# Web Scraping and Analysing of Three Different YouTube Videos

In this project, I have extracted the required data (title, likes, dislikes, comments) of videos from youtube.com and re-arranged in a proper manner to analyse these videos and get some insight from the data.

Web Scraping: Web scraping is basically a process of extracting data from website using some scripts or automaton tools/software.

Web scraping, also known as web data extraction, is the process of retrieving or “scraping” data from a website.

Tool Used:

1. BeautifulSoup: Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favourite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree.

Beautiful Soup parses anything you give it, and does the tree traversal stuff for you. You can tell it "Find all the links", or "Find all the links of class external Link", or "Find all the links whose URLs match "foo.com", or "Find the table heading that's got bold text, then give me that text."

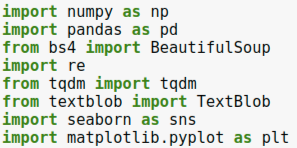
1. Text Blob: TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

Basic Steps to Extract Data:

1. Input HTML page
2. Convert it your favourite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree.
3. Identify the required data
4. Find the data
5. Store the data in specific format

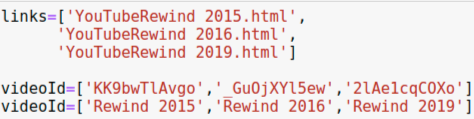
# Part 1: Web Scraping

## Step 1: Include required libraries



## Step 2: Define the HTML page links

I have already downloaded the html pages and in case of we want HTML pages online then define the URLs.



## Step 3: I have created a function for each task.

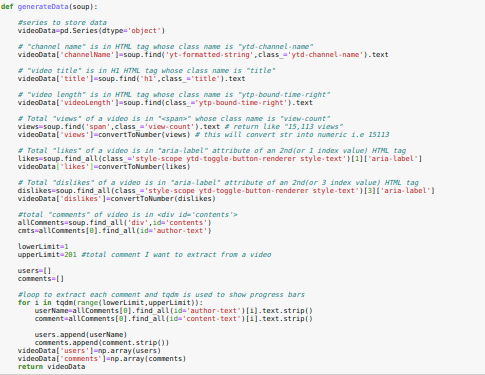
### Step 3.1: conventioneer()

This function will convert the text data into numeric format.

Example: “151,641,939 views” into “151641939”



### Step 3.2: generateData()



This function will collect

data from HTML page

using BeautifulSoup and

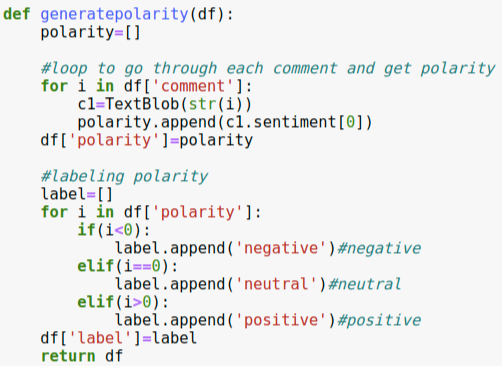
stored in videoData series.

### Step 3.3: generateDataframe()

This function will create

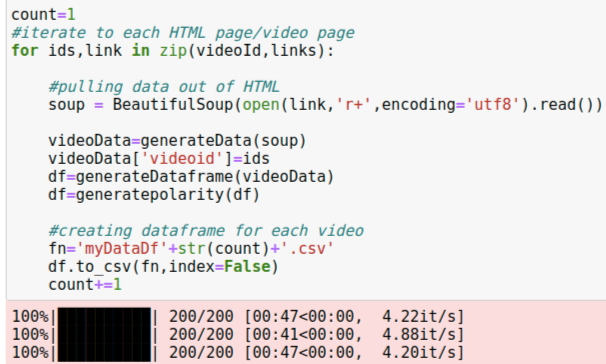
dataframe using above data.

### Step 3.4: generatepolarity()



Above function will generate polarity of each comments i.e. a comment is good, bad or neutral.

### Step 3.5: At last all functions are called and final output/data is generated.



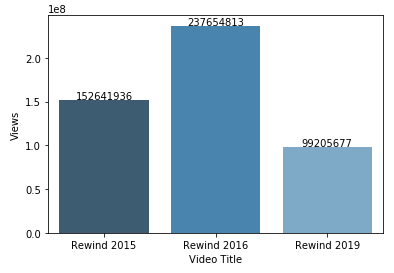
## Final Output:



# Part 2: Data Visualization/Analysing Data

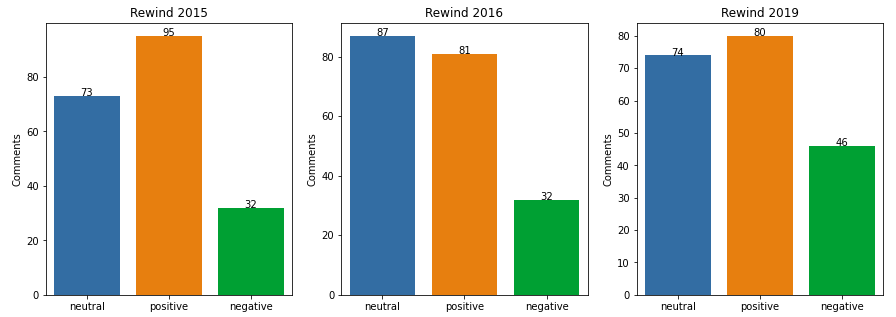
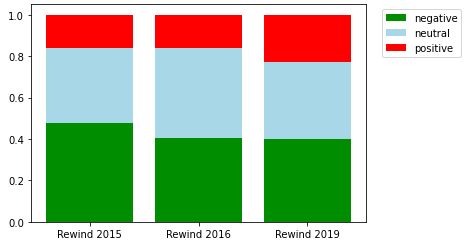
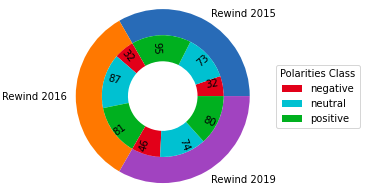
Note: Code of each graph is in mainFile.ipynb file

## 1) Analysing Views:



From the above graph, we can say that “Rewind 2016” got max views out of three videos.

## 2) Analysing Polarity:



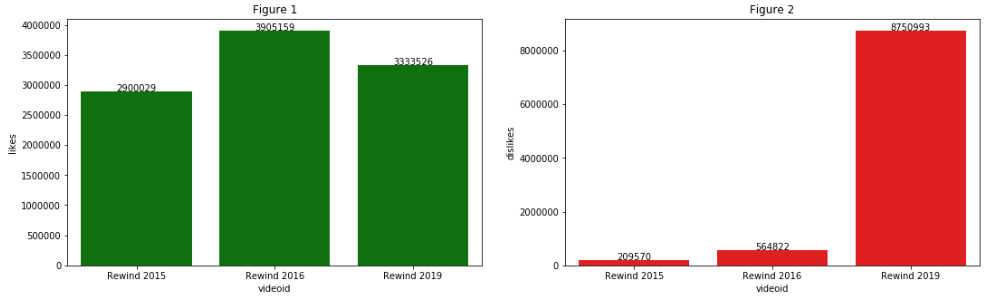
From the above graphs,

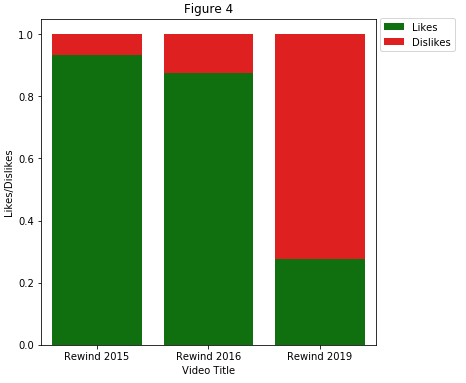
* Rewind 2015 has max positive comments and other two videos have nearly equal positive comments.
* Rewind 2016 has max neutral comments.
* Rewind 2019 has max negative comments. This video got more negativity.

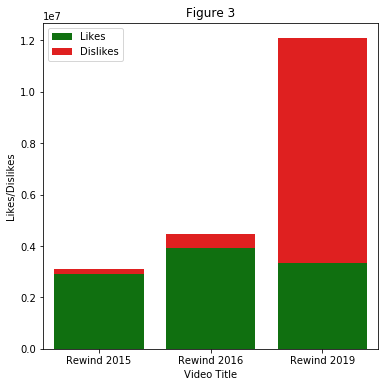
In this project, I had taken sample to 200 comments, so generalization to population that

“Rewind 2015” gets more positivity reviews and “Rewind 2019” gets more negative reviews.

## 3)Analysing Likes / Dislikes:







From above figures, we can say that

“Rewind 2016” has max likes or users enjoying this video and overall, all three videos have nearly equal likes.

“Rewind 2019” got high number of dislikes that means this video getting negativity or the content of the video is not liked by users.

## Conclusion:

Rewind 2015 and Rewind 2016 are better as compared to dislikes.

Rewind 2016 is performing good as it has good likes and positive comments.

Rewind 2019 performing bad because it got high number of dislikes and negative comments.