Project Overview

We used an IMDB review from each of the top 10 movies (list below) to see if top grossing movies are reviewed accordingly. The movies on the list are all pretty different, so we wanted to see how common words were similar or different in reviews for each movie using a counter function, and we also wanted to use sentiment analysis to rerank this top 10 list based on positive sentiments.

List of Top 10 films of all time (Worldwide Gross):

1. Avengers: Endgame (2019)
2. Avatar (2009)
3. Titanic (1997)
4. Star Wars: Episode VII - The Force Awakens (2015)
5. Avengers: Infinity War (2018)
6. Jurassic World (2015)
7. The Lion King (2019)
8. Marvel's The Avengers (2012)
9. Furious 7 (2015)
10. Frozen II (2019)

Source: https://www.filmsite.org/boxoffice.html

Implementation

Our process began by first realizing that we wanted to analyze movie reviews from IMDB for the top 10 grossing movies. We followed the process of importing the data from imdb and assigned values to each movie's name, the values being the reviews for the movies. For future user defined statements we created a list with all of the movie values and another list for each movie name.

We then started the process of creating our user defined functions to start mining into the data. The "new_list" function goes through the first review of every movie and splits the lines into each individual word (we also made sure to make every word lowercase for future analysis, so there is not a distinction between upper and lower). That function coupled with the "histogram" function creates a dictionary of how many times each word is used in each review and organizes them by most to least. We thought this would be relatively interesting to see what the top 5 words are in each review.

We lastly used the “main” function to put all of our previous functions into one place to change and add lines of code we were interested in exploring. In addition to the frequency of words, we were interested in the sentiment ranking for each of the movies to see which one really has the more positive rating to reorder the top 10 movies list. We did this by taking the same reviews used in the word analysis and ran the “SentimentIntensityAnalyzer” with “polarity_scores” to show the positivity vs negativity of each review. While we were trying to find a way to organize the information so it was not all just clumped together we created a print in the for loop in the main function that went through the ‘movies’ list we created, but quickly realized this did not work because the list carried values for each of the movie names instead of a string. In response we created a new list called “movies_string” where we created a list with each movie as a string. Then we put a line in the for loop that would print the movie name for each iteration and move to the next movie as the for loop progressed.

Results: Most Used Words

Endgame ('the', 23) ('to', 18) ('in', 12) ('and', 12) ('was', 11) Unique words: 331	Avatar ('the', 61) ('and', 43) ('of', 37) ('a', 32) ('to', 28) Unique Words: 597	Titanic ('the', 21) ('and', 12) ('to', 8) ('of', 7) ('a', 7) Unique Words: 187	Star Wars Episode VII ('a', 4) ('and', 2) ('is', 2) ('it', 2) ('the', 2) Unique Words: 43	Infinity War ('the', 10) ('is', 7) ('better', 6) ('film', 4) ('to', 4) Unique Words: 69
Jurassic World ('so', 6) ('many', 4) ('of', 3) ('who', 3) ('and', 2) Unique Words: 76	Lion King ('the', 34) ('and', 22) ('in', 16) ('a', 16) ('to', 13) Unique Words: 326	The Avengers ('the', 39) ('is', 26) ('and', 23) ('this', 19) ('of', 18) Unique Words: 211	Furious 7 ('the', 14) ('of', 11) ('to', 6) ('a', 6) ('was', 5) Unique Words: 136	Frozen II ('the', 9) ('had', 3) ('first', 2) ('it', 2) ('story', 2) Unique Words: 75

As part of our analysis, we looked at the top words used across all 10 movies, and were interested to find that “the” was the top word for 8 our of 10, and only not present in the top 5 for 1 movie. There were only 17 unique words in the table above out of 50. This just shows that no matter how different two movies are (like Titanic and The Avengers), words are so similar in reviews and are oftentimes incredibly generic. We were surprised to see how far we had to scroll

in some of the dictionaries to find words or character names that were related to the specific movie like Marvel or Zazu.

Results: Sentiment Analysis

Endgame: {'neg': 0.093, 'neu': 0.731, 'pos': 0.176, 'compound': 0.9947}

Avatar: {'neg': 0.058, 'neu': 0.795, 'pos': 0.147, 'compound': 0.9991}

Titanic: {'neg': 0.064, 'neu': 0.711, 'pos': 0.225, 'compound': 0.9959}

Star Wars: {'neg': 0.272, 'neu': 0.617, 'pos': 0.111, 'compound': -0.9035}

Infinity Wars: {'neg': 0.064, 'neu': 0.54, 'pos': 0.396, 'compound': 0.994}

Jurassic Park: {'neg': 0.229, 'neu': 0.745, 'pos': 0.026, 'compound': -0.9763}

The Lion King: {'neg': 0.074, 'neu': 0.825, 'pos': 0.101, 'compound': 0.9585}

The Avengers: {'neg': 0.063, 'neu': 0.697, 'pos': 0.239, 'compound': 0.999}

Furious 7: {'neg': 0.081, 'neu': 0.801, 'pos': 0.118, 'compound': 0.8512}

Frozen: {'neg': 0.078, 'neu': 0.859, 'pos': 0.063, 'compound': -0.1027}

We also performed sentiment analysis on each review to see if the top movies are rated accordingly, and to rerank them. As our previous analysis stated, the majority of movies had a very high neutral rating, in some cases as high as .8 or more. We did the “reranking” of movies based on the highest positive sentiment scores and were able to compare it to the original list and determine that not all movies are as good as their sales would make one think. Only one movie, Titanic stayed in the same place. It was really interesting to see how movies can be so overhyped before release and be reviewed so negatively by viewers.

Results: New Top 10 List:

1. Infinity Wars
2. The Avengers
3. Titanic
4. Endgame
5. Avatar
6. Furious 7
7. Star Wars
8. The Lion King
9. Frozen

10. Jurassic Park

Reflection

We did a great job collaborating and working on the code together. Pretty much everything was done in a pair programming format where we shared screens and worked through issues and bounce ideas off of each other instead of staring at our own individual screens for hours. I think this project will also be helpful for our project where we can analyze journal entries to find a user's most mentioned subjects, people, etc. and even compare their inputted mood to the sentiment in their daily journal entries. Something that caused a little bit of struggle in the project was not planning ahead with pseudo code to really map everything out. We went into the process going step by step and coming up with solutions on the spot. If we planned ahead maybe we could have taken less time to debug through code that was wrong.