

## 1. Project Overview

The purpose of this project is to allow users to get a quick grasp of the majority opinion of a movie by conducting text analytics on movie reviews on IMDB. We pickled the data from IMDB to load and save 25 reviews for each of the 4 movies, "The Shawshank Redemption", "The Godfather", "Disaster Movie", and "Saving Christmas". The first 2 movies are classics that have high ratings and the latter two have the opposite. For text analysis, the techniques that we chose were characterized by word frequencies, computing summary statistics, and NLTK.

## 2. Implementation

The project is divided into two main parts: data fetching and analysis. Since movie reviews are very unstructured data, one main component of our software is data cleaning. However, we recognized that different data processing is needed for our 3 methods of analysis, thus we chose to incorporate helper functions for each analysis method instead of cleansing all at once.

For data fetching, we use a built-in library *imdbpie* and *pickle* to retrieve 25 reviews for each of the four movies and save it as dictionary in the file format of .pickle.

In the first part of our analysis, we want to investigate the sentiment of our movies by counting word frequencies. We are expecting to be able to determine the style and the overall review of the movie based on the most commonly used words. We complete the data mining with two helper functions to concatenate comment word lists and remove unnecessary stopwords and special symbols. Then, we wanted a more sophisticated way to analyze the text because descriptive words can be more useful in our analysis than others. We discovered tokenization as a way of breaking down a text paragraph into building blocks with words for further analysis. Using *token*, we are also able to categorize and tag words by its attributes, such as adjectives, comparative

Minghui Li, Qi Ruan

10/21/2020

## Assignment 2 Report

adjectives, and superlative adjectives, to filter out only adjectives. For data cleaning, similarly, we use two helper functions to concatenate comment texts and remove stop words. The output of this analysis will allow us to see adjectives used reviews by frequency. Lastly, we are also interested in having a more straightforward view of the sentiments of the movie. We chose to use polarity-based sentiment analysis through *NLTK* to assign a score to the movie reviews' overall positivity, negativity, neutrality, and an aggregated score - compound. With our helper function, we normalize our results by conducting a sentiment analysis for each review and compute an average of the scores obtained. Using another helper function, *tabulate*, we organize our scores into a table format.

### 3. Results:

From our first analysis ("Characterizing by Word Frequencies", Fig 1), we successfully obtain the word list by frequency. However, we can see that many nouns, like "film", occupy the list, which is confusing and unhelpful for our analysis.

```
Characterizing by Word Frequencies ...
The Shawshank Redemption: {'film': 79, 'movie': 70, 'shawshank': 62, 'one': 44, 'andy': 39, 'best': 38, 'hope': 37, 'prison': 34, 'redemption': 28, 'time': 26, 'ever': 25, 'story': 24, 'many': 21, 'see': 21, 'life': 20, 'never': 20, 'people': 20, 'robbins': 19, 'red': 18, 'great': 18, 'like': 18, 'movies': 18, 'seen': 16, 'freeman': 15, 'going': 15, 'get': 15, 'say': 14, 'world': 14, 'every': 14, '-': 14, 'films': 14, 'darabont': 13, 'king': 13, 'tim': 13, 'good': 13, 'morgan': 12, 'two': 12, 'things': 12, 'makes': 12, 'us': 12, 'well': 12, 'years': 12, 'still': 12, 'acting': 12, 'way': 12, 'dudresne': 11, 'thing': 11, 'would': 11, 'times': 11, 'friends': 11, 'think': 11, 'find': 11, 'stephen': 10, 'much': 10, 'day': 10, 'better': 10, 'feel': 10, 'really': 10, 'first': 10, 'know': 10, 'hollywood': 10, 'even': 10, 'however': 9, 'yet': 9, 'could': 9, 'made': 9, 'novella': 9}
```

Fig. 1

Given the output of our first analysis, we figure we need to filter our results to only look at adjectives that reflect the sentiment. Our output ("compute\_summary stat", Fig. 2) present us with a clearer picture. For the movie The Shawshank Redemption, we see that adjectives such as "best" and "greatest" are frequently used in reviews, which matches our expectations since this is one of the greatest movies of all time. For comparison, the less acclaimed Disaster Movie ("compute\_summary stat", Fig. 3) is characterized by more negative adjectives such as "bad" and "worst". Immediately, we

Minghui Li, Qi Ruan

10/21/2020

## Assignment 2 Report

are able to grasp whether or not this is a good movie. In addition, we are also able to detect some characteristics such as “freeman” and “unhappy” for Shawshank.

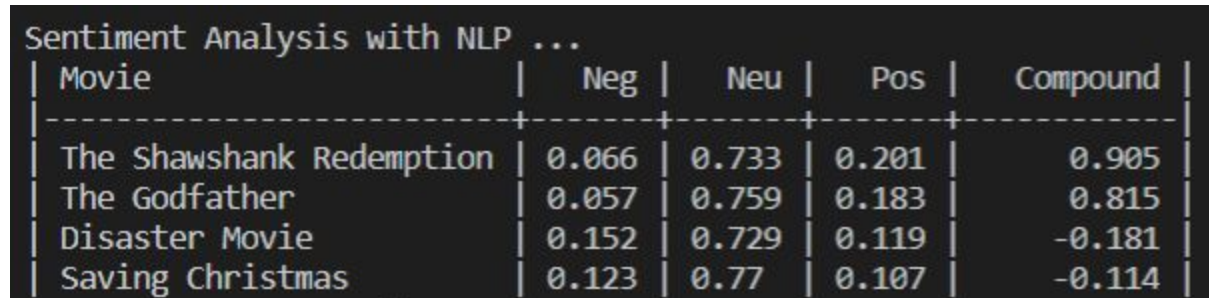
```
Computing Summary Statistics ...
The Shawshank Redemption: {'best': 34, 'shawshank': 24, 'andy': 22, 'many': 21, 'red': 20, 'great': 18, 'good': 13, 'morgan': 9, 'first': 8, 'busy': 8, 'different': 8, 'human': 7, 'better': 7, 'greatest': 7, 'true': 7, 'much': 6, 'strong': 6, 'freeman': 6, 'real': 6, 'fellow': 5, 'warden': 5, 'modern': 5, 'easy': 5, 'last': 5, 'single': 5, 'young': 5, 'brilliant': 5, 'frank': 4, 'emotional': 4, 'special': 4, 'top': 4, 'personal': 4, 'final': 4, 'free': 4, 'excellent': 4, 'unhappy': 4, 'magic': 4, 'tough': 3, 'classic': 3, 'sure': 3, 'new': 3, 'bad': 3, 'bob': 3, 'simple': 3, 'huge': 3, 'right': 3, 'favorite': 3, 'deep': 3, 'due': 3, 'short': 3}
```

Fig. 2

```
Disaster Movie: {'funny': 19, 'bad': 13, 'worst': 12, 'good': 11, 'worse': 9, 'many': 9, 'few': 9, 'least': 8, 'much': 8, 'unfunny': 7, 'better': 7, 'scary': 7, 'crappy': 6, 'awful': 6, 'high': 5, 'stupid': 5, 'real': 5, 'off-topic': 5, 'complete': 5, 'friedberg': 4, 'crystal': 4, 'sure': 4, 'old': 4, 'last': 4, 'horrible': 4, 'epic': 4, 'low': 4, 'wrong': 4, 'new': 4, 'entire': 4, 'first': 4, 'single': 4, 'lower': 4, 'little': 3, 'actual': 3, 'musical': 3, 'small': 3, 'several': 3, 'smart': 3, 'dead': 3, 'lazy': 3, 'flat': 3, 'original': 3, 'previous': 3, 'utter': 3, 'main': 3, 'nicole': 3, 'big': 3, 'angry': 3, 'incredible': 3, 'hard': 2, 'comic': 2, 'fu': 2, 'insulting': 2, 'indiana': 2, 'else': 2, 'pop': 2, 'minor': 2, 'next': 2,}
```

Fig. 3

Lastly, using NLP we were able to obtain a table of polarity scores (“nlp\_sentiment\_analysis”, Fig. 4), which will give us an idea of most people think of the movie. We are able to verify the accuracy of the output since we see The Shawshank Redemption and The Godfather have lower negative scores and higher compound scores than the other two movies. When we compare the results to the IMDB movie pages, we see that the latter two movies have overall lower ratings. Thus, we believe that the compound score comes in handy when we are trying to understand the ratings. In addition, this analysis will provide us more insight than ratings because we are able to see which direction the sentiment is polarized.



The screenshot shows a terminal window with a dark background and light-colored text. The title of the window is "Sentiment Analysis with NLP ...". Below the title is a table with five columns: "Movie", "Neg", "Neu", "Pos", and "Compound". The table contains four rows of data for the movies "The Shawshank Redemption", "The Godfather", "Disaster Movie", and "Saving Christmas". The values for "Neg", "Neu", and "Pos" are normalized scores between 0 and 1, while "Compound" is a combined score ranging from -1 to 1.

Movie	Neg	Neu	Pos	Compound
The Shawshank Redemption	0.066	0.733	0.201	0.905
The Godfather	0.057	0.759	0.183	0.815
Disaster Movie	0.152	0.729	0.119	-0.181
Saving Christmas	0.123	0.77	0.107	-0.114

Fig. 4

In the future, we think both the adjectives analytics and polarity scores are useful in analyzing the sentiments of a movie. Users of this program will be able to get a direct review of the ratings from polarity scores, whereas an in-depth view of sentiments through the most frequent adjectives used in reviews.

#### 4. Reflection:

Overall, the process of coding, collaborating, and debugging was smooth. We aligned our objective to focus on movie review analysis in the first meeting and agreed to pair program together over Zoom meetings on weekends. Our transparent communication helped us a lot in completing this assignment. For example, when one of us had trouble figuring out the logic in a *for loop*, the other would look over the code and identify the issue. We also had a long discussion when we were both confused on how NLTK operates to make sure we were on the same page. However, we learned that we should develop a better working framework before coding in the future. We did not plan ahead in detailed steps for each section or do unit testing, which actually gave us a lot of struggles when we were debugging multiple sections all at once. Therefore, we wish we knew the importance of planning ahead and performing unit testing so we could pinpoint the issue more quickly. The difference in time zones was also a significant obstacle because one of us needed to stay up late in order to work together over the call. This

Minghui Li, Qi Ruan

10/21/2020

### Assignment 2 Report

text mining and analysis assignment is very helpful for us because we are both concentrating in Business Analytics. As we learned how to perform data mining, data cleaning, and sentiment analysis by fetching data from a website, we can apply this skill in our future internships, such as performing customer sentiment analysis to shape marketing strategies.