Team Members: Aravind Patnam, Jeremy Tan, Timothy Le

Trending Here, Trending There

An in-depth exploratory analysis on trending and non trending youtube videos.

In this project, we used our datasets to discover some general exploratory data analysis and some in depth NLP. For exploratory data analysis, we first "played around" with the data in order to get a "feel" for the data. From this, we answered the following questions:

- How long do videos stay trending?
- What is the correlation between likes, comments, dislikes, views, title length, and description length?
- How is this correlation affected in different categories?
- What categories fail and what categories succeed?
- What categories are most popular?
- Which hours is it best to post your video?
- Which country has the most viewer engagement and active audience?
- What are the top videos for each country?
- Which are the most popular and unpopular youtube channels?

For our in depth NLP analysis, we answered the following questions:

- What are the most popular words in tags/descriptions?
- What can we gather from a sentiment analysis?
- How can we predict what topics will be trending?

Through these questions explored, we were able to gather different conclusions such as what videos can get trending status and which don't (from NLP analysis) and what types of videos a creator can make to hit trending successfully (from EDA analysis)

The data sources we used included the datasets we found on Kaggle, and also generated from our own Aravind's use of the youtube api. The ones on Kaggle only contained trending videos statistics; so to perform a holistic analysis, we utilized the youtube api and fine tuned our request payload to get very similar videos that were not trending. This allowed us to conduct a more, wholesome analysis. The list of trending datasets are contained in the zip file and have the word "**trending"** in them, and the non-trending ones have the keyword "**not_trending"** in them. We also provided a bank of positive and negative words which were used for the sentiment analysis. Here is the full list of the file names:
'USvideos.csv','CAvideos.csv', 'DEvideos.csv', 'FRvideos.csv', 'GBvideos.csv', 'INvideos.csv', 'JPvideos.csv','KRvideos.csv', 'MXvideos.csv', 'RUvideos.csv', 'not_trending_us_df.csv', 'not_trending_ca_df.csv', 'not_trending_de_df.csv', 'not_trending_fr_df.csv', 'not_trending_gb_df.csv', 'not_trending_in_df.csv', 'not_trending_jp_df.csv', 'not_trending_kr_df.csv', 'not_trending_mx_df.csv', 'not_trending_ru_df.csv', 'positivewords.txt'

Team Members: Aravind Patnam, Jeremy Tan, Timothy Le

Throughout the project, we found helpful code references to help guide us through this project:

https://realpython.com/python-data-visualization-bokeh/

https://rstudio-pubs-static.s3.amazonaws.com/79360_850b2a69980c4488b1db95987a24867a.html

https://towardsdatascience.com/the-next-level-of-data-visualization-in-python-dd6e99039d5e

https://towardsdatascience.com/basic-binary-sentiment-analysis-using-nltk-c94ba17ae386

https://www.datacamp.com/community/tutorials/text-analytics-beginners-nltk

https://towardsdatascience.com/getting-started-with-plot-ly-3c73706a837c

https://www.youtube.com/watch?v=zHcQPKP6NpM

https://www.youtube.com/watch?v=Ea_KAcdv1vs

https://developers.google.com/youtube/v3/docs/search/list

With these code references to guide us, we were able to plan what we wanted to show in our notebook as it gave us an understanding of what was possible for us. Starting from the top, Alex wanted us to make our data interactive; hence our trial and error findings of what module would work best for our notebook. In addition, for the NLP data in particular, we had no knowledge of how to do sentiment analysis or what an LDA even was. Through these readings, we were able to understand how to use our data to do such tasks. Finally, to grasp what kind of EDA we should do, we watched various videos such as the ones above to understand what we had to do to ascertain useful insights.

In addition, since we did web scraping so late in the course, Aravind had to figure out how to use the youtube api to scrape data . This was essential in order to do more analysis later on.

Throughout the project, we found helpful tools to help us visualize our data and make it interactive:

https://medium.com/@ChannelMeter/youtubes-top-countries-47b0d26dded

https://matplotlib.org/mpl_toolkits/mplot3d/tutorial.html

https://www.machinelearningplus.com/plots/top-50-matplotlib-visualizations-the-master-plots-python/

https://programminghistorian.org/en/lessons/visualizing-with-bokeh

https://plot.ly/python/v3/user-guide/

Over the course of this class, we learned how to use matplotlib and seaborn for graphs. We, however, wanted to make our project more interactive so we turned to using modules such as Bokeh and Plotly. These not only enable us to turn our static graphs dynamic, but also let the end user play around with the visualization. These two modules were essential in our

project as we were able to provide the user the interaction we would have as we played around with the datasets.

For our exploratory data analysis, our visualizations showed us a number of insights. To start, when a video hits trending, a video will usually fall off in one or two days. This especially happens in Eastern countries such as Russia, Japan and Korea, where they immediately fall off after one day. However, for western countries, we found that videos will stay trending for a number of days. This is especially true for Great Britain, where a video cans tay trending for as much as thirty days. This told us, and made us infer, that Great Britain had, if not the highest, user engagement. A later visualization we do later confirms this, where the majority of the views, likes, dislikes, and comments come from that country. Another useful analysis we found is through our correlation scatter matrix and correlation heat maps. We found there to be strong correlation between views and likes, and a somewhat strong correlation between comments and views, comments and likes, and comments and dislikes. These correlations, however, change by category as some categories show stronger or weaker correlations. An example of this is if you were to go to an unpopular category like "trailers," negative correlations appear in place of the previous, strong correlations. Most surprisingly, for categories that elicit human emotions, such as "Pet & Animals" and "Nonprofits & Activism" there is a strong correlation between likes, views, comments, and dislike. We can infer these type of videos cause these strong reactions due to how potent they can be in terms of bringing out the human emotion and thus “action.” Looking at the which categories fail or succeed, we found that in some countries it’s better to make a certain type of video. Although entertainment videos are the most abundant, these have a high success and fail rate. In order to really succeed, there are some categories that are more promising than others. An example of this is Germany, where in this country making a video that fall under “People & Blogs” will give you a higher chance of success to reach trending. In another visualization we show, these videos that give you a higher chance to reach trending does not necessarily mean they are the most popular in terms of views and likes. Taking Germany from the previous example, it’s actually “Music” that is the most popular category and “People & Blogs” don’t even reach the most popular videos (in terms of views and likes). This confirms that making a video in the most popular categories does not guarantee a spot in trending. Moving on, when seeing which videos get the most like per hour, we find that there are certain hours that’s best to post a video in order to get the maximum amount of likes. Although most people post at 4pm, it’s better to post at or after 7pm in order to get the most likes on your videos. Finally, by showing what channels grab most of __ and consistently reach trending, we can see that having a successful channel doesn’t mean having the most of __, but rather consistency in terms of content. A movie trailer may reach trending once , but channels like “Late Night with Seth Meyers” will repeatedly reach it because of their consistent delivery of content.

Some of the final results from the sentiment analysis and LDA are that most trending videos have a negative sentiment while most non-trending videos have a positive sentiment.

These sentiments are also represented via the topics that we found to be the most popular through our LDA model. These can be played around with by using the pyLDAvis visualization presented in the notebook.

Some of the libraries we used include: Pandas, Numpy, Bokeh, Scikit, nltk, requests, spacy, gensim, pickle, matplotlib, pyplot, wordcloud, PIL. These were the main libraries used, but later on decided not to use matplotlib as we wanted to make our graphs interactive.

Some of the more notable tasks that were accomplished included conducting a sentiment analysis using nltk and also the LDA topic inference model using spacy and genism. Since they were some of the stretch goals for our project, they were pretty hard to implement as they took a lot of time and trial and error. We had to make sure all our parameters for the model training were fine tuned properly. Additionally, another common struggle we all had was cleaning the input data so that we could feed it into the model or the other visualizations. Often times, the inputs required were very specific and getting data into that format was rather time consuming. In addition, the transfer from the classroom to an actual project showed us how hard it was to apply what we learned without any real experience. Moreover, converting our static visualizations into dynamic ones was a large learning curve, yet a fruitful one as we were able to produce some stunning visualizations. To sum it up, here are the tasks we ended up doing:

- Preprocessing
- Learning the dataset
- Scraping youtube for non trending data
- Learning how to do “proper” EDA
- Splitting questions to explore and gathering insights from those questions
- Learning Bokeh and Plotly
- Converting static graphs to dynamic

A somewhat minor challenge during this work was one of our members, Jeremy, having to leave for a wedding. This meant he missed meetings due to not being physically here and the time zone difference, but we were able to make it work by staying in contact and updating Jeremy about each meeting.