

Social Media Bonanza

Final Report

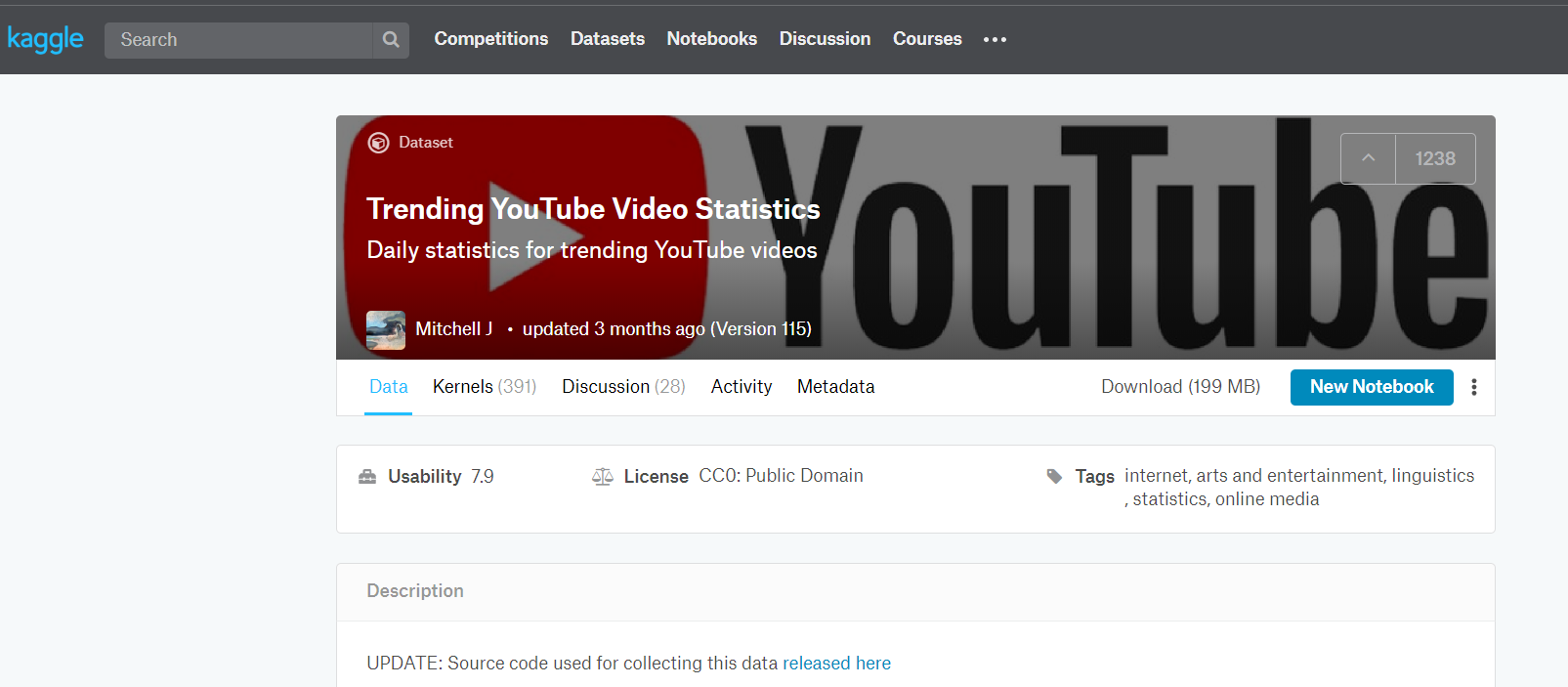
Blake Freeman, Jill Smith, Tomeka Morrison, Trong Nguyen

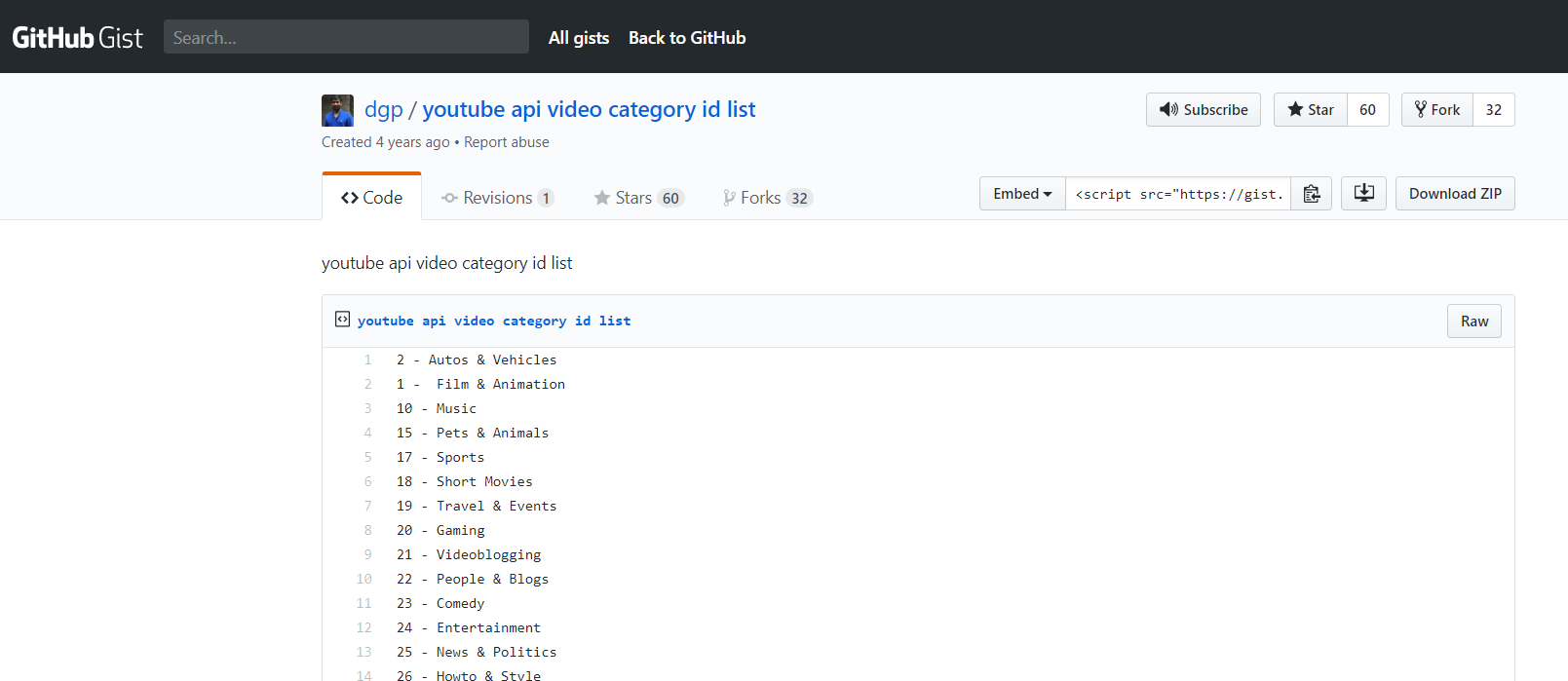
| ELT Project | 8/25/2019

# Extract

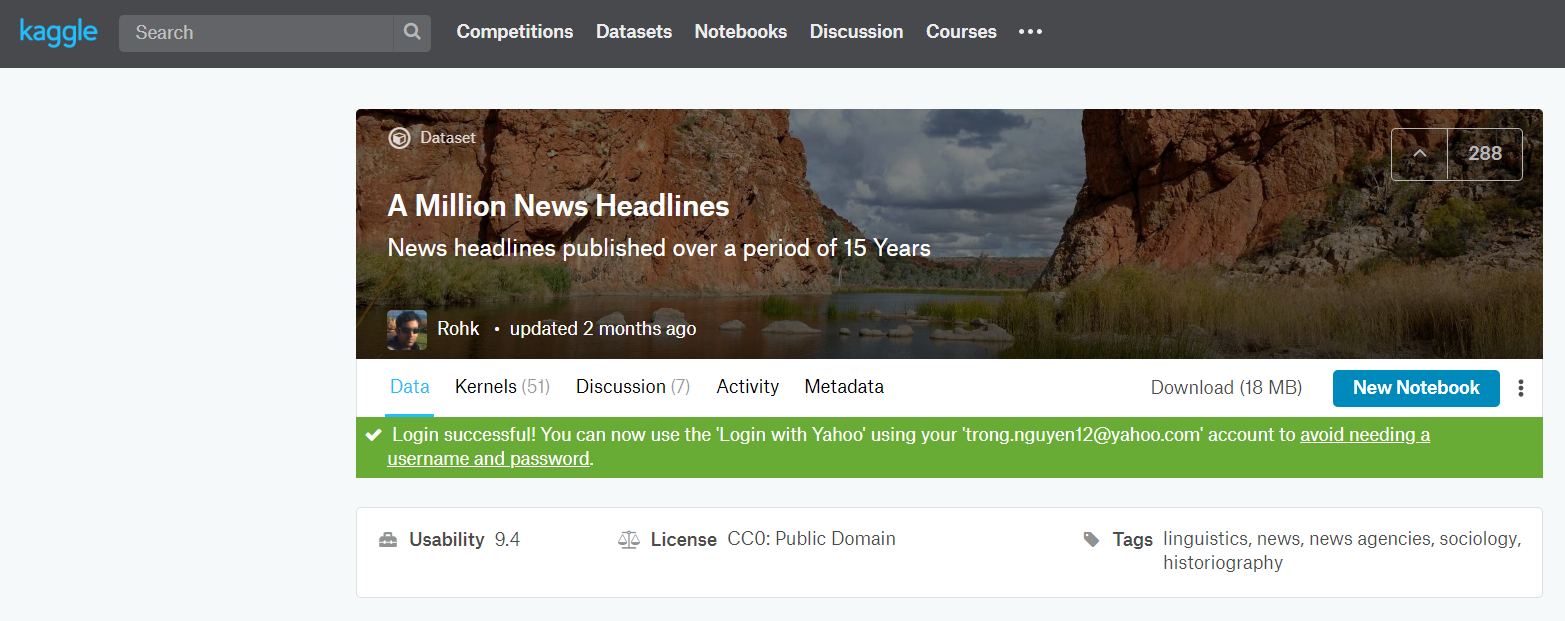
After brainstorming several ideas, we as a group decided that we wanted to see the relationship between news and social media. Basically, if a particular trending category in the news has any effect on a trending topic on YouTube.

For this project, our main YouTube data was extracted from *Trending YouTube Video Statistics* on the Kaggle website. The raw data came in the form of a .csv file that was downloaded and then pulled into Pandas. We also found the most popular YouTube API video category from a user called “dgp” on GitHub Gist.





For our trending news headline, our second data source was pulled from *A Million News Headlines* on the Kaggle website. The raw data came in the form of a .csv file that was downloaded and then pulled into Pandas.



* Need a heading? On the Home tab, in the Styles gallery, just click the heading style you want.
* Notice other styles in that gallery as well, such as for a quote, a numbered list, or a bulleted list like this one.
* For best results when selecting text to copy or edit, don’t include space to the left or right of the characters in your selection.

## Transform

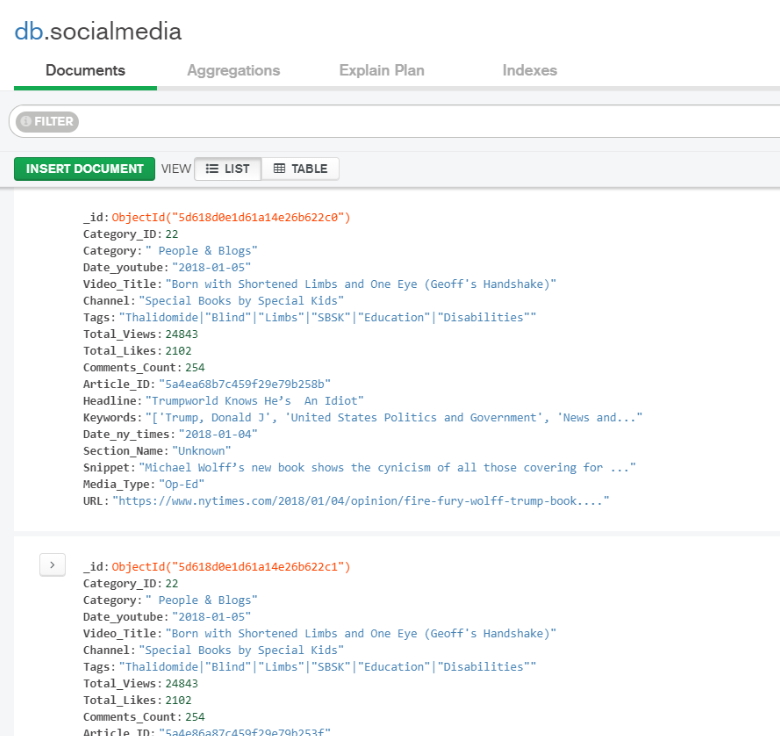
We knew that in order for our analysis work, we had to first transform the raw data from the .csv into a dataset that made sense. Using the Jupyter Notebook, we cleaned the three datasets by converting the time and date to just dates. And then from there, we filte0red out the dates that we wanted to analyze. We found that from 1/4/2018 – 05/01/2018, the dates in both the YouTube data and the News data both overlapped each other.

Next, there were a few categories in the News dataset that didn’t match up with the YouTube dataset, so we recategorizes them as best as we could. For example, the category metro or metropolitan about local news became just News & Politics.

The cleaned YouTube dataset and the cleaned News dataset was then combined and further transformed by dropping any columns that we thought were not needed. We also formatted some of the column headers to make the whole data frame look better.

## Load

To bring everything home, we loaded our final database into MongoDB. In the Jupyter Notebook, we imported pymongo and made a connection to the MongoClient. This took us a few tries, but we finally got it to work.



## Technical analysis breakdown

YouTube:

<https://www.kaggle.com/datasnaek/youtube-new> - Main Data

<https://gist.github.com/dgp/1b24bf2961521bd75d6c> - category ID's

News Articles:

<https://www.kaggle.com/therohk/million-headlines/downloads/million-headlines.zip/8> - Main Data

Clean\_News\_Articles (Jupyter notebook)

* Import articles csv files using pandas and create dataframes.
  + Files used: ArticlesJan2018, ArticlesFeb2018, ArticlesMarch2018, ArticlesApril2018
* Delete extraneous columns (refer to line 6)
* Append all four dataframes togehter to create a single dataframe
* Explore updated dataframe with function .count
* Groupby New\_Category to pull out all the category names and rename categories. (refer to lines 13 to 22)
* Saved cleaned dataframe as csv file

YouTube\_Clean (Jupyter notebook)

* Import YouTube csv and Category\_ID csv
* Use pandas to create dataframes from the csv files
* Use pandas to merge the dataframes on the category\_id
* Pull required columns from dataframe (refer to line 8)
* Use Datetime to update the 'trending\_date' column into datetime format
* Update the column names (refer to line 9)
* Save cleaned dataframe as csv file

Social\_Media\_Bonaza (Jupyter Notebook)

* Import clean YouTube\_Clean and Clean\_News\_Artilces\_v2 csv files
* Use pandas create dataframes from csv files
* Use pandas merged the dataframes on 'Category"
* Use JSON to convert the merged dataframe to a dictionary
* Use pymongo added the merged dataframe to MongoDB