1. Project Overview [Maximum 100 words]:

I used the IMDb API alongside the Natural Language Toolkit (NLTK) in order to run sentiment analysis on aggregate reviews of currently popular movies. I hoped to gain information about a film's popularity through its reviews, rather than user ratings.

2. Implementation [~2-3 paragraphs]:

I defined a Movie class and instantiated a new Movie class instance for each movie in IMDb's most popular movies list, setting its name, id and reviews based off of the API data received. I also set the movie's sentiment analysis score, using the reviews data I had collected alongside NLTK functionality.

In deciding how to store all of the information I was collecting and calculating for each movie, I considered both dictionaries and classes. I settled upon classes, thinking this would be the most logical way to set attributes for each movie, especially since the data I was receiving from IMDb were objects already.

3. Results [~2-3 paragraphs + figures/examples]:

- If you did some text analysis, what interesting things did you find? Graphs or other visualizations may be very useful here for showing your results.
- If you created a program that does something interesting (e.g. a Markov text synthesizer), be sure to provide a few interesting examples of the program's output.

The results of running my function are provided below. As you can see, not only am I able to source information on the most popular movies currently on IMDb, I am able to see what the sentiment of the current reviews landscape says about each one. Movies that haven't come out yet return no information.

```
Blade Runner 2049 has a sentiment analysis of: {'neg': 0.09, 'neu': 0.734, 'pos': 0.176, 'compound': 1.0}
Star Wars: The Last Jedi has a sentiment analysis of: No reviews yet
Game of Thrones has a sentiment analysis of: {'neg': 0.088, 'neu': 0.733, 'pos': 0.179, 'compound': 1.0}
Blade Runner has a sentiment analysis of: {'neg': 0.102, 'neu': 0.743, 'pos': 0.155, 'compound': 1.0}
It has a sentiment analysis of: {'neg': 0.146, 'neu': 0.709, 'pos': 0.145, 'compound': -0.9967}
American Horror Story has a sentiment analysis of: {'neg': 0.175, 'neu': 0.706, 'pos': 0.119, 'compound': -0.9999}
Justice League has a sentiment analysis of: No reviews yet
Star Trek: Discovery has a sentiment analysis of: {'neg': 0.116, 'neu': 0.758, 'pos': 0.126, 'compound': 0.9947}
Rick and Morty has a sentiment analysis of: {'neg': 0.066, 'neu': 0.702, 'pos': 0.232, 'compound': 1.0}
The Gifted has a sentiment analysis of: {'neg': 0.104, 'neu': 0.723, 'pos': 0.173, 'compound': 0.9998}
Kingsman: The Golden Circle has a sentiment analysis of: {'neg': 0.107, 'neu': 0.723, 'pos': 0.17, 'compound': 1.0}
The Flash has a sentiment analysis of: {'neg': 0.112, 'neu': 0.734, 'pos': 0.154, 'compound': 0.9999}
Stranger Things has a sentiment analysis of: {'neg': 0.077, 'neu': 0.74, 'pos': 0.183, 'compound': 1.0}
Big Mouth has a sentiment analysis of: {'neg': 0.083, 'neu': 0.73, 'pos': 0.188, 'compound': 0.9999}
Professor Marston and the Wonder Women has a sentiment analysis of: {'neg': 0.054, 'neu': 0.764, 'pos': 0.182, 'compound': 1.0}
The Dark Tower has a sentiment analysis of: {'neg': 0.078, 'neu': 0.761, 'pos': 0.161, 'compound': 0.9999}
Riverdale has a sentiment analysis of: {'neg': 0.102, 'neu': 0.745, 'pos': 0.153, 'compound': 0.9999}
Arrow has a sentiment analysis of: {'neg': 0.123, 'neu': 0.719, 'pos': 0.158, 'compound': 0.9998}
The Walking Dead has a sentiment analysis of: {'neg': 0.137, 'neu': 0.719, 'pos': 0.144, 'compound': 0.9952}
Outlander has a sentiment analysis of: {'neg': 0.119, 'neu': 0.71, 'pos': 0.171, 'compound': 0.9999}
Thor: Ragnarok has a sentiment analysis of: {'neg': 0.09, 'neu': 0.686, 'pos': 0.225, 'compound': 1.0}
The Orville has a sentiment analysis of: {'neg': 0.085, 'neu': 0.711, 'pos': 0.203, 'compound': 1.0}
Inhumans has a sentiment analysis of: {'neg': 0.116, 'neu': 0.724, 'pos': 0.16, 'compound': 0.9997}
Lucifer has a sentiment analysis of: {'neg': 0.133, 'neu': 0.688, 'pos': 0.179, 'compound': 0.9997}
Grey's Anatomy has a sentiment analysis of: {'neg': 0.087, 'neu': 0.734, 'pos': 0.179, 'compound': 0.9999}
```

4. Reflection [~1 paragraph]:

This assignment was incredibly fun and I learned a lot about working with APIs through Python and running analysis on text data. I was successful in obtaining all the data that I needed, and was able to integrate this data with the NLTK processing tools to analyze it. In terms of improvement, I would like to add more functionality and provide more visually readable outputs. For example, how do IMDb ratings compare with the sentiment analysis scores for reviews? Could I map this correlation? I scoped the project well, determining exactly how much I could do within the amount of time I had.