Predicting the performance of a movie using Movie trailer comments

Anubhav Rajauria
Department of Engineering Management
Leadership, Rice University
ar161@rice.edu

Benson Thomas
Department of Computer Science
Rice University
bt22@rice.edu

Ishita Kundaliya
Department of Computer Science
Rice University
ik20@rice.edu

1. Problem

Movies are a popular form of entertainment for people all over the world. When a movie is produced and its trailer is released, it is common for viewers to leave multiple comments on it, often making judgments based solely on the brief two-minute clip they have seen. These comments are believed to play a crucial role in determining the fate of the movie once it is released to the public.

The impact of viewers' comments on a movie's prospects is significant, as they can influence others' decision-making when it comes to choosing which films to watch. With the rise of social media and online forums, people have more platforms than ever before to express their opinions and influence others. Keeping this in mind we propose to analyze these comments on the trailers and use them to predict the rating of the movie from the set of classes (good, medium, bad). Our aim is to understand and utilize these comments and see if these comments can be used to make noteworthy predictions.

2. Literature Survey

In the paper Prediction of movies box office performance using social media [1] a similar approach is carried out where box office performance of movies is carried out. Multiple decision features were taken into consideration from the historical database of movie, follower count on the movie from twitter and sentiment analysis on the movie's YouTube comments. These decision factors were used to classify the movie into three classes(Hit, Neutral and Flop). Our modification to this approach was trying to use just the IMDB rating as the metric and develop patterns with YouTube trailer comments and the movie performance as the IMDB ratings can be easily acquired compared to the other features which was used in this paper.

In previous papers, YouTube comments have mostly been used to analyze the movie and extract information from the comments. Comments have been analyzed to understand what makes it good or bad like the presence of a certain plot, cast, etc. In the paper Sentiment Analysis on YouTube Movie Trailer [2], comments have been used to determine the impact on Box-Office Earning. In the above paper, the authors have extracted good/ bad movie indicators and then tried to predict the box office collection. In this paper there is a mention of the process however, no proper pipeline has been built. With our project we aim to analyze if the form of input changes the accuracy of predictions. In our project we are trying to put various factors in a model so that we can get accurate or somewhere close reviews about the movie using the YouTube comments which we have taken from Google API.

In the paper Predicting Success of a Movie from Youtube Trailer Comments using Sentiment Analysis [3], three different sentiment indexes have been defined which measure the sentiment of the movie reviews. The box office collection of these movies is then predicted based on these indexes. The observations have been made based on 20 movies. These models predict the box office collections and the standard error seems to be extremely high for the various sentiment indexes.

3. Hypothesis

We hypothesize that comments on the YouTube trailers of movies can tell us how good the movie is in the range of Good, Average or Bad.

4. Data Collection

To work with our hypothesis we required different datasets and not all of them were readily available. However there were a few datasets which helped us to start with our search.

Dataset of movie reviews and the sentiment(positive/ negative) on them- This dataset was used to create a sentiment analysis model which has been used as a baseline model to predict the sentiment of movie comments. Link to access dataset

Dataset including the movie details and rating of the movies- This dataset of over 800 movies is ready available on Kaggle with the name movie-success-rate.csv. Link to access dataset

Dataset of YouTube video IDs- To get the comments from YouTube videos it was necessary for us to get the video ID's. We got these IDs using beautiful soup on YouTube searches. These searches were of the form "movie_name + 'trailer'". This search returned numerous results and we extracted the ID of the first video. If the search was unsuccessful it returned "could not find". We stored these IDs in a csv file.

Dataset of cleaned comments on the YouTube movie trailers of the extracted video IDs- We then extracted the 100 most oldest comments for all the movies whose movie IDs were available using YouTube API. The reason we extracted the 100 oldest comments is because we wanted to deal with only those comments which were probably written before the release of the movie since once people have watched the movie they come back to the trailers to give their opinions on the whole movie rather than just the trailer. For some of the IDs the comments were disabled and hence, we could not extract their comments. After extraction of these comments they were cleaned and then stored in csv files. At the end we had a total of 64537 comments for over 740 movies.

5. Experimental setting for each use case

We tried several methods to try to prove our hypothesis. We first created a sentiment analysis model with the dataset of movie reviews and their sentiments. We cleaned the comments and then converted them into bag of words using CountVectorizer with a max of 4000 features. This was the input to Logistic Regression and LSTM model which predicted if a particular sentence is positive or negative. The reason we chose this dataset was so that our model would be curated to only predict reviews and comments related to movies. The accuracy of this model is 87%. We saved this model to use for our fruther experiments.

Experiment 1: In this experiment we first converted all our cleaned movie trailer comments to bag of words using the dictionary that was created with CountVectorizer with 4000 features on the movie reviews dataset. Then all these movies were sent to the saved sentiment analysis model and we got predictions if a particular trailer comment was positive or negative. Then for every movie we found the count of these positive and negative comments and made a new dataset which included the name of the movie, number of positive comments, number of negative comments and the rating of all these movies.

These movies have a rating between 0 to 10. However, for our experiments we had to convert these ratings to classes Good, Average and Bad. The classes are distributed such that ratings 0- 5 fall in class 'Bad', 5- 7 fall in class 'Average' and 7- 10 fall in class 'Good'. We assume IMDB ratings is a good metric to define how good the movie is and hence used it for this classification.

We then divide our dataset of number of positive comments, number of negative comments and rating into train and test set in the ratio 8: 2. We then created a model with 2 hidden layers and train it to predict the rating based on the positive and negative comments. Finally, prediction accuracy is calculated on the test set.

Experiment 2: In the previous experiment our final model was given only two features to work with and hence, it could not learn much. Keeping this in mind we decided to change our approach and give it better features in the form of comments itself rather than the number of positive and negative comments.

For the second experiment, we merged all the clean comments with the ratings. These ratings are again classified into 'Good', 'Average', 'Bad'. We now have a dataset with the movie title and each comment is assigned a rating. We then converted all these comments to bag of words using CountVectorizer with a maximum of 5000 features. To train these comments we again divided the dataset into train and test set with a 8:2 ratio. This split was done directly on the movie titles rather than the comments. Now all the comments that belonged to a particular movie title were sent to either the train set or the test set.

We also implemented one more approach to this experiment where we treated all YouTube comments of a movie as a feature instead of labelling each comment individually. Finally, we used multiple machine learning models like a 2 hidden layer neural network, K-Nearest Neighbors, Naive

Bayes, Logistic Regression, Support Vector Machines, Bagging and Boosting to evaluate which approach gave us better result and which model performed the best among the various machine learning models.

Experiment 3: For the third experiment we once again calculated the number of positive and negative comments for every movie using the same process as in Experiment 1 and found the ratio of positive and negative comments on the movie's YouTube trailer. We also converted the IMDB rating to the three classes as required. For this experiment we wanted to check if our new features created any significant change in the already available methods to calculate the success of movies.

To do this we first created a model which took multiple features of the movie like its genre, its run-time in minutes, etc as input and gave predictions on its rating from the set {Good, Average, Bad}. After this we added our features,(i.e.) number of positive comments and negative comments to this existing set of comments and trained a new model and calculated its accuracy using the pre-existing feature "Success" as the classification label. We experimented both the approaches with logistic regression and decision trees to see how different models perform.

6. Summarising experimental result

Logistic Regression and LSTM, a variety of Recurring Neural Network (Long Short Term Memory) were used to perform the Sentiment Analysis on the Movie Reviews and Sentiment Dataset. Logistic Regression gave an accuracy of 84% and the LSTM model gave us an accuracy of 87%.

Experiment 1: Our first experiment was to check if we could develop a relation between the performance of a movie and the count of positive negative comments on the movie's YouTube trailer. Using the Sentiment Analysis we had developed, we classified the comments to positive and negative and used a Neural Network model to predict how the movie would perform based on the ratio of positive and negative comments on the YouTube trailer of the movie. The model only gave us an accuracy of 60% hence no inference could be made from this approach.

Experiment 2: Our second experiment was to determine if we could develop any relation between the comments and the movie performance. For this we classified the IMDB rating of the movie as good(> 7.0), average(5.1 -7.0) and bad(< 5.0). Here, we tried out two different approaches.

Approach 1: The comments were treated as individuals and each comment of the movie was given the IMDB rating classification of the movie as the label and the model was trained. Predictions were made on this model and we obtained an accuracy of only 53% using Neural Networks.

Approach 2: Instead of treating the comment as individuals we found it optimal to use the entire comments of the movie as input to the model for each movie. All the comments of a movie were combined as one and bag-of-words for this input was created and served as input to the model. We believe this provides more information to the model than the previous approach. Neural Network model provided an accuracy of 73% which was better than the previous approach. Although we are unable to infer the performance of a movie with this approach we can clearly see that this was a better approach than the previous one.

As this approach seemed to have better performance, we implemented the same approach using different machine learning models like Logistic Regression, Naive Bayes, Support Vector Machine (SVM), K- Nearest Neighbor(with nearest-neighbors=3), Bagging and Boosting. The result of the experiment is shown below:

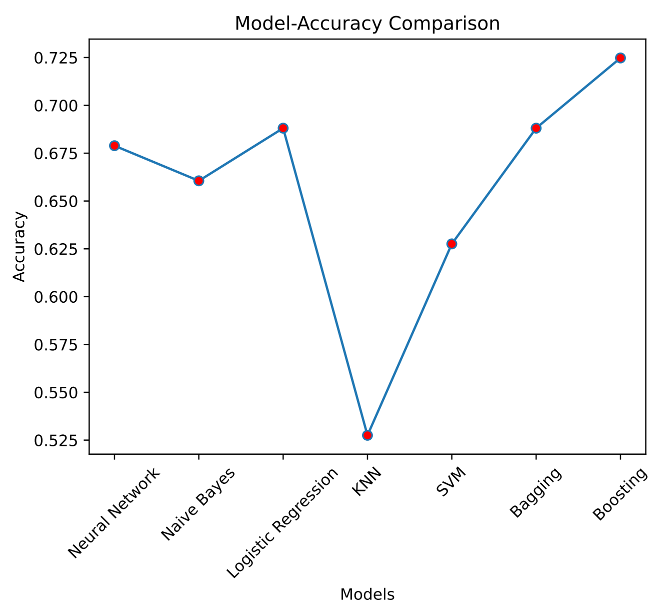

Figure 1. Performance of different models with 3 classes

From the above graph we can infer that Neural Network, Logistic Regression Bagging and Boosting performed considerably better than other models with Boosting performing the best out of them all with an accuracy of 72%.

We were curious and experimented how TF-IDF vectorizer would perform when compared to CountVectorizer and noticed that TF-IDF vectorizer couldn't perform as good as the Count vectorizer. This could be because of multiple reasons such as frequency based representation being more informative than the IDF weighting. Another reason for this would be amplification of noisy words by the TF-IDF vectorizer which is not the same with CountVectorizer as YouTube comments are generally noisy.

We also experimented with one more case i.e, we divided the ratings into 2 classes i.e, either good or bad with threshold kept at IMDB rating 6.8.

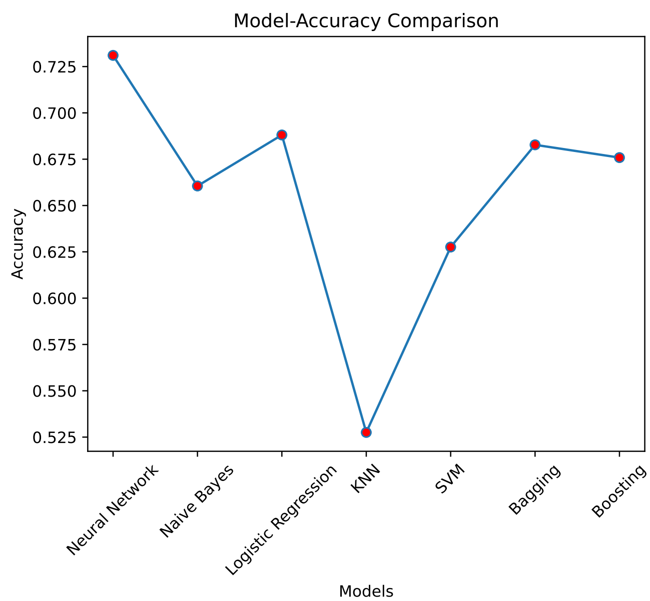

Figure 2. Performance of different models with 2 classes

We implemented this use case with different machine models, i.e, Neural Network, Naive Bayes, Logistic Regression, KNN, Support Vector Machines, Bagging and Boosting. From the plot we can confirm that the Neural Network Model performed the best with an accuracy of 73% compared to the other models while Logistic Regression, Bagging and Boosting performing comparatively decent.

Experiment 3: In this experiment we were mainly focused on finding out whether the addition of the positive negative ratio of the comments of each movie made any difference in the accuracy of predicting the performance of the movie. We used the movie success rate dataset that contained multiple features of a movie and we added the positive negative comment ratio of the YouTube trailer as features of the movie.

We carried out the experiment in multiple settings to compare the addition of positive negative comment ratio to the dataset:

- Machine Learning Models' Performance on the initial dataset
- Machine Learning Models' Performance on the dataset with positive negative comment ratio included
- Machine Learning Models' Performance on the dataset using IMDB Rating as the classifier without positive negative comment ratio
- Machine Learning Models' Performance on the dataset using IMDB Rating as the classifier with positive negative comment ratio

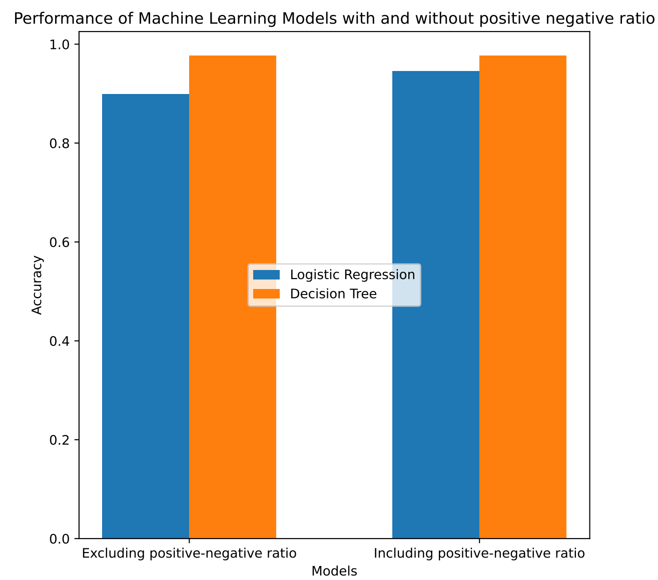

Figure 3. Performance of different models on the dataset with and without positive negative ratio

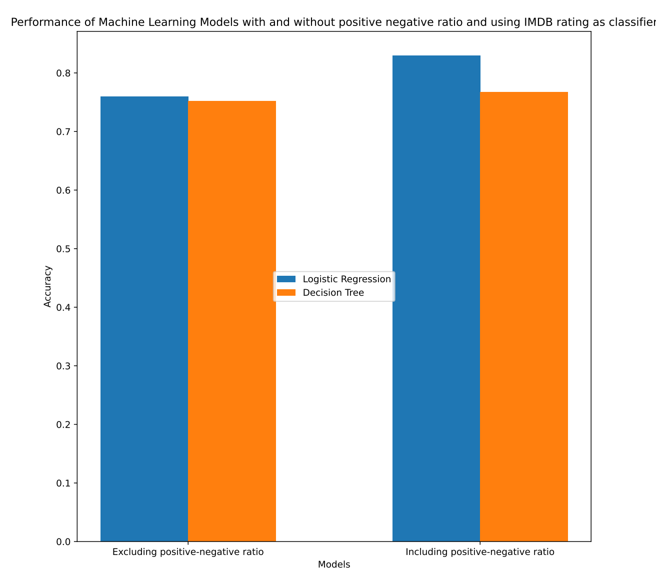

Figure 4. Performance of different models on the dataset using IMDB Rating as the classifier with and without positive negative ratio

7. Conclusion

From all our experiments we can conclude that YouTube trailer comments alone cannot help in the prediction of the rating(good, average, bad) of the movie. We see this clearly in experiments 1 and 2 where in we changed our approaches and used multiple models to see if a good enough result can achieved. However, these positive and negative comment values can be added to previous models and then be trained to give better accuracy. We also noticed that when the model used the feature "Success" of the dataset as the classification label it performed significantly better than when the model used "IMDB Rating Classification" as the classification label. The IMDB ratings of a movie are subjected to many criteria and not usually the public response to a movie. As our model focuses more on predicting the public response to a movie IMDB Rating might not be the best classification criteria. We believe if the similar tests were carried out using the "Success" feature as the classification label we might be able to achieve better performance with the machine learning models.

References

[1] K. R. Apala, M. Jose, C.-C. C. Supreme Motnam, K. J. Liszka, and F. de Gregorio. Prediction of movies box office performance using social media, 2013.

[2] R. Jain. Sentiment analysis on youtube movie trailer comments to determine the impact on box-office earning, 2017.

[3] H. Timani, P. Shah, and M. Joshi. Predicting success of a movie from youtube trailer comments using sentiment analysis, 2019.