**ETL Project**

**Youtube Channel Statistics**



Presented by,

Devi Siva Supriya

# **Summary:**

As we all know, Youtube videos are getting popular day-by-day. There are lot of channels with videos for all ages and in all genres. As part of this project, I thought to explore the statistics of a very few of the channels and challenge my skills learned from bootcamp.

# **Datasets:**

The data for the statistics is collected from various resources. To short list the number channels to collect the statistics, channel ids are collected from the publicly available dataset & by web scrapping. They are as follows,

* DataWorld
* Kaggle dataset
* Youtube API
* Website: <https://www.noxinfluencer.com/youtube-channel-rank/top-100-all-all-youtuber-sorted-by-subs-weekly>

**Dataworld:** The dataset had the history of records of the videos uploaded. It is a huge dataset. I took 1000records from that dataset for my statistics, as CSV file.

**Kaggle dataset**: The Youtube has videos in different categories. This dataset had the JSON file with the category id mapped to the category name.

**Youtube API:** This is the main source for all other data available in internet. I used API key to collect the required statistics for the channels.

**Website**: This website is loaded with Youtube channels’ data. I took advantage of the scrapping knowledge to get the top 100 youtube channels in spotlight.

# **Database used:**

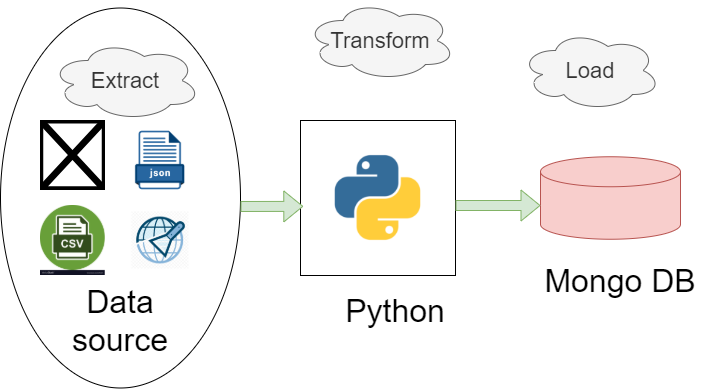
I wished to enrich my knowledge on NO SQL database. Hence, I chose **MongoDB** over POSTGRESQL [which I am familiar with]. After cleaning the data, all the details are stored as documents in collections within database name “youtube\_db”. The data is also uploaded in the cloud with database name: “youtube\_cloud\_db”.

# **Prerequisites:**

* API key in api\_keys.py
* Jupyter notebook
* Selenium
* Beautifulsoup
* Pymongo

# **Execution flow:**

* A main jupyter notebook is maintained and all the function to extract is called from the **main notebook.**
* The API call to collect details of 1000users is made from **api\_calls** notebook.
* The API call to collect details of 100 users from scrapped output is done as part of the main notebook itself since this tend to change after every scrapping.
* Later, all the loading into database/cloud is done in continuation to this from main notebook itself.



# **Libraries used:**

* import import\_ipynb
* import pandas as pd
* from pymongo import MongoClient
* from webdriver\_manager.chrome import ChromeDriverManager
* import requests
* from bs4 import BeautifulSoup as bs
* from selenium import webdriver
* import json

# **Steps followed in detail:**

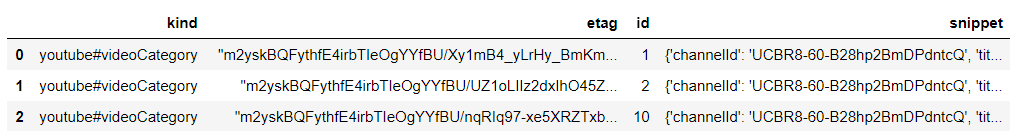
# **Extract:**

* From the dataworld datasets, records are downloaded in CSV format “Channel\_details.csv”.
* From the Kaggle datasets, “category\_id.json” with the video category details are downloaded.
* The website: <https://www.noxinfluencer.com/youtube-channel-rank/top-100-all-all-youtuber-sorted-by-subs-weekly> is scrapped to get the top 100 youtube channels and stored in dictionary.
* Youtube developers’ data is queried with API key using the URL,  
  [https://youtube.googleapis.com/youtube/v3/channels?part=snippet&part=statistics&id={id}&key={g\_key}](https://youtube.googleapis.com/youtube/v3/channels?part=snippet&part=statistics&id=%7bid%7d&key=%7bg_key%7d)

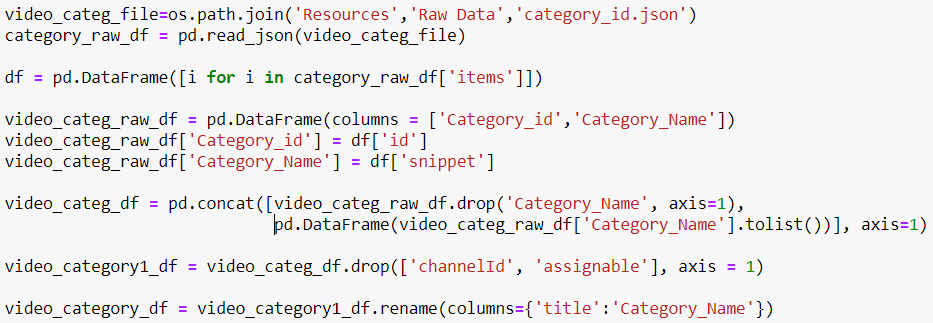
# **Tranform:**

1. The “category\_id.json” is read using pandas and converted into dataframe. The data from JSON file is like as follows. The snippet column has the category name and id is category id. Hence the remaining columns are dropped and required column is extracted.

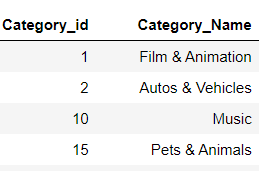
**Raw Data:**



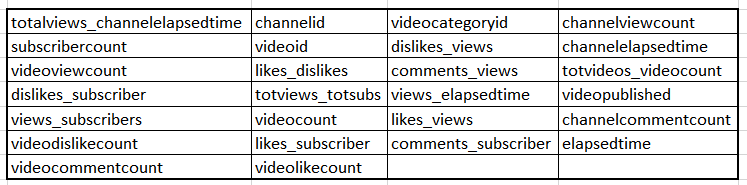
**Code**:



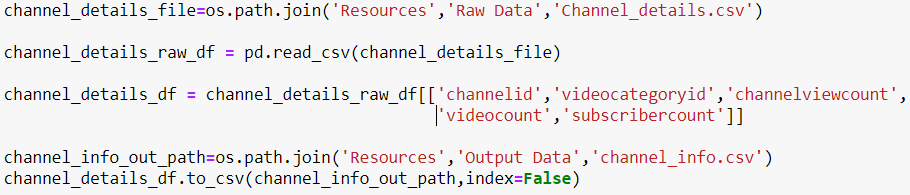
**Final output:**

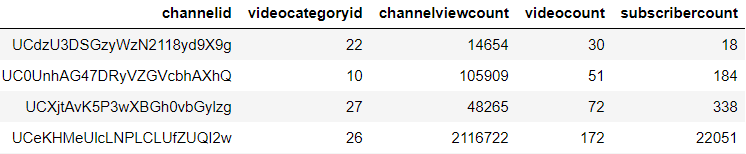
1. The “Channel\_details.csv” has 26 columns. Out of all the columns 5 columns are extracted and a dataframe is created.

**Input data:**  


**Code:**

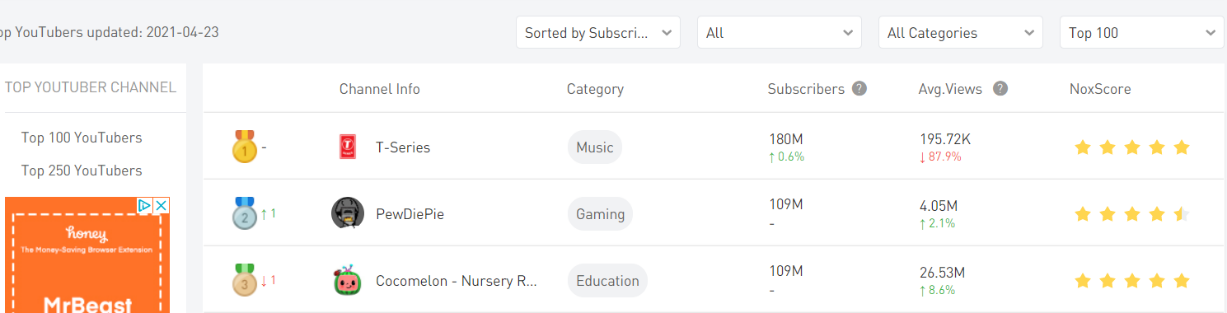


**Output data:**

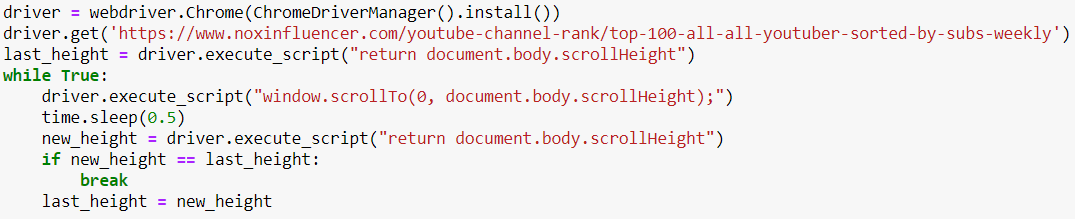
****

1. The website: <https://www.noxinfluencer.com/youtube-channel-rank/top-100-all-all-youtuber-sorted-by-subs-weekly> is scrapped and channel details are collected. Since the website infinite scrolling functionality, I used selenium driver and BeautifulSoup to scrape the website. In the final output there was space in the beginning and end of each value in the columns. They are stripped to make it data usable.

**Input data:**

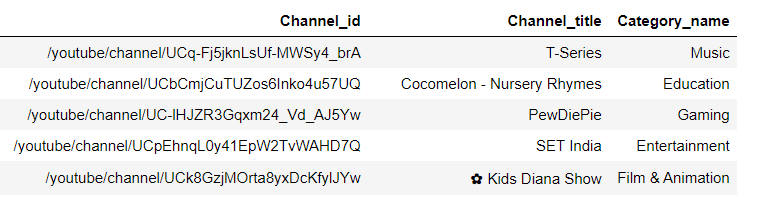


**Code**:

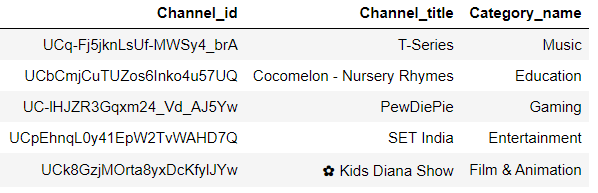




**Scrapped data:**



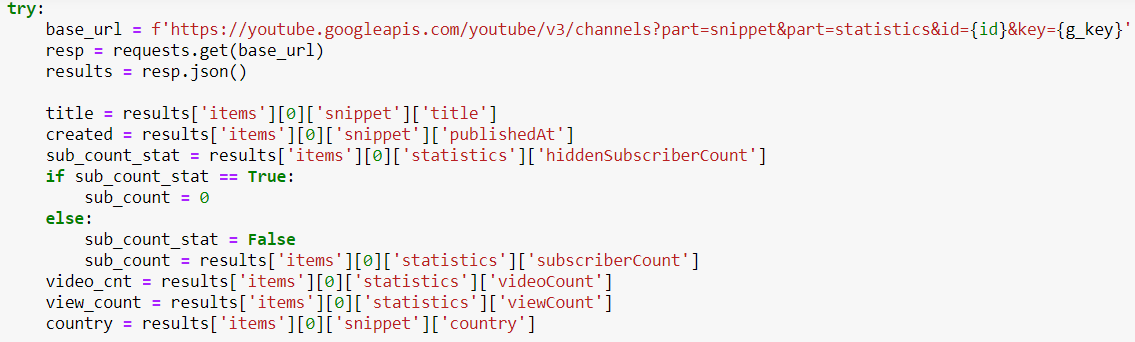
**Final output data:**



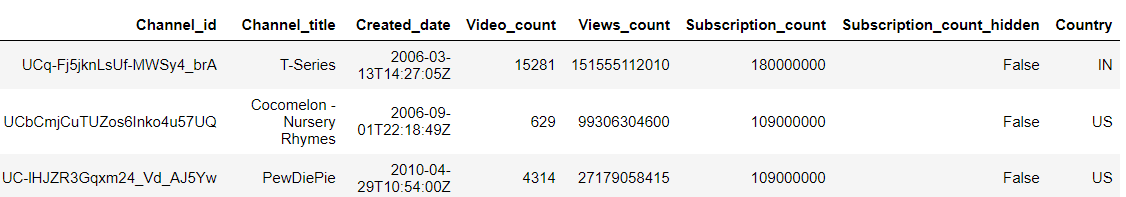
1. Now that we have the channel id from the dataset in csv file and from scrapped date, youtube API is queried and the statistics details are collected.

URL: [https://youtube.googleapis.com/youtube/v3/channels?part=snippet&part=statistics&id={id}&key={g\_key}](https://youtube.googleapis.com/youtube/v3/channels?part=snippet&part=statistics&id=%7bid%7d&key=%7bg_key%7d)

**Code:**



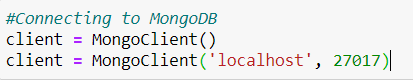
**Output data:**

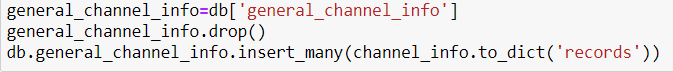
****

The response from the youtube API came across two problems. One is, for few channels subscriber count is hidden. Hence another field “hiddensubscribercount” has to be checked. In case if it is set True, I have to manually set the subscriber count to 0. Another is few of the channels are common across the countries and doesn’t have the country key. When I went through that channel information, found that they are significant ones [such as channel: WWE] and I don’t want to ignore those channels. Hence, I added those channels with country name filled as “NA” when a exception is raised with country error.

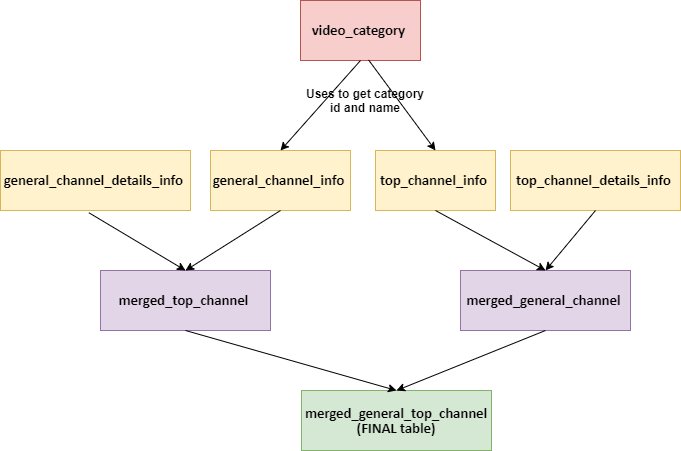
# **LOAD:**

Now with all the data ready, I decided to with MongoDB for the loading part.

* The database connection is made: 
* The database “youtube\_db” is created.
* The collections are created and inserted the data retrieved from all four sources. The collections are as follows. The sample code of creation and insertion of data.



* + general\_channel\_info – dataworld dataset
  + top\_channel\_info – website scrapping
  + video\_category – Json file from Kaggle
  + general\_channel\_details\_info – API data for the channel Id’s collected from dataset.
  + top\_channel\_details\_info – API data for the channel Id’s collected from web scrapping.
  + merged\_general\_channel – Merged raw data from dataset with data from API
  + merged\_top\_channel – Merged raw data from website with data from API
  + merged\_general\_top\_channel – Final collections with all the data collected from above.



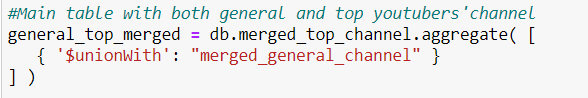
* + The data from dataworld dataset had only the category Id. With the help of the below code, update the collection “general\_channel\_info” with the category\_name. Similarly, in the data from web scrapping did not have the category id, updated the collection too.



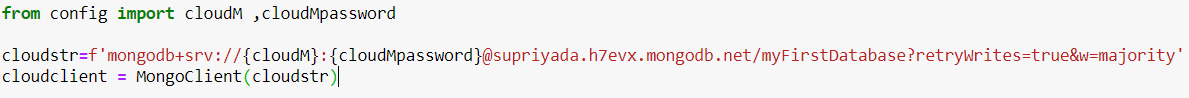
* + The data from the dataset and the data from API are merged into “merged\_general\_channel” using the $lookup(aggregation) method in MongoDB. The same method is used to merge the scrapped data and API data into “merged\_top\_channel”.

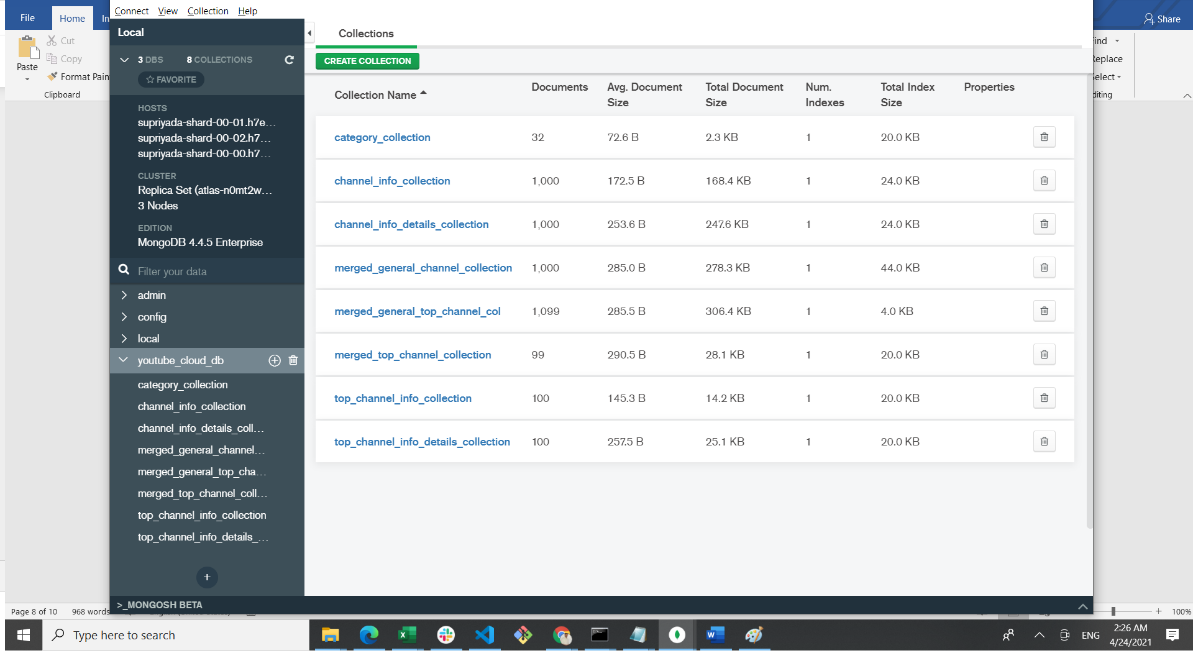


* + Now that both the data from web scrapping and dataset are similar and ready to use, with the help of $Unionwith method, combined both the data into final collection “merged\_general\_top\_channel”.



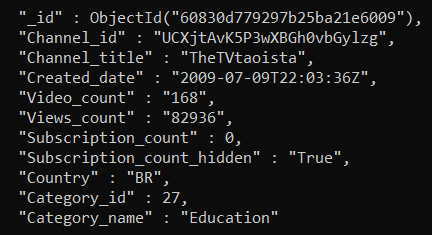
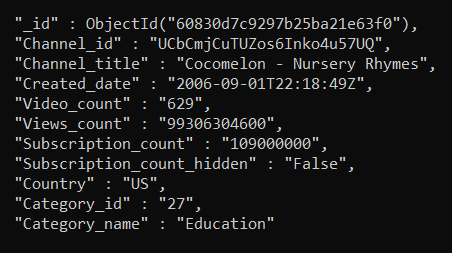
* The data is also loaded into the cloud. The URI string used is:



****

# SUMMARIZE:

The ETL is performed on the data. With the data from dataset [dataworld & Kaggle], API and web scrapping, channel Id are retrieved and with the help of that statistics of individual channel is obtained. Here is the sample of the document from the final collection. With this data I feel there can be a lots of analysis like channels with most videos/views/subscribers, category that has more videos/views, how long the channel took to reach maximum views and more interesting graphs.

Thank You for going through this document!