusion matrix for the linear SVM. Class 1 is for the movies that are successful while class 0 are movies determined to be flops. It is able to correctly classify successful movies almost up to 100% but it misclassifies 87.9% of the flop movies to be successful. Only 12.1% movies are correctly classified to be flops. The neural network also has trouble in correctly classifying movies that are flops but does much better compared to the SVM approach with respect to the classification of the flop movies. As we noted earlier this may be due to the very few flop movies we have in our data set compared to the movies that are successful and will be addressed in our future work.

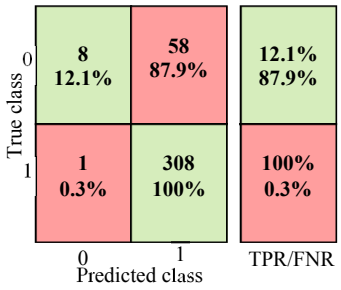

Fig. 6: Confusion matrix for linear SVM classification of our movie data set.

3) Discussion with Reference to the Existing Work

When compared to some of the previous work in the literature, we see that our neural network classification approach performs better and has higher accuracy in correctly classifying movies as a success or flop. Li et al. use 6 classes but has only 68.1% accuracy for exact classification [13]. Delen et al. use 9 classes and only scores 36.9% accuracy for direct classification [6]. These papers also use a 1-off scoring system in which they are very accurate. Because we only do a 2-class classification, 1-off scoring would give us 100% accuracy. So, we did not implement that. Kaur et al. is able to get 93.1% accuracy using 3 classes but do not consider social data [9]. Table II compares the results of our tests to the SVM and previous literature.

TABLE II. COMPARISON BETWEEN NN, SVM AND PREVIOUS RESEARCH

Network	NN	SVM	Li et al.	Delen et al.	Kaur et al.
Number of attributes	14	14	11	7	8
Number of classes	2	2	6	9	3
Number of hidden nodes	25	N/A	40	36	N/A
Number of inputs	375	375	241	834	111
Accuracy in %	88.8	84.2	68.1	36.9	93.1

VI. FUTURE WORK

One of the key observations in this preliminary work is that both our neural network approach and the SVM approach have difficulty in classifying the flop movies correctly. We like to include more movies from various parts of the IMDB movie list in our future work and not only the top 100 movies as we did in this work. We believe that will increase the classification accuracy of the flop movies. We also like to add more classes to categorize movies such as 'bomb', 'good', and 'flop' and more input data attributes about movie content such as the type of movie and visual effect.

We like to include more social media data as input such as sentiment analysis from Twitter as well as the number of people reached through marketing on Facebook reach, website hits, Facebook likes and the number of shares as used in Asur et al. [2]. With the introduction of Netflix, we could also look at the Netflix data to determine whether certain movies entering theatres could be successful as certain trends in Netflix content sometimes translate to the big screen. Netflix has also utilized big data techniques to determine which content it would produce, such as the hit show House of Cards [21]. Research on recommender systems are getting popular and neural network based approaches such as Restricted Boltzmann Machines are being widely used [9]. Including data from movie recommender systems can have an interesting effect on box office prediction.

VII. Conclusion

In this work we study the existing literature which lacks research on exploring and including the social media data in predictive systems for movies. We propose a multi-layer back-propagation neural network modeling approach to predict box office success of movies. We use not only the historical data as used in the related works but also data from multiple social media websites. We use 14 input data items as the input to the neural network which includes user review counts and ratings from multiple social media websites. We achieve better accuracy than most of the existing work in accurately classifying the movies in the proper class and our paper considers the social media data unlike the other comparable neural network based approaches. We also compare our approach with a SVM approach, a commonly used statistical two-class classification approach, and show that our neural network approach is able to achieve a higher overall accuracy. Therefore, our research shows that social media can improve the predictive power of neural networks in classifying movies as success or failure. For future work we like to use a larger data set as well as more social media information such as Twitter in order to improve results even further.

References

[1] Amato, F., López, A., Peña-Méndez, E. M., Vaňhara, P., Hampl, A., & Havel, J., 2013. "Artificial neural networks in medical diagnosis". *Journal of Applied Biomedicine,* Elsevier, vol. 11(2), pp. 47- 58.

[2] Asur, S. & Huberman, B.A., 2010. "Predicting the future with social media". In *proceedings of the International Conference on Web Intelligence and Intelligent Agent Technology,* vol. 1, pp. 492-499.

[3] Breuss, M., de Rijke, M., Oghina, A., & Tsagkias, M., 2012. "Prediciting IMDB movie ratings with social media". In *proceedings of the 34th European Conference on IR Research: Advances in Information Retrieval, ECIR*, Barcelona, Spain, Eds. R. Baeza-Yates & A.P. de Vries & H. Zaragoza & B.B. Cambazoglu & V. Murdock & R. Lempel & F. Silvestri, LNCS vol 7224, pp. 503-507, Springer.

[4] Carr. D. 2013. "Giving Viewers What They Want". The New York Times. Retrieved from http://www.nytimes.com/2013/02/25/business/media/for-house-of-cards-using-big-data-to-guarantee-its-popularity.html

[5] Chang, W., & Lee, K.J., 2009. "Bayesian belief network for box-office performance: A case study on Korean movies". *Expert Systems with Applications,* Elsevier, vol. 36(1), pp. 280–291.

[6] Delen, D. & Sharda, R., 2006. "Predicting box office success of motion pictures with neural network". Expert Systems with Applications, Elsevier, vol. 30 (2), pp. 243–254.

[7] Egeli, B., 2003. "Stock market prediction using artificial neural networks", Decision Support Systems, vol 22, pp. 171-185.

[8] Film Industry. (n.d.). Retrieved from http://www.statista.com/topics/964/film/

[9] Ghiassi, M., Skinner, J., & Zimbra, D., 2013. "Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network", *Expert Systems with Applications*, Elsevier, vol. 40(16), pp. 6266–6282.

[10] Hinton, G., Mnih, A., & Salakhutdinov, R., 2007. "Restricted Boltzmann Machines for Collaberative Learning". In *proceedings of the 24th international conference on Machine learning,* Corvallis, OR, pp. 791-798.

[11] Internet Movie Database,2016. Retrieved from http://www.imdb.com/.

[12] Kaur, A. & Nidhi, A.P., 2013. "Predicting movie success using Neural Network". *International Journal of Science and Search,* India, online, vol. 2(9),pp. 69-71.

[13] Li, Z., Jianhua, L., & Suying, Y., 2009. "Forecasting box office revenues of movies with BP neural network". Expert Systems with Applications, Elsevier, vol. 36(3), Part 2, pp. 6580–6587.

[14] Metacritic, 2016. Retrieved from http://www.metacritic.com.

[15] Netflix. (n.d). Retrieved from http://www.statista.com/topics/964/film/

[16] Online Movie Database (OMDB) API version 0.5.0, 2016. Python library at https://pypi.python.org/pypi/omdb.

[17] OpusData, 2016. Retrieved from http://www.opusdata.com/.

[18] Rotten Tomatoes, 2016. Retrieved from http://www.rottentomatoes.com/.

[19] Rico, J. 2016. "How The Break Even Point on a $200 Million Movie is $400 Million". About Entertainment. Retrieved from http://warmovies.about.com/od/War-and-Action-Movie-Budgets/fl/How-the-Break-Even-Point-on-a-200-Million-Film-is-400-Million.htm

[20] Twitter, 2016. Retrieved from https://twitter.com/.

[21] YouTube, 2016. Retrieved from https://www.youtube.com/.

[22] Zhang, W., and Skiena, S., 2009. "Improving Movie Gross Prediction through News Analysis". In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology,* vol. 1, pp. 301-304, IEEE Computer Society, Washington, DC, USA.