# ANALYSIS ON YOUTUBE TRENDING VIDEOS

A Project Report

Submitted in the partial fulfillment of the requirements for the award of the degree of

# Bachelor of Technology in

Department of Computer Science and Engineering

By

Nihal Agarwal 2010030413

Jaideep Sharma 2010030374

T Venkata Sai Sathvik 2010030361

Shaik Abdul Shaan 2010030153

under the supervision of

# Mr. Chanda Raj Kumar Rao, MTech

# Assistant Professor, CSE



Department of Computer Science and Engineering

K L University Hyderabad,

Aziz Nagar, Moinabad Road, Hyderabad – 500 075, Telangana, India.

March, 2022

**Declaration**

The Project Report entitled “ANALYSIS ON YOUTUBE TRENDING VIDEOS” is a record of bonafide work of **Nihal Agarwal (2010030413), Jaideep Sharma (2010030374), T Venkata Sai Sathvik (2010030361), Shaik Abdul Shaan (2010030153)**, submitted in partial fulfillment for the award of B. Tech in the Department of Computer Science and Engineering to the K L University, Hyderabad. The results embodied in this report have not been copied from any other Departments/ University/ Institute.

|  |  |
| --- | --- |
| NIHAL AGARWAL | 2010030413 |
| T VENKATA SAI SATHVIK | 2010030361 |
| JAIDEEP SHARMA | 2010030374 |
| SHAIK ABDUL SHAAN | 2010030153 |

*iii.*

**Certificate**

This is to certify that the Project Report entitled “ANALYSIS ON YOUTUBE TRENDING VIDEOS” is being submitted by Nihal Agarwal (2010030413), Jaideep Sharma (2010030374), T Venkata Sai Sathvik (2010030361), Shaik Abdul Shaan (2010030153), submitted in partial fulfillment for the award of B. Tech in Computer Science Engineering to the K L University, Hyderabad is a record of bonafide work carried out under our guidance and supervision.

The results embodied in this report have not been copied from any other departments/ University/Institute.

## Signature of the Supervisor

Mr. Chanda Raj Kumar Rao, MTech

Assistant Professor

## Signature of the HOD Signature of the External Examine

*iv.*

**ACKNOWLEDGEMENT**

First, I would like to thank our beloved Founder and chairman, of Koneru Lakshmaiah University for giving us this opportunity to complete our project within the University in the guidance of our faculty.

I also would like all the fellow students that completed their social internship in KL University with their patience and openness, they created an enjoyable and challenging working environment.

I am highly indebted to our Principal, **Dr. L Koteswara Rao** who has been constantly pushing us and providing us opportunities to accomplish this internship.

I would like to thank my Head of department **Dr. Chiranjeevi Manike** for his exemplary guidance, monitoring and constant support through the social internship. We thank **Mr. Chanda Raj Kumar Rao, Assistant Professor** of our department who has supported throughout this project holding a position of supervisor.

I am extremely grateful to all the teaching and non-teaching staff of our department without whom we won’t have made this project a reality. We would like to extend our sincere thanks to our parents who supported us making this project a grand success.

*v.*

**ABSTRACT**

In this modern era, many users are addicted to YouTube which is available both as application and through website. In detail, YouTube is a content sharing platform through the means of video in any format which allows users to watch or share any specific video with anyone. Usually there are three genres of content, that goes trending on YouTube, ranging from educational, entertainment and informative. Generally, when it comes to the entertainment genre there are users who would like to see gaming, music, short videos and many more. Students also frequently use YouTube for their study purposes to establish a good secure knowledge. YouTube is not limited to a particular age group, caste, creed or country, anyone can use it throughout the world. We provide analysis through data science to determine videos which are trending based on number of likes, views, and comments. The analysis is done using user features such as Views, Comments, Likes, and Dislikes. Analysis can be performed using algorithms like Linear Regression, classification and other Machine learning models and python libraries like pandas, matplot library to classify the YouTube Data and obtain useful information.

**keywords**: YouTube, Trending Analysis, Visualization, Machine Learning, Pandas, Big Data.

*vi.*

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| S. No. | TITLE | Page No. |
| 1 | Introduction | 1 |
| 2 | Literature Survey | 2 |
| 3 | Hardware and Software Requirements | 4 |
| 4 | Functional and Non-Functional Requirements | 5 |
| 5 | Proposed System | 8 |
| 6 | Implementation | 9 |
| 7 | Result Discussion | 25 |
| 8 | Conclusion and Future Work | 26 |
| 9 | specific Contribution | 27 |
| 10 | References | 28 |

# **List of Tables**

|  |  |  |
| --- | --- | --- |
| Fig. No. | Figure Title | Page No. |
| 2.1 | Literature Survey | 2,3 |

# **List of Figures**

|  |  |  |
| --- | --- | --- |
| Fig. No. | Figure Title | Page No. |
| 7.1 | Bokeh Output-1 | 25 |
| 7.2 | Bokeh Output-2 | 25 |

*vii.*

# **1. INTRODUCTION**

1.1 **PROBLEM STATEMENT**

YouTube is a widely used and famous online video platform in the world today. YouTube has a list of trending videos that is updated constantly. Analysing these trending videos can give content creators greater perspective and

knowledge for increasing their popularity and brand of their channels. Companies and businesses using social media and digital platform can also use this analysis to boost their growth in business by publishing videos or sponsoring

appropriate channels at right time.

1.

# **2. LITERATURE SURVEY**

2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AUTHOR** | **TITLE** | **PUBLISHED SOURCE** | **METHODS** | **FINDINGS** |
| [Johanes Fernandes Andry](https://ieeexplore.ieee.org/author/37088964712), [Stefan Azriel Reynaldo](https://ieeexplore.ieee.org/author/37089199629), [Kevin Christianto](https://ieeexplore.ieee.org/author/37089202437), [Francka Sakti Lee](https://ieeexplore.ieee.org/author/37089203555), [Julia Loisa](https://ieeexplore.ieee.org/author/37089202538),  [Aman Budi Manduro](https://ieeexplore.ieee.org/author/37089204146). | Algorithm of Trending Videos on YouTube Analysis using Classification, Association and Clustering. | [2021 International Conference on Data and Software Engineering (ICoDSE)](https://ieeexplore.ieee.org/xpl/conhome/9648407/proceeding), 3-4 Nov. 2021, DOI:  [10.1109/ICoDSE53690.2021.9648486](https://doi.org/10.1109/ICoDSE53690.2021.9648486). | Authors has been carried out with existing data mining software to meet the results, find out that rating given by YouTube for getting to the trending list are “views”, “likes”, dislikes”, “comments”. | Finding the highest positive rated video by analysing the different factors of the video such as views, likes, comments and dislikes by the data mining, k means algorithm, classification and clustering. |
| Swati Gayakwad, Rajas Patankar, Dashrath Mane. | Analysis on YouTube Trending Videos | [International Research Journal of Engineering and Technology (IRJET), Volume: 07 Issue: 08 | Aug 2020 .](https://www.irjet.net/archives/V7/i8/IRJET-V7I8732.pdf) e-ISSN: 2395-0056, p-ISSN: 2395-0072. | Analysis for this paper is done by using 3 different ways of analysis. They analyzed the basic statistics of YouTube trending videos by downloading data through YouTube API, then differentiated trending and non trending. | They used three algorithms to analyse the trending videos and non trending videos such as classification, regression, and clustering and similarity matching. Even different factors were analysed. They analysed the best time to upload videos. |
| Sana Khanam, Safdar Tanweer, Syed Sibtain Khalid. | Youtube Trending Videos: Boosting Machine Learning Results Using Exploratory Data Analysis. | The Computer Journal, bxab142, <https://doi.org/10.1093/comjnl/bxab142>, Published:   20 October 2021. | We present our analysis by measuring, mining, analyzing and comparing key aspects of time-series YouTube data with respect to its view and audience response statistics from 40 000 trending YouTube videos collected over 205 days. | We have performed an exploratory data analysis (EDA) on all its aspect to get data insights and used statistics to find similarities between them to understand viewing pattern of different video categories.  We also compare and observe the variation of activity over time with the nature of the event that affects the quality of our analysis. |
| [Iman Barjasteh](https://www.researchgate.net/profile/Iman-Barjasteh), [Ying Liu](https://www.researchgate.net/profile/Ying-Liu-295), [Hayder Radha](https://www.researchgate.net/profile/Hayder-Radha). | Trending Videos: Measurement and Analysis. | Cornell University, <https://arxiv.org/abs/1409.7733>. September 2014. | The study is based on collecting and monitoring high-resolution time-series of the viewership and related statistics of more than 8,000 YouTube videos over an aggregate period of nine months. | They employed Granger Causality (GC) with significance testing to conduct this analysis. Unlike traditional correlation measures, our directional-relationship analysis provides a deeper insight onto the viewership pattern of different categories of trending videos. |
| Aakash Ashok Niture Supervisor: Mr. Pierpaolo Dondio | Predictive analysis of YouTube trending videos using Machine Learning | <https://esource.dbs.ie/bitstream/handle/10788/4260/msc_niture_aa_2021.pdf?sequence=1&isAllowed=y>. : 11/01/2021 | Since trending video statistics consists of number of Views, Likes, Dislikes and Comment counts, the research performed Linear regression model of Machine Learning for predictive analysis of number of views for YouTube trending videos  Achieving maximum accuracy of 62.53%. | In addition, the study performs a comparative analysis of a number of classification models namely Random Forest, SVM, Decision Tree, Logistic Regression and Gaussian Naïve Bayes, to determine which model suits better for predicting the number of days a video will take to get trending from its upload time and the number of days a video will trend on the trending list. |
| Muhammad Nihal Hussain, Serpil Tokdemir, Samer Al-khateeb, Kiran Kumar Bandeli, and Nitin Agarwal | Understanding Digital Ethnography: Socio-computational Analysis of Trending YouTube Videos | <http://sbp-brims.org/2018/proceedings/papers/latebreaking_papers/LB_14.pdf>. | The original dataset from Kaggle has the following attributes: URL of the video, video ID, title of the video, title of the channel that published the video, category in which the video belongs to, number of views, number of likes, number of dislikes, number of comments the video received at the time data was collected, and the date the video was trending | In the second phase, we enhanced the dataset obtained from Kaggle by adding the description of the video, date the channel was created, and the number of subscribers of the channel, using YouTube API. It is a common practice among prominent YouTubers to associate their various social media accounts with their YouTube channel. |

**Table 2.1. Literature Survey**

3.

# **3 HARDWARE AND SOFTWARE REQUIREMENTS**

**3.1 HARDWARE REQUIREMENTS:**

1. Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz 2.50 GHz

2. 8.00 GB RAM or higher.

3. 64 Bit operating system or higher.

4. 1 TB Hard free drive space.

**3.2 SOFTWARE REQUIREMENTS:**

1. Operating System: Windows 10

2. Web Browser: Brave/ Google Chrome

3. Python programming language

4. Jupyter notebooks for python

4.

# **4. FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS**

4.1 **ARTIFICIAL INTELLIGENCE**

Artificial Intelligence (AI) is a branch of computer science that deals with problem-solving by the aid of symbolic programming. It has greatly evolved into a science of problem-solving with huge applications that deals in business, health care, and engineering. Expert systems are one of the major applications (pivotal) of AI. The major advantage of AI is that it reduces the time that is needed for drug development, in turn, it reduces the cost that are associated with drug development. It enhances the return on the investment and may decrease in cost for the end user. The year 1956 is usually considered to be when AI was born, as it was in 1956 that Dartmouth college had organized the famous conference.

4.2 **DATA VISUALIZATION**

Data visualization is the presentation of data in a pictorial or graphical format. It enables decision makers to see analytics presented visually, so they can grasp difficult concepts or identify new patterns. With interactive visualization, you can take the concept a step further by using technology to drill down into charts and graphs for more detail, interactively changing what data you see and how it’s processed. Data visualization can also; Identify areas that need attention or improvement, clarify which factors influence customer behaviour.

4.3 **PANDAS**

Pandas is a Python library created by Wes McKinney, who built pandas to help work with datasets in Python for his work in finance at his place of employment. Pandas stands for ‘panel data’. Pandas is an open-source library, which means that anyone can view its source code and make suggestions using pull requests.

4.4 **JUPYTER NOTEBOOKS**

Jupyter Notebook is a client-server application used for running notebook documents in the browser. Notebook documents are documents able to contain both code and rich text elements such as paragraphs, equations, and so on.

5.

4.5 **WARNINGS LIBRARY**

Warning messages are typically issued in situations where it is useful to alert the user of some condition in a program, where that condition (normally) doesn't warrant raising an exception and terminating the program.

4.6 **NUMPY LIBRARY**

NumPy is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open-source project and you can use it freely. NumPy stands for Numerical Python.

4.7 **MATPLOTLIB LIBRARY**

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible.

4.8 **SEABORN LIBRARY**

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. For a brief introduction to the ideas behind the library, you can read the introductory notes or the paper.

4.9 **BOKEH LIBRARY**

Bokeh is a Python library for creating interactive visualizations for modern web browsers. It helps you build beautiful graphics, ranging from simple plots to complex dashboards with streaming datasets. With Bokeh, you can create JavaScript-powered visualizations without writing any JavaScript yourself.

4.10 **OUTLIERS**

A statistical outlier is any datapoint in a dataset that is beyond a pre-defined distribution range, usually representing a measurement error or abnormal data that should not be included.

6.

4.11 **PROJECT AREA**

The proposed solution is to analyze the trending videos on YouTube taking some external factors in consideration such as likes count, views, and comments. The execution of the project shows us the analysis of videos that are trending on YouTube using data visualization to understand the big data. It visualizes based on various factors and gives the user whether it might be a viewer or content creator. The analysis and data visualization is done using python libraries in jupyter notebooks.

7.

# **5. PROPOSED SYSTEM**

Yt\_Trending

{

importing the libraries used for analysis and visualisation;

eg(pandas,warnings,numpy,matplotlib,seaborn,bokeh)

we can directly visualise the data but doing so we can get much noise and inaccuracy in the data points

we cannot get the right analysis and minimum and maximum number of likes and views needed for the trending data;

Detecting the outliers:

{

Using the quantile method to detect the outliers;

to use the quantile method first we need to convert the views and likes which are in e^7-e^8 numericals in log format into 10^5 maximum;

by using the quantile/percentile method, get the maximum threshold for the video to be on trending;

by using the quantile/percentile method, get the minimum threshold of likes foe the video to be on trending;

now using the maximum and minimum threshold, extract the final dataset by filtering it out and storing it into other datframe using pandas;

}

Bokeh visualisation for YT\_Trending

{

now convert the likes and views in the form of log format using the numpy library for python;

bokeh visualisation uses the glyph data points where it is plotted for the analysis based on the channel name, views and likes for the channel's video;

now the different videos have different sub categories in the videos, we need to group them by and check whether which category has the highest number of trending videos;

after we get the highest category videos, we need to differentiate the dataframe for that category and put all the data in it;

using the dataframe we need to get the highest number of views, likes and comments for the video to get on trending;

not only highest likes, views and comments even the minimum threshold to get on trending list;

we can also use analysis methods to check whether the videos titles contain the captalised word or not and what character is used more;

also the trending can be shown by looking for the length 0f the title;

we can also use other histogram, bar plots to get the views having maximum likes and minimum likes for the following data;

}

Analysis

{

We can also use the barplot to see which channel has the highest number of videos and the channel name for the videos, and if it has the highest number of videos whether its videos are on trending;

}

Hence this is an analysis project we can use the following analysis methods to complete our project;

}

}

8.

# **6. IMPLEMENTATION**

**6.1 DATASET COLLECTION** ([Trending-YouTube-Scraper](https://github.com/mitchelljy/Trending-YouTube-Scraper))

* We had collected the dataset needed for our project using Kaggle notebooks and downloaded the dataset which we want to work on.
* We had 10 datasets for different countries, but we had chosen to work on the dataset for India and US.
* We will perform trending video analysis on the benchmark dataset and filter out the results.
* The data is already scraped and cleaned using pandas and Jupyter notebooks using NLTK library.
* Here is the link for the pre-processed dataset.

**6.2 DATASET TOOLS USED**

* To scrape the dataset, they have used different tools and libraries that gives us a clean dataset with a maximum precision.
* They used YouTube API KEY to the key identifies your project and provides API access, quota, and reports.
* The different modules for the data extraction and cleaning, except the request’s module.
* They had assigned 10 country codes for 10 datasets to filter out the data given in it.
* Running the script using all the technique helps us in cleaning the data.



9.

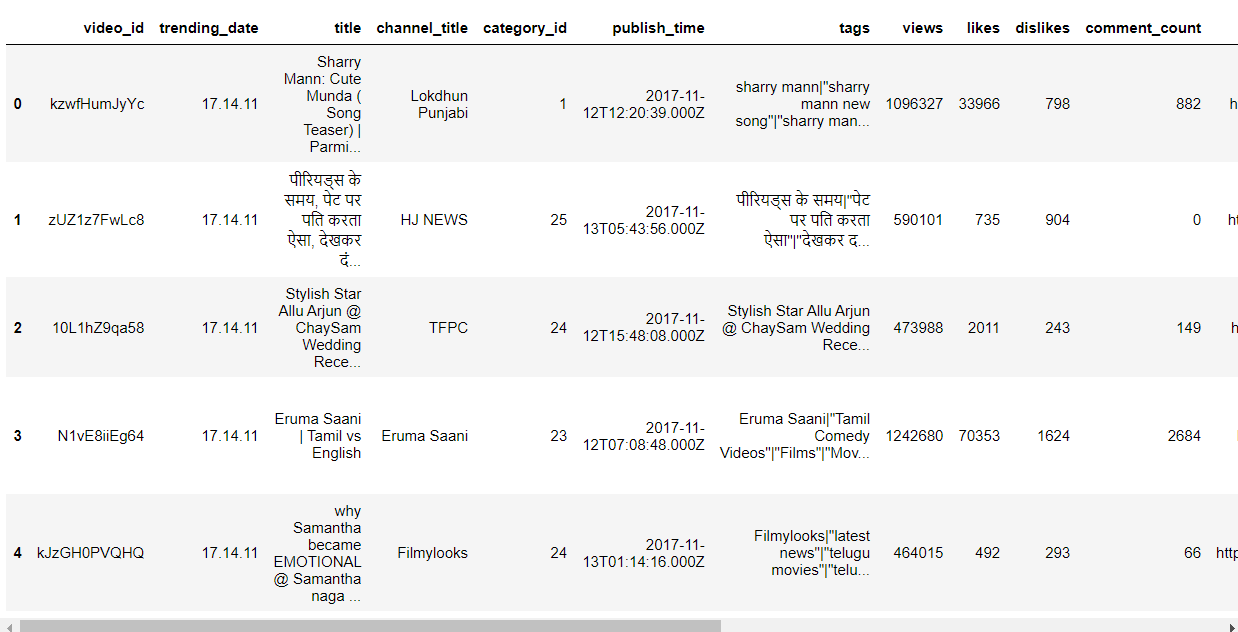
# importing the libraries required  
import pandas as pd  
import warnings  
import numpy as np  
from matplotlib import pyplot as plt  
import seaborn as sns  
import matplotlib as mpl

pwd

'C:\\Users\\Jaideep\\Downloads\\chadvik'

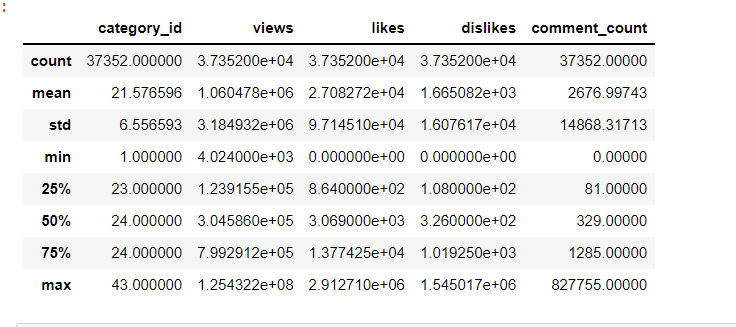
# importing the first csv file  
df = pd.read\_csv('INvideos.csv',encoding='utf-8')

# we need to print the first 5 data for the csv file  
df.head()



# to show number of rows and columns  
df.shape(37352, 16)

# TO DESCRIBE ALL THE ATTRIBUTES IN THE LOG FORMAT  
df.describe()



10.

# to remove the likes and views in higher order for e^n  
df['log\_views'] = np.log(df['views'])  
# df['log\_likes'] = np.log(df['likes'])

# importing the bokeh libaries that are useful to complete the dataplotting  
from bokeh.plotting import figure , output\_file, show  
from bokeh.models import ColumnDataSource  
from bokeh.models.tools import HoverTool

# showing the bokeh circle using the comment count and the likes count and the source is dataframe which we extracted  
fig = figure()  
p = fig.circle(x = 'likes', y = 'comment\_count', source = df)

#### showing the figure usibng the bokeh mode

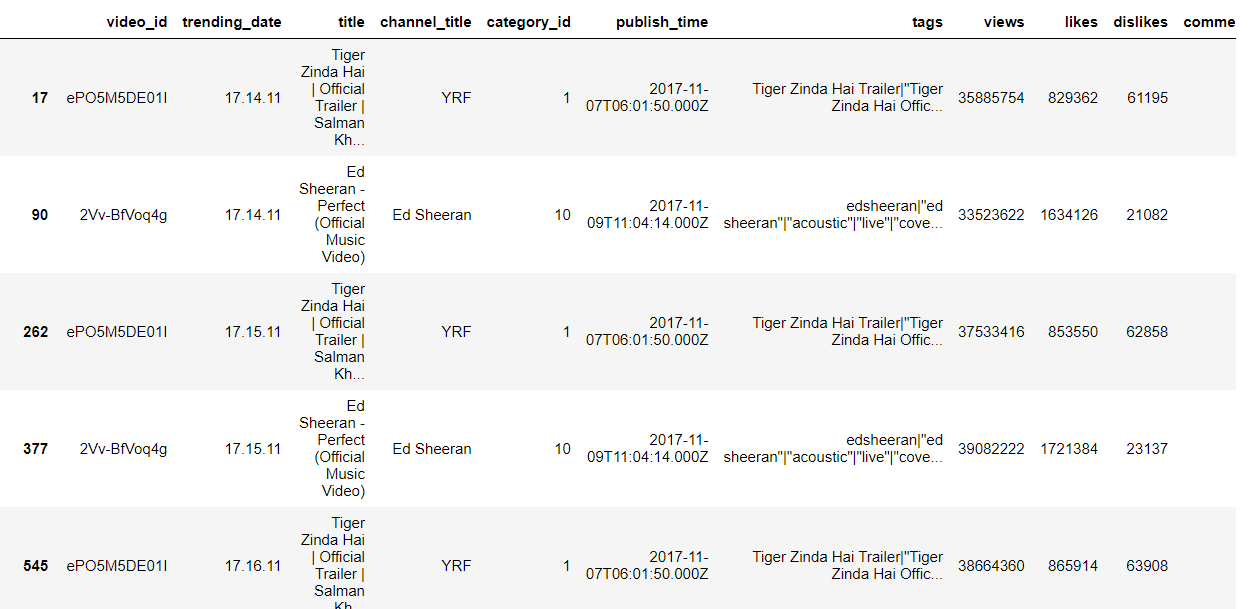
show(fig)

# **OUTLIER DETECTION USING THE PERCENTILE AND QUANTILE**

max\_threshold = df['likes'].quantile(0.96)  
max\_threshold

158879.0

df[df['likes'] > max\_threshold]



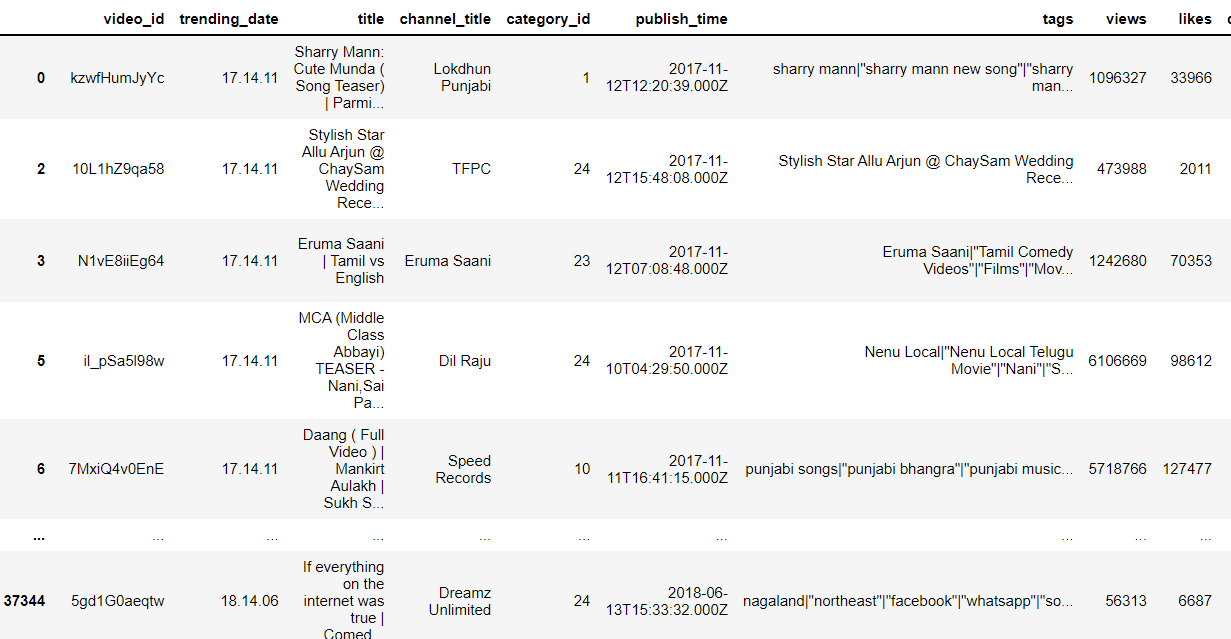
11.

  
  
[1494 rows x 17 columns]

min\_threshold = df['likes'].quantile(0.40)  
min\_threshold

1879.0

df[df['likes'] > min\_threshold]



12.

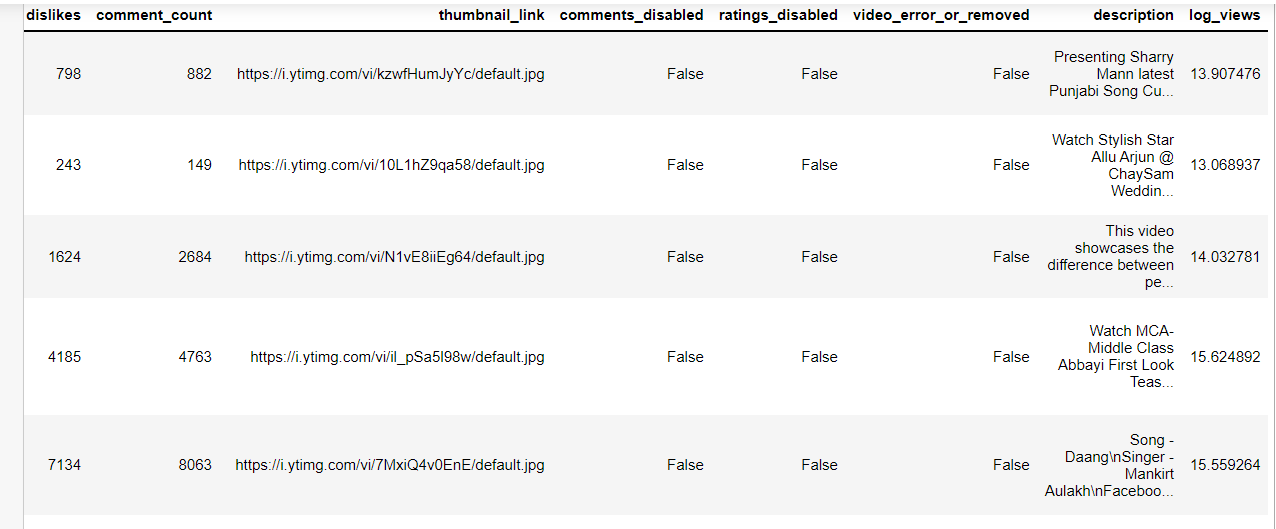


[22408 rows x 17 columns]

# Extracting the final dataset using the outlier detection

df = df[(df['likes']<max\_threshold) & (df['likes'] > min\_threshold)]  
df



  
[20912 rows x 17 columns]

13.

# to remove the likes and views in higher order for e^n  
df['log\_views'] = np.log(df['views'])  
df['log\_likes'] = np.log(df['likes'])

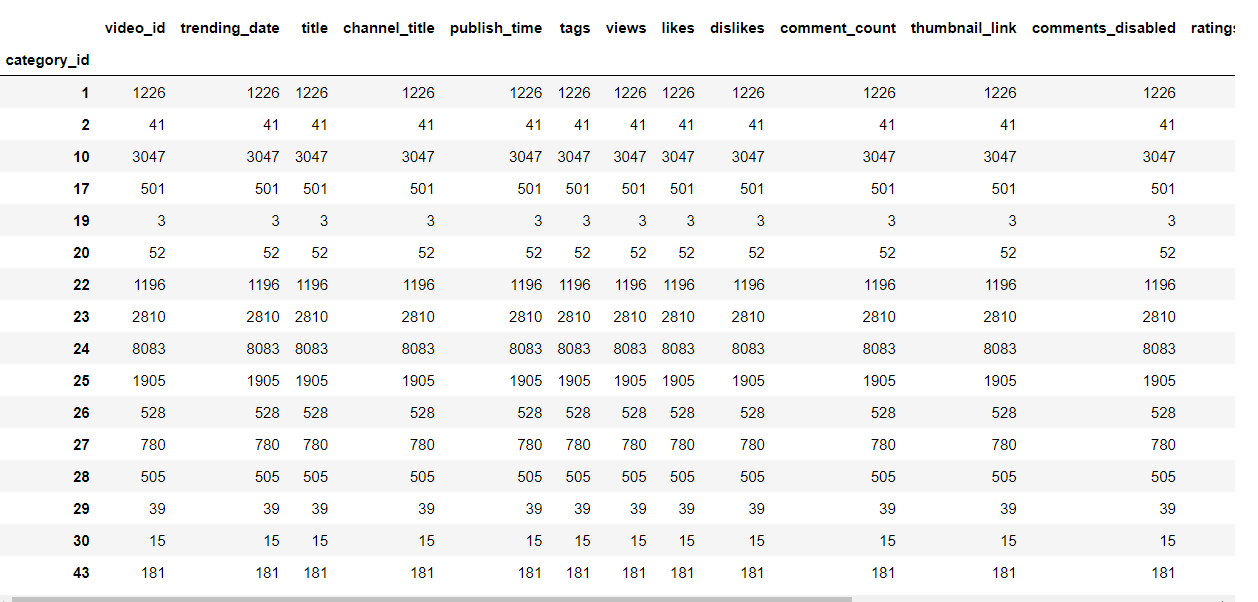
# For hovering over information  
  
TOOLTIPS = [  
 ("Title", "@title"),  
 ("Channel","@channelTitle"),  
 ("Views", "@view\_count")  
]  
  
fig1 = figure(tooltips = TOOLTIPS)  
p = fig1.circle(x='log\_views', y= 'log\_likes',source = df)  
  
# adding axis labels and coloring the graph plots  
fig1.xaxis.axis\_label = 'Likes'  
fig1.yaxis.axis\_label = 'Views'  
glyph = p.glyph  
glyph.size = 15  
glyph.fill\_alpha = 0.3  
glyph.fill\_color = "green"  
glyph.line\_color = "grey"  
glyph.line\_dash = [6, 3]  
glyph.line\_width = 2

# glyphs are the data points where it is plotted on the graph to understand what video has what trending update.

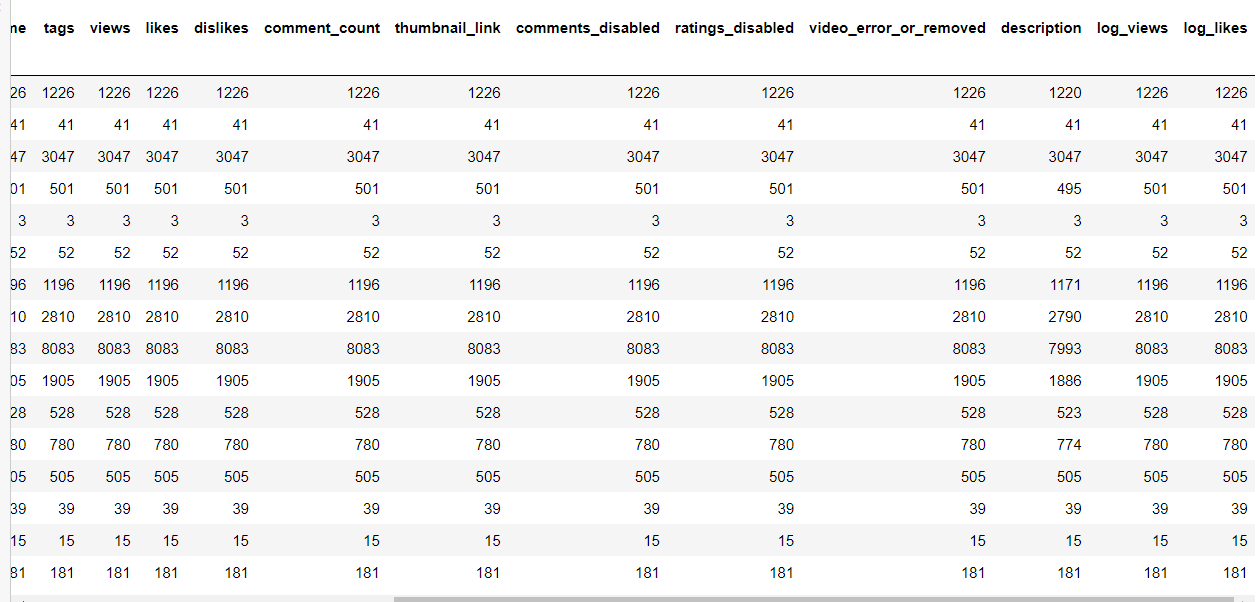
show(fig1)

# **CATEGORY ID GROUPING**

df.groupby('category\_id').count()



14.

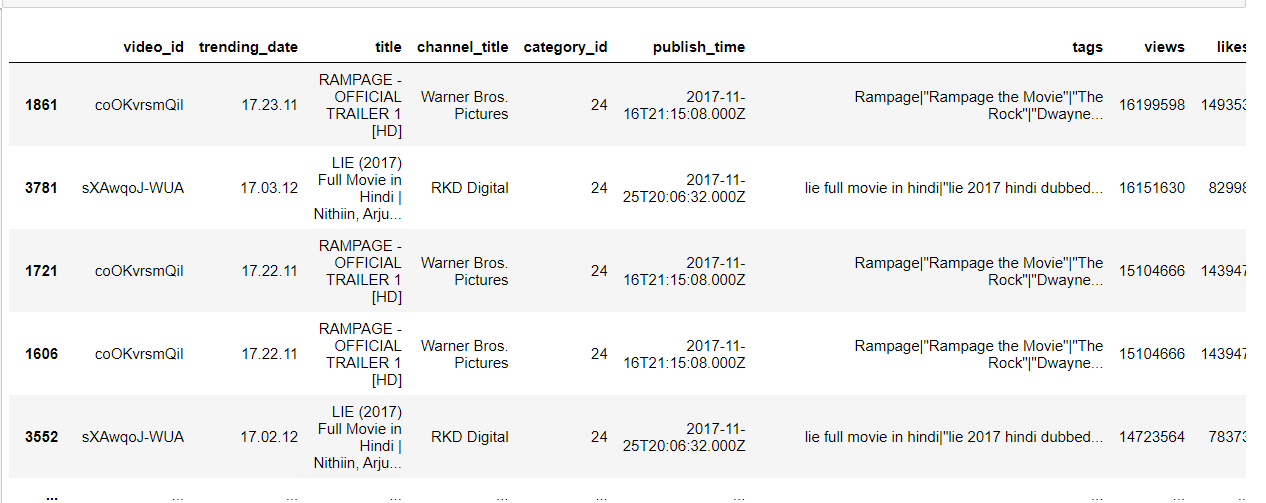


# ID -> Category Name  
# 1 --> Film and animation  
# 2 --> Autos and vehicles  
# 10 --> Music  
# 15 --> Pets and Animals  
# 17 --> Sports  
# 19 --> Travel and Events  
# 20 --> Gaming  
# 22 --> People and Blogs  
# 23 --> Comedy  
# 24 --> Entertainment  
# 25 --> News and politics  
# 26 --> How to style  
# 27 --> Education  
# 28 --> Science and Technology  
# 29 --> Non-profits and Activism

# Here we have the entertainment category as the highest view count and next the music category so we will create other dataframe for it and sort it by view count.

entertainment = df[df['category\_id'] == 24]

entertainment.sort\_values('views', ascending=False)

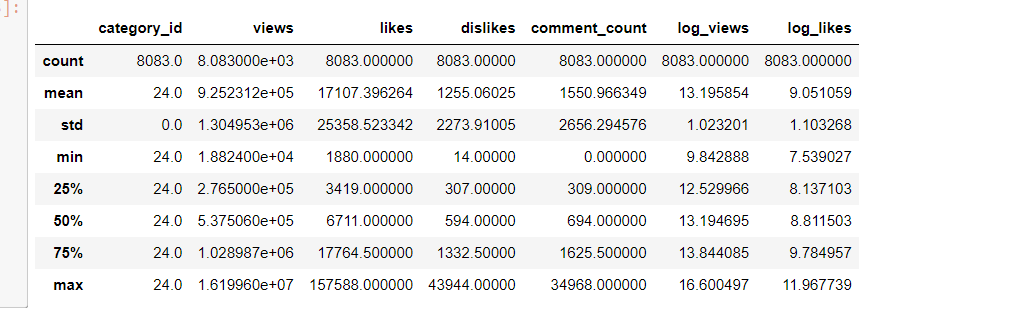
 [8083 rows x 18 columns]

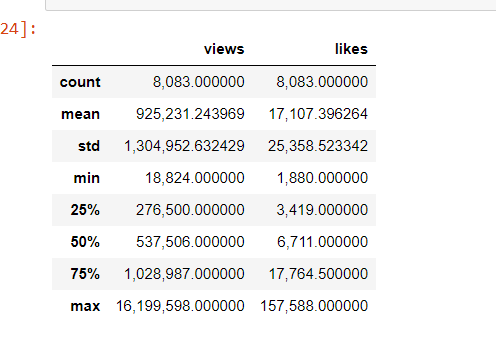
15.

entertainment.views.dtype

dtype('int64')

entertainment.describe()

  
 log\_views log\_likes   
count 8083.000000 8083.000000   
mean 13.195854 9.051059   
std 1.023201 1.103268   
min 9.842888 7.539027   
25% 12.529966 8.137103   
50% 13.194695 8.811503   
75% 13.844085 9.784957   
max 16.600497 11.967739



pd.options.display.float\_format = '{:,f}'.format  
entertainment[['views','likes']].describe()

views likes  
count 8,083.000000 8,083.000000  
mean 925,231.243969 17,107.396264  
std 1,304,952.632429 25,358.523342  
min 18,824.000000 1,880.000000  
25% 276,500.000000 3,419.000000  
50% 537,506.000000 6,711.000000  
75% 1,028,987.000000 17,764.500000  
max 16,199,598.000000 157,588.000000

16.

# for entertainment category to be in trending you need minimum of 18000 views and 2000 likes, and the highest is 1.6 crore views and 85k likes

print ("Amongst the trending entertainment videos , the lowest trending trending number of views was {}".format(min(entertainment['views'])))

Amongst the trending entertainment videos , the lowest trending trending number of views was 18824

print ("Amongst the trending entertainment videos , the highest trending trending number of views was {}".format(max(entertainment['views'])))

Amongst the trending entertainment videos , the highest trending trending number of views was 16199598

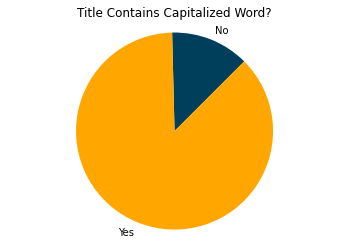
print ("Amongst the trending entertainment videos , the trending video was {}".format(max(entertainment['views'])))

Amongst the trending entertainment videos , the trending video was 16199598

def contains\_capital(s):  
 for w in s:  
 if w.isupper():  
 return True  
 return False

entertainment["contains\_capitalised"] = entertainment["title"].apply(contains\_capital)   
value\_counts = entertainment["contains\_capitalised"].value\_counts().to\_dict()  
  
fig2, ax = plt.subplots()  
\_ = ax.pie([value\_counts[False], value\_counts[True]], labels=['No', 'Yes'],   
 colors=['#003f5c', '#ffa600'], textprops={'color': '#040204'}, startangle=45)  
\_ = ax.axis('equal')  
\_ = ax.set\_title('Title Contains Capitalized Word?')

C:\Users\Jaideep\AppData\Local\Temp/ipykernel\_14668/2292556925.py:8: SettingWithCopyWarning:   
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead  
  
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy  
 entertainment["contains\_capitalised"] = entertainment["title"].apply(contains\_capital)



17.

# from above we can see that almost 80% of the videos has title as capital letter when it is in trending for entertainment category.

# Let’s add another column to our dataset to analyze the length of titles of videos, then plot the title length histogram to get an idea of the length of trending video titles:

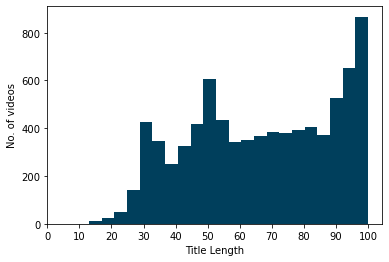
PLOT\_COLORS = ["#268bd2", "#0052CC", "#FF5722", "#b58900", "#003f5c"]

entertainment["title\_length"] = entertainment["title"].apply(lambda x: len(x))  
  
fig2, ax = plt.subplots()  
\_ = sns.distplot(entertainment["title\_length"], kde=False, rug=False,   
 color=PLOT\_COLORS[4], hist\_kws={'alpha': 1}, ax=ax)  
\_ = ax.set(xlabel="Title Length", ylabel="No. of videos", xticks=range(0, 110, 10))

C:\Users\Jaideep\AppData\Local\Temp/ipykernel\_14668/1925263200.py:1: SettingWithCopyWarning:   
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead  
  
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy  
 entertainment["title\_length"] = entertainment["title"].apply(lambda x: len(x))

C:\Users\Jaideep\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

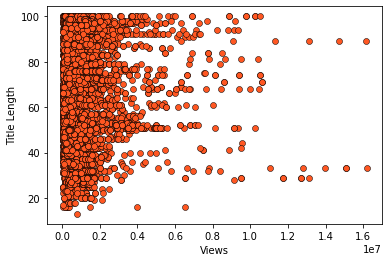
warnings.warn(msg, FutureWarning)



# From the above graph we can see that the more videos have more than or equal to atleast 100 characters to be in the trending list.

fig2, ax = plt.subplots()  
\_ = ax.scatter(x=entertainment['views'], y=entertainment['title\_length'], color=PLOT\_COLORS[2], edgecolors="#000000", linewidths=0.5)  
\_ = ax.set(xlabel="Views", ylabel="Title Length")

18.



# from the above scatter plot we can see that there is no relationship between the length and the views, but if we look keenly there are videos which are more than 80 characters that has 160000000 views, and there are videos which have 100 characeters but still <20000 views

# to remove the likes and views in higher order for e^n  
entertainment['log\_views'] = np.log(entertainment['views'])  
entertainment['log\_likes'] = np.log(entertainment['likes'])

C:\Users\Jaideep\AppData\Local\Temp/ipykernel\_14668/793671460.py:2: SettingWithCopyWarning:   
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead  
  
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy  
 entertainment['log\_views'] = np.log(entertainment['views'])  
C:\Users\Jaideep\AppData\Local\Temp/ipykernel\_14668/793671460.py:3: SettingWithCopyWarning:   
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead  
  
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy  
 entertainment['log\_likes'] = np.log(entertainment['likes'])

# For hovering over information  
  
TOOLTIPS = [  
 ("Title", "@title"),  
 ("Channel","@channel\_title"),  
 ("Views", "@views")  
]  
  
fig4 = figure(tooltips = TOOLTIPS)  
p = fig4.circle(x='log\_views', y= 'log\_likes',source = entertainment)

19.

# adding axis labels and coloring the graph plots  
fig4.xaxis.axis\_label = 'Likes'  
glyph = p.glyph

glyph.size = 15  
glyph.fill\_alpha = 0.3  
glyph.fill\_color = "green"  
glyph.line\_color = "grey"  
glyph.line\_dash = [6, 3]  
glyph.line\_width = 2

show(fig4)

import seaborn as sns

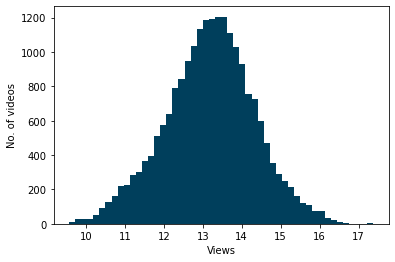
figg, ax = plt.subplots()  
\_ = sns.distplot(df["log\_views"], kde=False, color=PLOT\_COLORS[4],   
 hist\_kws={'alpha': 1}, bins=np.linspace(0, 2.3e9, 47), ax=ax)  
\_ = ax.set(xlabel="Views", ylabel="No. of videos", xticks=np.arange(0, 2.4e9, 1e7))  
\_ = ax.set\_xlim(right=2.5e8)  
\_ = plt.xticks(rotation=90)



# We note that the vast majority of trending videos have 5 million views or less. We get the 5 million number by calculating 0.1×10^8/2=5×10^6 Now let us plot the histogram just for videos with 25 million views or less to get a closer look at the distribution of the data

fig, ax = plt.subplots()  
\_ = sns.distplot(df[df["log\_views"] < 25e6]["log\_views"], kde=False,   
 color=PLOT\_COLORS[4], hist\_kws={'alpha': 1}, ax=ax)  
\_ = ax.set(xlabel="Views", ylabel="No. of videos")

20.



# Now we see that the majority of trending videos have 1 million views or less. Let's see the exact percentage of videos less than 1 million views

df[df['views'] < 1e6]['views'].count() / df['views'].count() \* 100

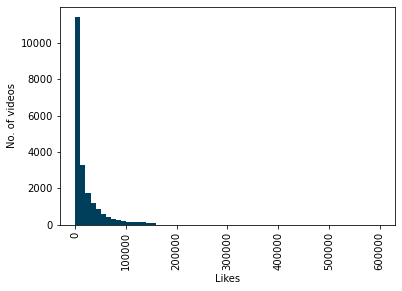
71.1792272379495

# So, it is around 70%. Similarly, we can see that the percentage of videos with less than 1.5 million views is around 71%, and that the percentage of videos with less than 5 million views is around 80%.

# After views, we plot the histogram for likes column

plt.rc('figure.subplot', wspace=0.9)  
fig, ax = plt.subplots()  
\_ = sns.distplot(df["likes"], kde=False,   
 color=PLOT\_COLORS[4], hist\_kws={'alpha': 1},   
 bins=np.linspace(0, 6e5, 61), ax=ax)  
\_ = ax.set(xlabel="Likes", ylabel="No. of videos")  
\_ = plt.xticks(rotation=90)

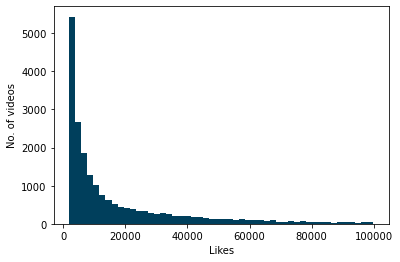
21.



# We note that the vast majority of trending videos have between 0 and 100,000 likes. Let us plot the histogram just for videos with 1000,000 likes or less to get a closer look at the distribution of the data

fig, ax = plt.subplots()  
\_ = sns.distplot(df[df["likes"] <= 1e5]["likes"], kde=False,   
 color=PLOT\_COLORS[4], hist\_kws={'alpha': 1}, ax=ax)  
\_ = ax.set(xlabel="Likes", ylabel="No. of videos")

22.

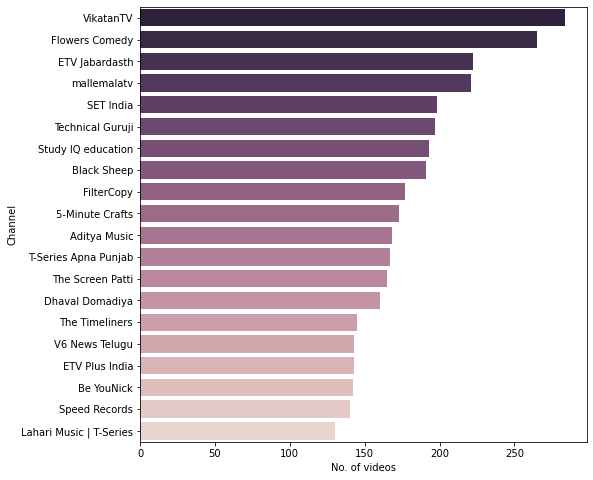


df[df['likes'] < 4e4]['likes'].count() / df['likes'].count() \* 100

84.26740627390971

cdf = df.groupby("channel\_title").size().reset\_index(name="video\_count") \  
 .sort\_values("video\_count", ascending=False).head(20)  
  
fig, ax = plt.subplots(figsize=(8,8))  
\_ = sns.barplot(x="video\_count", y="channel\_title", data=cdf,  
 palette=sns.cubehelix\_palette(n\_colors=20, reverse=True), ax=ax)  
\_ = ax.set(xlabel="No. of videos", ylabel="Channel")

23.

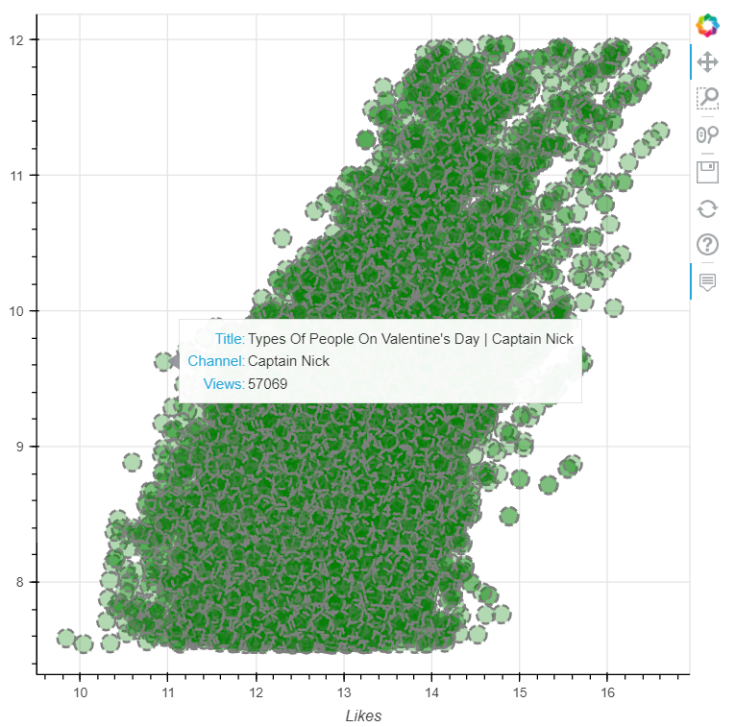
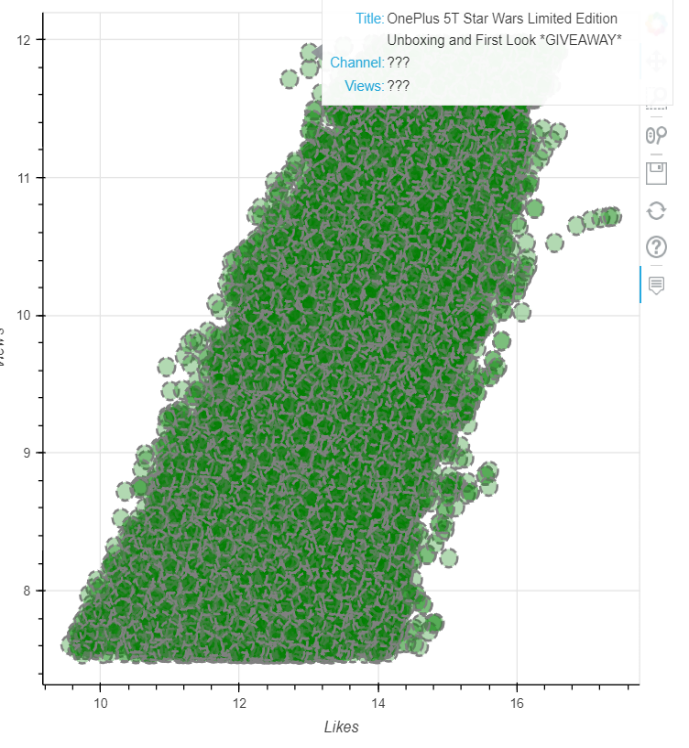


from collections import Counter

title\_words = list(df["title"].apply(lambda x: x.split()))  
title\_words = [x for y in title\_words for x in y]  
Counter(title\_words).most\_common(25)[('|', 26838),  
 ('-', 6921),  
 ('2018', 3225),  
 ('||', 2044),  
 ('Song', 1528),  
 ('The', 1459),  
 ('Episode', 1417),  
 ('Movie', 1328),  
 ('Video', 1218),  
 ('Full', 1194),  
 ('&', 1054),  
 ('Latest', 1031),  
 ('Songs', 1015),  
 ('Official', 1010),  
 ('Trailer', 946),  
 ('New', 909),  
 ('Punjabi', 891),  
 ('Telugu', 805),  
 ('in', 704),  
 ('2', 702),  
 (':', 660),  
 ('News', 633),  
 ('2017', 630),  
 ('of', 625),  
 ('to', 593)

24.

# **7. RESULTS DISCUSSION**



**Fig 7.1. Bokeh Output-1 Fig 7.2. Bokeh Output-2**

# 25.

# **8. CONCLUSION AND FUTURE WORK**

8.1 **CONCLUSION**

The above implementation shows analysis of the YouTube videos, which is based on the different parameters such as category name, length of the title, views, likes. If a person watches a video, the first thing he notices is the above mentioned parameters. The data visualization on these parameters gives us the appropriate analysis of videos, which helps the viewer to make a choice to watch the video or skip it. It will also help the channel owner to make videos based on the interest of the viewer. The graphs and tables generated gives a proper remark on the video.

8.2 **FUTURE WORK**

The above work shows the analysis of the trending videos on YouTube. In the future, we shall try to implement a user interface for making it interactive for the user. Which will make it more comfortable for visualization.

26.

# **9. SPECIFIC CONTRIBUTION**

|  |  |
| --- | --- |
| **NAME** | **CONTRIBUTED** |
| T Venkata Sai Sathvik | Data Analysis and Visualization code and creating GitHub repository. |
| Nihal Agarwal | literature survey, creating the project documents, identification of basic requirements of the project. |
| Jaideep Sharma | Report document, data set extraction and cleaning. |
| Shaik Abdul Shaan | Data Visualization code and maintaining the GitHub repository |

27.

# **9. REFERENCES**

1. [2021 International Conference on Data and Software Engineering (ICoDSE)](https://ieeexplore.ieee.org/xpl/conhome/9648407/proceeding), 3-4 Nov. 2021, DOI:[10.1109/ICoDSE53690.2021.9648486](https://doi.org/10.1109/ICoDSE53690.2021.9648486).
2. [International Research Journal of Engineering and Technology (IRJET), Volume: 07 Issue: 08 | Aug 2020 .](https://www.irjet.net/archives/V7/i8/IRJET-V7I8732.pdf) e-ISSN: 2395-0056, p-ISSN: 2395-0072.
3. The Computer Journal, bxab142, <https://doi.org/10.1093/comjnl/bxab142>
4. Cornell University, <https://arxiv.org/abs/1409.7733>.
5. <https://esource.dbs.ie/bitstream/handle/10788/4260/msc_niture_aa_2021.pdf?sequence=1&isAllowed=y>
6. <http://sbp-brims.org/2018/proceedings/papers/latebreaking_papers/LB_14.pdf>

28.