Viral Video Data Analysis

Project Deliverable 1

Team Members: Abhishek Ashwin Bhale (20144060)

Shivani Shashikant Achrekar (20212861)

Harshita Menon Achat (20212860)

Divya Peddapeta (20228697)

Oklahoma State University

MSIS 5600 Data Science Programming 1

Dr. Bryan Hammer

December 10, 2019

**Table of Contents**

Executive Summary…………………………………………………………………………..…1

Statement of Scope….……….……………………………………………………………….….1

Project Schedule…………………………………………………………………………………3

Data Preparation…………………………………………………………….…………………...4

* Data Access…………………………………………………..……………………….…4
* Data Consolidation………………………………………………………………………4
* Data Cleaning……………………………………………………………………………9
* Data Transformation…………………………………………………………………..…9
* Data Reduction…………………………………………………………………….….….9
* Data Dictionary…………………………………………………………….…………...10

Visualizations…………………………………………………………………………….….….11

Descriptive Statistics and Analysis……………………………………………………….….…14

Text Mining and Sentiment Analysis…………………………………………………….….….15

Appendix…………………………………………………………………………………..……23

**Executive Summary**

YouTube is one of the largest video sharing platforms where users and the general public can watch, like, share, comment and upload their videos. YouTube platform consists of two types of users: Video creators (people who have channels and upload videos to them) and Video viewers (people who watch videos, interact with videos and subscribe to channels). Many users use it for entertainment purposes, some to watch tutorials or for keeping up with their favorite artists' latest music videos and so much more. (source: <https://www.lifewire.com/youtube-101-3481847> )

As of February 2017, there were more than 400 hours of content uploaded to YouTube each minute, and one billion hours of content being watched on YouTube every day. As of August 2018, the website is ranked as the second-most popular site in the world, according to Alexa Internet.

Videos uploaded on YouTube get circulated through the viral process of Internet sharing. These videos may be informative, comic or sometimes deeply emotional. Our objective through this project is to perform analysis on these viral videos.

**Statement of Scope**

The purpose of this project is to analyze the viral YouTube videos based on the comments, likes, dislikes and the view counts given by the users. This analysis will be done on the viral videos that show up when searched for the Viral Videos on YouTube. Text mining and Sentiment analysis will be performed on the comments of the top viral videos.

**Project Objectives**

* To perform Visualization on the numeric data to explain what the data tells us.
* To perform Descriptive Statistics on views, likes and dislikes on viral videos. This will help us to analyze the opinions of the users on these videos.
* To perform Text Mining and Sentiment Analysis on the comments of the videos to determine the users’ emotions like happiness, anger, sarcasm, etc.
* To build a classification model using a Target Variable and predictor variable.
* To perform Named-Entity-Recognition analysis and generate word clouds.

The analysis which we performed on the videos published in the month of October is just a sample where we can generalize the analysis to any number of videos.

**Variables**

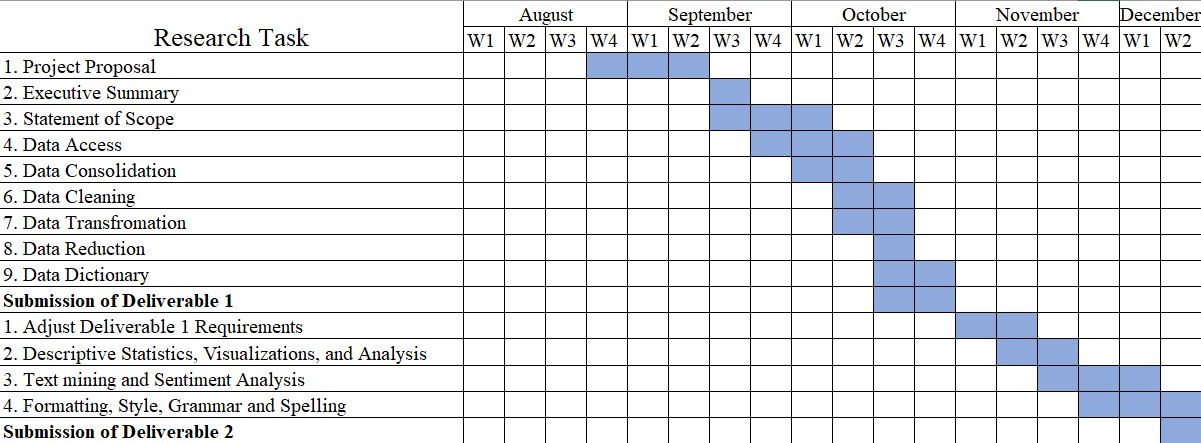
We are going to scrape the name of the video, video uploader, number of views, likes, dislikes, video id, length of the video in seconds, duration and the published date. These are the variables on which we will perform Descriptive Statistics. Name of the video, video uploader and video id are the qualitative text variables.

We will also scrape the comments from the viral videos to perform Text mining and Sentiment Analysis on them. For the ease of analysis, we have selected 5 videos from the search results.

**Project Schedule**

We have used two GANTT charts for our project schedule. The first chart was for our previous schedule and the second chart is for the revised schedule. Every week we meet on Wednesdays and plan to meet on every other following Wednesday and plan to work for 3 to 4 hours. All the tasks are done together as a group. The Project might take around 5 weeks more to get completed. In every meeting new plan is created and we work on these plans for the following week. As we had exams in the month of September there was a slight change in the schedule which lead to a delay in our schedule. The previous chart depicts that there was a Video presentation to be completed but as a revised schedule, the video presentation got cancelled. And the word cloud function was being added and that took up a while. Though there was a change in the pattern and schedule of the project, we were resilient to the changes and have gathered all the data so that the project could be completed on time.





**Data Preparation**

**Data Access**

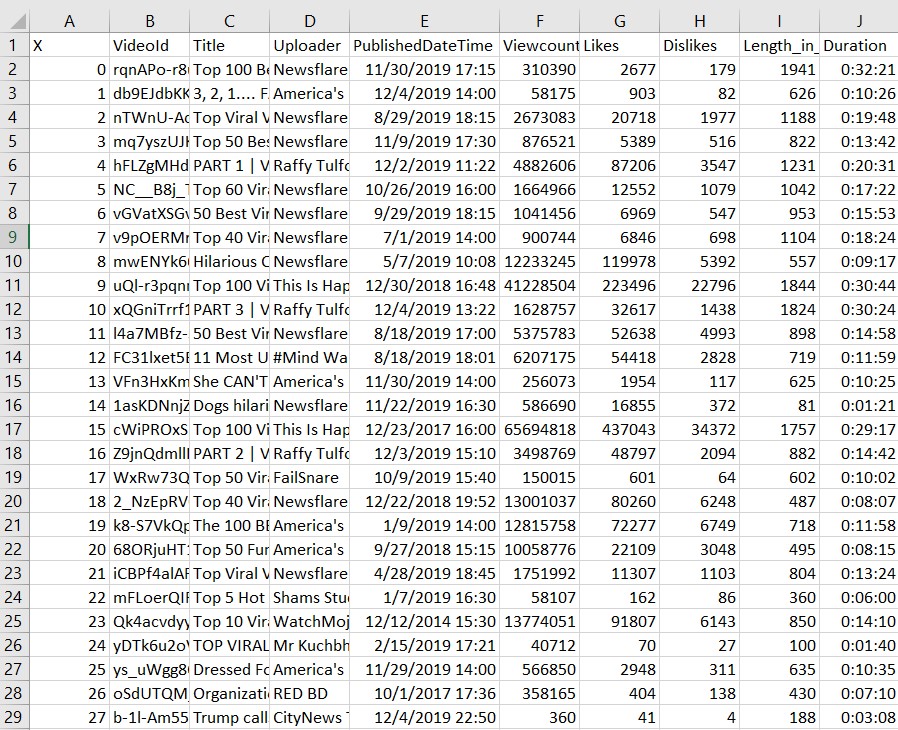
Here, we searched for the ‘Viral Videos’ on YouTube as our search query and analyze the top 100 videos that show up after the search. For accessing the data, we are using a code which will scrape the links to the top videos that were obtained with the help of Selenium and CSS selectors. Out of all the links that are obtained using Selenium, we will only be using the top 100 links for the videos. We used a library in Python named pafy which gives out the likes, dislikes, view count, name of the video and the publisher and many more. All of the data comes from YouTube itself.

Source: (<https://www.youtube.com/>, <https://pythonhosted.org/Pafy/>)

**Data Consolidation**

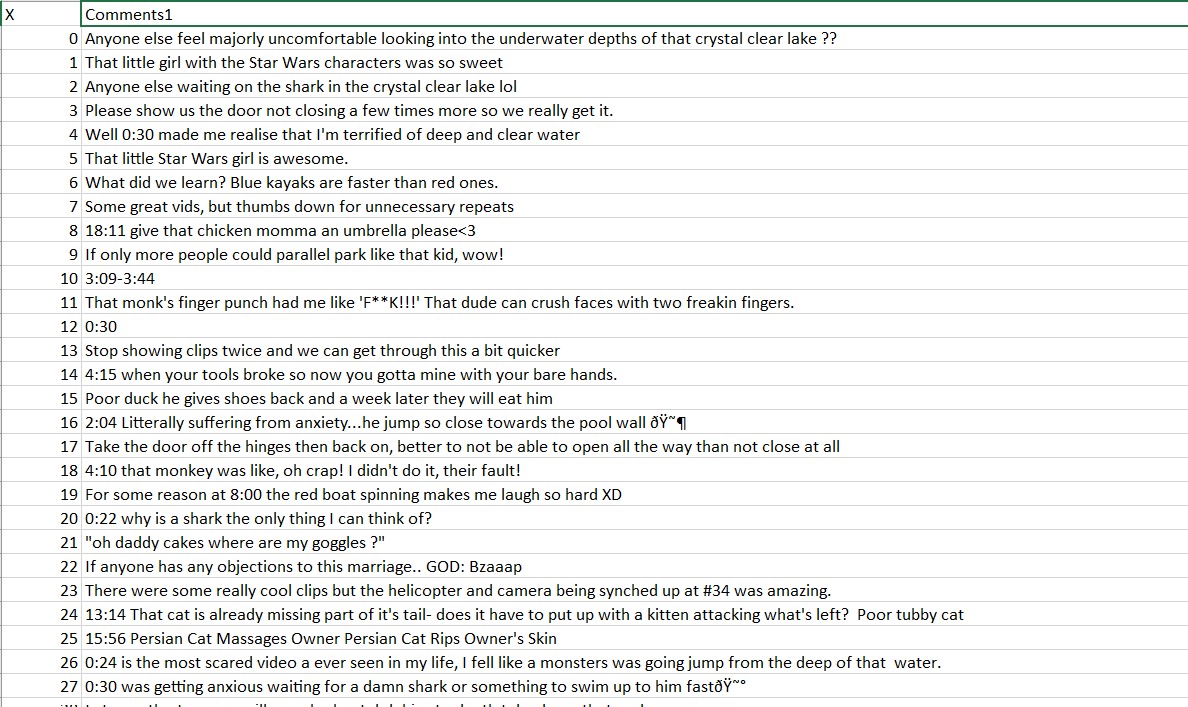
We have done the consolidation process using Python code which we have attached in the Appendix at the end of this document.

CSV file of YouTube Data.

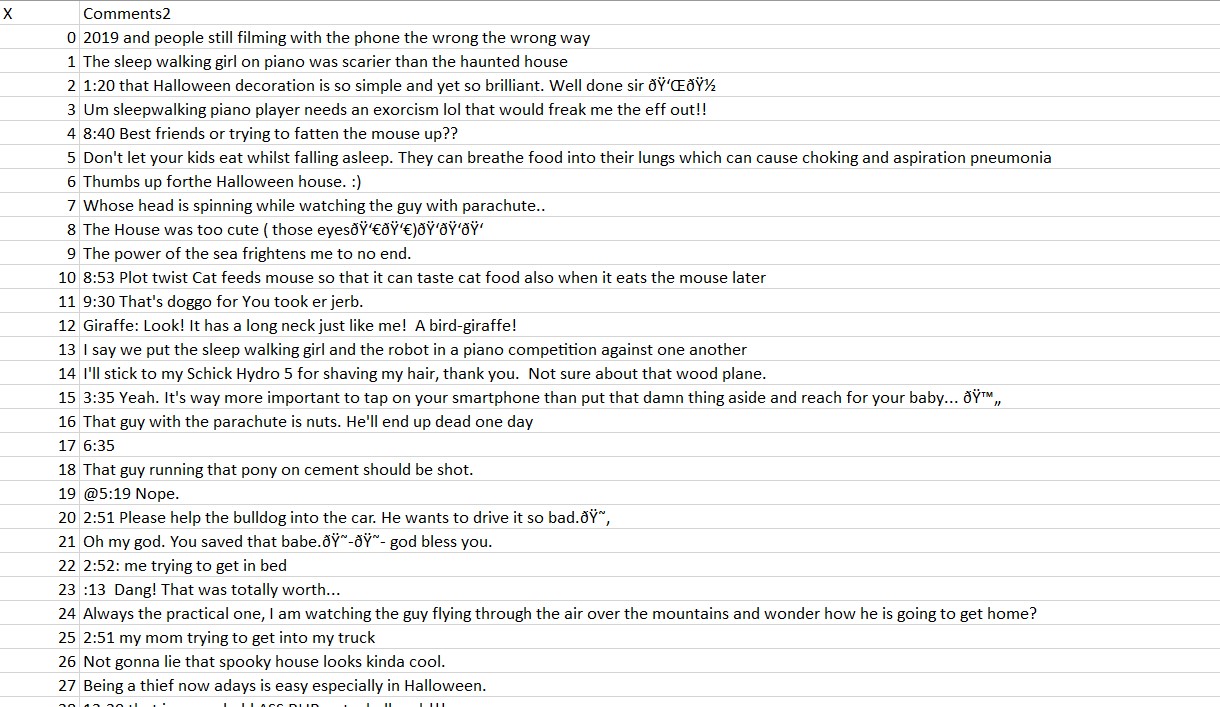
****

We have created 5 different CSV files for the comments on different videos to make the analyis simple.

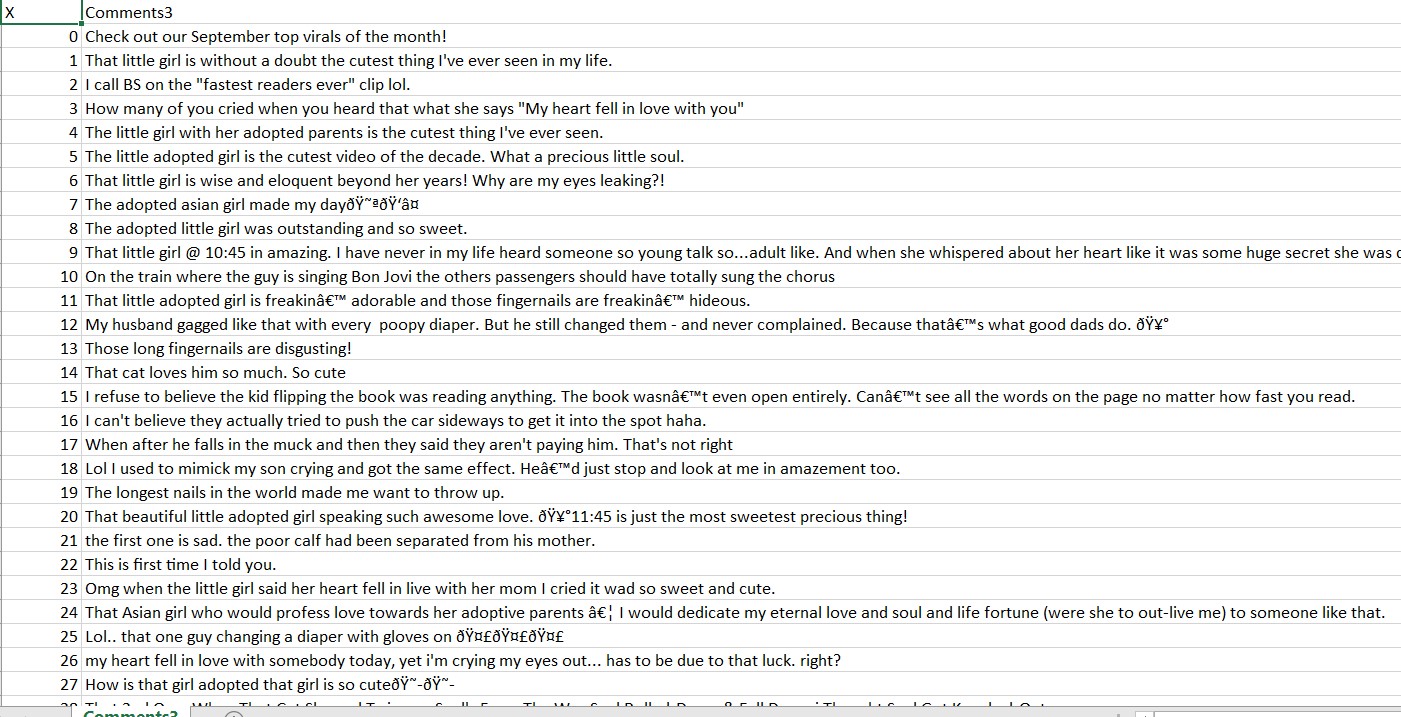
CSV file for comments on the 1st video.

****

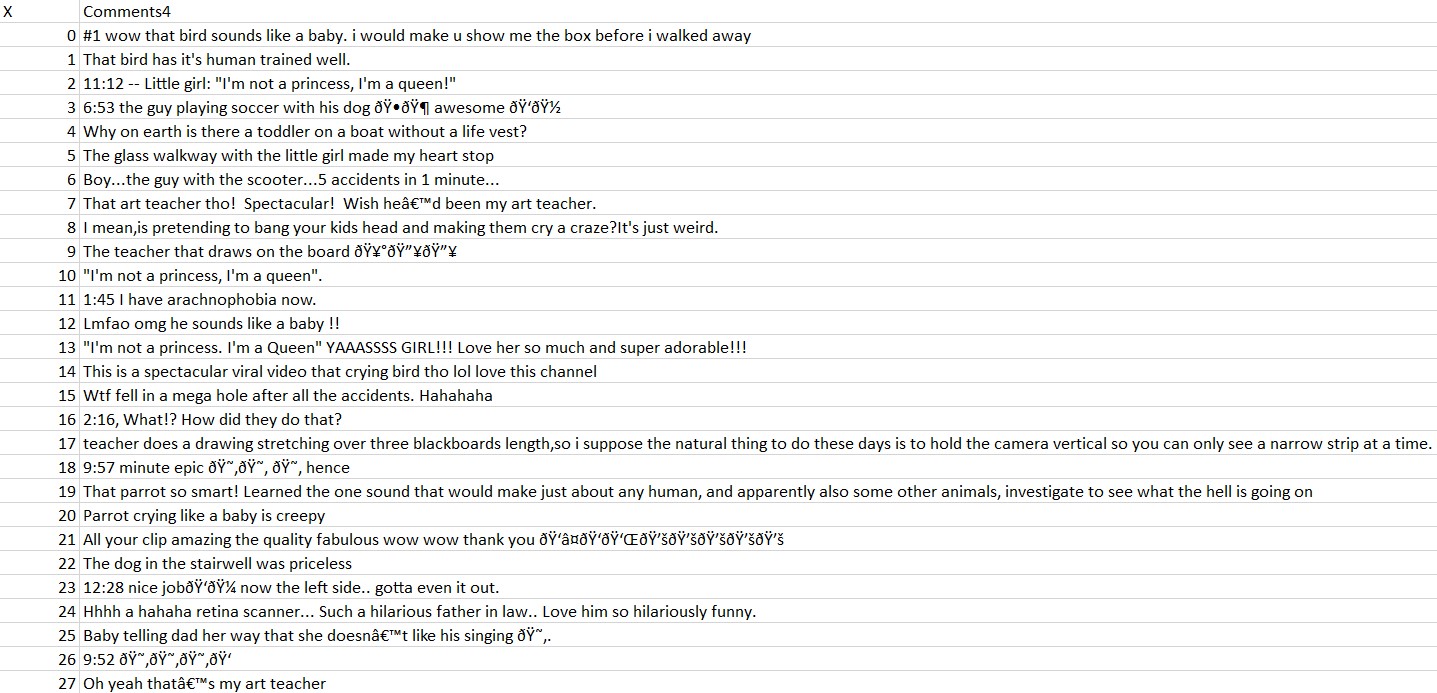
CSV file for comments on the 2nd video.

****

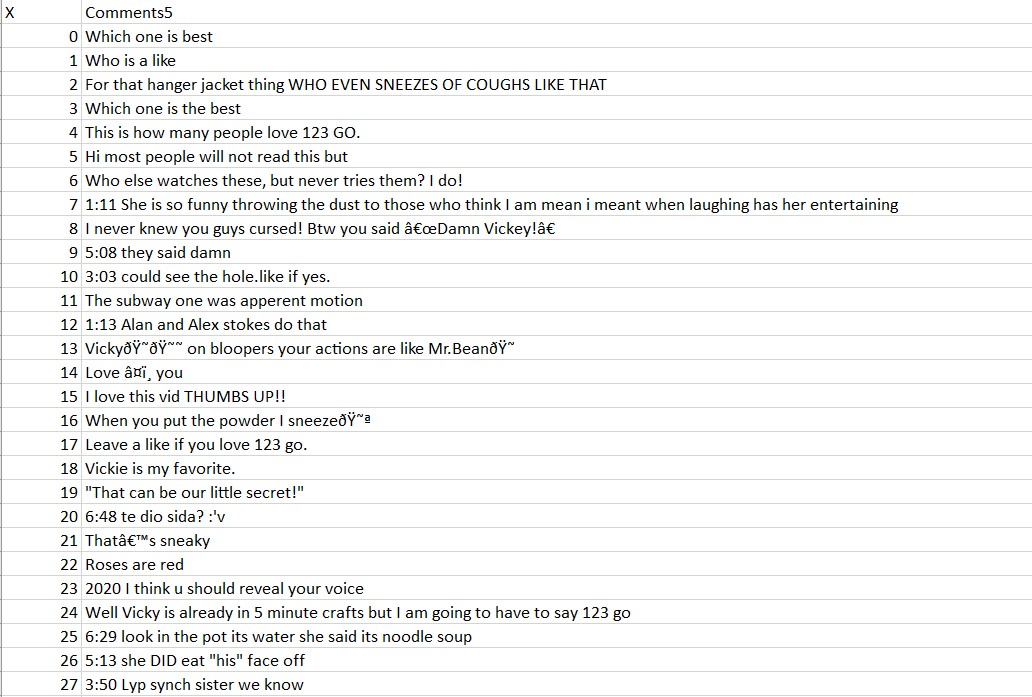
CSV file for comments on the 3rd video.

****

CSV file for comments on the 4th video.

****

CSV file for comments on the 5th video.

****

**Data Cleaning**

There are few videos that have missing data for Likes and Dislikes columns. So, as to handle this missing data, we did Listwise deletion to perform descriptive analytics and visualization. The comments consisted of some unwanted data like the timings when the comments were posted and emoticons also appeared in the comments’ section. We used regular expressions to remove these timings and emoticons. In this way, we cleaned the comments and made them useful for text mining and sentiment analysis. There is no more specific data that needs to be cleaned or any variable that needs to be dropped.

**Data Transformation**

For the variable Viewcount, the data that is obtained is in lakhs, to plot this data we are normalizing it by dividing it by 1000. We are doing this so that the visuals of the plot can be easily analyzed.

We are transforming the comments that were obtained for the viral videos by converting them to strings. This will help us in performing Text Mining and Sentiment Analysis.

**Data Reduction**

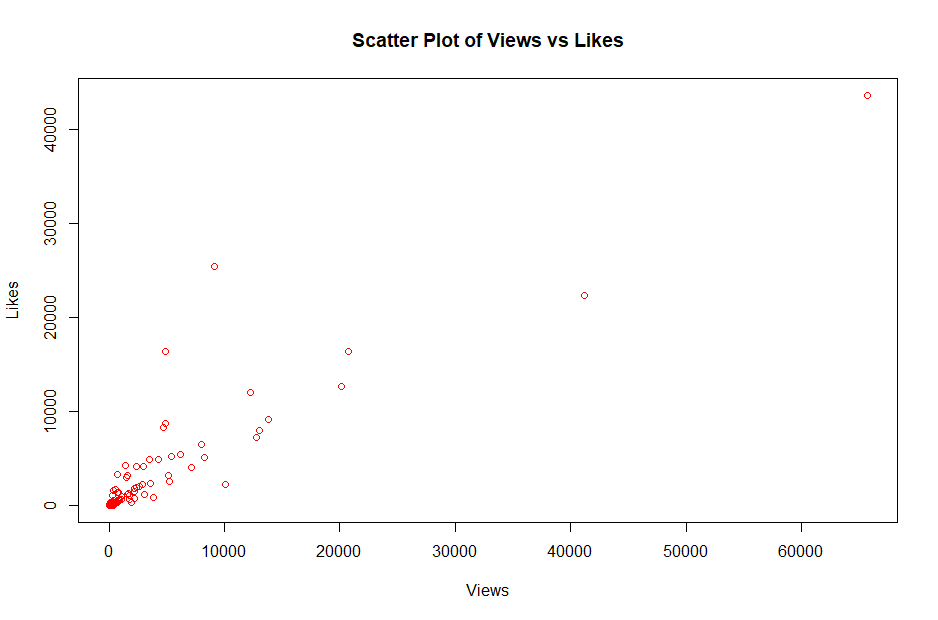
As of now, we are pulling data from only 100 YouTube videos. And the number of variables we have are 9. So, this is the exact data that we will need for our analysis. If needed, we can increase the number of YouTube videos that we will be pulling the data from. For now, we won’t be performing any Data Reduction on the data that we have scraped.

**Data Dictionary**

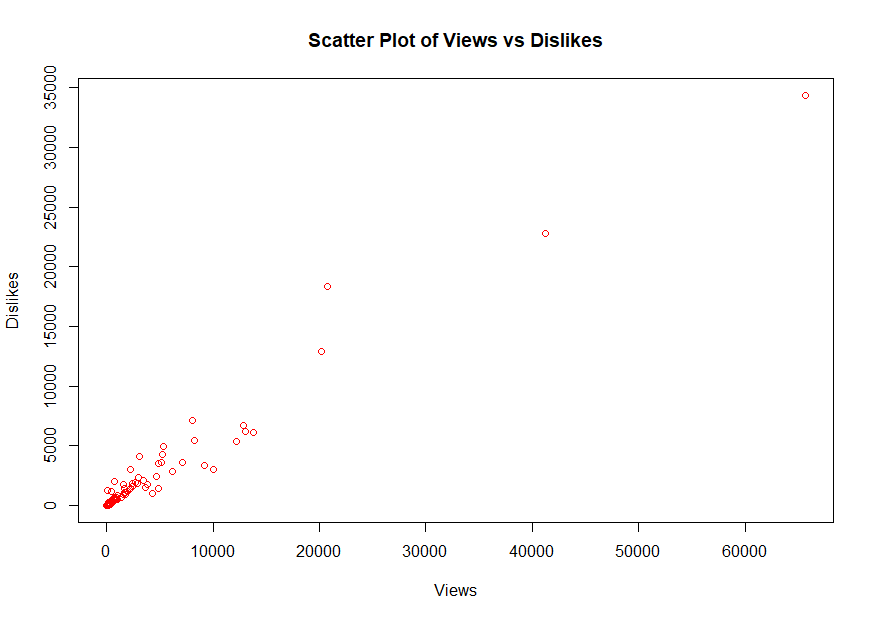
|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute Name** | **Description** | **Data Type** | **Source** |
| VideoId | It is the alphanumeric unique id of the individual videos | String | https://www.youtube.com/ |
| Title | Name of the video | Char(30) | https://www.youtube.com/ |
| Uploader | Name of the channel that published the video | Char(30) | https://www.youtube.com/ |
| PublishedDateTime | The time and date the video was uploaded | Date time format as String | https://www.youtube.com/ |
| Viewcount | The number of users who viewed the video | Integer | https://www.youtube.com/ |
| Likes | Users who liked the video | Integer | https://www.youtube.com/ |
| Dislikes | Users who disliked the video | Integer | https://www.youtube.com/ |
| Length\_in\_sec | The length of the video in seconds | Integer | https://www.youtube.com/ |
| Duration | The duration of video in HH:MM:SS format | String | https://www.youtube.com/ |
| Comments | Feedback given on videos | Char (30) | https://www.youtube.com/ |

**Visualizations**

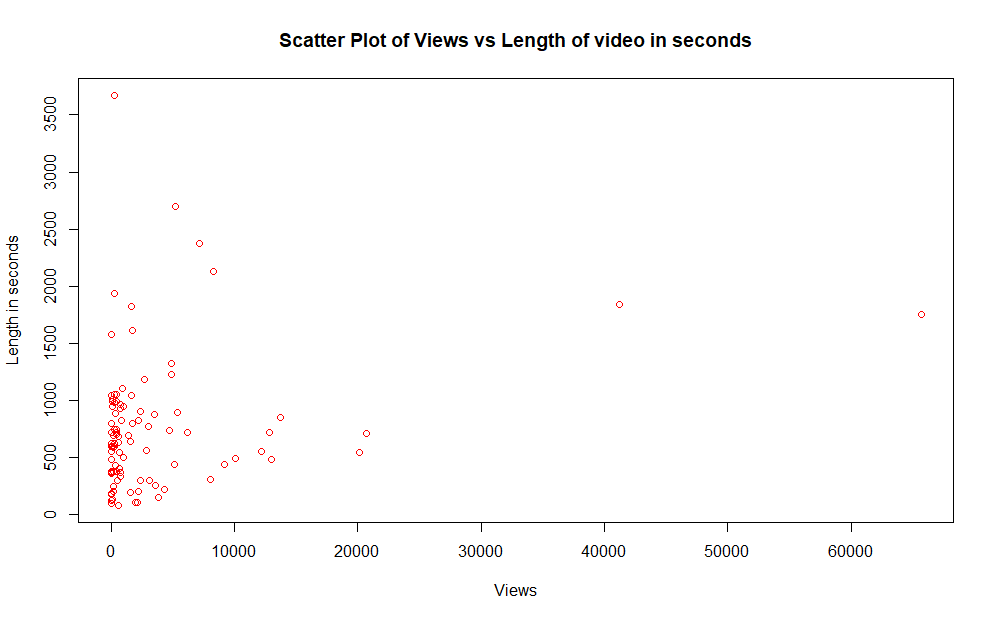
Data visualizations help us depict and understand data using graphs and various kinds of plots. The pictorial representation of data helps us to interpret data more effectively. A scatterplot shows the relationship between two continuous variables. In our project, the proposed target variables are number of Likes, Dislikes and Length\_in\_sec. The predictor variable is the Viewcount.

****

The graph above depicts the relationship between Views and Likes. The scatterplot depicts a strong positive linear relationship between Views of videos and Likes of videos. But there appears to be a couple of outliers in the data. Most of the videos’ views range from 10 views to 6,000,000 views. The views are in millions, so we have divided them by 1000. The scatterplot also depicts potential outliers. The Likes on the video are in the range of 10,000 so for better visualization we have divided it by 10. A few videos have more than 6,000,000 views and by here we eventually conclude that most of the times as the views increase the Likes also increase i.e., if more people tend to watch the video, they would tend to have an opinion about the video.



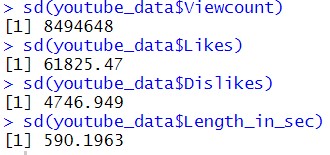
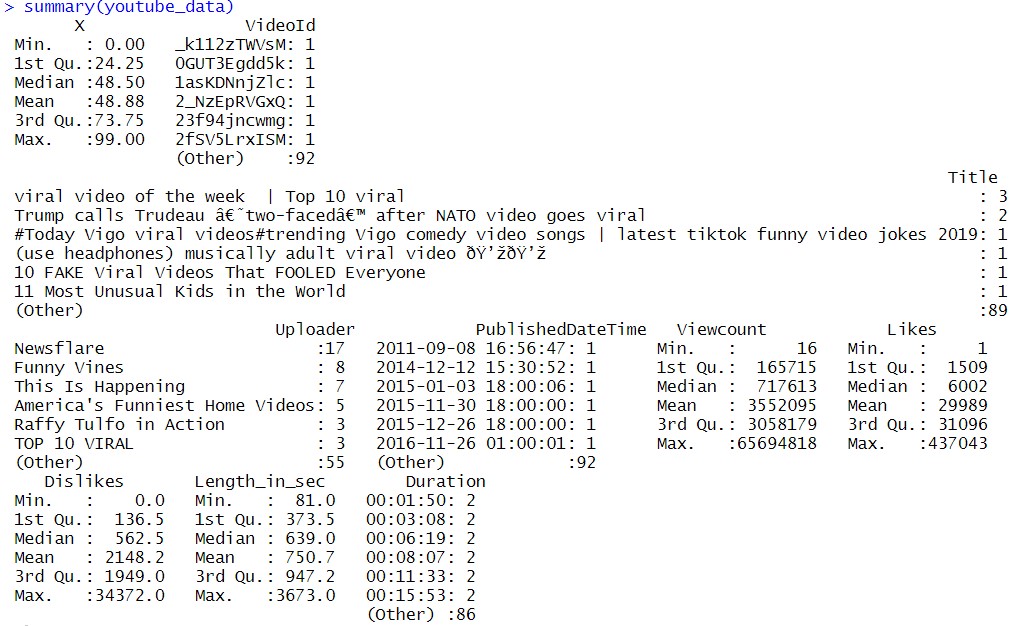
The graph above depicts the relationship between Views and Dislikes. The scatterplot depicts a strong positive linear relationship between Views of videos and Dislikes of videos. But there appears to be a couple of outliers in the data. Most of the views of the videos range from 10 views to a 6 million views. The view values are divided by 1000. The scatterplot also depicts potential outliers. A few videos have more than 20,000 views and by here we eventually conclude that most of the times as the views increase there is a chance of people not liking the video and hence the dislikes increase. i.e., if more people tend to watch the video, they would tend to have an opinion about the video.



From the graph above it can be interpreted that, the duration of the video does not affect the views. Irrespective of the duration in minutes the number of views does not change. The plot is a vertical line depicting that the views are constant. There are also a few potential outliers in the data where the views for an 80-minute video is more than 60,000 according to the plot.

In our modelling technique, we are selecting Viewcount as our predictor variable and Likes, Dislikes and Length in seconds as our dependent variable.

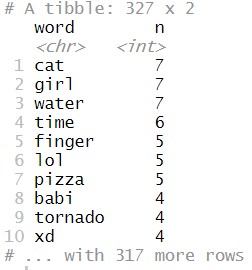
**Descriptive Statistics and Analysis**

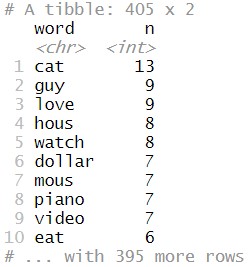
****

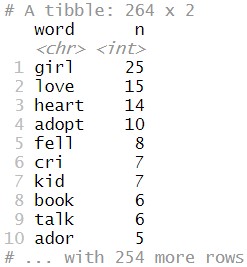
The minimum view counts on a video is 16 while the maximum is 65694818. On an average, the videos have 3552095 views. There is a video with a single like and one has 437043 with maximum likes. The mean value for number of likes is 29989 and for number of dislikes is 2148. The standard deviation for Dislikes is 4746.949. The shortest YouTube video is of 81 seconds where as the longest You tube video is of 3673 seconds.

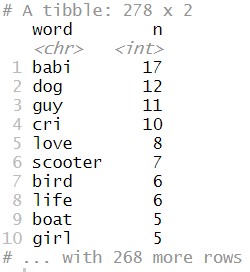
**Text Mining and Sentiment Analysis**

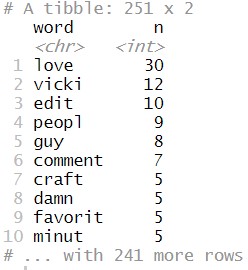
Following are the outputs of the top ten words generated after performing text mining on the comments of the selected 5 videos.

****

****

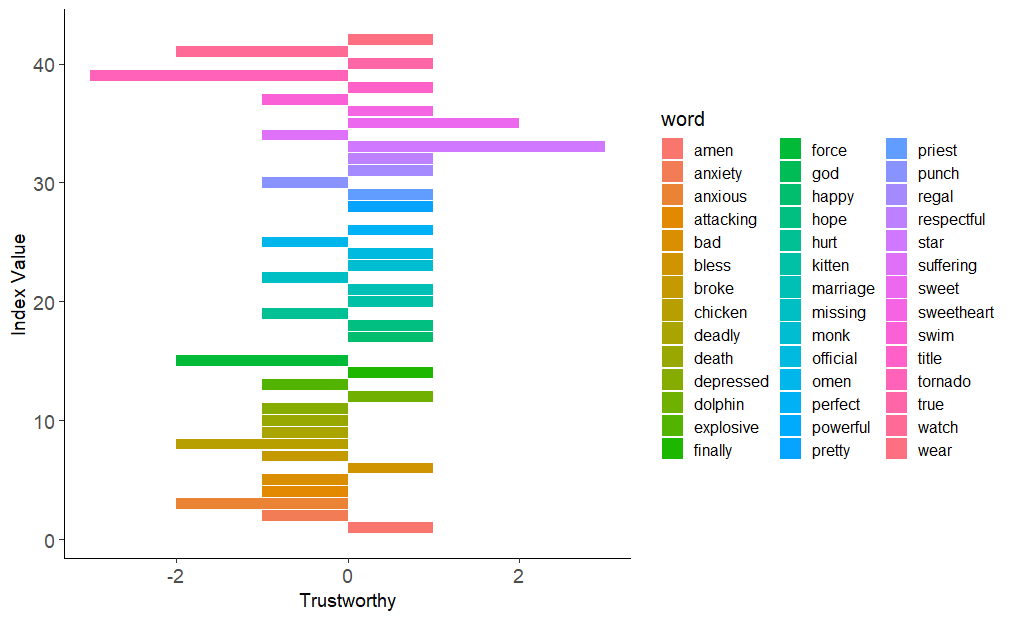
****

****

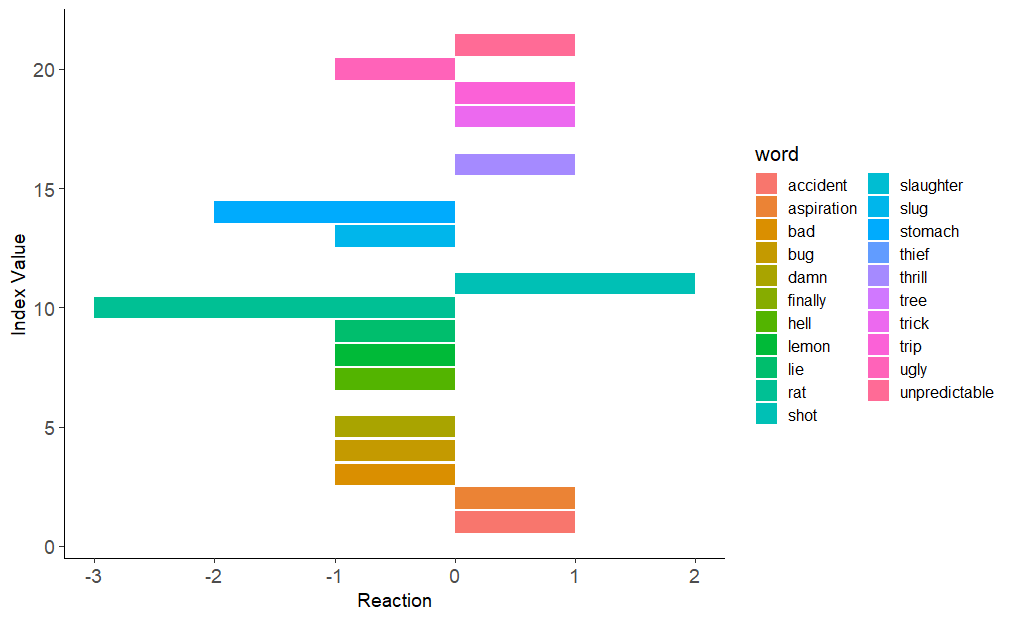
****

Following are the outputs for the Sentiment Analysis that was performed on the comments of the YouTube data.

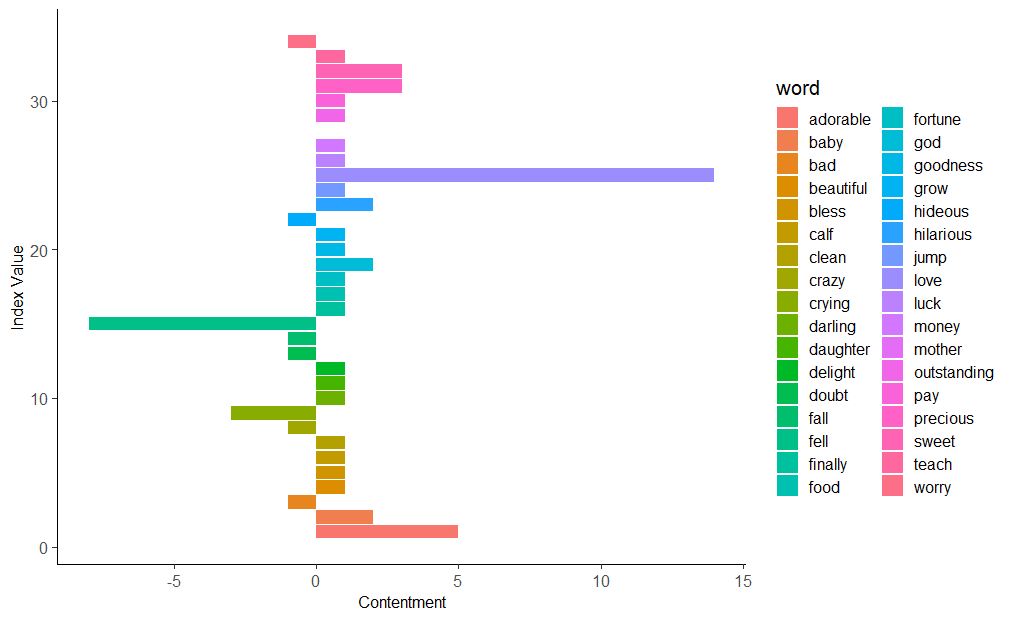
For the comments on the first video, we have taken trust and fear as the two sentiments for assessing the feeling of Trustworthiness.

****

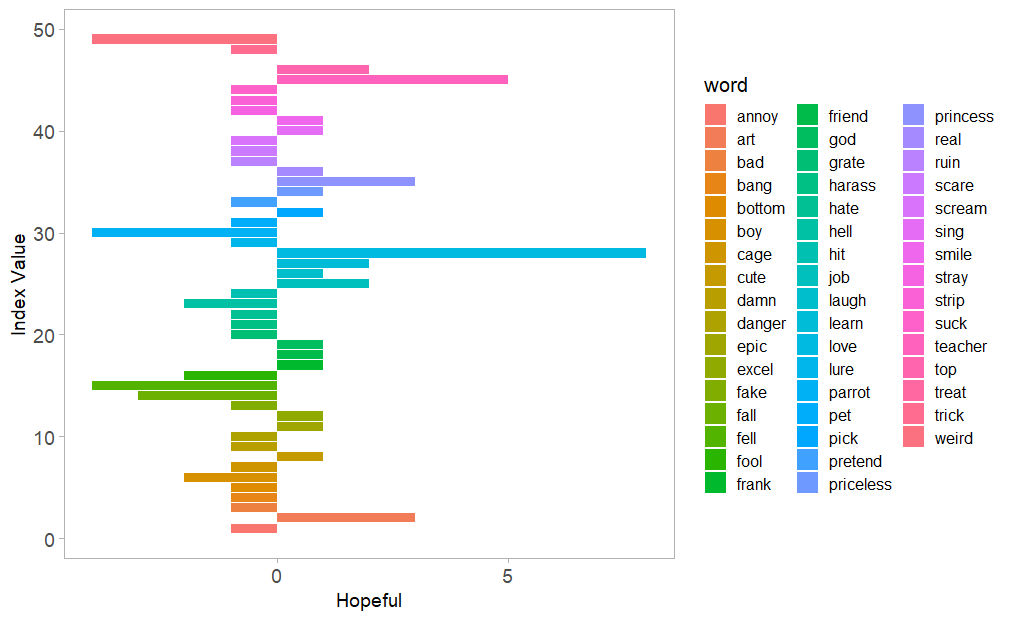
Sentiment analysis on the second video is performed using the sentiments surprise and disgust giving us the kind of reaction that the users have.

****

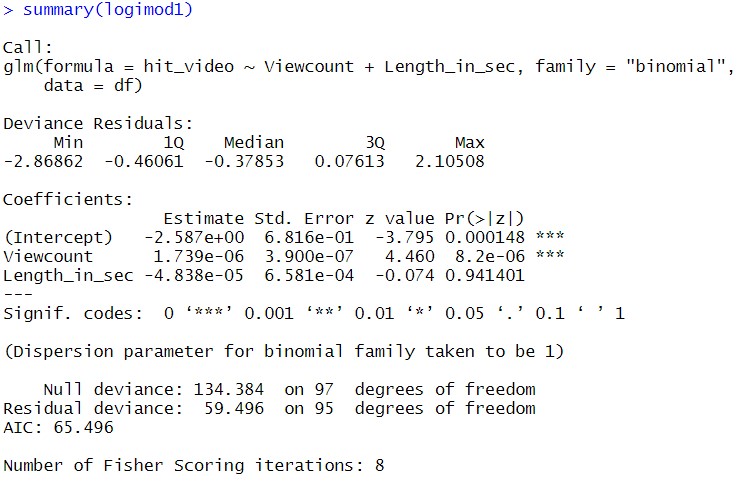
For the third video, joy and sadness are the two sentiments which we have used to assess the feeling of Contentment.

****

Similarly, for the comments on the fourth video, we have done sentiment analysis using positive and negative as the two sentiments to analyze the feeling whether the users are hopeful or not.

****

**Classification Model**

**Word Cloud for Verbs**



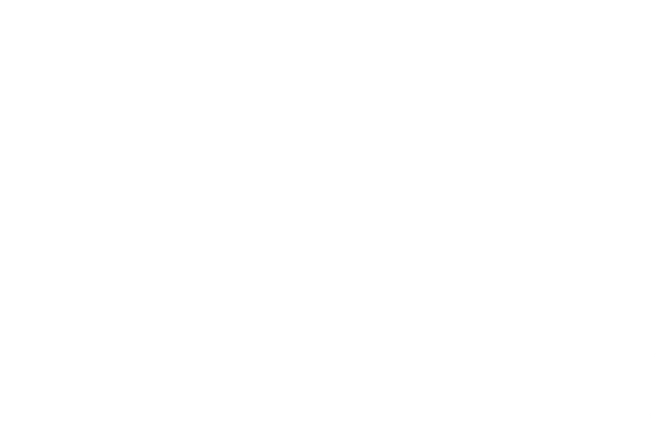


In this word cloud, love, looking, feel, realise, closing, waiting are the entities that stand out to us.

The entity ‘love’ has occurred in the top 10 words of the 4 videos. So, it’s the word that catches the eye.

**Word Cloud for Adjectives**





In this word cloud, little, many, lol, uncomfortable, deadly and terrified are the entities that stand out to me.

The entity ‘lol’ has appeared in the top 10 word of the first video and we can see that it stands out in the word cloud.

**Appendix**

**Data Consolidation**

## Putting all the data into the dataframe#############################################################

linku = []

id = []

name = []

count = []

like = []

dislike = []

lent = []

dur = []

aut = []

pub = []

for i in range(long):

url = links[i]

v = pafy.new(url, gdata = True)

video\_id = v.videoid

id.append(video\_id)

v\_title = v.title

name.append(v\_title)

views = v.viewcount

count.append(views)

likes = v.likes

like.append(likes)

dislikes = v.dislikes

dislike.append(dislikes)

v\_length = v.length

lent.append(v\_length)

v\_duration = v.duration

dur.append(v\_duration)

v\_author = v.author

aut.append(v\_author)

v\_published = v.published

pub.append(v\_published)

## Creating the DataFrame##############################################################################

list = {'VideoId':id, 'Title':name, 'Uploader':aut, 'PublishedDateTime':pub, 'Viewcount':count, 'Likes':like, 'Dislikes': dislike, 'Length\_in\_sec': lent, 'Duration': dur}

youtube\_data = pd.DataFrame(list)

youtube\_data.to\_csv(r'C:\Users\abhis\OneDrive\Desktop\Assignments\Project\youtubedata.csv')

# Consolidation code for comments of the first video

come = []

for y in comments0:

come.append(y.text)

comment\_data1 = pd.DataFrame(come)

comment\_data1.to\_csv(r'C:\Users\abhis\OneDrive\Desktop\Assignments\Project\youtubecommentdata1.csv')

Similarly, we have consolidated the remaining 4 videos.