Jingyu He

Professor Li

Problem Solving with Software Design

11th November 2018

Project Writeup and Reflection – Text Mining and Analysis

- Project Overview

This assignment is completed by using *imdbpie* and *nltk* libraries to complete an analysis between the general ratings of a movie versus the sentiments expressed by the viewers and critics. I hope to use this analysis tool to prove whether a score can be predicted by the viewer's sentiment score. In the process of making the program, I hope I can get used to techniques such as using classes, storing and capturing information in dictionaries and lists. Moreover, it is interesting for me to get familiar with APIs from IMDB and Metacritic, to obtain ratings and written reviews of specific movies.

- Implementation

The code mainly contains five major steps: tally & save movie titles, translate movie titles to imdb-proprietary serial numbers, obtain & store ratings and written reviews of each selected movies, perform nltp-analysis on written reviews and perform data comparisons. Due to the lack of filer-searching functionalities of imbdpie, I manually looked up the top-rated 250 movies on IMDb on the website (https://www.imdb.com/chart/top?ref_=nv_mv_250). With some simple copy and reformatting, the list of top 250 films are made into a .csv file. (imdb_top_250.csv)

After the movie titles are stored in a format that is friendly to be processed by imdbpie, a python script named 'loaddata.py' is created to fetch all the information I need offline: the lookup ID for each of the movies, the ratings of the movie, most popular user reviews and most popular Metacritic reviews. The code intends to load the data from IMDb database once instead of constantly accessing data from the server, which is likely to trigger the blacklisting mechanism. I imported the library of pickle to help store the data in the folder.

After all the data is loaded in the directory, a python script named 'testdata.py' is created to perform the actual analysis. The core library used for the code is nltk, which processes the sentiment element of sentences and give out a score related to the positivity or negativity of the speaker. By averaging all users' and critics' normalized nltk score, the script can now calculate 1. If the critics tend to overpraise movies and 2. If the critics' opinion corresponds closer to the IMDb ratings. The calculations of correlation, in this case, are simply counting and comparing scores between users, critics, and IMDb. If the nltk normalized score is higher than users' score, then it is safe to assume that critics overpraise this specific movie. Similarly, if the difference between the normalized IMDb score and the user's nltk score is higher than the one's from the critics, that movie entry will count as the critics have a closer judgment than the users. Finally, by calculating both numbers, we can have answers to the two questions posed.

- Results

During the process of sampling movies and extracting ratings, we found that some movies do not have enough user reviews or have a Metacritic score because of either their age or

because of their categorization not as a movie. Some non-movies can make into the list due to the lack of filters in the imdb.search_for_title() function.

After filtering out non-movies and entries without reviews, we can see that the answers to both of the questions all favor towards the critics (see spreadsheet below). First, the critics only rate 16.83% of movies with a sentiment score higher than the normal users (43 movies out of 202 movies sampled), which means that from the sample the users have a higher tendency to express their love to the movie with a more aggressive sentiment.

Moving on to the second question whether the critics or the users have a more realistic view (a closer score to the rating from IMDb), the program returns the result that 167 of 202 movies have critics' scores closer to the IMDb ratings. Therefore, it is expected that the critics' score is 82.67% accurate in a movie.

	Top 250 Sample	Samples with meta/user reviews
# of Samples	250	202
	Overpraised?	Closer to IMDb rating?
Critics	34	167
Users	168	35
Total	202	202

- Reflection

This assignment served me as a very good opportunity to learn more about some of the real-world application usage of coding in Python. The aspect of text-analysis provided me with the experience of quantifying people's speech and make developers easier to categorize the positivity/negativity of the user's language. On the other hand, using APIs from the third party is very eye-opening to me: even though that APIs can pull up very useful data, developers still have to customize the system to create custom queries or other functionalities to ensure that the APIs can achieve what developers intend to do. Finally yet importantly, actually using pickle as a quick method to "dump" data offline helped me understand ways that can help me expedite the computing process of my code.