

Analysis of Youtube Trending Videos

Mudit Aggarwal 2020UCD2157, Rupali Bisht 2020UCD2121

April 20, 2023

Abstract

In this project we go through the process of collecting, storing and Analysing youtube trending videos from 12th August 2020 till date. We demonstrate the construction of data ingestion pipeline to regularly call the youtube API for trending videos and storing the information on a managed Microsoft Azure instance in a docker container. We further analyze the data to extract insights about the general viewing practice of people during the COVID-19 Pandemic period. We draw insights about which category of youtube videos are generally seen on the youtube trending tab, we also analyze which channels are regularly seen on the trending tab. Finally, We also train a BERT(Bidirectional Encoder Representations from Transformers) powered model from title of collected youtube videos to estimate the category to which a video belongs to.

1 Overview

Youtube is one of the most popular social media sites with over 122 million daily active users and more than 1 billion hours of content spread across the globe. [globalmediainsight](#).

One of the most popular and important features of the youtube website is unarguably the trending tab (recently name changed to explore). This section contains some of the most popular/upcoming videos of a region. It was one of the first features added to the youtube site which enabled content creators on the platform to reach to a wider audience.

2 Problem Statement

Youtube trending videos dataset is an important asset which allows us to monitor the growth of the platform and change in paradigm of the amount and type of content present on the platform.

But there is a significant problem, there is no publicly available official historical dataset of youtube trending videos. The youtube API only allows for "mostPopular" video list of the date of the API call.

So this project is aimed to create a tool to regularly query and store youtube trending videos data set so as to eventually create a historical data set and perform analysis on it to understand YouTube usage and content paradigms.

Thus the steps taken to accomplish this are:

1. Create a persistent PostgreSQL database in a docker container on Microsoft Azure cloud platform



Figure 1: Youtube, the most popular Video streaming service.



Figure 2: Some major cloud service providers.

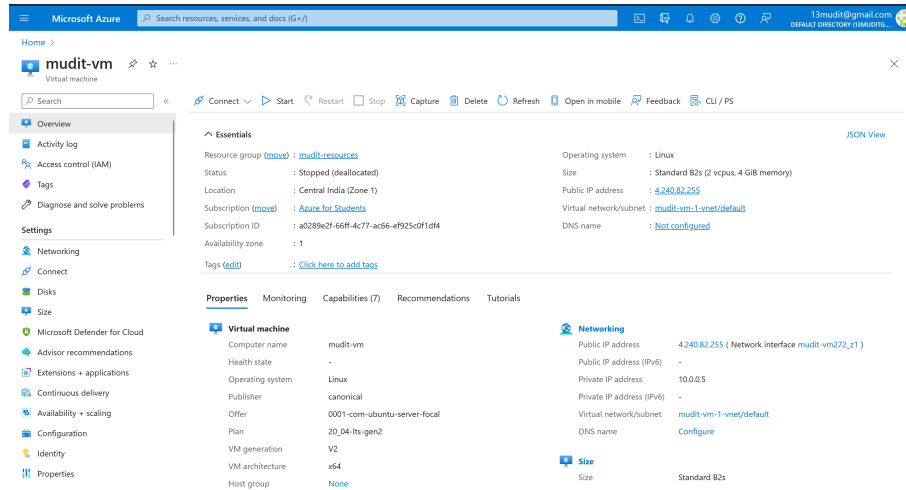


Figure 3: Azure VM console overview.

2. Run a regular cloud edge computing function to query Youtube API and store the current date most popular videos for 10 countries/regions
3. Create an Analysis script to query that data and draw insights from it
4. Train a BERT based Language model to identify the category of a given youtube title

3 Create a Cloud PostgreSQL Server

3.1 Introduction

Cloud services like Microsoft Azure, Google Cloud Platform, AWS, Alibaba Cloud provide a persistent machine running in a server. These machines are very useful for tasks like data storage or edge computing as they are fully maintained and we don't have to rely on our local machine's resources.

These services are generally aimed at larger organisations as it is much more convenient for them to deploy to a cloud instance than set up their own servers. However, These services can also be very useful for home users who may not have/may not have an incentive to permanently invest in powerful enough hardware that can perform reliable and continuous computation.

3.2 Creating a cloud instance on Azure

Of the popular Cloud Service Providers in 2, we decided to go with Microsoft Azure as it provides a \$100 credit to Students enrolled in an accredited institute without the need of any credit card.

To get started with Azure, we created a Microsoft Azure account and started a Cloud Virtual Machine 3

This enabled us to control a virtual machine with 2 Compute cores and 4 GiB of RAM

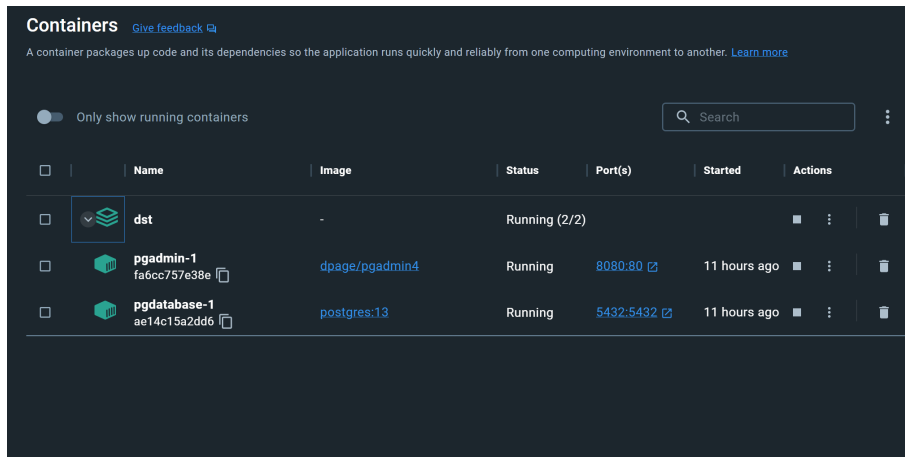


Figure 4: Running docker containers on local machine.

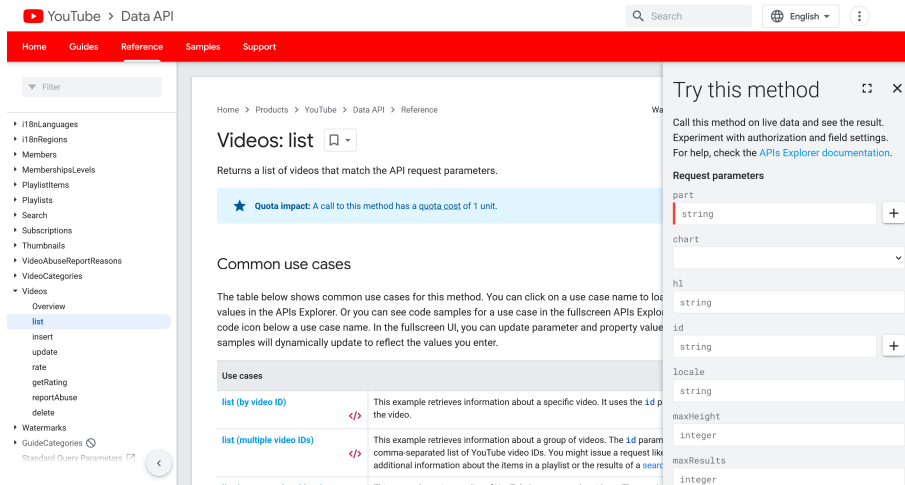


Figure 5: Youtube Data API v3.

3.3 Create a Docker Container to Host PostgreSQL server

Next step was to start a PostgreSQL server on the created VM. Instead of directly running the PostgreSQL server instance, we checked and monitored its use and performance on our local machine, then to ensure identical behaviour we set up a docker container and deployed that container on the cloud instance.

Docker is a powerful tool used to create isolated systems which allows the user to test their products without any fear of outside intervention/hard to their hardware.

Docker containers, in practice works exactly the same as isolated virtual machines, but a big difference between virtual machines and Docker containers is that docker containers are much quicker to deploy and scale.

4 Interfacing with Youtube API

4.1 Introduction

Yotube API is a part of a multitude of freely available services under Google APIs. Google APIs documentation and reference is very well documented and easy to use. To learn more [Google API explorer](#)

4.2 Youtube API

To get started with YouTube API, refer to [youtube api v3 reference](#)

Particularly for Trending videos the API query will look like:

```
request = youtube.videos().list(
    part="snippet,statistics",
    chart="mostPopular",
    regionCode=country_code,
    maxResults=50,
)
response = request.execute()
```

And the response will look like:

```
{
  "kind": "youtube#videoListResponse",
  "etag": etag,
  "nextPageToken": string,
  "prevPageToken": string,
  "pageInfo": {
    "totalResults": integer,
    "resultsPerPage": integer
  },
  "items": [
    video Resource
  ]
}
```

We Also query the the Youtube Video Categories index, which contains the snippet names and category numbers of all the video Categories.

This index can be directly joined with Video List response to get the category name of any given video

Category response query:

```
def get_video_categories(country_code):
    request = youtube.videoCategories().list(
        part="snippet",
        regionCode=country_code
    )
    return request.execute()
```

Category Response:

```
{
  "kind": "youtube#videoCategoryListResponse",
  "etag": etag,
  "nextPageToken": string,
  "prevPageToken": string,
  "pageInfo": {
    "totalResults": integer,
    "resultsPerPage": integer
  },
  "items": [
    videoCategory resource
  ]
}
```

5 Uploading Data to Cloud Storage

Finally after extracting data we upload the data to the PostgreSQL server in Microsoft Azure using Pandas and SQLAlchemy module of python with pyscopg2 binding for SQLAlchemy. This packages can be downloaded directly from Python Package Index.

We also use the historical data collected by [Rishav Sharma](#) and upload it to the cloud database too.

Due to bandwidth limits and large size of Data, it takes upto 30 mins to upload all the data

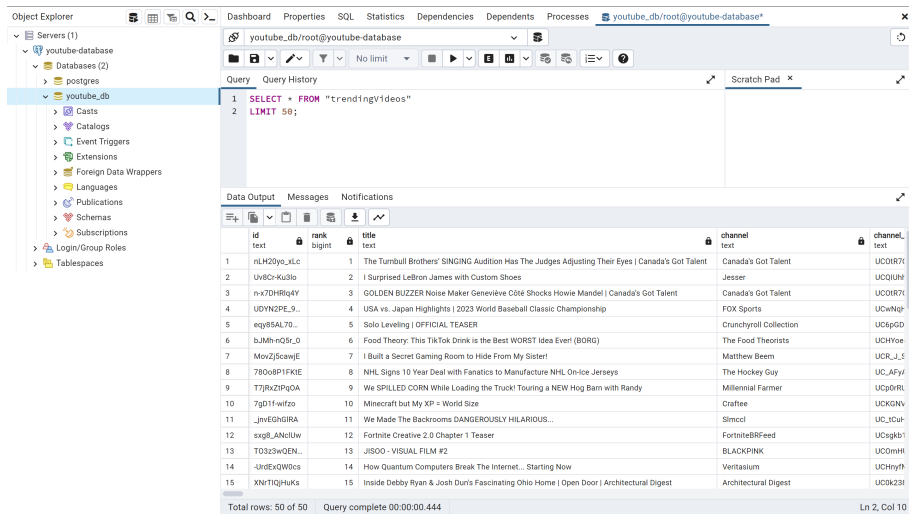


Figure 6: pgadmin connected to PostgreSQL server.

6 Analysis

6.1 Introduction

In this section we will work in a Jupyter notebook to draw insights and train a machine learning model using BERT on the stored database. Some of the packages and technologies we will use are:

1. Transformers model from Huggingface
2. Pytorch module for training our model
3. SQLAlchemy for connecting to the database
4. Other common packages and modules including matplotlib, numpy, pandas, etc

6.2 Exploratory Data Analysis

We can very easily extract information regarding youtube usage patterns across different countries for the time period of our data as can be seen in fig 7.

We can see from the graph that the bumps in the views for some countries directly correspond to the lockdown restrictions imposed on those countries. We can also conclude that youtube is predominantly watched in English speaking countries because of the significant lower view counts for countries like Russia, Japan and France as compared to Great Britain, US and India.

We can also do analysis of which category of youtube videos are the most popular and thus generally tend to be on the youtube trending page. As can be seen in fig 8. We can also see from the same graph that alot of video categories do not have any trending videos at all, this is due to the discontinuation of these categories by youtube itself. These exist in the youtube data api solely for analysis of older youtube videos data.

Finally we can check the youtube channels with some of the highest number of days on the youtube trending page

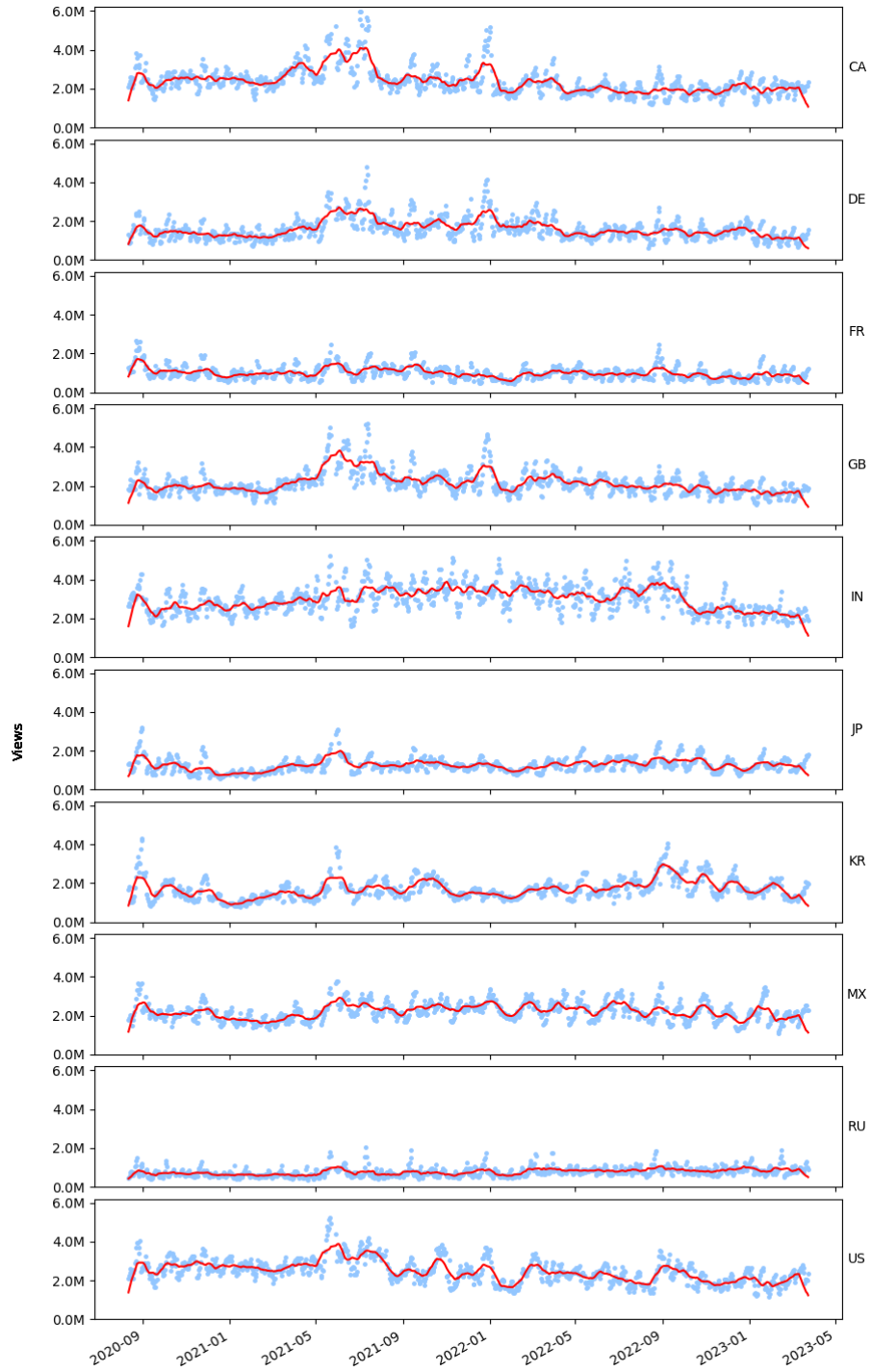


Figure 7: Trend of views per day per country.

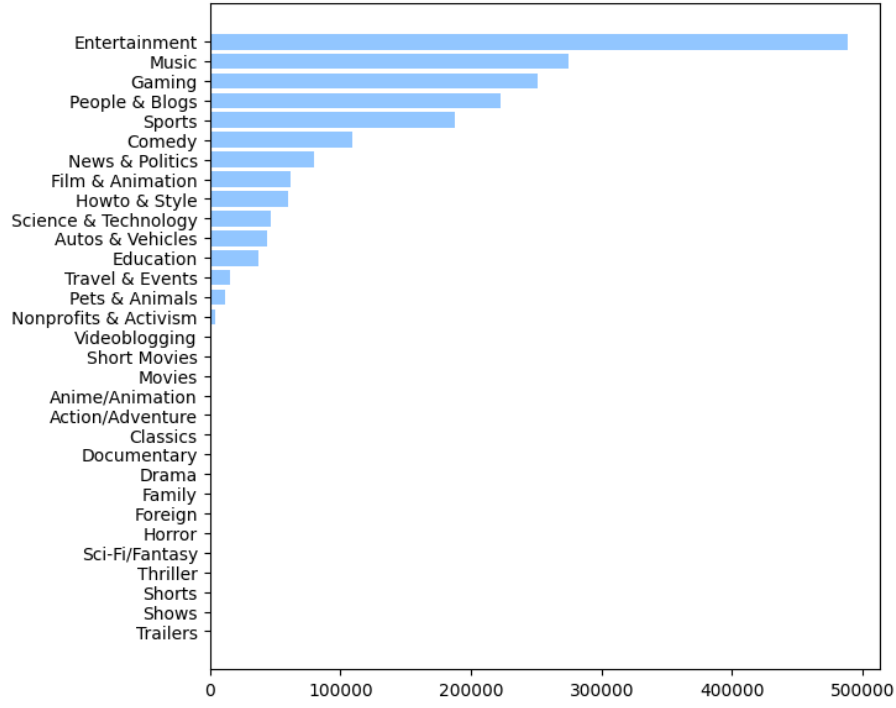


Figure 8: Frequency on the trending tab for different categories.

| | channel | trending count | countries | avg views | avg rank |
|----|---------------------|----------------|-----------|--------------|------------|
| 0 | BANGTANTV | 4312 | 10 | 1.096245e+07 | 106.921614 |
| 1 | JYP Entertainment | 3919 | 10 | 1.777034e+07 | 104.735647 |
| 2 | SMTOWN | 3868 | 10 | 1.800122e+07 | 101.823164 |
| 3 | HYBE LABELS | 3662 | 10 | 2.190246e+07 | 96.327417 |
| 4 | FORMULA 1 | 3153 | 9 | 4.363194e+06 | 105.787187 |
| 5 | NFL | 2924 | 10 | 6.492250e+06 | 95.740766 |
| 6 | 東海オンエア | 2717 | 1 | 1.836095e+06 | 75.396393 |
| 7 | BLACKPINK | 2701 | 10 | 3.475369e+07 | 94.876342 |
| 8 | NBA | 2578 | 9 | 2.223982e+06 | 101.742048 |
| 9 | The United Stand | 2437 | 4 | 3.471158e+05 | 96.193681 |
| 10 | Clash of Clans | 2421 | 10 | 5.883090e+06 | 99.887650 |
| 11 | Apex Legends | 2347 | 8 | 2.797414e+06 | 97.522795 |
| 12 | Vijay Television | 2245 | 1 | 2.411118e+06 | 84.974165 |
| 13 | Mnet K-POP | 2144 | 10 | 3.009970e+06 | 104.048041 |
| 14 | MrBeast | 2144 | 9 | 3.198041e+07 | 84.438433 |
| 15 | BT Sport | 2124 | 7 | 1.248324e+06 | 84.506121 |
| 16 | Sky Sports Football | 2120 | 5 | 8.593057e+05 | 72.044340 |
| 17 | TUDN México | 2092 | 6 | 1.220577e+06 | 90.370937 |
| 18 | beIN SPORTS France | 2056 | 5 | 5.657368e+05 | 86.535019 |
| 19 | WWE | 2028 | 8 | 2.397727e+06 | 106.636588 |

The table shows 20 of the most trending channels on youtube, we can infer from this table and from general knowledge, this is very different from the conventionally most subscribed channels on youtube. This is because of the fact that trending video count correlates directly with the number of uploaded videos for a channel.

For instance, the top entries in this table correspond to the very famous Korean pop music channels, whose music is enjoyed across all the countries/regions we have studied for, thus their trending videos count is cumulative across all these regions. Next we can see sports channels in this list which again are enjoyed across a wide location.

Both of these category of videos have in common the fact that their video upload frequency is relatively high and their videos remain on the trending page for multiple days if not weeks.

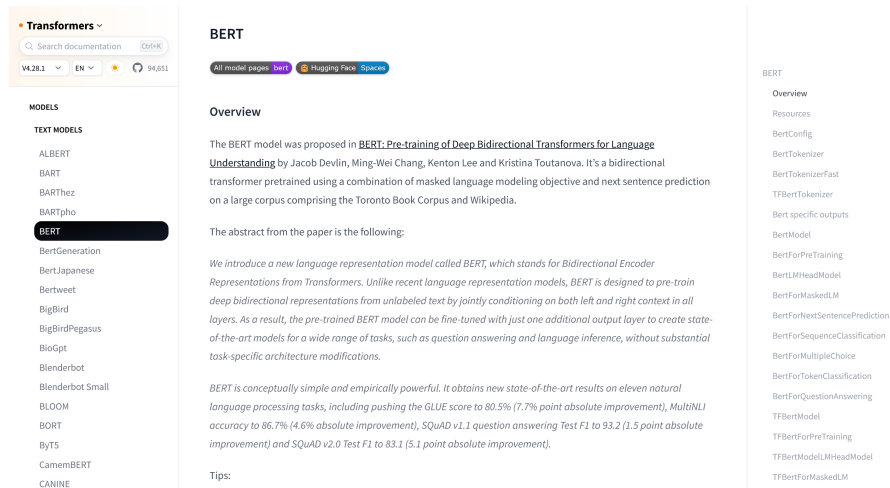


Figure 9: Huggingface BERT landing page.

6.3 Language Model using BERT

As we have seen, all youtube videos have a category tag associated with them which the video creator specifically chooses for their channel. Thus in practice this dataset can act as a hand labelled dataset for text calssification.

We try to find out whether there is any correlation between the title of a video and the category tag associated with it. If there is we will build a language model to generalize it.

For training this model, we use BERT(Bidirectional Encoder Representations from Transformers) multilingual (fig 9), which supports tokenization in 104 languages, since the nature of our data is by definition multilingual (across 10 countries) this is an appropriate model for us to use.

Our dataset contains upwards of 400k unique category labelled videos, training which requires a GPU, since we donot have access to a GPU we first tested training on CPU using an i7 12th gen processor.

For a sample dataset of only 500 videos, the training took 51 minutes to complete an epoch.

we are currently in the process of training the model on a GPU using google colab.