Sentiment Analysis of Instagram Posts

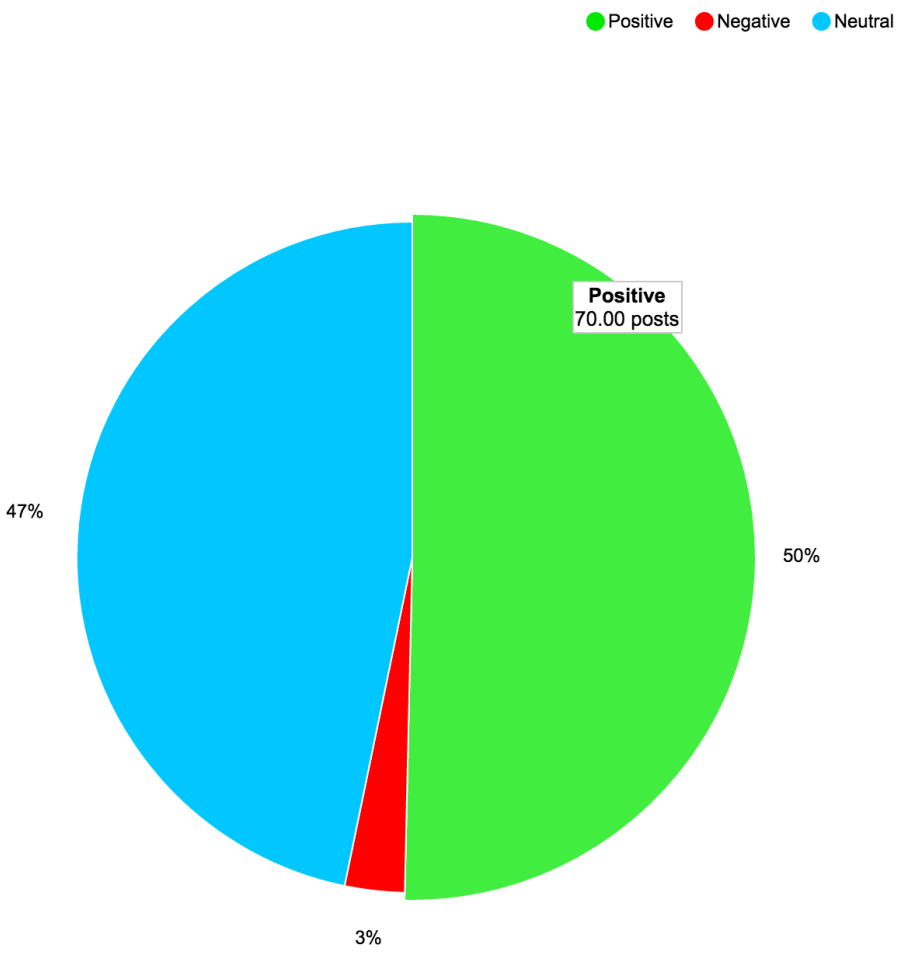
My Python script fetches all Instagram posts for a given tag ("capitalone" in this case) from the past *N* days, where the user specifies *N*. From each post, which is identified by a unique media id, my code extracts:

- The user who posted the post
- The number of likes on the post
- The number of users who follow this user
- The number of users this user follows
- The number of posts this user has made
- The caption on this post

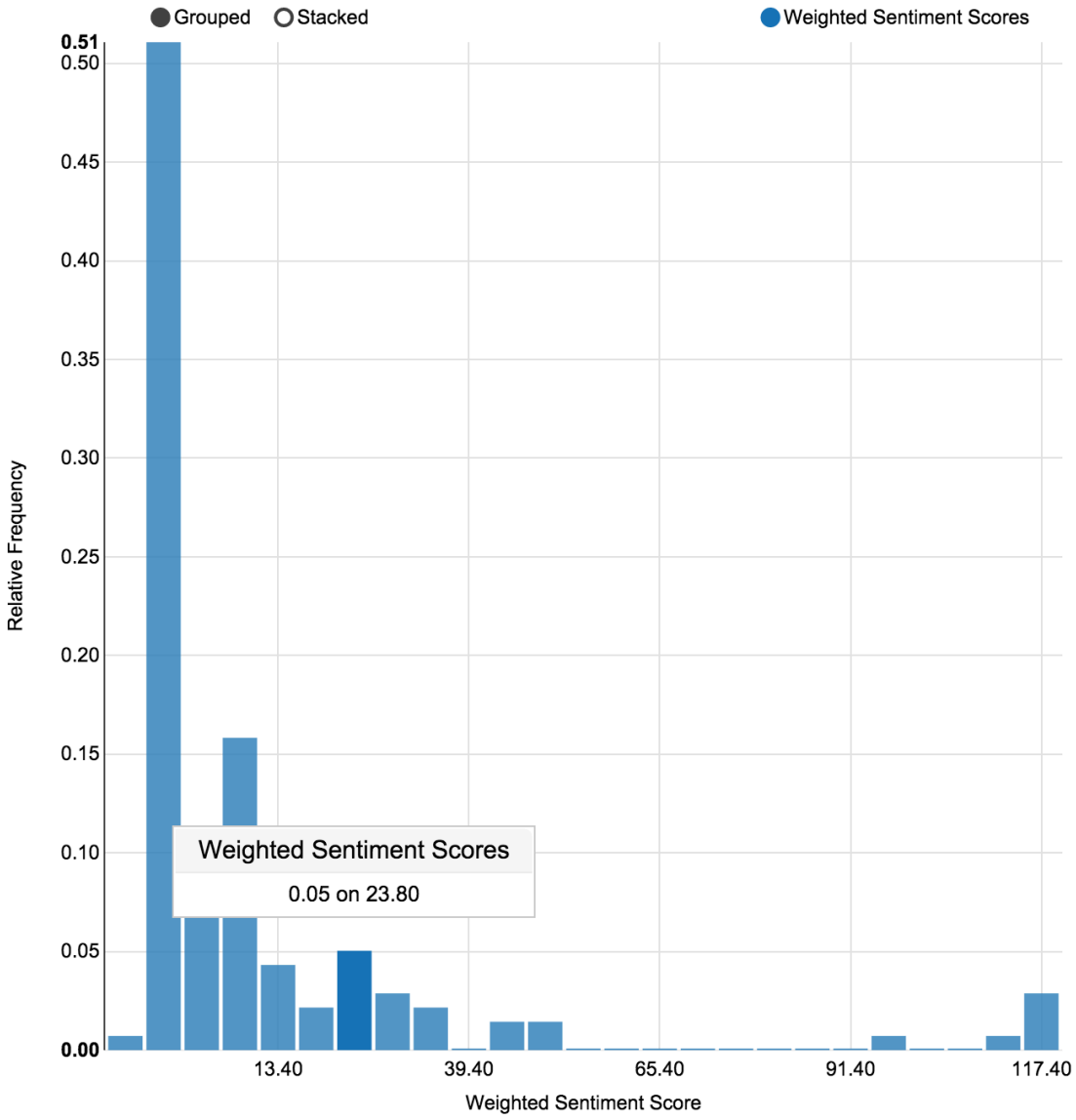
All of this information is stored in a Python dictionary where each media id maps to a nested dictionary with each of these six pieces of information as keys mapping to their respective values. Assuming Instagram API calls run in constant time, this segment of my code executes in O(n), where n is the number of posts.

I then pass the caption to Python's open source Natural Language Toolkit (NLTK), which employs Valence Aware Dictionary for sEntiment Reasoning (VADER) to perform sentiment analysis (one can find more information about VADER in the following paper: http://comp.social.gatech.edu/papers/icwsm14.vader.hutto.pdf). VADER indexes strings of text with positive, neutral, and negative percentages based on a corpus of human ratings of lexical features (words, emoticons, etc.) and grammatical constructs (e.g. exclamation points add emphasis). Vader will output a continuous vector of positive, neutral, and negative percentages, which I convert to one of three classes – 1 for positive, 0 for neutral, -1 for negative – based on the values of the percentages. I then compute a "weighted sentiment score," which is the sentiment classification multiplied by the number of likes the post has. This derived metric takes into account community agreement with the sentiment expressed by the post. High absolute values of weighted sentiment score indicate significant consensus around the sentiment of the post. Assuming sentiment classification occurs in constant time, this segment runs in O(n) as well.

Lastly, my code creates visualizations for the relative frequencies of each sentiment classification and for the distribution of weighted sentiment scores using NVD3, a Python wrapper for d3.js. The relative frequencies of each sentiment classification are displayed in the following pie chart in frequencyPieChart.html:



The distribution of weighted sentiment scores is displayed in the following histogram in weightedDistibutionBarChart.html:



In this way, my script fetches all posts for a given tag over a specified time range, extracts useful metadata from each post, performs sentiment analysis on the caption of each post, and outputs visualizations describing the overall sentiment around the tag.

Thank you for the opportunity to participate in this competition

Python Library Dependencies:

- instagram
- nltk
- nvd3

.