Analyzing YouTube Trends

YouTube keeps a list of the top trending videos on its website that it updates every day. To determine what videos trend, it uses a combination of factors including the number of views, shares, comments, and likes (Variety Magazine). Although one would think that the algorithm used to determine trending (assuming that this is first a classification algorithm - trending or not trending - followed by a ranking algorithm to determine where the video must lie on the trending list) is fair and unbiased, there have been studies conducted that determined that certain channels get prioritized over others. If a video's popularity is defined by trending, views, and likes, what factors determine the popularity of a YouTube video in the United States? Can we attempt to understand how polar and subjective certain videos are, compared to others? My goal is to build a model that performs sentiment analysis on the title and description of YouTube videos that trended between November 2017 and June 2018.

1. Data

I procured the data for this project from Kaggle, since I wanted to pick a source that was reputable and authentic (enough people had examined it beforehand and found no significant errors in its reporting). The dataset was scraped using YouTube's API by Mitchell J, and it contains information on videos trending between 11/14/2017 and 06/14/2018. It examines a series of statistics including number of likes, number of views, video description, and tags.

Feel free to examine the data source here: https://www.kaggle.com/datasnaek/youtube-new

Even though there are only 6351 unique videos covered in this dataset, the dataset contains around 40900 rows. This is because every time a particular video trends, a new entry is created in the dataset. This allows us to examine and compare the trend frequency of different videos.

2. Method

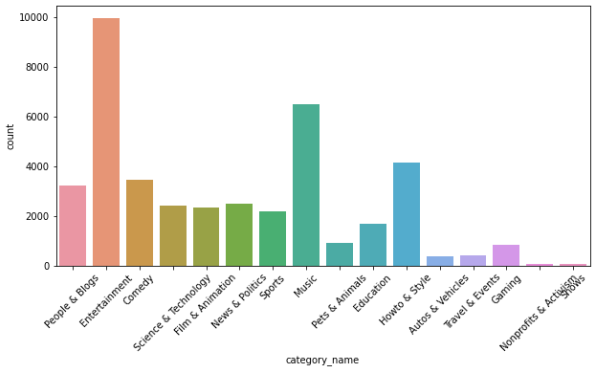
Sentiment analysis refers to understanding the context and emotion behind specific text, and it helps us determine the ratio of positive engagements to negative engagements for a specific topic. This is especially interesting in the realm of YouTube data, since most Youtubers have a starkly different tone which they employ to set their channel apart. There are various tools that can be used for sentiment analysis. NLTK (Natural Language Toolkit), a library commonly employed for textual analysis, contains several text classifiers that are good at sentiment analysis. Textblob is another python library that contains a lot of useful text classification tools. I decided to use the classifiers from this library because the learning curve is much less steep and it is good for a first pass through the data.

3. Data Cleaning

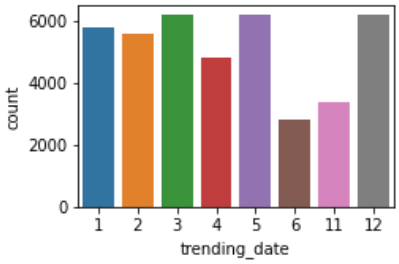
Overall, the dataset was pretty clean, which is what I expected since the author had taken a lot of care to ensure that it was a high-quality dataset. The primary problem with this dataset was that there were several trending videos which were missing descriptions. Now this could either mean that a particular video did not have a description (the creator did not provide one) or that the scraper failed to extract the description text. I examined the percentage of values that were missing and found that only a very small percentage of rows had their descriptions missing. I decided to fill in the missing values with 'No Description' to account for the possibility that the video simply did not contain one.

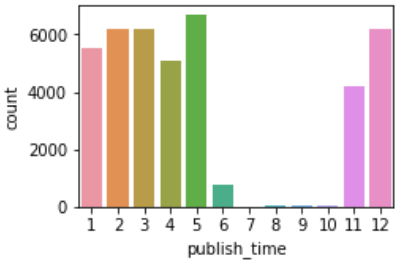
4. Exploratory Data Analysis

The dataset has several variables, some numerical like the number of views a particular video got on a particular day and some categorical like the category of the video. I wanted to examine how the videos in the dataset were distributed based on category, and I observed that the most popular categories are Entertainment, Music, and How-to & Style.



I also wanted to understand the distribution of trending videos over the entire time period that I contain data for – which months saw the greatest number of trends? Which months saw the greatest number of videos being published?



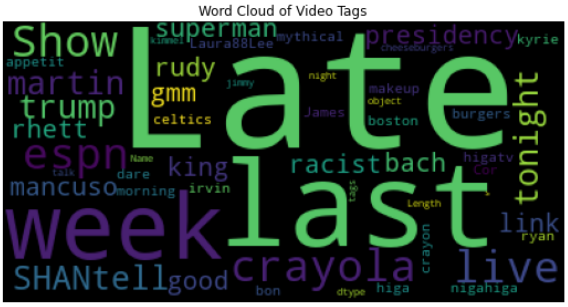


Keeping in mind that the dataset only contains videos that trended for half of June and half of November, May seems to have the greatest number of videos published (maybe to celebrate the start of summer?), and we see a high number of trending videos in March, May, and December. Even though our dataset does not contain any videos that trended in months 7, 8, 9, and 10, we still see a small number videos published during these months. This might lead us to our first insight – videos tend to trend right after they have been published, first and foremost.

Since I wanted to primarily conduct sentiment analysis on video descriptions, I decided to create a video cloud to understand what words tended to appear most often in the video titles and video descriptions.

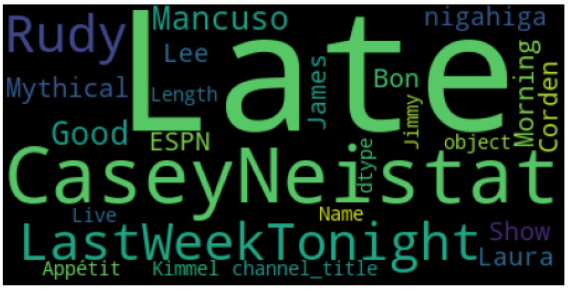
Word Cloud of Video Titles

Word Cloud of Video Tags



Word Cloud of Video Descriptions

Word Cloud of Channel Titles



I see that the word ‘video’ appears most often in the descriptions (unsurprisingly) and there also appears to be several instances of ‘https’ (since a lot of YouTubers tend to want to link their channel or other videos of theirs in their descriptions).

5. Feature Engineering

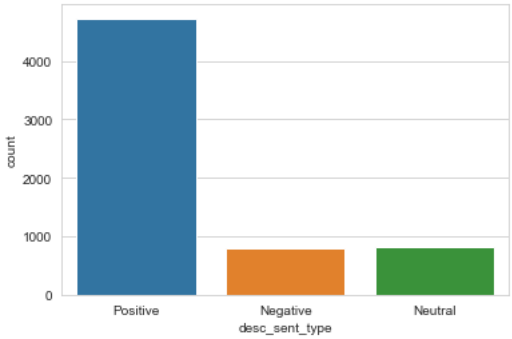
The sentiment analyzer I want to build only makes use of the textual features of the dataset, and those are the ones I need to clean and extract useful features from. I performed several cleaning steps.

(a) Expanding contractions of words (for example, changing “you're” to “you are”)
(b) Removing all the hyperlinks from the text (i.e. links starting with “http” or “https”)
(c) Changing “&” to “and”
(d) Removing the newline character (‘\\n’)
(e) Removing all non-alphabet characters.
(f) Removing unnecessary whitespace (by reducing it to just 1 space between words).
(g) Removing stopwords (by using NLTK's stopwords list).

I wrote a function and used regular expressions to clean each video’s title and description.

6. Modeling

Hurrah! We are finally at the final and most interesting step. I decided to use Textblob to extract the sentiment and the polarity of each description. This will tell me what sentiments tend to be the most polarizing, which sentiments tend to be the most commonly observed in which categories, etc. I observed that most of the videos in the dataset tend to be positive, with a small number of them being negative/neutral.

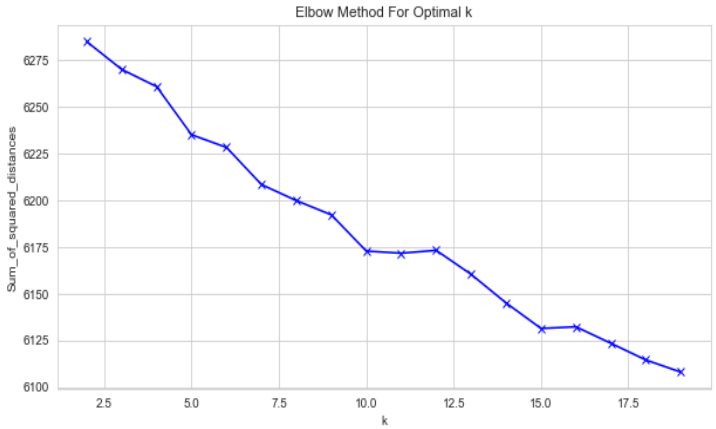


I wanted to understand what kind of videos resulted in a negative sentiment classification, and when I examined them closely I realized that several of these videos did not seem negative to me at all. Most of the descriptions, after being cleaned and removed of connecting words, just consisted of the names of other channels, other social media sites, and the words ‘subscribe,’ ‘like,’ and ‘video’ over and over again. I decided to try a more powerful sentiment analyzer on the same data to see if this might give me more precise results, and I employed the NaiveBayesAnalyzer for my second pass. This is an NLTK model that is trained on a movie

reviews corpus. Running this took a lot more time than the first model (I ended up needing to run it overnight). This analyzer classifies each video as 'pos' or 'neg,' and once again, I observed that most videos in the dataset tend to be classified as positive. However, more videos were classified as negative by the NaiveBayesAnalyzer than by the default PatternAnalyzer. This could just be because the former model does not have a 'neutral' classification category and so is able to classify more videos as overtly negative.

I wanted to understand what made each sentiment analyzer different from the other – were they picking up different words? Was the difference attributable to the specific corpus that each of them had been trained on? So I looked at the specific differences between the results of each. As I predicted, a lot of the videos that the default analyzer classified as 'neutral' were classified by the NaiveBayesAnalyzer as 'neg.' However, there were also several videos that the default analyzer classified as 'positive' than were classified by the more powerful analyzer as negative. I was not able to discern any insight just from looking at the data, and so I did some more research on the exact differences between the two models. Each analyzer employs a pre-defined set of categorized words that helps it 'learn' the sentiment of specific texts, and I think this is what caused the difference between the two analyzers.

As a final text analysis, I wanted to see if there was a way that I could cluster the different video descriptions and see if there was a way I could predict what videos belonged to the same category just through clustering. I knew this might be a tough task since, as I observed earlier, there isn't a lot of meat left on the YouTube descriptions after removing stopwords, emojis, and hyperlinks. I used a vectorizer to break each description into a matrix consisting of the frequencies of each word, and I then clustered the descriptions using K-Means clustering. I used the Elbow Method plot to find the optimal number of clusters.



As you can see, the plot is a little confusing to read. Upon discussing this with my mentor, I decided to use 10 clusters to group the descriptions into, and we came upon this number after closely studying the plot and keeping in mind the number of categories the videos were currently classified as. The videos were split up into 10 clusters but each cluster did not contain a lot of significant information to differentiate it from the others. When I looked at the top words in each cluster, several of them were the same – video, subscribe, facebook, twitter, Instagram. Ultimately, the conclusion I arrived at was that YouTube descriptions are not a good source of clean text for analysis but provide a fun source of social insights.

7. Recommendations

- My insights can be used by data science enthusiasts who also happen to be parents and are interested in the kinds of videos their kids are accessing.
- Most of the videos I observed were positive so I don't think they have too much to worry about, but doing this kind of sentiment analysis not just on the general videos available on YouTube but on specific creators and channels will help them and anyone understand what they're watching.
- YouTube and various ad companies can also use this kind of sentiment analysis to understand what kinds of videos seem to have the greatest viewer engagement and can then specifically monetize those videos and channels.

8. Next Steps

- In the future, I would love to build a better and more powerful sentiment analyzer that can accurately classify videos as positive and negative.
- I would use this analyzer to create a model that, on taking the age of the user, will determine if the video is too negative for young users and either issue a warning or block it.