# Capstone Project 1: Apply Inferential Statistics

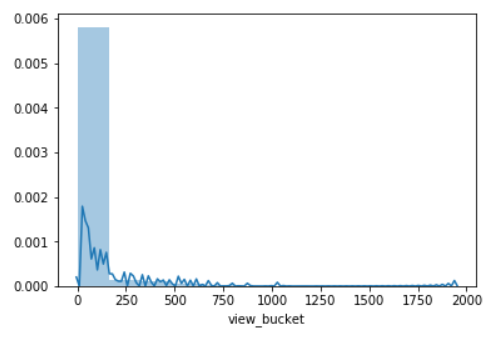
*Youtube View Count Prediction*

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

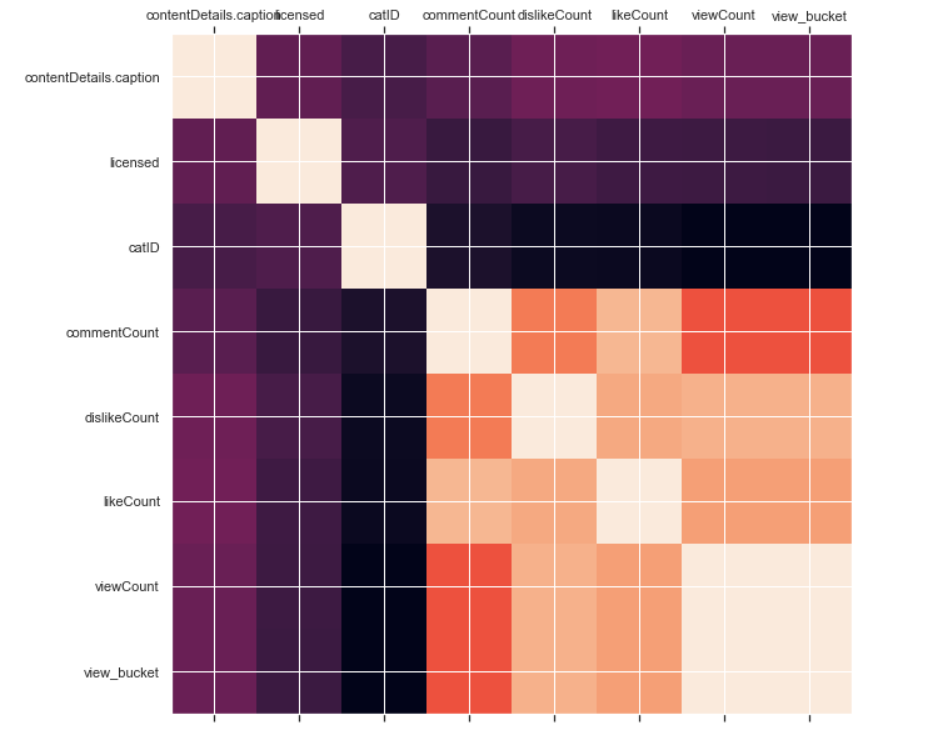
Statistical Techniques:

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.

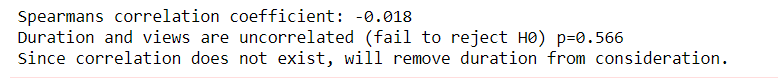
Correlation Matrix: Observed higher correction with comment, dislike, with view count. This is an obvious correlation.



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. Below is a sample of the Spearmans correlation coefficient along with p value that was calculated for each of the features in order to assess their statistical significance. The significance level that was chosen for all features was 5%. 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. On the sample screen below, the p value is .566 which means that a chance of a Type II error is considerably low.



Next Steps: Since my project involves both numeric and NLP data, the NLP portion of the data also needs to be analyzed. My current approach to experimenting with the NLP data is to use a shotgun approach to trying out different models and preprocessing techniques.

1. Utilize both CountVectorizer and TFIDVectorizer for vectorization
2. Utilize a pipeline to run a slew of ML models (number of experiments would be 2 \* 6 + hyperparameter tuning experiments if applicable) :
   1. Logit
   2. MultinomialNB
   3. SVM
   4. GBM
   5. Random Forest
   6. LSTM
3. Assess the accuracy of each model - Basically look for the highest score from each of the experiments.
4. Run a classification report on all three of the best performing models.
5. Integrate the numeric + NLP data to rerun the top three pipelines to assess the performance and pick the best model.