

ETL Final Report

Group Members : Kanwal, Janet, Amrrita

Introduction:

In the day and age of countless videos, YouTube records its top trending videos. Youtube uses a combination of factors including measuring users interactions (number of views, shares comments and likes) to determine this. The trending videos differ from the most viewed videos overall for the year. With the amount of channels on YouTube, it also begs the question of top channels, since they are the one putting out the videos – that could end up being the top trending videos. YouTube’s popularity does not seem to be dwindling, and this data could be used by those interesting to gain insights.

This report outlines the steps required to reproduce our ETL process of trending videos and top channels on YouTube .

Extract :

Sources of Data:

1) Top Videos in Canada

<https://www.kaggle.com/datasnaek/youtube-new#KRvideos.csv>

Columns present:

Columns
A video_id
A trending_date
A title
A channel_title
🔍 category_id
📅 publish_time
A tags
views
likes
dislikes
comment_count
🔍 thumbnail_link
✓ comments_disabled
✓ ratings_disabled
✓ video_error_or_removed
A description

Years : 2011 – 2018















Format: CSV

2)Top channels

<https://www.kaggle.com/babikov/youtube-channels-100000>

Columns Present:

Date : Extracted up until 9months ago.

Columns	
	category_id
	category_name
	channel_id
	country
	description
	followers
	join_date
	location
	picture_url
	profile_url
	title
	trailer_title
	trailer_url
	videos

Format : CSV

Description :

1) Import dependencies.

```
# Importing required modules
import pandas as pd
from sqlalchemy import inspect, create_engine
```

Figure 1

2) Put each CSV into a dataframe using Pandas.

```
# Location of csv documents
Cad_trending_loc = "CSV files/CAVideos.csv"
Subscribers_loc = "CSV files/channels.csv"
```

```
# Reading csv files using pandas
cad_trend = pd.read_csv(Cad_trending_loc)
subscribers = pd.read_csv(Subscribers_loc)
```

Figure 2

Transform:

Type of Transformation needed for this data: cleaning, merging.

Description:

1) Cleaned the trending video data to keep only relevant columns, which we then renamed.

```
# Cleaning Data for trending videos
cad_trend = cad_trend[["channel_title", "title", "category_id", "views", "likes", "dislikes", "comment_count"]]
cad_trend.head()
```

	channel_title	title	category_id	views	likes	dislikes	comment_count
0	EminemVEVO	Eminem - Walk On Water (Audio) ft. Beyoncé	10	17158579	787425	43420	125882
1	iDubbzTV	PLUSH - Bad Unboxing Fan Mail	23	1014651	127794	1688	13030
2	Rudy Mancuso	Racist Superman Rudy Mancuso, King Bach & Le...	23	3191434	146035	5339	8181
3	nigahiga	I Dare You: GOING BALD!?	24	2095828	132239	1989	17518
4	Ed Sheeran	Ed Sheeran - Perfect (Official Music Video)	10	33523622	1634130	21082	85067

```
# Changing column names
cad_trend.rename(columns = {"channel_title": "Channel Name", "title": "Video Name", "category_id": "Category ID", \
                           "views": "Views", "likes": "Likes", "dislikes": "Dislikes", "comment_count": \
                           "Number of Comments"}, inplace = True)
cad_trend.head()
```

	Channel Name	Video Name	Category ID	Views	Likes	Dislikes	Number of Comments
0	EminemVEVO	Eminem - Walk On Water (Audio) ft. Beyoncé	10	17158579	787425	43420	125882
1	iDubbzTV	PLUSH - Bad Unboxing Fan Mail	23	1014651	127794	1688	13030
2	Rudy Mancuso	Racist Superman Rudy Mancuso, King Bach & Le...	23	3191434	146035	5339	8181
3	nigahiga	I Dare You: GOING BALD!?	24	2095828	132239	1989	17518
4	Ed Sheeran	Ed Sheeran - Perfect (Official Music Video)	10	33523622	1634130	21082	85067

Figure 3

2) We then did the same for the subscriber data.

```
# Cleaning Data for subscribers videos
subscribers = subscribers[["title", "category_name", "followers", "videos"]]
subscribers.head()
```

	title	category_name	followers	videos
0	PewDiePie	Entertainment	69896406	3649
1	T-Series	Music	69471946	12820
2	Justin Bieber	Entertainment	41858494	132
3	5-Minute Crafts	Howto & Style	40474509	2350
4	WWE	Sports	36301947	37928

```
# Changing column names
subscribers.rename(columns = {"title": "Channel Name", "category_name": "Category Name", \
                             "followers": "Followers", "videos": "Number of Videos"}, inplace = True)
subscribers.head()
```

	Channel Name	Category Name	Followers	Number of Videos
0	PewDiePie	Entertainment	69896406	3649
1	T-Series	Music	69471946	12820
2	Justin Bieber	Entertainment	41858494	132
3	5-Minute Crafts	Howto & Style	40474509	2350
4	WWE	Sports	36301947	37928

Figure 4

3) We did a `dropna` function prior to merging both tables to check blank entries, NaNs and ensure accurate and correct merging. The format of data in both tables matched, and hence did not require data formatting to standardize the two.

4) Then merged both tables on Channel Name to create a merged table called Youtube.

```
# Merging documents
Youtube = pd.merge(cad_trend, subscribers, on = "Channel Name")
Youtube.head()
```

	Channel Name	Video Name	Category ID	Views	Likes	Dislikes	Number of Comments	Category Name	Followers	Number of Videos
0	EminemVEVO	Eminem - Walk On Water (Audio) ft. Beyoncé	10	17158579	787425	43420	125882	Music	26572348	82
1	EminemVEVO	Eminem - Walk On Water (Audio) ft. Beyoncé	10	20539417	840642	47715	124236	Music	26572348	82
2	EminemVEVO	Eminem - Walk On Water (Audio) ft. Beyoncé	10	22702386	869304	50018	123235	Music	26572348	82
3	EminemVEVO	Eminem - Walk On Water (Audio) ft. Beyoncé	10	24578152	891283	51978	125444	Music	26572348	82
4	EminemVEVO	Walk On Water/Stan/Love The Way You Lie (Medle...	10	1535199	153770	2439	16875	Music	26572348	82

Figure 5

- 5) We did not utilize any aggregate functions here since we are not analyzing the data or filtering for a limiting (maximum, minimum, or other) date or factor since the data from both the tables already works

Load

Final tables/ collections used in production database:

Merged table named Youtube.

Description:

- 1) Open PG Admin (We use PostgreSQL here since our data was merged prior, and did not have NaNs, or missing fields)

```
# Location of csv documents
Cad_trending_loc = "CSV files/CAvideos.csv"
Subscribers_loc = "CSV files/channels.csv"
```

```
# Reading csv files using pandas
cad_trend = pd.read_csv(Cad_trending_loc)
subscribers = pd.read_csv(Subscribers_loc)
```

Figure 6

- 2) Create a new Database . Open Query Tool to create :

```
CREATE TABLE youtube (
    "Channel Name" TEXT,
    "Video Name" TEXT,
    "Category ID" INT,
    "Views" INT,
    "Likes" INT,
    "Dislikes" INT,
    "Number of Comments" INT,
    "Category Name" TEXT,
    "Followers" INT,
    "Number of Videos" INT
)
```

3) Create a connection to database in PostgreSQL.

```
# Connect to Local data base
rds_connection_string = "postgres:postgres@localhost:5432/youtube_db"
engine = create_engine(f'postgresql://{rds_connection_string}')

# Engine table name
engine.table_names()

['youtube']
```

Figure 7

4) Check for a successful connection to the database and confirm that the tables have been created using engine. Confirm successful Load by querying database.

```
# Use pandas to Load database into pgAdmin server
Youtube.to_sql(name='youtube', con=engine, if_exists='append', index=False)

# Making sure data is imported
pd.read_sql_query('select * from youtube', con=engine).head()
```

	Channel Name	Video Name	Category ID	Views	Likes	Dislikes	Number of Comments	Category Name	Followers	Number of Videos
0	EminemVEVO	Eminem - Walk On Water (Audio) ft. Beyoncé	10	17158579	787425	43420	125882	Music	26572348	82
1	EminemVEVO	Eminem - Walk On Water (Audio) ft. Beyoncé	10	20539417	840642	47715	124236	Music	26572348	82
2	EminemVEVO	Eminem - Walk On Water (Audio) ft. Beyoncé	10	22702386	869304	50018	123235	Music	26572348	82
3	EminemVEVO	Eminem - Walk On Water (Audio) ft. Beyoncé	10	24578152	891283	51978	125444	Music	26572348	82
4	EminemVEVO	Walk On Water/Stan/Love The Way You Lie (Medle...	10	1535199	153770	2439	16875	Music	26572348	82

Figure 8

Inspiration/ What Data Could Be Used For:

- If someone were to further inspect, and perhaps get the number of most trending videos by category name, a group by would do this.
- Additionally, videos could be categorized on their comments.
- To gauge pattern of viewership toward popular categories

Further cleaning or modifying could allow one to get answers to questions like:

- Which channels provided the most number of trending videos?
- How are views, likes, dislikes, comment count, title length, and other attributes correlate with (relate to) each other? How are they connected?
- Which video category (e.g. Entertainment, Gaming, Comedy, etc.) has the largest number of trending videos?