**ETL and Youtube**

ETL Project - Predictions by Youtube trending videos. Members: Toby Wong, David Bartlett, Levar McKnight, Jerry Youngblood

# Proposal:

A social media marketing division of Team Barret is tasked with spreading their advertisements to the correct target audience from across the world. Their team of analysts have decided to study the data on Youtube’s trending video page to see what is currently most popular.

The team of analysts wants to ensure that the data they collect are from different sources so they can make proper predictions on upcoming trends.

# 

**ETL (Extract, Transform, Load) Summary for YouTube Trending Data**

**David Bartlett**

**Extract:**

Source for .csv and .json files:

*Mitchell, J (2017). Trending YouTube Video Statistics. From kaggle.com:* [*https://www.kaggle.com/datasets/datasnaek/youtube-new*](https://www.kaggle.com/datasets/datasnaek/youtube-new)

The .csv file contains data relating to trending YouTube videos such as: like/dislike count, view counts, upload dates, trending date, category, etc for the United States. Where the .json contains the category titles and id’s which correspond with the category\_id column in the .csv.

**Transform:**

For the transforming, the .csv was pulled into a jupyter notebook and converted to a DataFrame using Pandas. From there, unnecessary columns were dropped and the publish\_time column was changed to a datetime format.

As stated previously, the category\_id column only contained numbers that acted as a key for corresponding category titles located in a separate .json file. To change these numbers to the titles, the json file was brought into the notebook and put into a list. Using a for loop, the tuples of the id number and matching category title were pulled into a new list, then into a dictionary with the id numbers as the keys. The category\_id column type was changed from integer to string since the numbers are being replaced by the title strings. Then a loop was used with the .loc function to loop through the DataFrame and replace the numbers with the titles. The finished DataFrame was then put into a final dictionary for loading to MongoDB.

**Load:**

To load the data, a connection was established to the Mongo Client, a reference to the database – named youtubeTrending – and collection – named youtube – was made, and the final dictionary was inserted into the collection. Finally, a query was done with .find() to see the entries, which showed the load to the database was successful.

**YouTube Videos - Ads and Trailers / ETL Summary / Jerry Youngblood**

**Extract:**

For data on YouTube ad videos and movie trailers, the following dataset was used:

*Ay, Y. E. (2020). Most Popular Youtube Ads. From kaggle.com: https://www.kaggle.com/datasets/yamaerenay/most-popular-youtube-ads-20172020*

This dataset was formatted as a JSON file containing several elements about YouTube video ads and movie trailers for the years 2017-2020. The file was structured as a list of 57 dictionaries, each with three key-value pairs followed by a fourth that was paired with a list of nested dictionaries. The file was read into a Jupyter notebook, and from there the file data was extracted. Data for the year 2020 was in focus for this project, considering that year being one when people were in their homes for longer periods of time and accordingly when viewership of videos and movies increased.

**Transform:**

With the ads dataset read into a Jupyter notebook, an initial dataframe was assembled to view more of the data, and from that was observed four subsets of data specific to the year 2020. Of those four, one was for the second quarter only of 2020 and was therefore excluded. For each of the three remaining subsets, the data wrangling ensued through cleaning, filtering, and dis-aggregating. For each subset of data, a dataframe was assembled by normalizing the JSON data, then refining the dataframe to reduce it to its most salient elements. The final dataframe in each sub-set allowed for observations to be made across three areas for 2020. First was a top 10 ranking of YouTube ads overall. Second was a top 10 ranking of public service announcement ads (PSAs). And third was a ranking of the top 9 movie trailers contemporaneous with the Oscars. Following the final dataframe for each sub-set, observations were made in a markdown cell about the top ads and movie trailers according to their respective rankings, brands and titles, and the creative and media agencies involved in ad/trailer creation and distribution.

**Load:**

Each subset of transformed data for the top rankings of ads, PSAs, and movie trailers was subsequently loaded into a non-relational database. After observations were drawn from each data subset, its final dataframe was converted to a dictionary and loaded to a corresponding collection in MongoDB Compass. Confirmation of the load was demonstrated via the find operator, and a pprint of the contents accordingly produced. The name of the database is, “youtube\_videos,” and the names of its collections are: “psa\_top10\_2020,” “top10\_ads\_2020,” and “top\_trailers\_2020.”

ETL/ Toby Wong

Extract:

<https://www.kaggle.com/datasets/advaypatil/youtube-statistics>

<https://www.kaggle.com/datasets/rsrishav/youtube-trending-video-dataset>

The Datasets are in JSON and CSV file formats where the CSV file contains comments, likes, and views that are tied to specific keywords and the JSON contains category IDs that correspond with the keywords.

Transform:

Looking at how the data can translate into understanding what videos are popular and how the views and likes can show the trends on youtube. Taking the data and dropping extra information such as the VideoID, and Unnamed column headers to make it easier to understand.

Load:

Since we are using MongoDB, I created a connection to mongoDB and referenced the database before querying using pymongo.

**Extract, Transform, and Load Summary of YouTube’s Channel Dataset**

**Levar McKnight**

Reference:

Gupta , Harshit H. “**YouTube's Channels Dataset.” *Kaggle*, 9 December 2023,**

[**https://www.kaggle.com/datasets/harshithgupta/youtubes-channels-dataset**](https://www.kaggle.com/datasets/harshithgupta/youtubes-channels-dataset)**.**

Extract:

The dataset that was gathered from Kaggle.com was last updated three years ago. It is a JSON collection of the top trending videos on YouTube from mostly the previous decade.

Transform:

Per our proposal, I decided to use views as a factor to determine how to place our advertisements. To simplify this process, I dropped columns that I felt were irrelevant to view counts. For example, columns that involved comments were excluded because not all viewers leave comments. Columns such as view counts and video categories were left in order to make it easier to determine what type of videos attracted the most viewers.

Load:

After the dataset was cleaned, it was loaded into MongoDB. Due to the size of the dataset, a query was not done in Jupyter. Instead, successful loading was confirmed by checking the collection in MongoDB. The cleaned up database can be later analyzed to determine where to place our advertisements for maximum viewership.