Intelligent Prediction of Ratings and Profit Estimation of Upcoming Movies

Snehal Patel (*Author*)
Department of Computer Science,
Illinois Institute of Technology,
Chicago-IL
spate141@hawk.iit.edu

Aditya Kulkarni (*Author*)
Department of Computer Science,
Illinois Institute of Technology,
Chicago-IL
akulka17@hawk.iit.edu

***Abstract*— Success of a movie in terms of ratings and profit, is linked to a number of factors notably the cast, directors, production house, advertising and movie category. Predicting a movie's rating requires an overall consideration of its impact on social networks and thorough analysis of past relevant resources. In our analysis, we are using three approaches to predict a movie's success. The first approach is a IMDb-based 'Prediction model', the second approach is using Twitter API for to analyze upcoming movie-related tweets and the third approach involves the use of YouTube API in analyzing the likes/dislikes of the upcoming movie's official trailers. Our main objective is to predict the success of an upcoming movie based on studying and analyzing the influence of Twitter, YouTube and IMDb-based Prediction model.**

Keywords—rating; estimation; prediction, tweets, trailer;

I. Introduction

Predicting an upcoming movie's rating and estimating its financial success is crucial on the part of producers and investors who usually find it difficult to estimate the revenue based on expenses involved in movie production. Also, on the part of viewers, based on the rating of a movie, viewers could decide if going for a movie is worth spending the money.

The factors associated with the success of an upcoming movie are taken into consideration during the prediction analysis process. Our prediction algorithm consists of three approaches. The first approach is an IMDb-based 'Prediction model' that computes final rating using weighted sum formula. The second approach is using Twitter API for performing sentiment analysis of upcoming movie-related tweets and using classifiers to predict the movie's rating. The third approach involves the use of YouTube API in analyzing the likes, dislikes & viewer count of the upcoming movie's top 10 viewed trailers.

Our analysis is structured into three phases. In the first phase, we used Twitter API to collect trending tweets of an upcoming movie and performed sentiment analysis on those tweets to generate positive and negative scores. In the second phase, we used YouTube API to analyze likes and dislikes of top trending trailers. In the third phase, we calculated score for the upcoming movies rating & its estimated revenue from the movie data we collected in the time range of year 2000 to 2015.

II. Data

A. *IMDb movies data using OMDB API*

We used OMDB API to collect IMDb movie-related information of about 8000 movies from year 2000 to year 2015. The API response consisted of 34 movie parameters for each movies. For our analysis, we needed the following parameters,

- Actors
- Directors
- Producers
- Genre
- Ratings from IMDb
- Rotten Tomatoes ratings
- Budget

Further, the data we got from OMDb API was saved in excel worksheet categorized by movies parameter. For better accessibility, we converted the data for easy accessibility & saved it into JSON format.

B. *Twitter API*

Using Twitter API we collected about 10,000 tweets relevant to the upcoming movie.

C. *YouTube API*

For analysis using YouTube API, we collected following parameters of top 10 most viewed trailers of the upcoming movie,

- View count
- Like count
- Dislike count
- Viewers count

III. Methods & Experiments

A. *Collect movies data from OMDb API*

We have used OMDb which is an API to access data from IMDb, Meta critics & Rotten Tomatoes website to fetch movie information. Movie information consist of different types of 34 parameters. OMDb API has no request limit to get the movies information. But from our experiments we found that with more than 500 movies information took higher time to get the response. So, to overcome this problem first we created a list of movies title & their released years from year 2000 to 2015 and saved it into separate excel file. After then we used that information to get movies data from OMDb API in request of 500 movies at a time. Received information saved in separate excel file. At the end, we merged all those excel files into one single file for efficient use.

We are using openpyxl API to manage excel files in our project. Excel data is converted back into JSON format for faster data processing. All movies data collected, merged & preprocessed were saved into final *"data_new.json"* file.

B. *Get the upcoming movie's name & its released year information from user.*

At the beginning of the code user needs to enter movie name & its released year information. Later this information will use for fetching movie related tweets from twitter & getting trailer information from YouTube.

After user entered the movie title & released year. Data regarding the movie will be fetched from OMDb API. From that response we are just using actors, directors, genre, and production house parameter for further calculation.

C. *Collect movie related tweets from Twitter*

To collect the tweets from Twitter API, we need to follow the instruction provided by https://dev.twitter.com/ to get the credential to access Twitter API to fetch tweets. After successfully establishing the connection with twitter, we are sending request to Twitter API to get 'search/tweets' for particular query, i.e. in our case movie name.

For this project we are collecting 10,000 tweets. We have saved all tweets into pickle to ensure any accidental loss of information. All tweets were saved into "tweets_10k.pkl" pickle. Similarly as the movie information stored into JSON format, we have stored all tweets into "json_data.txt" file for further use.

D. *Computing Sentimet of tweets*

Getting the sentiment of tweets is one of the key functionality. We tried various approaches to compute the sentiment of tweets, some of them are:

1. Tokenizing tweet & finding important words from tweets. Comparing those words with AFINN Data set & obtain the final sentiment of entire tweet.
2. Using sentiment 140 API to classify the entire tweet as positive or negative.

From above two approaches, we found that sentiment140 results were more accurate compared to first approach. So, finally we decided to go with 2nd approach to get the sentiments of the tweet. The format in which Sentiment140 take request is as follow: (Example 1)

```
{"data":    [{"text":    "I    love
Titanic."},
                  {"text":   "I   hate
Titanic."}]}
```

We converted all 10,000 tweets which we have collected into same format & send those tweets as request to Sentiment140 API. The response will be the same as the request, except a new field "polarity" added to each object. For example, the response for Example 1 above will look like the following:

```
{"data": [{"text": "I love
Titanic.",    "id":1234, "polarity":
4},
          {"text": "I hate
Titanic.", "id":4567, "polarity":
0}]}
```

The polarity values are:

- 0: negative tweet
- 2: neutral tweet
- 4: positive tweet

All tweets obtained from sentiment 140 API with their polarity are saved into JSON file "json_data_response.txt". We have saved all tweets into separate list, positive, negative & neutral tweets.

E. *Tweets analysis using classifier.*

For the analysis of tweets, we have divided all positive and negative tweets into two parts:

(1) train_all (2) test_all

We are tokenizing all tweets to convert each terms into tokens. Labels for each tweets are defined by the length of positive & negative tweets. Positive tweets are labeled as 1 & Negative tweets are labeled as 0. Unique terms are identified by applying CountVectorizer() from scikit learn library. A new nxm csr matrix is generated. We have used LogisticRegresion() classifier to compute the accuracy of model. Later, that classifier is used to predict the accuracy of test tweets.

F. *Weighted Measurement of Movies data*

To compute the upcoming movie rating and its estimated revenue, we have used weighted sum of individual scores of various movies parameters as follows:

Initial input features	
Features	**Weights**
Actors	3.5
Directors	3
Genre	2
Production House	1.5
Total	10

$$Score = \sum_{i=1}^{n} M_i * W_i \,/\, \sum_{i=1}^{n} W_i$$

$$m_i - \textit{score of a movie that artist was in}$$
$$W_i - \textit{contribution of artist p in movie } m_i$$

The Score for individual parameter is calculated from above equation. Which later used for predicting upcoming movie's rating & its estimated business.

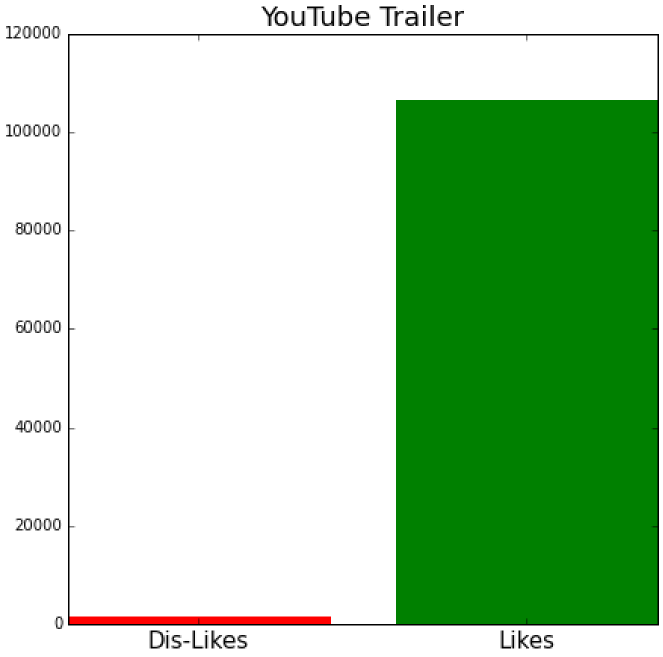
G. *YouTube Data analysis.*

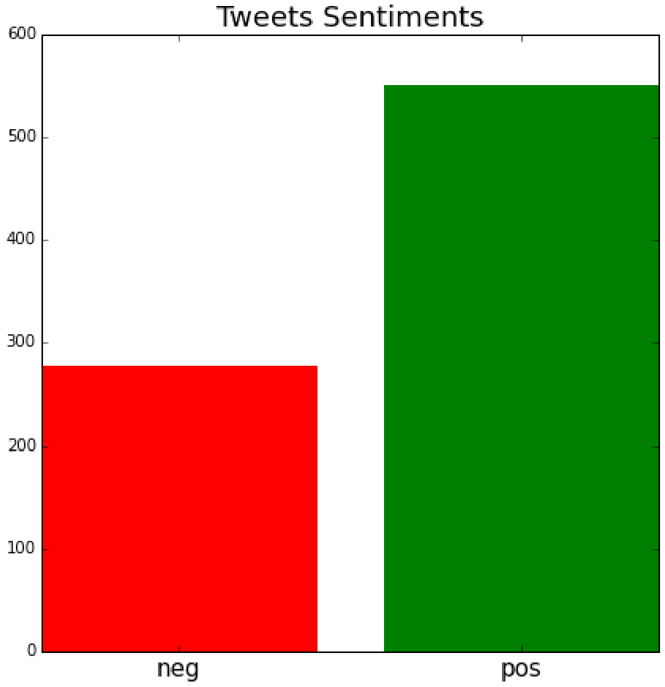
We are collecting video data from YouTube using pafy API & collecting top 10 video information. Information included:

1. Video Likes
2. Video Dislikes
3. Video view counts
4. Video rating

After collecting all this information, we have generated average likes, dislikes & view count measurements. From all this measurements individual rating is generated for all video and later can be used to determine overall rating for particular search.

Sample graph outputs for YouTube & Twitter sentiments is shown below:





IV. Related work

Most of the projects dealing with 'Movie Rating and Success Prediction' primarily worked on collecting and analyzing IMDb data and fitting classifiers for predicting results. In our project, we have used two approaches in addition to the Prediction model based on IMDb data. We have successfully analyzed and classified YouTube statistics and Twitter tweets relevant to the upcoming movie for better understanding of the impact of social networks on the success of an upcoming movie.

V. Conclusions and Future Work

For this project, we have used three major social network platform to get data. (1) Twitter (2) YouTube (3) IMDb. All data was in raw format which needed to perform data mining & prepare that data for required use. We have applied various scientific majors to prepare the data to use for specific purpose. Based on a comparative analysis of accuracies of individual ratings from three different approaches and the accuracy of combined rating of all the approaches, we can conclude that Twitter & YouTube influences the rating of any upcoming movie’s rating.

VI. References

[1] Twitter API: https://dev.twitter.com/

[2] YouTube API: https://developers.google.com/youtube/

[3] OMDb API: https://github.com/dgilland/omdb.py

[4] Sentiment140: http://help.sentiment140.com/api

[5] OpenPyExcel: https://openpyxl.readthedocs.org/en/latest/

[6] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). Learning Word Vectors for Sentiment Analysis. The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011).

[7] Sci kit learn: http://scikit-learn.org/stable/