# 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.

## Beginners Guide to YouTube

YouTube videos have a number of features. The title text is a short description of the video, and for music videos often (but not always) follows a format like Song - Artist, on average about 7 words. The video has tags to help it appear in searches, and typically consist of genre-like information such as trap, rap, pop, etc, with 14 tags as average. The description text contains more detailed information in theory, although commonly this has links to the artists page on facebook, soundcloud, itunes, spotify or other promotional material. Descriptions can be quite wordy, reaching nearly 100 words on average.

# 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

1. 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:

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

Dropped some unnecessary items

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

Duplicates (by vidId) are also dropped. Given the very loose method of searching for videos, I was pleasantly surprised to see that out of about 10,000 records scraped, about 8600 unique videos remained.

As a note, tags were concatenated into a single string with a comma to separate them. This makes it more clear later on when counting the total number of tags

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.

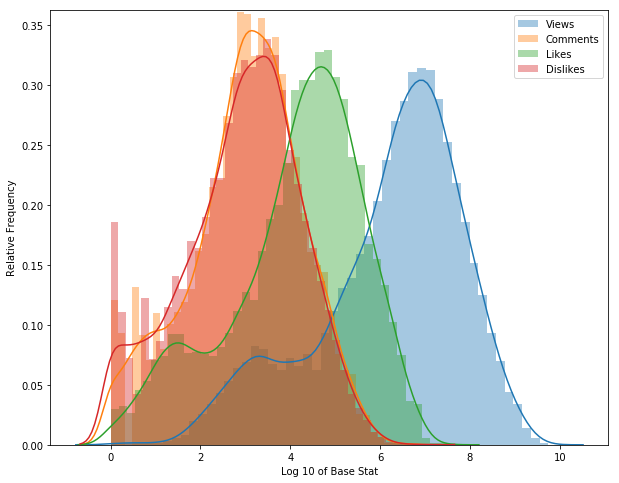
# Exploratory Data Analysis

## Part 1

Some basic facts

1. Mean / Median views: 55 M / 3 M

The very first thing I noticed about the data at this point is the spread along orders of magnitude. Since part of the goal of this project involves predicting whether a video is a hit or not, decided to transform the data by taking the base 10 log of the summary statistics to make it easier to interpret results (than using square root or natural log, for example). The next chart is striking.



Some basic info here:

|  |  |  |  |
| --- | --- | --- | --- |
| **Log** | Mode | Bump | Mode / Bump |
| Views | 6.96 | 3.25 | 2.14 |
| Likes | 4.40 | 1.60 | 2.75 |
| Dislikes | 2.94 |  |  |
| Comments | 3.13 |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Log Ratio | Views | Likes | Dislikes | Comments |
| Views |  | 1.58 | 2.37 | 2.22 |
| Likes | 0.63 |  | 1.50 | 1.41 |
| Dislikes | 0.42 | 0.67 |  | 0.94 |
| Comments | 0.45 | 0.71 | 1.06 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Actual** | Mode | Bump | Mode / Bump |
| Views | 9.08E+06 | 1.78E+03 | 5105 |
| Likes | 2.53E+04 | 3.98E+01 | 635 |
| Dislikes | 8.71E+02 |  |  |
| Comments | 1.34E+03 |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Views | Likes | Dislikes | Comments |
| Views |  | 3.6E+02 | 1.0E+04 | 6.8E+03 |
| Likes | 0.28% |  | 2.9E+01 | 1.9E+01 |
| Dislikes | 0.01% | 3.44% |  | 6.5E-01 |
| Comments | 0.01% | 5.31% | 154.17% |  |

Key observations:

1. Log likes are a pretty direct function of the view count (63% log to log, or .3% in actual terms)
2. The comment count is 50% higher than the dislike count
3. Comment / dislikes can be described almost completely by likes, which can be described by views (in log space)
4. The Like to View ratio tapers off as views increase because it is described in log space

For example, let’s look at Venom, by Eminem and Despacito, by Louis Fonzi (as of 10/18/18).

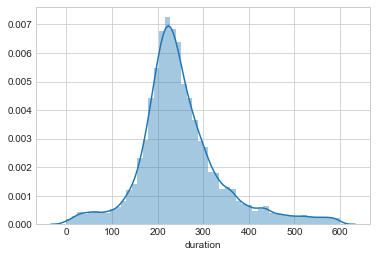
The stats look like this:

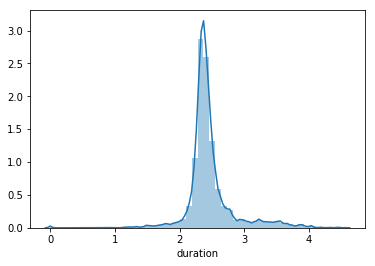
1. Venom - Views: 80 M, Likes: 2.9 M, Dislikes: 100k , Comments: 185 k
2. Despacito - Views: 5.6 B, Likes: 30M , Dislikes: 3.5 M, Comments: 2.6 M

Despacito has 70 times as many views as Venom, but only 10 times as many likes. In both cases dislikes are within 50% of the comments (it looks like a lot more people dislike Despacito than we would expect - more on that later).

Eminem was set up for failure, however. He could have tripled his views if he had a featuring in artist. Videos with some variant of ‘featuring’ in the title have an average of 129 M, compared to 55 M overall and 43 M for videos without a featuring artist. This works out to an average multiplier of almost exactly 3, which I dub the Pitbull effect. Pitbull has made a career out of being featured in other peoples songs, is incredibly crisp for some reason, and no one is totally sure why it works but it does.

Note: This affect is actually probably larger since I’m not sure exactly how it works in the Spanish language world. Again, like Pitbull.

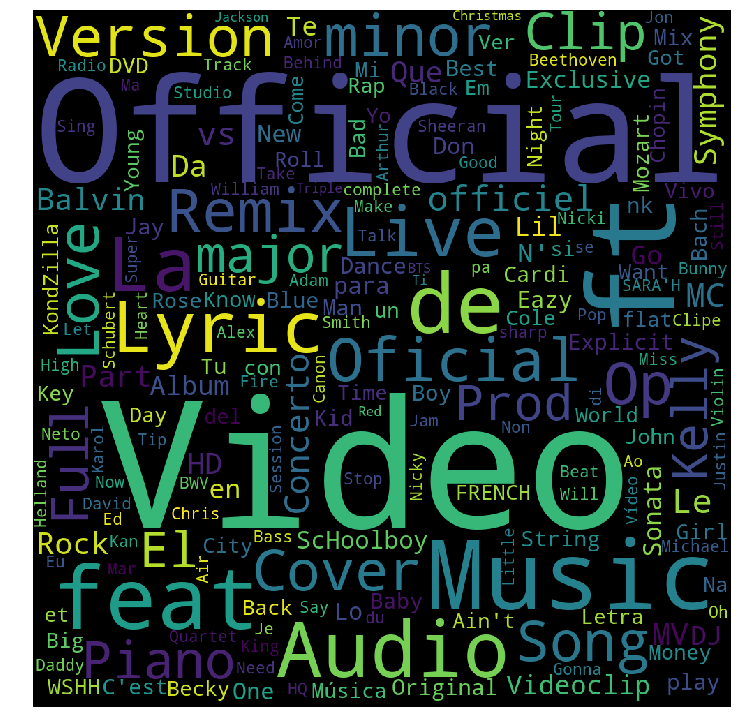




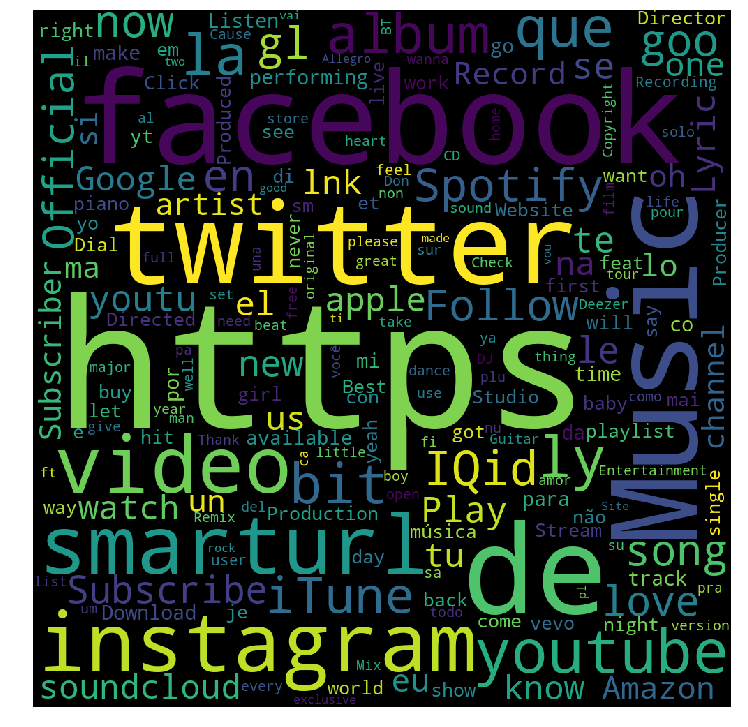
Way more videos over 10 minutes (~3 on the log scale) than expected. Turns out people like to have their study jams on YouTube. 5% of all videos (about 400 total in this set) are over 20 minutes long.

## Some Text Stuff: Title vs. Tags vs. Description

A YouTube video can be posted with 3 text categories, and I wanted to know if there’s any real difference. Lets look.







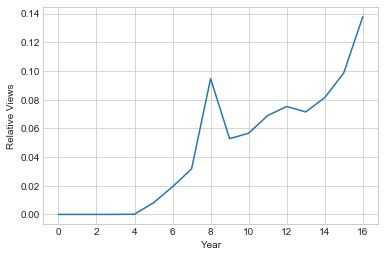
In brief, the title gives a broad description of the video category (lyrics, live, official, cover),

tags contain helpful genre or content information, and the description is full of self-promotional trash.

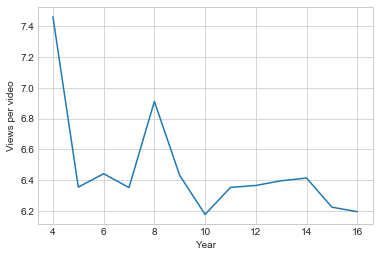
(J Balvin and people named Kelly are popular, love songs as always)

Overall, the sentiment score is pretty similar on average between these, and is unsurprisingly positive. The word bag used for this did not include Spanish, so it may not be totally accurate, although the score should still be informative in relative terms.

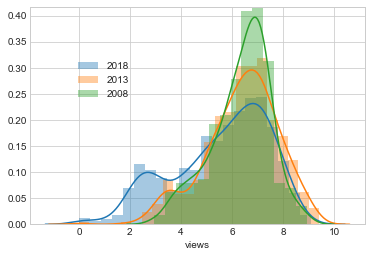
## Time Series



Relative views per year. Mainly reflects increase in videos uploaded per year.



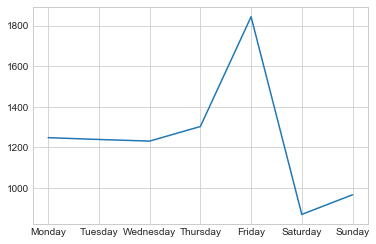
2008 - the golden era. Interestingly, average views per video has stayed above comfortably above 1 million despite the increase in total videos uploaded per year over time. Mainly, this is due to the videos in the set being more broadly distributed, and having more low end answers:



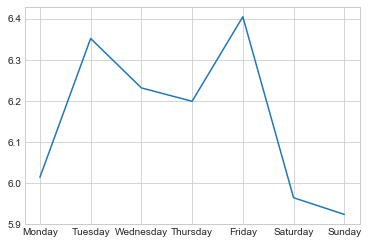
The cause of the low end bump is more apparent here. The search query method used to scrape the videos was far from perfect, as I was looking for individual letters by relevance. I believe YouTube uses views and upload date when determining relevance, and so the low count videos generally have some single letter in the title somewhere and are more recent. For example:

|  |  |
| --- | --- |
| **Views** | **Title** |
| 240 | B tone - LEAVE STORY Official Video1 |
| 159 | WHY HOLD BACK - Trunk Boiz member B\*Janky |
| 112 | Jonathan C. Meier - Savage (Audio) |
| 58 | HeartLand Didgeridoo - key of C - (4701) |
| 157 | Genomineerden gouden C 2017 |
| 26 | DELAIN /SING TO ME /c/ MARCO HIETALA 6/ en MAL... |
| 87 | Tram,Bix and Lang - For no reason at all in C ... |
| 270 | Catalogue d'Emojis Teaser |

Finally, there appears to be a preponderance of videos released on Fridays:



And a drop-off for views for videos released on the weekend:

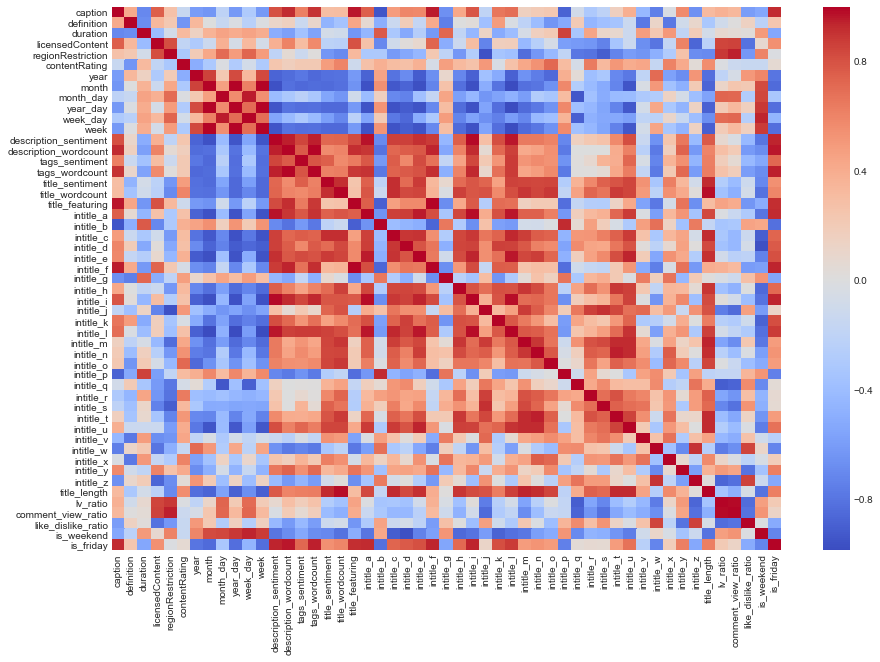


Apparently, either people like to have social lives or else no one is in the Vevo offices on the weekend.

## Part 2

In this phase, I looked at all my features to see which ones might be providing confusing statistics to the model. Refer to the Excel file in /background/supplemental for the full table. Basically, I went through all the variables to see which had distinctive means between each view class to see which might have real explanatory power.

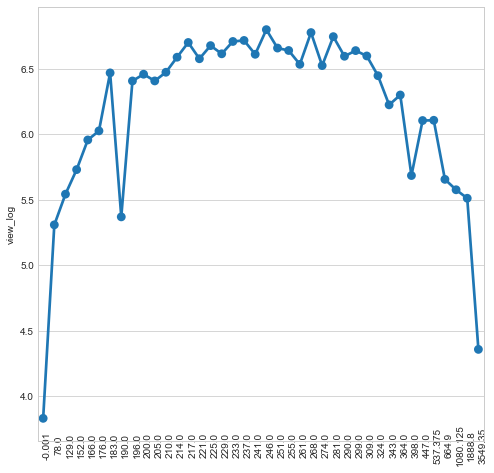
In addition, after doing some down select, I Iooked at cross correlation to see if there was some overlap.



Some items of interest here are the correlation of a lot of variables over time, presumably because people interact with youtube differently now. For example, title lengths have gotten shorter, presumably as more people learned how to use tags. Similarly, sentiment goes with length, since sentiment is calculated as the sum of the sentiment of each individual word.

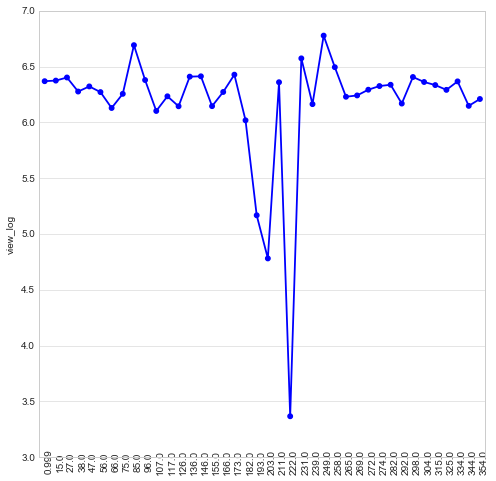
This suggests we need to focus on models robust to feature correlation, as well as removing most of the time series and some of the text analysis features.

### Some brief additional conclusions



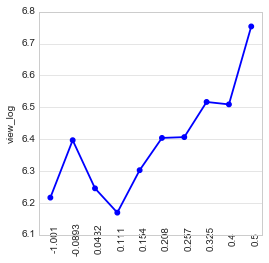
Looking at view count around the center of the duration distribution, most videos in the traditional radio-friendly length range of 2.5 - 5 minutes had average views in the 10s of millions.

Conversely, YouTube hits appear to reject the summer hit trope of radio play:

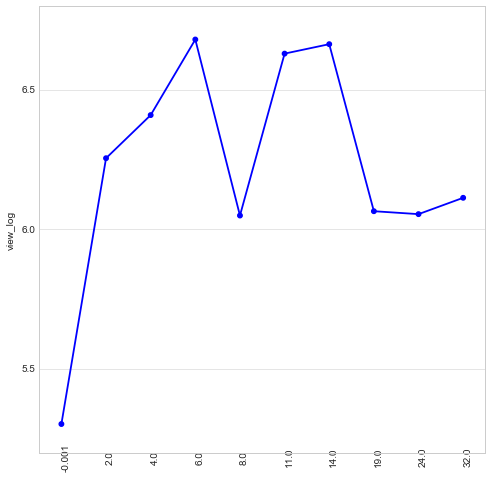


There’s a pretty distinctive dropoff for videos released in mid-late summer. Presumably, these are students working on passion projects. The notable exception is KPop, which has higher engagement in general.

Having a more positive tone in the description generally helps:



But adding a large number of tags doesn’t:



Anything more than 5 keywords seems to just confuse people.

# Feature Engineering

## Measures

These are basically potential regression targets, and are measures of engagement with the video.

I created one additional feature, view\_class, which represents the order of magnitude of views. This is done by rounding the the views to the nearest whole log10. Its possible that this could have been re-done at some point. For example, a 30 million and 250 million count video end up in the same category this way. I felt that it was best to leave it as is since the data behaves so nicely like this, and it is easy to quickly interpret groups.

## Text

### Description and Tags

Description and tags are handled similarly. The sentiment of each was taken. Given the nature of the scorer, this will tend to make it possible for longer posts to score higher on polarity on average. I left this in since I felt it more closely reflected how I believe the search algorithm works and to capture the effect of spamming words. Additionally, I feel that having more words gives an opportunity for more words to cancel each other out. Given that descriptions are longer than tag strings, and that description sentiment was ultimately a better predictor, this decision seems to be better than taking sentiment per length or some similar aggregation.

Next, I created text vectors for each uses sklearn’s CountVectorizer. With 8,500 records, these vectors quickly became un-managebly large for processing on a local machine. Instead, I focused on words that appeared more than 200 times after lemmatization (a number picked to get a reasonably large number of results without crashing anything). This should hopefully counteract overfitting since any one word won’t be too rare relative to a training split.

I did spin up an AWS instance to attempt to get a bigger machine crunching on the text with mixed results. The ultimate inclusion of that model is pending as of 10/18/18.

### Title

Similar to Description and Tags, overall title length and polarity are scored. In addition, I added a column for the Pitbull effect to detect whether a video had a featuring artist.

In the spirit of data mining, I wanted to see if certain kinds of words were more appealing to people since I did not have access to a full text vector. I also wanted to avoid overfitting on ‘Ed Shereen’ or ‘Cardi B’ as predictors. In that spirit, I added a column for the occurrence of each letter of the alphabet (ultimately removing most after EDA). A few did stand out - the letter f was correlated at .13 with views.

## Ratios

Here’s the meat. I added ratios for each feature, computed after taken the log. The like / view ratio was quite strong, correlated with view log at .69. In addition, I got rid of some non-sensical results where videos with 0 views had multiple likes or comments. Apparently, YouTube wipes views if detects spam, but does not exclude it from search results.

## Output

In addition to including a list of features at the end of this notebook, I also grouped them thematically:

1. Time (year, month, is weekend etc.)
2. Text (sentiment and length measures)
3. Title letters (count of each letter in title)
4. Ratios
5. Other (booleans and duration)
6. Measures (likes, views, log views etc.)
7. Non-model columns (non-quantitative information)

# Model

## Ratio explanatory power

Without discussing anything else, the ratios almost completely describe the number of views. Without any other features and without any kind of tuning, a model built on this had r squared around .87. Adding the other features adds only .03 to the regression score. Upon further analysis, it became clear that the ratios were really a function of the view count. In general, the more popular a video, the more likely people are to interact with it. The final nail in the coffin for including these features is that they’re useless for planning. I can make a video that lasts exactly 4 minutes, but I can’t manufacture a video with a set like to view ratio.

## Unpopular videos, low stats

As I went through the data during EDA, it became clear that videos with less than a 1000 views had a lot in common with high view count videos. The first pass with models had the r squared around .2. Removing videos less than 500 views (only about 500 in total) nearly doubled this to .4. GradientBoost and RandomForrest appeared to be the strongest models, presumably due to non-linear relationships like duration and the correlation of a number of text features with year. This realization helped kick off my suspicion of the like to view ratio, since a lot of very small videos had high ratios.

## Confirming Feature Engineering

During feature selection, I was able to eliminate a number of otherwise tempting features: 'year\_day', 'month', 'tags\_sentiment','title\_sentiment', 'title\_wordcount'. Taking these out of the model either slightly improved or had no effect on performance.

## Type By Type Discussion

Across the board, PCA did not improve performance, so I guess I did a good job with eliminating stuff during feature engineering.

Gradient boost scored most consistently high across different versions of the data.

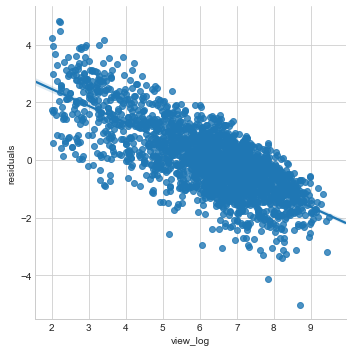
The neural net was finicky, and responded erratically to different hyper-paramaters. Presumably, more features and samples are needed to get this to really work.

Random Forrest was almost as good as Gradient Boost, and responded quite well to grid searching. The un-gridsearch model was the best a identifying high viewcount videos.

Bagging GradientBoost did not appear to help too much, which leads me to believe that the top score may be a function of the random state to a small degree.

Interestingly, Maroon 5’s Sugar confounded models the most. It sits at 2.7 B views, but all the models firmly agreed it was a 30 million view video at most.

Looking at the residuals,



The models clearly had a hard time identifying the ‘x-factor’ that makes something a hit. Sure enough, the videos it underpredicted the most had high like to view ratios.

# So what *can* we use this information for?

## Forcing some signal

Here’s what I do know: some features do provide some information about views. I used the feature importance from Gradient Boost to identify stronger predictors, and linked it up to the coefficients of a basic linear Ridge model to get a sense of direction. To that end, here are some potentially actionable insights, ranked by importance:

1. Duration - shorter is better
2. Year - older videos have higher total views, probably due to a combination of 1. more time to accumulate views 2. Feature of the ETL phase which probably failed to retrieve old videos with lower view counts
3. Day of year: release earlier in the year, although recalling the midsummer dip, it is probably safe to say winter videos do better
4. Longer descriptions with more positive tone do better
5. The number of tags is much more important than their sentiment
6. Longer titles do not do well
7. Licensed content is more viewed
8. Including a caption seems to help visibility
9. Use the letter ‘a’ in the title a bunch, but not ‘p’
10. Content cool enough to be prohibited in certain regions is more popular

## Another thought…

I set out with a side goal in mind - can I identify cool new songs as they are released? To the end, I made a second pull on the data, and isolated songs uploaded in the weeks after I initially pulled in the data. Looking at videos, a lot of those that are predicted over a million are already at that value.

One thing this effort underscored was a big lack of an import feature of the videos - when did the song included actually come out, vs. when was it uploaded? The model is fairly accurate at picking out views for newer music. In fact, it scored slightly higher on the new data than on the original test data. However, it starts to fall down when presented with older songs and obscure uploads (especially those from foreign countries).