# Overarching

Imports uses common build for consistency and ability to quickly create and conform new notebooks

Scope: This analysis is restricted to videos tagged by YouTube as Music.

# ETL

General: <https://developers.google.com/youtube/v3/docs/>

Categories: <https://gist.github.com/dgp/1b24bf2961521bd75d6c>

This stage is built using the YouTube Data API, v3.

Most YouTube API features are targeted at uploaders, and channel owners in particular. Many features, including the majority of Analytics, are restricted to the owner of a video or channel.

In general, the API references a custom function (build) to construct the query along with authorization. The build type can be stored. Search methods are then appended to this to return information.

I restricted my search to category 10: music. This is a tag that is assigned by YouTube at upload

* search().list
  + Takes some kind of search term as primary input
  + works like typing a query into the youtube search bar, returning a json file that has high-level summary information
  + https://developers.google.com/youtube/v3/docs/search/list
* videos().list
  + Takes unique video id as input
  + gets summary statistics of specific video, along with detailed

Token Costs

* Search: List - 100 tokens

The main search was executed by searching relevant videos for each letter of the alphabet to get as wide a cross reference as reasonable. Each letter returned about 320 results.

Beyond down selecting data at the query, some additional cleaning was done in cleanup. When the model was first done, I had to break halfway through the clean up, so the working model was saved.

Next came interpreting. For example, duration is stored as text in the format PT #H #M #S, with 0 values not stored. For example, 2 hours 23 minutes is PT2H23M, and 1 hour 1 second is PT1H1M.

Some basic feature engineering was done here as well, including adding some integer value date time features, and the conversion of duration, licensedContent, and caption to integers:

* Definition (video quality): 0 for standard def, 1 for high def
* licensed Content (whether the content is claimed under copyright): 0 for none, 1 for yes
* caption (has video caption to display in query): 0 for false, 1 for true

Dropped some unnecessary items

* categoryID: category10 was stored for all videos
* channelID: outside of scope
* dimension: whether video was 2D or 3D. Only 3 or 4 HD videos existed in the set

Note: This process was repeated when pulling new videos to test the model against.

Not included, unfortunately:

Time series data is heavily restricted. For example, comment and view count by date would have been interesting. Google trends could have served as an approximation, but querying this programmatically requires establishing a specific login protocol. Loosely speaking, it is worth noting that most videos with much more than a million views appear to get the majority of their views in the first month or so after publishing.

# Exploratory Data Analysis

## Part 1

## Part 2

after initial modelling pass

# Feature Engineering

# Model