A
Report On

YouTube Analysis

Social Media Analytics

Submitted By:

1. Anjitha Antony(C0796673)
2. Shilpa Thomas(C0800534)
3. Rincy Jose(C0790298)
4. Ashna Kunnathully Prasannan(C0786430)

Under the guidance of

Prof.Debashish Roy

Contents

1. Abstract

Nowadays, the impact of online video streaming platforms is very high compared to the past days. Websites such as YouTube offers a great platform to share their knowledge, ideas, and interesting information with their viewers. This paper deals with the analysis of YouTube Data on Trending Videos. The analysis is done using user features such as Views, Comments, Likes, and Dislikes.

2. Introduction

YouTube is a well-known and widely utilized internet video platform in today's world. YouTube maintains a list of popular videos that is updated regularly. Analysing the data can give insights into YouTube trending videos, to see what is common among those. People who desire to boost the popularity of their YouTube videos could benefit from these insights. This paper can help in finding, measuring, analysing, and comparing key aspects of YouTube trending videos.

3. Data Collection:

3.1 Dataset

Datasets are essential for the advancement of numerous computational domains, providing results with scope, robustness, and confidence. The data set that we were using during the study was gathered from the internet. The dataset was retrieved from the Internet. Which contains information about the videos such as YouTube id, movie id, and titles. The dataset comprised 25623 instances with 3 attributes.

	youtubeId	movieId	title
0	K26_sDKnvMU	1	Toy Story (1995)
1	3LPANjHlPxo	2	Jumanji (1995)
2	rEnOoWs3FuA	3	Grumpier Old Men (1995)
3	j9xml1CxgXI	4	Waiting to Exhale (1995)
4	ltwvKLnj1B4	5	Father of the Bride Part II (1995)

Figure 1: Sample of Dataset

We have visualized our data using pandas profiling.

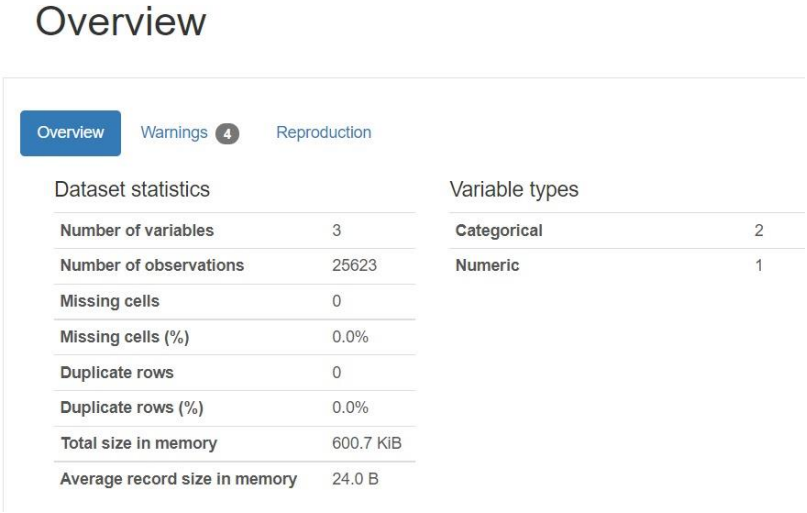


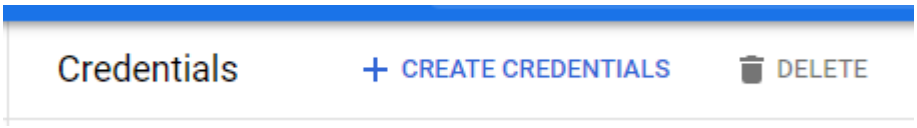
Figure 2: Overview of Dataset

3.2 Extraction of YouTube Videos

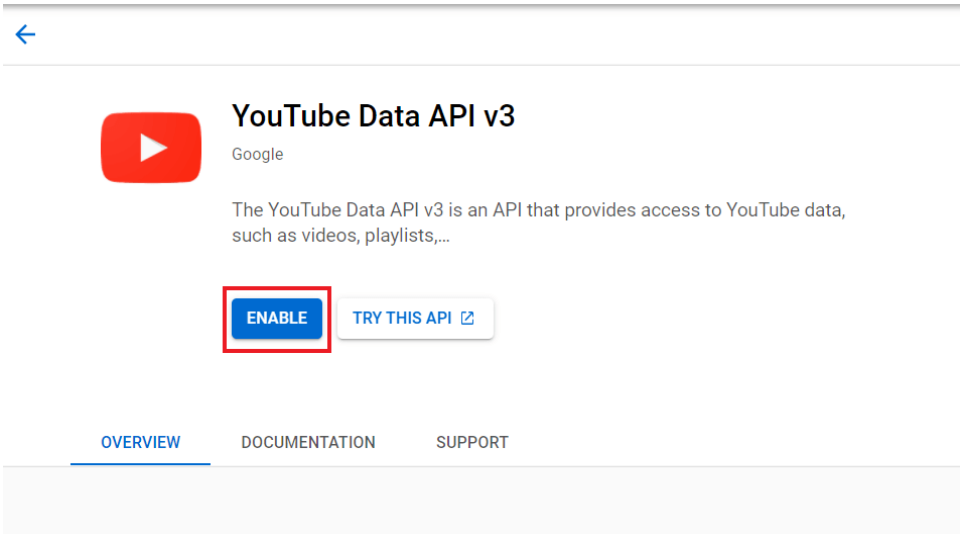
The extraction phase comprises YouTube API enabling, Obtaining Video Information, Keyword-based search, retrieving channel details followed by extracting the comments.

3.2.1 YouTube API key creation & enabling

To generate an API key navigate to Google's API Console (https://console.developers.google.com/apis/) and from the credentials, create a new credential as shown below.



In the next phase, select the desktop app as the application type and gave a name for the project then create. The newly generated API key is displayed and click on Enable.



4. Data Pre-processing:

The dataset was pre-processed before analysis to ensure that it was in a usable format. The null values were checked, and it was found that there were no null values present in the data which was considered for analysis.

```
1 # checking for missing values
2 videolinks_data.isnull().sum()
```

```
youtubeId    0
movieId      0
title        0
dtype: int64
```

5. Data Analysis:

The analysis phase comprises YouTube API enabling, Obtaining Video Information, Keyword-based search, retrieving channel details followed by extracting the comments.

The libraries required for the project are pandas, Json, Matplotlib, requests, and seaborn.

5.1 Getting Video Details

This step involves collecting information from YouTube videos such as the title, description, upload time, and statistics such as the number of views, likes, and dislikes. By using a video_Id K26_sDKnvMU, checked the different parameters of the YouTube video.


```
{'kind': 'youtube#videoListResponse', 'etag': 'oGuzP739H1f9z0RSHv0WlaR7HZw', 'items': [{'kind': 'yout
ube#video', 'etag': 'tKfhnzxARPjKN3ibc4k_Wj8rHrI', 'id': 'K26_sDKnvMU', 'snippet': {'publishedAt': '2
015-05-26T17:11:42Z', 'channelId': 'UC1o_r-f-ZG-x6U9c6MfUfHQ', 'title': 'Toy Story (1995) Trailer 2
(VHS Capture)', 'description': 'Trailer for Toy Story (1995) captured from the The Santa Clause (199
4) VHS tape. This tape is labeled 3633 and released in (1994).', 'thumbnails': {'default': {'url': 'h
ttps://i.ytimg.com/vi/K26_sDKnvMU/default.jpg', 'width': 120, 'height': 90}, 'medium': {'url': 'http
s://i.ytimg.com/vi/K26_sDKnvMU/mqdefault.jpg', 'width': 320, 'height': 180}, 'high': {'url': 'http
s://i.ytimg.com/vi/K26_sDKnvMU/hqdefault.jpg', 'width': 480, 'height': 360}, 'standard': {'url': 'htt
ps://i.ytimg.com/vi/K26_sDKnvMU/sddefault.jpg', 'width': 640, 'height': 480}}, 'channelTitle': 'retro
VHS trailers', 'tags': ['1995', 'Capture', 'promo', 'preview', 'teaser', 'trailer', 'vhs', 'vcr', 'ta
pe', 'retro', 'Company Logo'], 'categoryId': '1', 'liveBroadcastContent': 'none', 'localized': {'titl
e': 'Toy Story (1995) Trailer 2 (VHS Capture)', 'description': 'Trailer for Toy Story (1995) captured
from the The Santa Clause (1994) VHS tape. This tape is labeled 3633 and released in (1994).'}}, 'sta
tistics': {'viewCount': '102906', 'likeCount': '100', 'dislikeCount': '12', 'favoriteCount': '0', 'co
mmentCount': '13'}}], 'pageInfo': {'totalResults': 1, 'resultsPerPage': 1}}
The number of likes= 100
The number of dislikes= 12
The number of views= 102906
The number of comments= 13
```


To calculate the duration of the video, we are converting it into a timeframe as follows.

```
youtube_stats_df = pd.read_csv("C:\\Users\\rinuj\youtube_videos.csv",names= [ 'youtubeId','description'
       'likeCount', 'dislikeCount','commentCount','Duration'])
#youtube_stats_df.head()
youtube_stats_df.shape
```

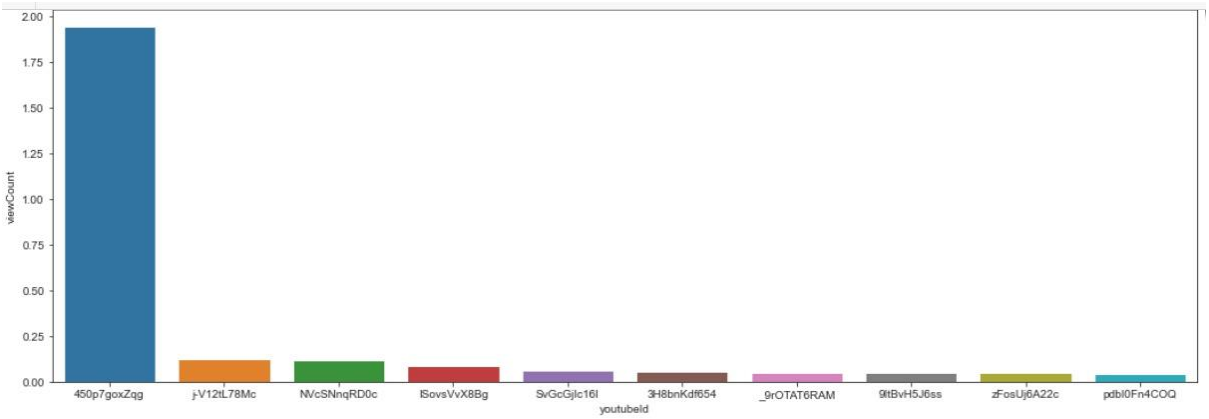
5.2 Deleting Duplicate YouTube IDs

For removing the duplicate values in the youtube ids, we use

```
youtube_data_df=youtube_data_df.drop_duplicates(subset=['youtubeId'])
youtube_data_df.head()
```

5.3. Top 10 most viewed videos from the dataset

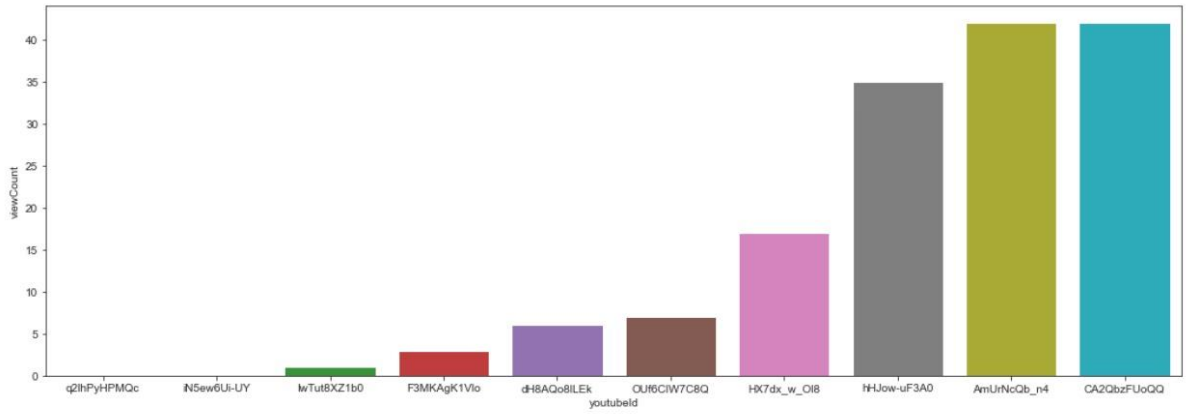
We are sorting the YouTube ids in ascending order using the view count and visualizing the results with YouTube ids on the x-axis and view count on the y-axis.



5.4. Top 10 least viewed videos from the dataset

We are sorting the YouTube ids in descending order using the view count and visualizing the results with YouTube ids on the x-axis and view counts on the y-axis.

```
youtube_stats_df.sort_values(by=['viewCount'],ascending=True).head(10)
```



5.5. Title of most liked video

We're utilizing the like count to sort the youtube ids in ascending order and outputting the results.

```
youtube_stats_df.sort_values(by=['viewCount'],ascending=False).head(1)["title"]
#youtube_stats_df.sort_values(by=['viewCount'],ascending=False).head(1)
```

```
90163    All of Me (2013)
Name: title, dtype: object
```

5.6. Title of least liked the video

We're utilizing the like count to sort the YouTube ids in descending order and outputting the results.

```
#print(youtube_stats_df.sort_values(by=['viewCount'],ascending=True).head(1))
print(youtube_stats_df.sort_values(by=['viewCount'],ascending=True).head(1)["title"])
```

```
69643    Hear My Song (1991)
Name: title, dtype: object
```

5.7. Title of the video with the highest duration

The duration of the YouTube videos is translated to seconds, and then they are sorted in ascending order.

```
youtube_stats_df.sort_values(by='Duration in seconds',ascending=False).head(1)
```

	youtubeId	description	viewCount	likeCount	dislikeCount	commentCount	Duration	movieId	title	Duration in seconds
85615	RBB_6gpUE-Q	Trailer for the independent film "Getting to K...	3772.0	2.0	3.0	0.0	PT12H49M24S	91444	Getting to Know You (1999)	46164.0

6. Extracting the comments from the YouTube ids

Extracted the YouTube videos from the corresponding YouTube ids using the API key and the set 100 comments count for each YouTube ids, then append with a snippet to get the results.

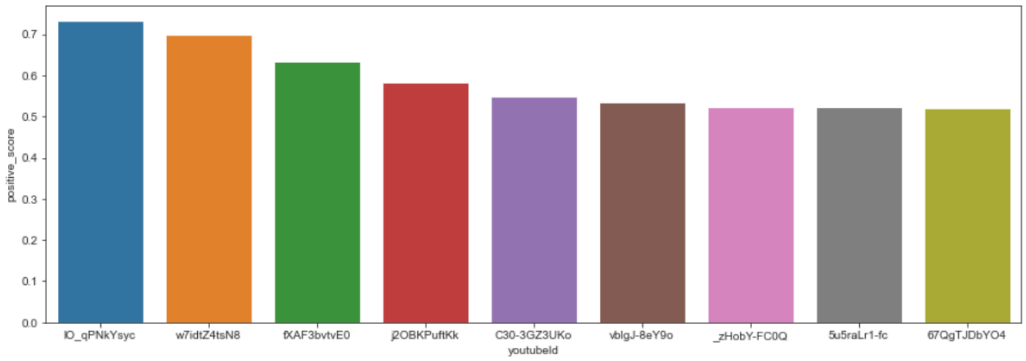
Then we have displayed, the top-10 videos that have the highest positive sentiment scores, which we calculated from the comments using the bar diagram.

```
df1.sort_values(by=['positive_score'],ascending=False).head(10)
```

	youtubeId	movieid	comments	polarity	positive_score
145605	vblgJ-8eY9o	1140	['Im here from school lol', 'Truly powerful an...	{'neg': 0.0, 'neu': 0.465, 'pos': 0.535, 'comp...	0.535
145779	ThH8WocPRM0	57	['Check out this great photo of the beautiful ...	{'neg': 0.0, 'neu': 0.61, 'pos': 0.39, 'compou...	0.390
145832	ThH8WocPRM0	57	['Check out this great photo of the beautiful ...	{'neg': 0.0, 'neu': 0.61, 'pos': 0.39, 'compou...	0.390
145641	ocCWEBSC4-0	1179	['A perfect modern film noir.', 'great!', 'OH ...	{'neg': 0.0, 'neu': 0.681, 'pos': 0.319, 'comp...	0.319
145650	CTDvQ_3VaG0	1189	['Is this what they used to scare the shit out...	{'neg': 0.134, 'neu': 0.555, 'pos': 0.311, 'co...	0.311
145622	OobLM9z3PLQ	1160	['W.C. Fields never gets old.', 'Any good pool...	{'neg': 0.0, 'neu': 0.704, 'pos': 0.296, 'comp...	0.296
145780	0or7hSz-7gc	58	['Hello my dear old friend , Hope you...	{'neg': 0.0, 'neu': 0.72, 'pos': 0.28, 'compou...	0.280
145833	0or7hSz-7gc	58	['Hello my dear old friend , Hope you...	{'neg': 0.0, 'neu': 0.72, 'pos': 0.28, 'compou...	0.280
145688	DfqPjRMsRP0	19	['Excellent. ♡ T.E.N.', 'This movie makes me l...	{'neg': 0.085, 'neu': 0.637, 'pos': 0.278, 'co...	0.278
145711	DfqPjRMsRP0	19	['Excellent. ♡ T.E.N.', 'This movie makes me l...	{'neg': 0.085, 'neu': 0.637, 'pos': 0.278, 'co...	0.278

6.1. Sentiment analysis of comments

Here we are using VADER LEXICON for the analysis. VADER is a sentiment analysis program that uses a lexicon and rules to analyze social media sentiments. Sentiment analysis was performed using positive and negative polarity ratings, and the results were presented in the form of a bar graph.



7. Conclusion

Our findings for measuring, assessing, and comparing essential characteristics of YouTube popular videos were reported in this study. Understanding these statistics will aid YouTube in not only developing better video processing algorithms but also in making judgments for individual YouTubers.

8. References

- C. Reed, T. Elvers, and P. Srinivasan, "What's Trending? Mining Topical Trends in UGC Systems YouTube as a Case Study", MDMKDD , New York, 2011, No. 4.
- M. Thelwall, P. Sud, and F. Vis, "Commenting on YouTube videos: From guatemalan rock to El Big Bang", Journal of the American Society for Information Science and Technology , 2012, Vol 63,pp. 616-629.
- YouTube data API documentation: https://developers.google.com/youtube/2.0/reference
- R. Zhou, S. Khemmarat, and L. Gao, "The impact of YouTube on recommendation system on video", IMC New York, pp. 404-410, 2010
- https://console.developers.google.com/apis/