

Final Report

Predicting Movie Success based on Tweet Sentiments

*Group Members:*

Parth Patel

Sarvesh Shah

Ajinkya Bobade

*“I pledge my honor that I have abided by the Stevens Honor System.”*

May 4, 2016.

Table of Contents

[Abstract](#_njtoh7nazfla)

[Introduction](#_xk7lcxj7jfe5)

[Data Collection](#_maihu3et4061)

[Approach](#_sjgqdqfk74tr)

[Results](#_f1uzwve2q9c9)

[Conclusion](#_5buzm7k2m9kj)

[References](#_lun85dd5x3j1)

# Abstract

There are many 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 analyze 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 March, April, and May. 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 box-office. 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 real-time 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.

# 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. Deadpool
2. Zootopia
3. The Jungle Book
4. Barbershop: The Next Cut
5. Captain America: Civil War

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 within the last few months and 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

The approach for this project will be written in a ‘readme’ kind of way so that it serves dual purpose: 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
* getratings.py
* plotting.py

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

1. First step is to create a new folder on your local drive. Label it whatever you wish. Download these two text files: [Positive](https://pythonprogramming.net/static/downloads/short_reviews/positive.txt) and [Negative](https://pythonprogramming.net/static/downloads/short_reviews/negative.txt) into your new folder. These files will serve as your sentiment dictionary for the rest of your project. Once you download them, you can update these files to make them more accurate.
2. Next step is to download the sentiment\_mod\_builder.py code and save it into your folder. Run the code once it’s fully downloaded. This code will create pickle files within your folders. Pickle is a helpful tool in Python to convert python objects into character streams. Read up on Pickle [here](https://pythontips.com/2013/08/02/what-is-pickle-in-python/).
3. Now, download and execute sentiment\_mod.py code. This code utilizes all the pickle files and helps the entire live tweeting process run faster.
4. We are ready to start creating our twitter database files now. Download the twittersemantics.py code. Don’t run it yet! This code uses the sentiment\_mod as an import so that the process is a bit faster. When you examine the code, the default movie we have there is “Zootopia” and it will run all the tweets with the hashtag of “zootopia” 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 real-time based on hashtags. If the tweet doesn’t have a hashtag, it won’t show in the output.
   1. To change the hashtag for another movie, simply find this line:

and change zootopia to whatever movie hashtag you want to track/collect. After changing that, you may want to change the name of the output file in this line:

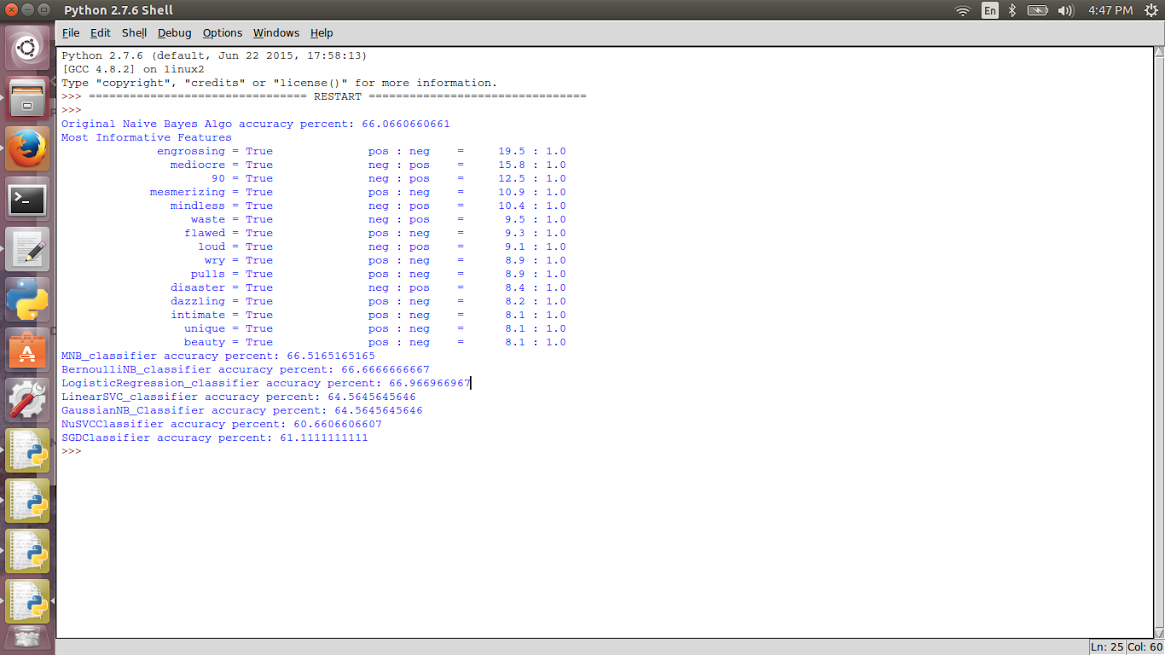
 to any title you want for your database file.

Run the code once you have made your changes. 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 telling you that it is 80% confident that the tweet has a positive sentiment. Once you have collected about a 1000 tweets, we suggest you to stop running the code. We did this for every movie in our list (5 times). Our outputs were stored in 5 different .csv files that served as our database for tweets for each movie we had selected. Check out the Results section for a screenshot of how the database output should look like.

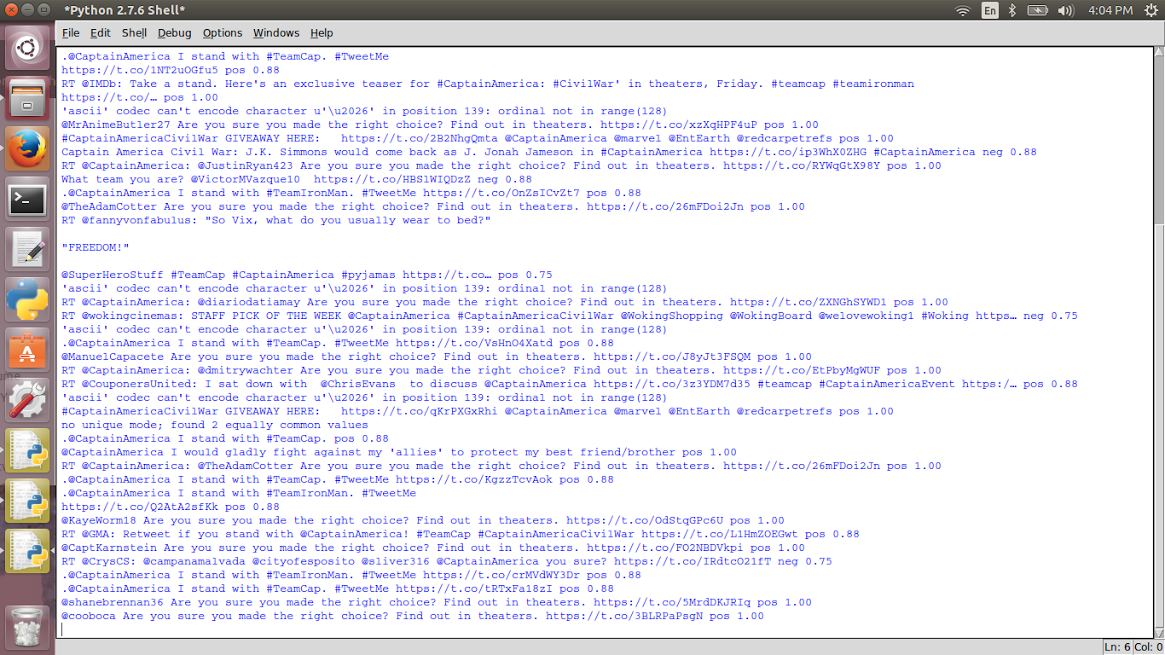
1. We are onto our last step now! Once you have your database, download All Ratings plots.py code. Don’t run it yet. This code asks for input files: so enter your name of the database files here. This code reads your database files and separates positive and negative sentiments into its own list and writes them into another output file. This code also fetches the IMDB movie user rating from their website, so change this line of code to get the rating of the movie you got the database for: Run the code now. Your folder should now have a ratings.txt file with a list of movies you had selected. Along with this list, you will also see a positive and negative sentiment scoring (out of 10) for each movie and its IMDB user rating. You will also get a bar graph like plot 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

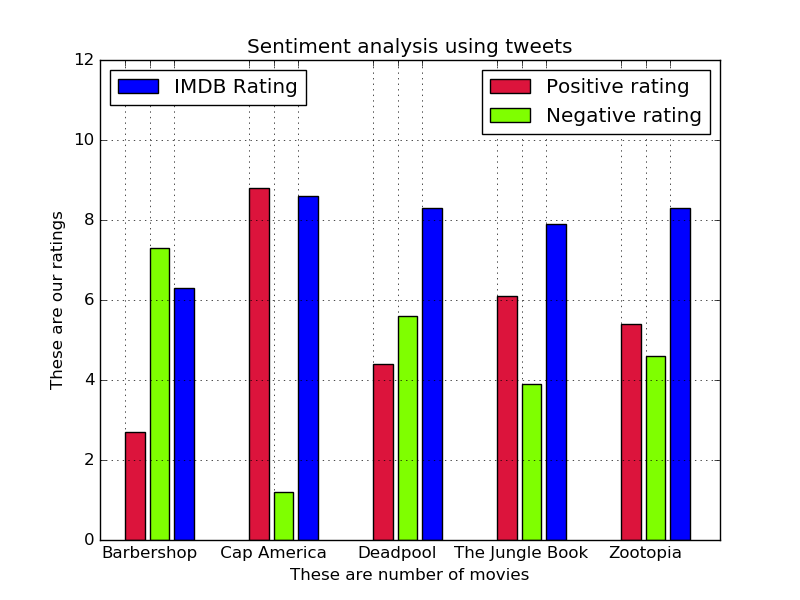
With sci-kitlearn, we used 8 algorithms to determine the confidence for the sentiment. Below are the 8 algorithms and the accuracy for each algorithm:



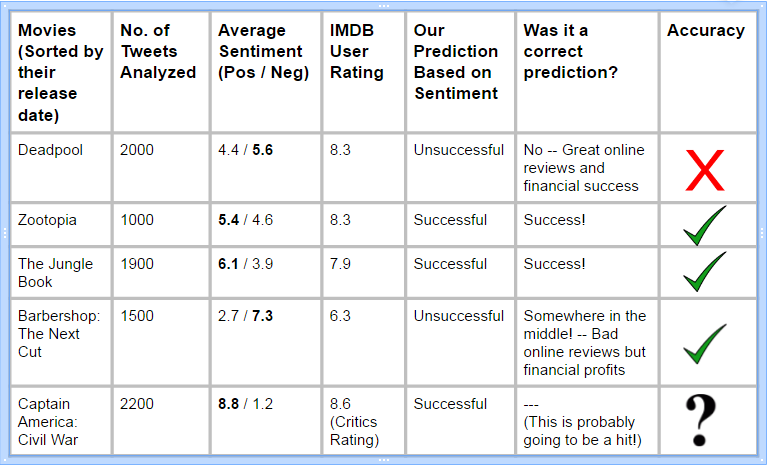
Here’s an example of how data collection for the particular movie should look like:



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



In the above graph, we see that Barbershop and Deadpool have more negative sentiments than the other movies. This means our initial prediction based on sentiment alone was that these movies will be unsuccessful. For the Barbershop, the eventual IMDB user rating was below a 7, which confirms our hypothesis that a negative sentiment via tweets leads to an unsuccessful movie. With Deadpool, we see that the IMDB user rating is above an 8 which means the movie was successful and our prediction accuracy was incorrect. Movies such as Zootopia and The Jungle Book had a slightly more overall positive sentiment than a negative one and so they were predicted to be successful, i.e. a user rating above an 8. The eventual IMDB user rating were above an 8 for both the movies and so once again this confirms our hypothesis that positive sentiments directly correlate with good IMDB user ratings and financial success. This graph is better represented in a table with actual numbers below.



‘Our prediction based on sentiment’ column was solely based on the movie’s overall average sentiment rating. If the sentiment was negative, then the prediction was unsuccessful and vice-versa. As mentioned above, we went wrong with the Deadpool prediction. Deadpool emerged a huge hit even though the sentiment on Twitter about the movie was quite the opposite. We did predict the other movies correctly. Captain America’s prediction will also most likely be a hit because of the overwhelming positive sentiment on Twitter.

# Conclusion

To conclude, this project was an excellent learning curve for all of us. Working with new libraries such as scikit-learn, 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 (considering Captain America will be a huge success). What we all gleaned 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 the sentiment about a movie through tweets and it’s eventual success and movie producers would benefit a lot from this kind of a tool.

# References

We exclusively used NLTK tutorials and Scikit-learn to develop code for this project:

<https://pythonprogramming.net/naive-bayes-classifier-nltk-tutorial/>

<https://pythonprogramming.net/sklearn-scikit-learn-nltk-tutorial/>

<https://pythonprogramming.net/python-pickle-module-save-objects-serialization/>

<http://scikit-learn.org/stable/tutorial/machine_learning_map/>