# Capstone Project 1: Milestone Report

# Youtube View Count Prediction

**Introduction:**

All businesses want to offer the right product, at the right place, right time, and at the right price. In order to increase their chances, they have to optimize their marketing strategy and increase their brand awareness and online presence. The top three things any S&P 500 company will focus on to reach their customers online:

1. Their website – Fundamental information customers need to see
2. Search engine optimization –find that website
3. Social media- Facebook, Twitter, Youtube, etc. – builds engagement and awareness

This project will focus on one particular channel of number 3 – advertising on Youtube videos to answer the question of how many potential customers can be reached by purchasing advertisements on a particular video. A ranking system was created in order to classify videos into categories based on their current level or ability to become viral.

| Rank | # of Views | # of Days Since Creation |
|---|---|---|
| White Belt | 20,000 | 300 |
| Blue Belt | 100,000 | 400 |
| Purple Belt | 500,000 | 450 |
| Brown Belt | 2,000,000 | 475 |
| Black Belt | 10,000,000 | 500 |

Since predicting view counts could potentially have multiple applications or use cases, we can look at the problem statement from both the perspective of a potential Youtube customer and also from Youtube themselves. When defining a problem statement, it's useful to see several dimensions towards how the solution can solve more than one type of problem.

**Problem Statement from Marketer Perspective:**

1. Which youtube video related to my business will achieve greater than 20,000 view counts within 10 months based on title, category ID, description, tags, and other video features?
2. Which youtube videos are going to exceed 10 million views within 17 months (viral video) based on title, category ID, description, tags, and other video features?

The answer to these questions could assist a marketer in making a decision regarding which type of Youtube advertisement to purchase such as pay-per-click, or non-skippable, etc. As an example, on higher view count videos, perhaps a pay-per-click model is more optimal.

**Problem Statement from Youtube Perspective:**

1. Which youtube video should we consider charging an premium for advertisement?
2. How can we optimize the pricing system to charge more for viral videos vs. low rank videos?

Note: All of the requirements in the problem statements above can be modified to suit the user's own interests.

**Constraints / Important Notes:**

1) Due to limitations within API and not using a random word generation, This was not a random sample of searched words. Due to API limits, I could only return 50 entries of data per call. For this project, the API was called 20 times manually to get 1000 rows of data. The search words that were utilized were: dude perfect, music, cats, movies, python, mac, Christmas, Donald trump, vacation, sports, iphone, horror, laugh, gaming, golf, Minecraft, mars, video blogging, speedrun, Siemens.
2) If we step back and think about this problem, probably the most important feature will be the User ID of the youtube video because a particular user will have historic evidence of view counts based on their video production capabilities. Unfortunately, this feature is not available through the Youtube API so we're going to deal with this problem without this feature.
3) Since individual view counts per video are going to be very scattered, to reduce the scatter of the view counts during EDA, I chose to "bucketize" the viewcounts into buckets of 200,000 views.

**Decision Maker:** Marketing Manager for a company looking to advertise on Youtube or Youtube Sales / Pricing Manager

**Success Criteria:** Come up with prediction model with 60-70% accuracy

## Data Wrangling

Due to the requirement of utilizing the Youtube API in order to download the data for this project, several data wrangling procedures had to be utilized in order to access and clean up the data. The steps utilized in cleaning the data are outlined below:

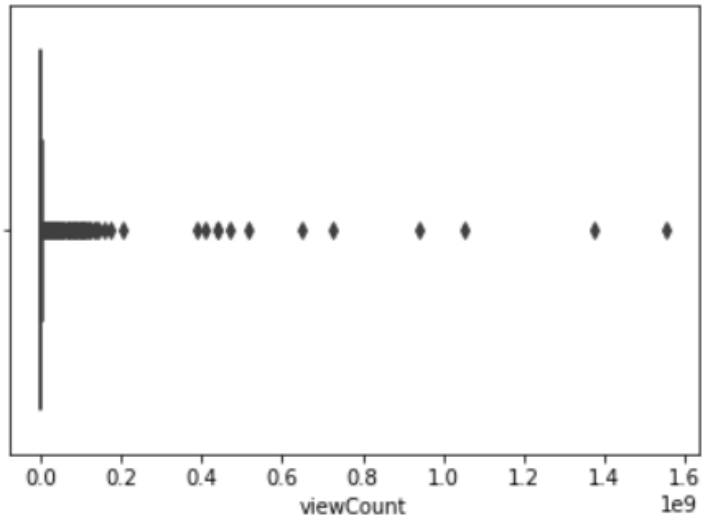
1. Utilized Google / Youtube API online instructions to authenticate an email with Oauth2. These credentials are necessary in order to access the API with a daily quota limit.
2. Through the API, supplied 20 different random words based on categories provided by the Youtube general categories. Each search produces 50 results, therefore the dataset had 1000 rows of data that I saved as a CSV.
3. Once the data was downloaded from the API, utilized json_normalization to flatten the data and pass this into the pandas dataframe.

Initial shape of the data from CSV file: 1000 rows (1000 videos collected) and 43 columns (video factors)

```
Unnamed: 0                            int64
search_word                          object
contentDetails.caption                 bool
contentDetails.definition            object
contentDetails.dimension             object
contentDetails.duration              object
contentDetails.licensedContent         bool
contentDetails.projection            object
etag                                 object
id                                   object
kind                                 object
snippet.categoryId                    int64
snippet.channelId                    object
snippet.channelTitle                 object
snippet.defaultAudioLanguage         object
snippet.defaultLanguage              object
snippet.description                  object
snippet.liveBroadcastContent         object
snippet.localized.description        object
snippet.localized.title              object
snippet.publishedAt                  object
snippet.tags                         object
snippet.thumbnails.default.height     int64
snippet.thumbnails.default.url       object
snippet.thumbnails.default.width      int64
snippet.thumbnails.high.height        int64
snippet.thumbnails.high.url          object
snippet.thumbnails.high.width         int64
snippet.thumbnails.maxres.height   float64
snippet.thumbnails.maxres.url        object
snippet.thumbnails.maxres.width    float64
snippet.thumbnails.medium.height      int64
snippet.thumbnails.medium.url        object
snippet.thumbnails.medium.width       int64
snippet.thumbnails.standard.height float64
snippet.thumbnails.standard.url      object
snippet.thumbnails.standard.width  float64
snippet.title                        object
statistics.commentCount            float64
statistics.dislikeCount            float64
statistics.favoriteCount             int64
statistics.likeCount               float64
statistics.viewCount               float64
dtype: object
```

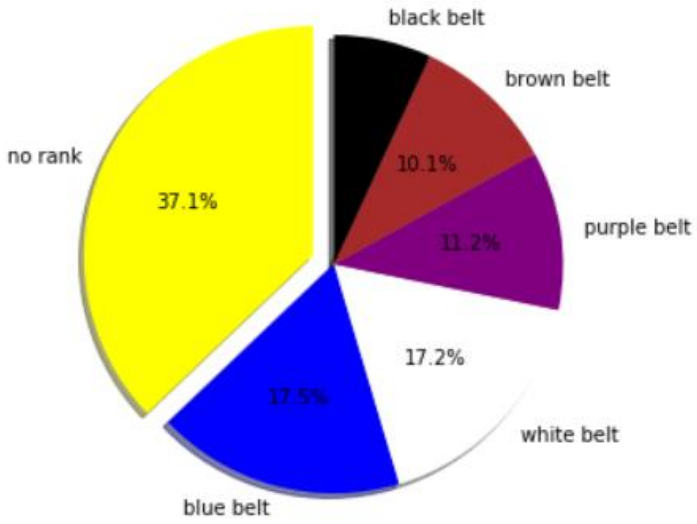
4. Analyzed all of the columns, shape, head of the data to assess the current data and then removed all of the columns that were irrelevant that had no relationship to view Count and could not be used for the prediction.
5. Analysis of the view counts was then performed to assess the outliers. The rationale for this is the concern that there will be a few videos from the search that represent overly huge view counts that is not representative of an average video on youtube. From the analysis, 10 of 11 of the identified outliers were music videos. These videos would skew my model – I made the decision to remove the outliers since predicting view count is not life critical, I chose to discard these values. This action should theoretically improve the overall model.

A box plot of view counts shows us 11 outliers. 10 out of these 11 are music videos. This is going to throw off our model so I chose to exclude this data.



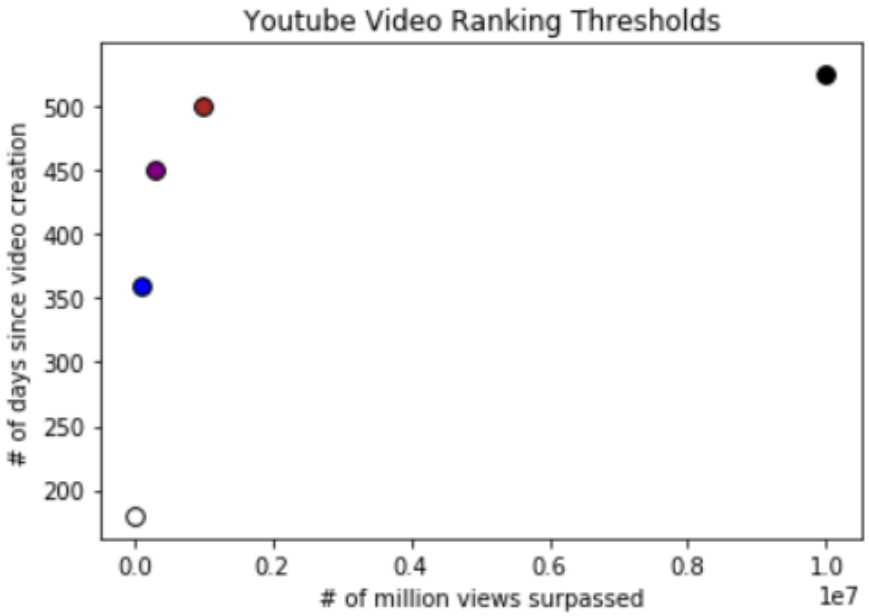

6. Missing values of the continuous data (e.g. likeCount, commentCount, dislikeCount) were filled in using the mean() function. I felt this would be the most appropriate method of addressing the continuous data since I did not want missing values to skew my model. Also, certain columns had missing lists or strings,, therefore I filled in an empty list or string for those columns using the fillna() function.
7. Initially use count based features of snippet tags. NLP analysis will be considered as a next step.
8. Conversion of the video publish date column to age in days was completed.
9. Conversion of a string to calculate the total duration of a video was required with the use of regular expression and a list comprehension plus zip function to multiply / calculate the total time in seconds.

10. Since predicting a precise view Count is rather unlikely, the dependent variable was categorized into a ranking system. I utilized a pie chart to show the distribution of dependent variables.



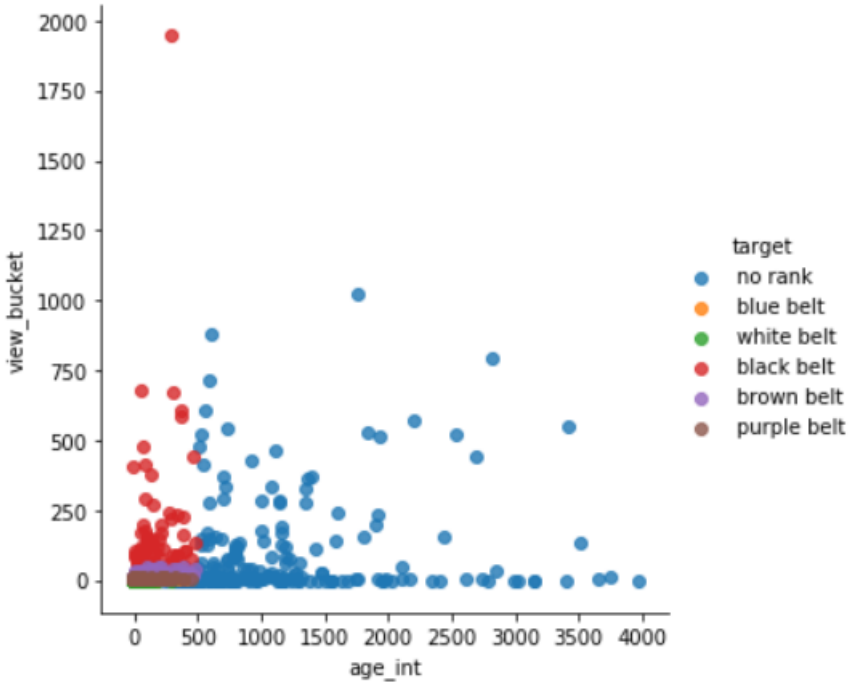

Also, a scatter plot of the ranking thresholds was visualized:

Note: A mathematical model for the ranking system could be considered in the future.

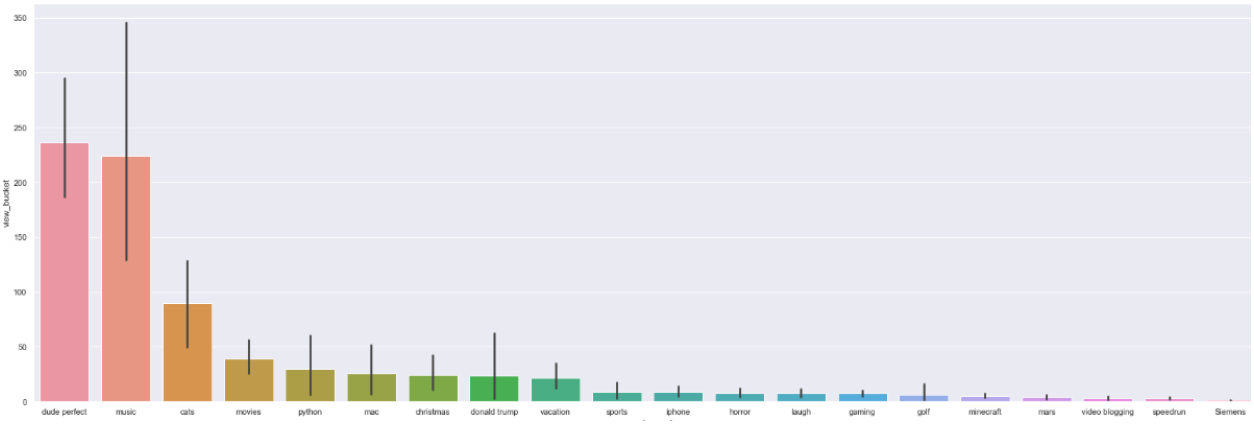


Other Interesting Visualizations:

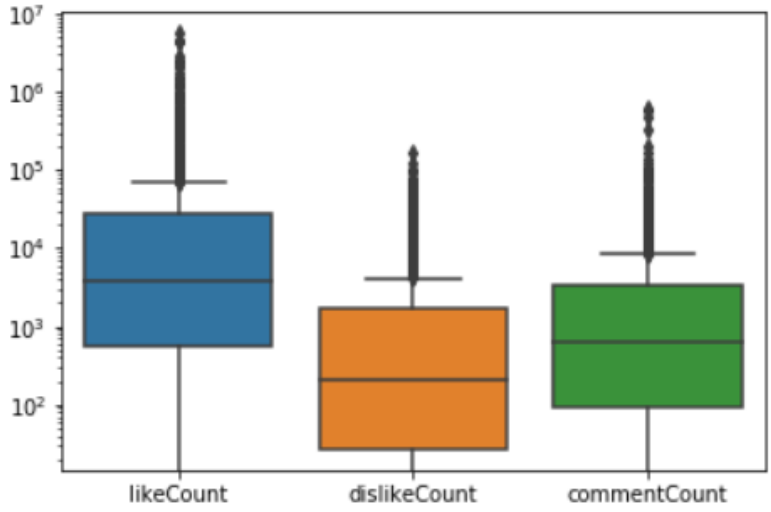
Plot of Age vs. View Count with Rank as Hue: There are many videos that do not make rank because they did not achieve the target views within the cutoff date but they do have a lot of views over a long period of time. These videos could have a different pricing model for ads.



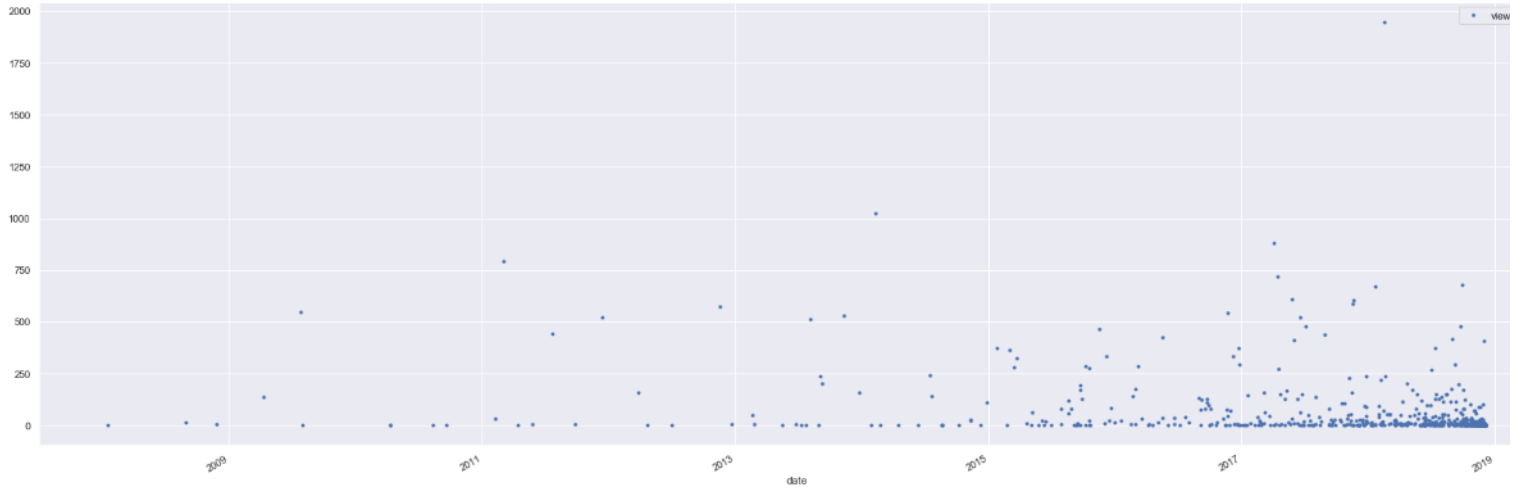

Bar plot of the view counts by search word:

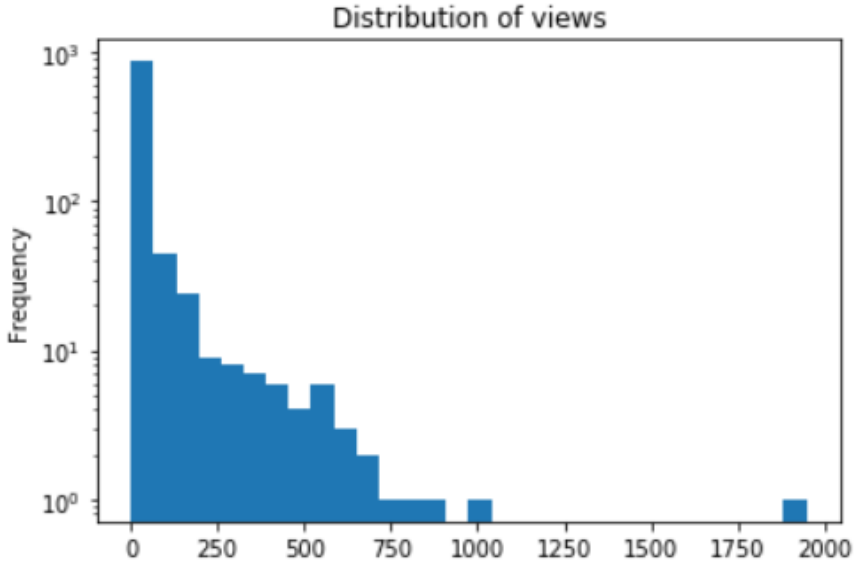


Box plot of like, dislike and comment counts for all videos. This makes sense since most people will click ‘like’ a video rather than ‘dislike’ a video.




Further data visualization shows interesting patterns in the data / view count: There appears to be more recent videos based on the search. This makes sense since the search function should return recent or more relevant videos.



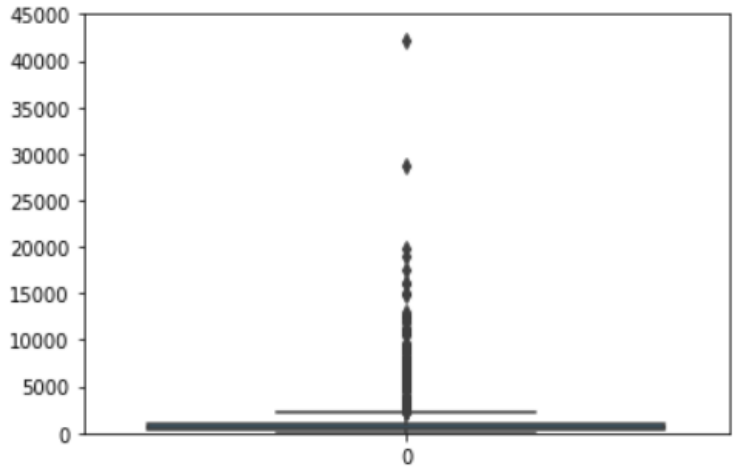



```
Average view_bucket for all 1000 videos 36.97
Observation: One more outlier is observed
```

No further action taken to remove any more outliers.

Average length of video was visualized just for informational purposes.

```
Average length of a video in seconds 1389.9817997977755
Maximum length of a video in seconds is 42112
Standard deviation of a video in seconds is 2850.836893189657
```




11. Once the data was in a clean format, we can apply inferential statistics

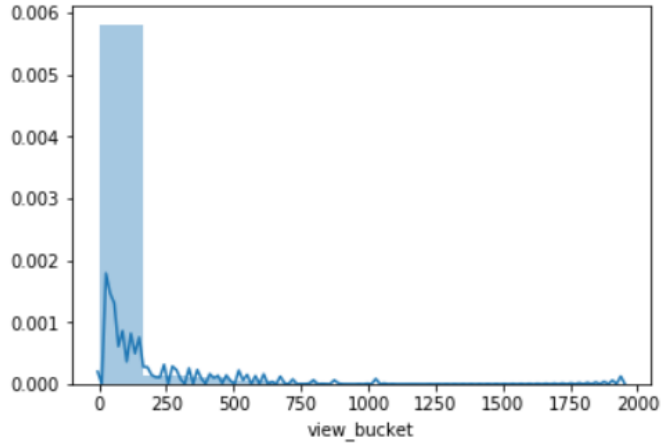
## Apply Inferential Statistics

The purpose of my model is to predict the view count of a youtube video depending on attributes about the video:

1. Search word
2. Title
3. Age
4. Description
5. Tags
6. Count based features of the title, description, tags

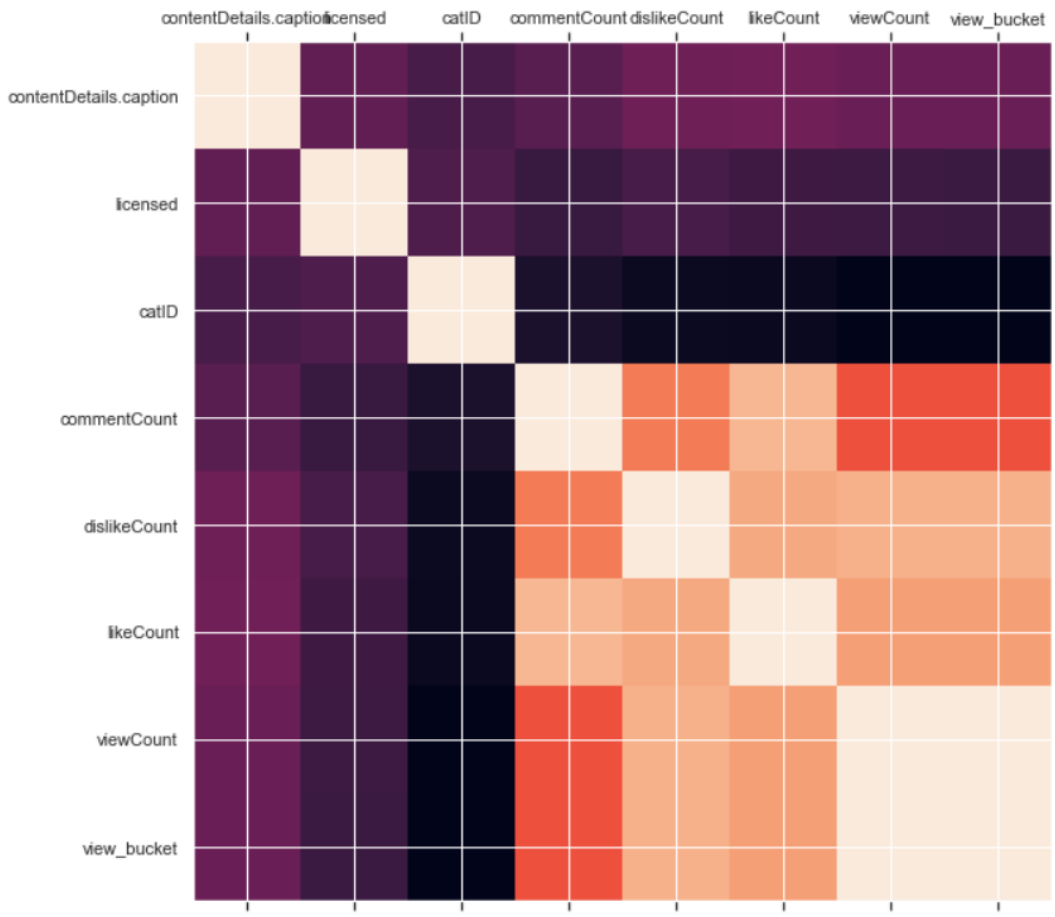
Statistical Techniques:

Distribution Analysis: One of the first things I tried to look for in my data is any normal distribution for the view counts of 1000 videos. It was immediately apparent that a right skew due to outliers occurred in my data due to a number of high view count videos.




Descriptive measures: The mean, and max of video length was looked at to see the average and max length of a video. The mode of the view bucket was looked at to see most of the videos fall under the first bucket. In other words, most of the videos have small view counts.

Correlation Matrix: Observed higher correction with comment, dislike, with view count. This is an obvious correlation.




Problem encountered: A random sample was not utilized because I was searching for specific search terms that I knew about.

Possible Solution: Utilize a random word generator to search for truly random terms for whatever the random generator produced. It would be great to automate this in the call to Youtube API along with increasing each search size beyond 50 entries. Right now, I had to manually search 20 words to get 1000 entries of data.

Dealing with the data I have: I started looking for any relationship between the independent and dependent variables such as video duration, licensed content, category ID, like, dislike, comment count.

1) For ordinal data, I utilized Spearmans correlation coefficient and for numeric data I utilized Pearson's correlation coefficient.
2) The significance level that was chosen for all features was 5%.
3) The p values were calculated for each of the features in order to assess their statistical significance.

4) If null hypothesis was rejected, then the feature was included for further modeling efforts. If the null hypothesis failed to be rejected, then the feature was excluded from further experiments.

Outcome of the PCC and SCC, I will include category ID, like count, dislike count, comment count in the numerical modeling. For the NLP modeling, I will include only the category ID.

## Conclusion & Next Steps

1. **NLP EDA and modeling will be critical to predicting viral video.** My current approach to experimenting with the NLP data is to use a shotgun approach to trying out different models and preprocessing techniques.
    a. Utilize both CountVectorizer and TFIDVectorizer for vectorization
    b. Utilize a pipeline to run a slew of ML models (number of experiments would be 2 * 6 + hyperparameter tuning experiments if applicable) :
        i. Logit
        ii. MultinomialNB
        iii. SVM
        iv. GBM
        v. Random Forest
        vi. LSTM
    c. Assess the accuracy of each model - look for the highest score from each of theexperiments.
    d. Run a classification report on all three of the best performing models.
2. **Integrate the numeric + NLP data** to rerun the top three pipelines to assess the performance and pick the best model.
3. **Test the model –** Run the API a few more times and test the accuracy of the model on some new data.
4. **Make final conclusions** / next steps beyond the project