Final Report

Predicting Movie Success based on Tweet Sentiments

Under Guidance of

Prof. Rong Duan

by

Ankit Luv Mittal

Ajinkya Bobade

Table of Contents

Abstract

There are multiple factors that determine a movie's financial success and one of the major factors is if the movie is anticipated by the masses. The higher the anticipation, the higher the financial success of the movie in its early days. One of the ways to determine if a releasing movie is being anticipated by the masses is to analyse the sentiments about the movie via tweets. The purpose of this project is to determine if there's a correlation between the anticipation sentiment before a movie releases and its eventual user rating on IMDB after its release. It is assumed, in this project, that if the movie has high user ratings on IMDB (greater than 8), then it was financially successful. In this project, we decided on a list of 5 popular movies that released in last few months. The tweets regarding these 5 movies went through our NLTK based sentiment analysis code one by one. This code determined if a particular tweet is positive or negative. We eliminated the option of a neutral sentiment on a tweet to keep our sentiments polarized. After going through our ratings and plotting codes, we get a resulting graph where we see the movie and its user rating along with its positive and negative sentiment numbers. If the number for the positive sentiments is more than the negative, and the IMDB user rating for movie is above an 8, then we can conclude that there is some correlation between the anticipation sentiment of the movie and its eventual user rating. Same goes with the movie rating being below 8 and negative sentiments outnumbering the positive sentiments. We used Python and its abundance of libraries to accomplish the goal of this project.

Introduction

Predicting a movie's box office success has always been trivial business. Movie producers want to make films to attract large crowds so that they can make profits on their initial investments. At the end of the day, no matter how good the movie is to the critic, the producers only care if they get a solid return on investment on the movie. This means movies need to please the masses instead of entertaining only a particular niche of the population. It is risky for the producers to wait till the movie release date to determine the anticipation of the crowd. Producers can utilize social media platforms such as Twitter to generate some numbers regarding the crowd anticipation on a particular movie. Twitter is a social media platform where people express their opinion freely and concisely (due to the character limitations). There are about 200,000 movie related tweets on a daily basis, which means there's good amount of data out there to determine a particular movie's anticipation. Identifying the sentiment behind the tweets about a particular movie could help determine its success at the boxoffice.

For example, if there's a collective negative sentiment out of 1000 tweets for a particular movie that is releasing soon, then it's safe to predict that the movie won't perform well at the box office and the producers will incur losses. If the sentiment analysis is accurate, then this tool could be really useful for the movie producers to gain insight on the sentiment of the masses about their upcoming movie. If they notice that the sentiment is bad, then the movie producers have a chance to improve their marketing tactics which can hopefully create more excitement and anticipation for the movie. Since we are collecting all the tweets in realtime to develop our tweet database, the process takes a bit longer because we have to wait for the tweets to come in. This tool could potentially end up saving the producers a lot of money by avoiding their losses.

Key Hypothesis : Movies with an IMDB user rating higher than 8 is expected to have an overall positive sentiment rating and good financial success. Similarly, movies with a rating below 7 will have an overall negative sentiment and go into losses.

Contribution : We worked on this project together and have equally contributed towards the completion of this project. We met weekly for discussion and to ensure successful implementation of the project.

Data Collection

First, we had to determine which movies to select to conduct this project. We decided to get the most popular movies from this year so that we can get enough tweet data and keep it relevant. We chose the first 4 movies (that have already released) to first check if there's actually a correlation, and we chose the last movie to predict its user rating based on the sentiment. These are the 5 movies we picked in the order of their release date:

1. Barbershop: The Next Cut
2. Captain America: Civil War
3. Jack Reacher
4. Doctor Strange
5. Moana

Since we were waiting for the tweets to come in real time, we couldn't afford to use other relatively non popular movies, such as Keanu for example, to build our database. We had to use popular movies that are still running in the nearby theatres to get the most tweets out of live tweeting. Once we had our movies finalized, we began collecting our database. This process is described in detail in Step 4 of the Approach Section.

Approach

Our step by step approach of writing code and running the project, and instructions for a new user to run the code. We have written all our following codes that are listed below with the help of NLTK tutorials online:

- sentiment_mod_builder.py
- sentiment_mod.py
- twittersemantics.py
- All Ratings plot.py

We also used a number of libraries to run our codes: NLTK, numpy, pickle, scikitlearn, statistics, tweepy, time, datetime, json, imdbpy, scipy, and matplotlib. All these libraries along with Python need to be installed before you begin.

1. Create a new folder on the local drive. Label it with the desired name. Load these two text files into the new folder:

Positive(https://pythonprogramming.net/static/downloads/short_reviews/positive.txt)
Negative(https://pythonprogramming.net/static/downloads/short_reviews/negative.txt)

These files will serve as your sentiment dictionary for the rest of the project. Once we load them, we can update these files to make them more accurate.

2. Download and run the sentiment_mod_builder.py code and save it into the folder. This code will create pickle files within the folder. Pickle is a helpful tool in Python to convert python objects into character streams. Read up on Pickle here .

3. We are ready to create our twitter database files now. Load and run twittersemantics.py code. This code uses the sentiment_mod as an import so that the process is a bit faster. Examining the code, the default movie we have there is “Barbershop” and it will run all the tweets with the hashtag of “Barbershop” filtered in English language. We can change the movie to run the tweets for any hashtag since the code finds tweets that are tweeted in realtime based on hashtags. If the tweet doesn’t have a hashtag, it won’t show
in the output.

To change the hashtag for another movie, simply find this line:

```
twitterStream.filter(track=["Barbershop"],languages=['en'])
```

and change Barbershop to whatever movie hashtag, we want to track/collect.

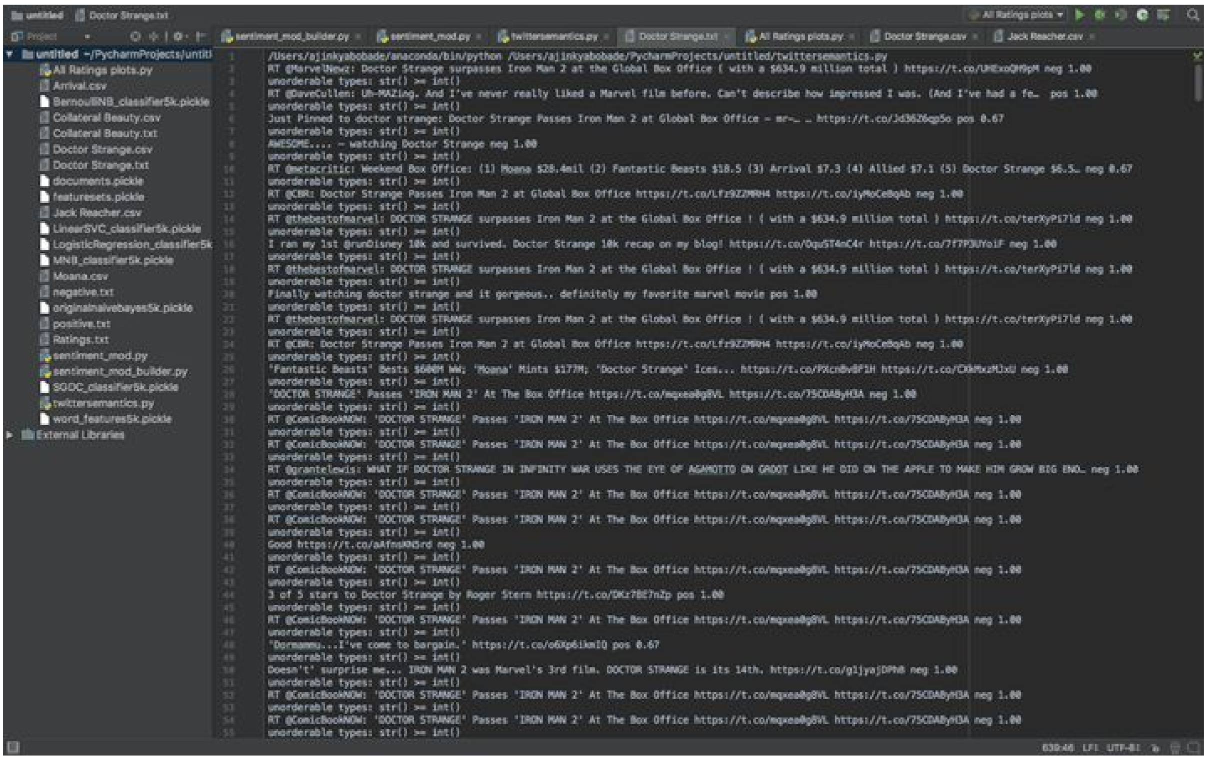
Run the code once changes are done. This process will take some time depending on the popularity of the hashtag, so have patience and wait for at least 1000 tweets for a proper database collection. The output should have the original tweet, a negative or a positive sentiment along with the tweet, and a confidence rating. The confidence rating shows the user its confidence (out of 1). For example, if the sentiment is pos and the confidence rating is 0.8, then the NLTK sentiment analysis code is trying to convey that it is 80% confident that the tweet has a positive sentiment. Once we have collected about a 1000 tweets, we suggest to stop running the code. We repeated changing names for every movie in our list (5 times). After collecting the reviews on Python terminal we make a text file with movie name (for eg. Barbershop.txt) and save the data from the terminal to the text file.

Our outputs were stored in 5 different .txt files that served as our database for tweets for each movie we had selected.

5. We are onto our last step now! Once database is ready, execute All_Ratings_plots.py code. The code asks for input files which are the name of the database files here. The code reads the database files and separates positive and negative sentiments into its own list and writes them into another output file. This code also fetches the IMDB movie user rating from their website. The folder should now have a Ratings.txt file with a list of movies we selected. Along with this list, there is a positive and negative sentiment scoring (out of 10) for each movie and its IMDB user rating. A bar graph is plotted where all the movies are in the x axis and the positive (red) and negative (green) sentiment scoring and IMDB user rating (blue) are in the y axis.

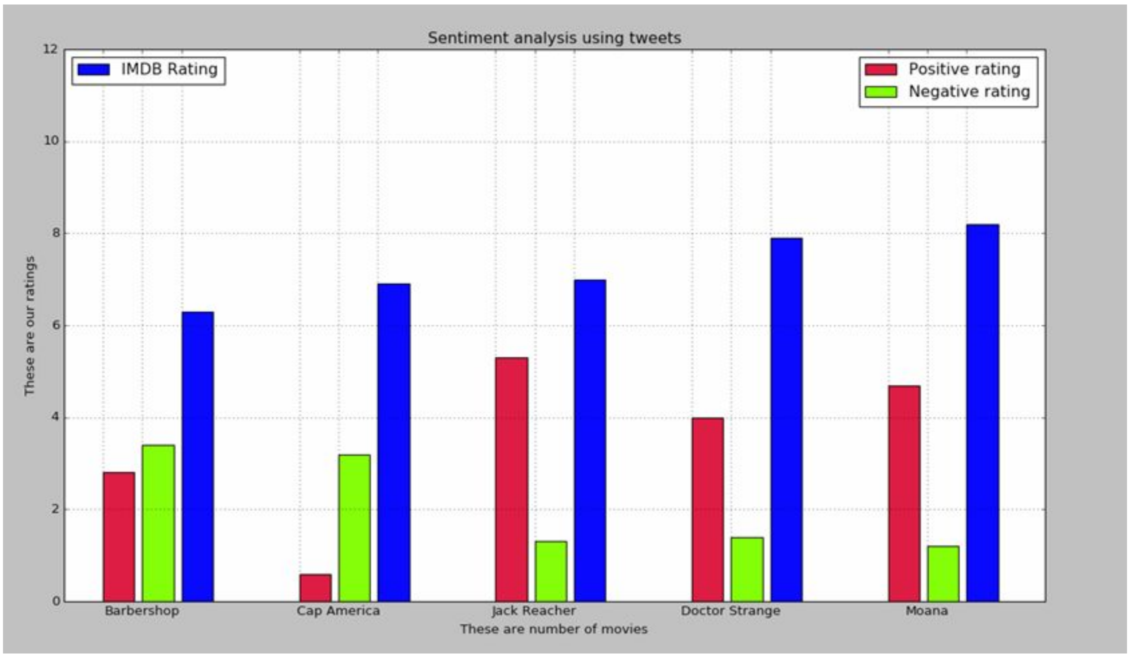
Results

With scikitlearn, we used multiple algorithms to determine the confidence for the sentiment. Here's an example of how data collection for the particular movie looks like:



```
/Users/ajinkyabobade/anaconda/bin/python /Users/ajinkyabobade/PycharmProjects/untitled/twittersemantics.py
RT @MarvelNewz: Doctor Strange surpasses Iron Man 2 at the Global Box Office ( with a $634.9 million total ) https://t.co/UHExoQH9pM neg 1.00
unorderable types: str() >= int()
RT @DaveCullen: Uh-MAZing. And I've never really liked a Marvel film before. Can't describe how impressed I was. (And I've had a fe…  pos 1.00
unorderable types: str() >= int()
Just Pinned to doctor strange: Doctor Strange Passes Iron Man 2 at Global Box Office - mr-… … https://t.co/Jd3626qp5o pos 0.67
unorderable types: str() >= int()
AWESOME.... - watching Doctor Strange neg 1.00
unorderable types: str() >= int()
RT @metacritic: Weekend Box Office: (1) Moana $28.4mil (2) Fantastic Beasts $18.5 (3) Arrival $7.3 (4) Allied $7.1 (5) Doctor Strange $6.5… neg 0.67
unorderable types: str() >= int()
RT @CBR: Doctor Strange Passes Iron Man 2 at Global Box Office https://t.co/Lfz9Z2MRH4 https://t.co/iyMoCeBqAb neg 1.00
unorderable types: str() >= int()
RT @thebestofmarvel: DOCTOR STRANGE surpasses Iron Man 2 at the Global Box Office ! ( with a $634.9 million total ) https://t.co/terXyPi7ld neg 1.00
unorderable types: str() >= int()
I ran my 1st @runDisney 10k and survived. Doctor Strange 10k recap on my blog! https://t.co/OquST4nC4r https://t.co/7f7P3UYoiF neg 1.00
unorderable types: str() >= int()
RT @thebestofmarvel: DOCTOR STRANGE surpasses Iron Man 2 at the Global Box Office ! ( with a $634.9 million total ) https://t.co/terXyPi7ld neg 1.00
unorderable types: str() >= int()
Finally watching doctor strange and it gorgeous.. definitely my favorite marvel movie pos 1.00
unorderable types: str() >= int()
RT @thebestofmarvel: DOCTOR STRANGE surpasses Iron Man 2 at the Global Box Office ! ( with a $634.9 million total ) https://t.co/terXyPi7ld neg 1.00
unorderable types: str() >= int()
RT @CBR: Doctor Strange Passes Iron Man 2 at Global Box Office https://t.co/Lfz9Z2MRH4 https://t.co/iyMoCeBqAb neg 1.00
unorderable types: str() >= int()
'Fantastic Beasts' Bests $600M WW; 'Moana' Mints $177M; 'Doctor Strange' Ices... https://t.co/PXcnBv8F1H https://t.co/CXWMxzMJxU neg 1.00
unorderable types: str() >= int()
'DOCTOR STRANGE' Passes 'IRON MAN 2' At The Box Office https://t.co/mqxea0g8VL https://t.co/75CDAByH3A neg 1.00
unorderable types: str() >= int()
RT @ComicBookNOW: 'DOCTOR STRANGE' Passes 'IRON MAN 2' At The Box Office https://t.co/mqxea0g8VL https://t.co/75CDAByH3A neg 1.00
unorderable types: str() >= int()
RT @ComicBookNOW: 'DOCTOR STRANGE' Passes 'IRON MAN 2' At The Box Office https://t.co/mqxea0g8VL https://t.co/75CDAByH3A neg 1.00
unorderable types: str() >= int()
RT @grantelewis: WHAT IF DOCTOR STRANGE IN INFINITY WAR USES THE EYE OF AGAMOTTO ON GROOT LIKE HE DID ON THE APPLE TO MAKE HIM GROW BIG END… neg 1.00
unorderable types: str() >= int()
RT @ComicBookNOW: 'DOCTOR STRANGE' Passes 'IRON MAN 2' At The Box Office https://t.co/mqxea0g8VL https://t.co/75CDAByH3A neg 1.00
unorderable types: str() >= int()
RT @ComicBookNOW: 'DOCTOR STRANGE' Passes 'IRON MAN 2' At The Box Office https://t.co/mqxea0g8VL https://t.co/75CDAByH3A neg 1.00
unorderable types: str() >= int()
Good https://t.co/aAfnsKN5rd neg 1.00
unorderable types: str() >= int()
RT @ComicBookNOW: 'DOCTOR STRANGE' Passes 'IRON MAN 2' At The Box Office https://t.co/mqxea0g8VL https://t.co/75CDAByH3A neg 1.00
unorderable types: str() >= int()
3 of 5 stars to Doctor Strange by Roger Stern https://t.co/DKz7BE7nZp pos 1.00
unorderable types: str() >= int()
RT @ComicBookNOW: 'DOCTOR STRANGE' Passes 'IRON MAN 2' At The Box Office https://t.co/mqxea0g8VL https://t.co/75CDAByH3A neg 1.00
unorderable types: str() >= int()
'Dormammu...I've come to bargain.' https://t.co/o6Xp6ikmIQ pos 0.67
unorderable types: str() >= int()
Doesn't' surprise me... IRON MAN 2 was Marvel's 3rd film. DOCTOR STRANGE is its 14th. https://t.co/g1jyaJDPN8 neg 1.00
unorderable types: str() >= int()
RT @ComicBookNOW: 'DOCTOR STRANGE' Passes 'IRON MAN 2' At The Box Office https://t.co/mqxea0g8VL https://t.co/75CDAByH3A neg 1.00
unorderable types: str() >= int()
RT @ComicBookNOW: 'DOCTOR STRANGE' Passes 'IRON MAN 2' At The Box Office https://t.co/mqxea0g8VL https://t.co/75CDAByH3A neg 1.00
unorderable types: str() >= int()
```

Once we get all the database files ready in .txt format, and used it as an input for the All Ratings plots.py code, we get a visual output in the form of a bar graph.



In the above graph, we see that Barbershop has more negative sentiments than the other movies. This means our initial prediction based on sentiment alone was that this movie will be unsuccessful. For the Barbershop, the eventual IMDB user rating was below 7, which confirms our hypothesis that a negative sentiment via tweets leads to an unsuccessful movie. With Moana, we see that the IMDB user rating is above 8 which means the movie was successful parallel to our prediction. Movie Captain America had a slightly more overall negative sentiment than a positive one and so it was predicted to be unsuccessful, i.e. a user rating below 7. The eventual IMDB user rating was below 7 for it and so once again this confirms our hypothesis that positive sentiments directly correlate with good IMDB user ratings and financial success and vice-versa. This graph is better represented in a table with actual numbers below:

Movies (Sorted by their release date)	Average Sentiment (Pos / Neg)	IMDB User Rating	Our Prediction Based on Sentiment	Was it a correct prediction?	Accuracy
Barbershop: The Next Cut	2.8 / **3.5**	6.3	Unsuccessful	YES	✓
Captain America: Civil War	**0.8** / 3.3	6.8	Unsuccessful	YES	✓
Jack Reacher	**5.3** / 1.4	6.9	Successful	No	✗
Doctor Strange	4.0/1.5	7.9	Successful	YES	✓
Moana	4.6/1.3	8.2	Successful	Predicted	?

‘Our prediction based on sentiment’ column was solely based on the movie’s overall average sentiment rating. If the negative sentiment rating was dominant or more than positive one, then the prediction was unsuccessful and vice versa. The Jack Reacher’s overall rating’s prediction went wrong as it was supposed to be emerged as a huge hit though the actual IMDB rating says quite the opposite. We did predict the other movies correctly. Moana’s prediction will also most likely be a hit because of the overwhelming positive sentiment on Twitter but we left accuracy as “**?**” for it because it’s released recently and ratings might change with time.

Conclusion

To conclude, this project was an excellent learning curve. Working with new libraries such as scikitlearn, NLTK, and tweepy was a challenge, but a rewarding experience after looking at our overall results. Revisiting the purpose of the project, we do think there’s correlation between the tweeting sentiments and movie’s eventual IMDB user rating and its success or failure at the box office. We cannot quantify the exact amount of correlation, however with the 5 movies we picked, we did achieve 80% success rate. What we learned from this project was that the people who tweet about movies are mostly the same people who take interest in rating these movies online and also end up influencing people if they should watch the movie or not, thus impacting the movie’s financial business. Therefore, yes there’s a good amount of correlation between the sentiment about a movie through tweets and it’s eventual success. Movie producers would benefit a lot from this kind of a tool.

Challenges

There are multiple challenges, we faced, while working on this project but we would like to reiterate only two which are still a hurdle for this tool to be successfully used.

- The challenge was to pick a relevant movie where we could get enough data for our tweet database. For example Jack Reacher was not that popular movie and thus predicting wrong rating.
- Movies with same name creates ambiguity. For example Captain America: Civil War was the movie, we tried to predict but as usually people don’t tweet with the full name, we searched for tweets with tag of “Captain America” which included the tweets from the first movie “Captain America” & others with same name and also, gave us the IMDB rating of the first movie “Captain America”. Thus predicting the movies with similar names or sequels is also a challenge.

References

NLTK tutorials and Scikitlearn to develop the project:
https://pythonprogramming.net/naivebayesclassifiernltktutorial/
https://pythonprogramming.net/sklearnscikitlearnnltktutorial/
https://pythonprogramming.net/pythonpicklemodulesaveobjectsserialization/
http://scikitlearn.org/stable/tutorial/machine_learning_map/
https://pythonprogramming.net/static/downloads/short_reviews/positive.txt
https://pythonprogramming.net/static/downloads/short_reviews/negative.txt