Final Report
Northwestern University
EECS 349
Alan Wojciechowski, Ryan Chilton, Ryan Hodin
Contact: alanwojciechowski2017@u.northwestern.edu

Abstract

Our goal is to determine what quantitative factors lead to a successful Youtube channel. With the rise of the Youtube Celebrity, there is merit in determining what separates the obscure channels from the ones with tens of millions of subscribers. With the potential to make money off increased subscriber count, the factors that lead to popularity are extremely important for starting channels. In particular, we examined the success of Let's Play channels, a genre of gaming videos that involve a mixture of walkthroughs, commentaries, and reviews of different games. We created a program in python to collect quantitative data from Youtube's API and another program to format it so we could run it through Weka.

Our target function initially deemed a channel a success if it has over one million subscribers. Using attributes such as video count, hidden subscriber count, average video duration, average video title length, average video description length, average posting time, sd/hd ratio, and caption ratio, we found no meaningful model that performed better than ZeroR. The key result we found was that J48 split on caption ratio twice, giving an accuracy of 93.55%, which was still lower than ZeroR at 95.16%. Channels with caption ratio greater than 0.0 and then greater than 0.13 were labeled a success. We later relaxed our target function to consider a channel a success if it had over 250,000 subscribers and ended up with similar results.

Report

We began our search by trying to figure out the best way to get data about Youtube channels. A few websites seemed promising, but we ultimately decided on using the Youtube Data Api because it allowed us to search for channels and videos with specific keywords. We used Python to request data and searched for channels with the keywords “Let’s Play” in them. The program would iterate over each channel, and for each channel it would request a variety of information. We held onto all the important data about a channel, which also included the ID of a playlist of all a channel’s uploads. This ID would be used to fetch the playlist of all their videos, allowing the script to iterate over each of the channel’s videos and store relevant information in a sub-object of the channel object in memory. After all channels had been processed, what was left would be an array of channel objects, each of which had the data returned by YouTube plus an array of video objects with data from YouTube. The program proceeded to simply output the array of channels to a JSON file.

The Youtube Data Api only allowed a max result size of 50 channels with 50 videos per channel. In order to get the next page of results, we needed to implement our own method of pagination. Using the max result size of 50 per page proved to be unsuccessful after many attempts. We suspected that it had to do with the amount of data that was being returned. After many attempts and changing max results to many different increments, we eventually got a result of 312 channels using max result size of 20 that got appended to a 27MB json file. We

then parsed the json file into a csv file in order to format our results in a way that was accepted by Weka. Our main data file ended up having 312 channels along with 24 attributes we felt might have potential in determining success.

The 24 attributes were topic ids(list), topic categories(list), comment count(numeric), view count(numeric), video count(numeric), subscriber count(numeric), hidden subscriber count(boolean), video topic categories(list), video topic ids(list), video tags(list), average video comment count(numeric), average video view count(numeric), average video favorite count(numeric), average video like count(numeric), average video dislike count(numeric), average video duration(numeric), average title length of video(numeric), average video description length(numeric), average posting time(numeric), sd/hd ratio(numeric), projection ratio(numeric), caption ratio(numeric), licensed content ratio(numeric), and 3d/2d ratio(numeric). We removed all attributes that had the same value for all channels, attributes that were directly related to success, and attributes that were lists of string values. In the end we had 8 attributes that were video count, hidden subscriber count, average video duration, average video title length, average video description length, average posting time, sd/hd ratio, and caption ratio.

From there, we split our data into training and test sets. After researching different ways to split our data, we chose to randomly split our data into 80% training data and 20% test data. In order to determine success, we focused on the subscriber count attribute because we wanted to focus on the long-term success of a channel, rather than the virality of a channel. We initially deemed a channel a success if it had over one million subscribers. By looking at top gaming channels via SocialBlade.com, we figured that if a channel had over one million subscribers then it would belong in the top 1000 gaming Youtube channels. After splitting our data on that threshold, we ended up having 3 out of 62 channels in our test set labeled a success. Thus, ZeroR displayed an accuracy of 95.16%(Fig.1). No algorithm that we ran on the data beat ZeroR. The best model we got was using a decision tree (Fig.2) with accuracy 93.55%, which produced a model which split on caption ratio (Fig.3). This indicated to us that having a higher proportion of videos with captions might correlate with success, a hypothesis we would later research. All other models either performed worse than ZeroR , such as Naive Bayes, or ended up predicting the most common class value like ZeroR . We hypothesized that our set of attributes did not help determine success in a channel.

Examining this hypothesis, we used Weka to visualize our dataset. We found some interesting clustering in several attributes (Fig. 4). In particular, successful channels exhibited a below-average mean video duration and a very low number of SD videos as compared to HD videos. Additionally, successful channels seemed to cluster towards a longer average title, a shorter average description, and a tendency to post videos later in the day. However, none of these attributes were used by any classifier. Upon further investigation into each attribute's graph with success (Fig. 5), we observed that only caption ratio had a clear rule that had all successes past a certain caption ratio. None of the other attributes had a strong rule, so it made sense to us why our decision tree split on the caption ratio, and other models ignored the remaining attributes.

Displeased with our results, we relaxed our threshold to 250,000 subscribers to see if we could give a model a chance to beat a lower ZeroR accuracy. We split the data with the same 80/20 split into train and test data and ran it through Weka. We found ZeroR gave us an

accuracy of 82.2581% (Fig.6). J48 split again on caption ratio in the same way as it did with the threshold of one million subscribers, but with an accuracy of 80.6452% (Fig.7). Unfortunately, every model that we tested performed worse, or labeled all instances the most common label, like ZeroR. All attempts at feature selection failed, as any combination of attributes used together did not change the results we were getting.

In conclusion, the most meaningful result we found was the decision tree split on caption ratio. After some analysis and research, we hypothesize that having a higher ratio of videos with captions helps with a channel's success due to Google's SEO algorithm. SEO, or Search Engine Optimization, is the site's way of choosing the order to list videos or other results when a term is searched. As this takes into account the ratio of captioning for the video. A captioned let's play will get more valuable ranking real-estate than a non-captioned one. Thereby, when someone is looking for a let's play of a new release, a captioned video is more likely to get their attention, and thereby, their subscription.

Given additional time and resources, investigating attributes such as age, gender, style of commentary, type of video, seeing the player on screen, type of game, and other qualitative features might help lead to a better indicator of Youtube success. These attributes cannot be fetched automatically, and would have to be done by watching many videos. From our analysis, we concluded that different video lengths, descriptions lengths, and other modifications of our eight main attributes did not have have a major indication in success. It was disappointing for us to come away with not having any major indicator of success, but also understandable given the small rate of success with Let's Play channels and the unpredictable nature of becoming a Youtube sensation.

The work was evenly divided among all group members, and all members gave input during each step of the project. Ryan Hodin handled fetching data from the Youtube Data API, Alan Wojciechowski handled parsing the json data into a readable format for Weka, and Ryan Walker handled transforming the data into appropriate train/test sets and ran the data through Weka. All group members handled analysis and interpretation of results.

Github Repo

Fig.1

```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances          59               95.1613 %
Incorrectly Classified Instances         3                4.8387 %
Kappa statistic                          0
Mean absolute error                      0.1308
Root mean squared error                  0.2188
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances               62

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.952      1.000    0.975      0.000    0.500     0.952     no
                 0.000    0.000    0.000      0.000    0.000      0.000    0.500     0.048     yes
Weighted Avg.    0.952    0.952    0.906      0.952    0.928      0.000    0.500     0.908

=== Confusion Matrix ===

  a  b   <-- classified as
 59  0 |  a = no
  3  0 |  b = yes
```

Fig.2

```
Time taken to build model: 0.03 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.02 seconds

=== Summary ===

Correctly Classified Instances          58               93.5484 %
Incorrectly Classified Instances         4                6.4516 %
Kappa statistic                         -0.0248
Mean absolute error                      0.127
Root mean squared error                  0.2544
Relative absolute error                 97.0695 %
Root relative squared error            116.241  %
Total Number of Instances               62

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.983    1.000    0.951      0.983    0.967      -0.029   0.345     0.941     no
                 0.000    0.017    0.000      0.000    0.000      -0.029   0.345     0.039     yes
Weighted Avg.    0.935    0.952    0.905      0.935    0.920      -0.029   0.345     0.898

=== Confusion Matrix ===

  a  b   <-- classified as
 58  1 |  a = no
  3  0 |  b = yes
```

Fig.3

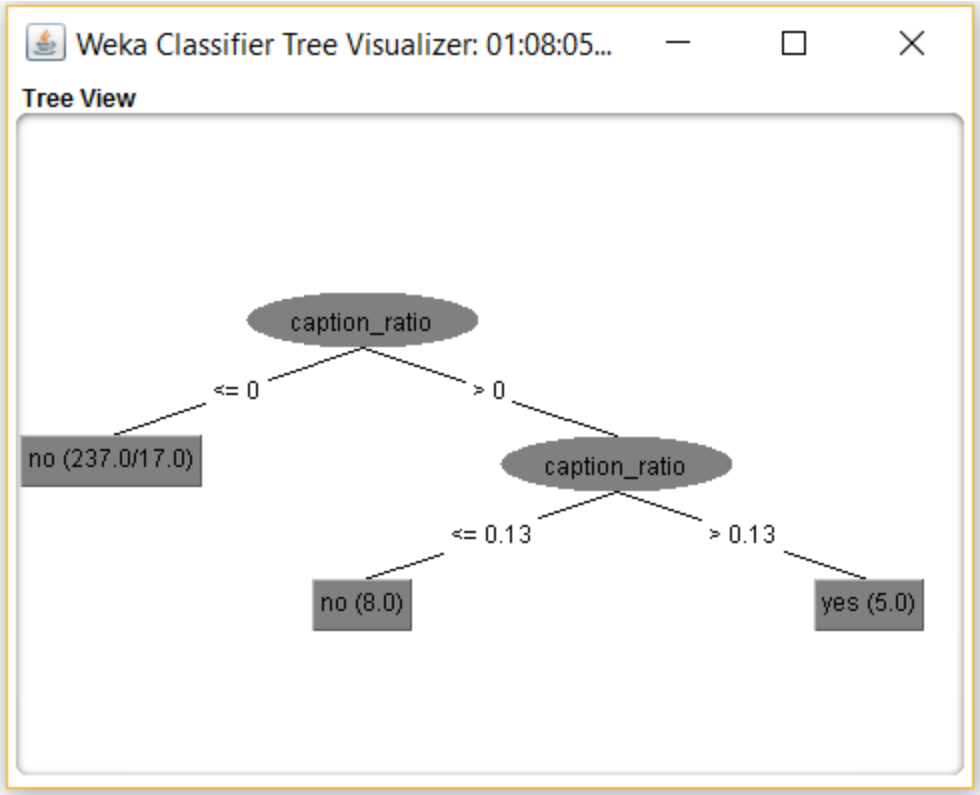
Weka Classifier Tree Visualizer: 01:08:05...
Tree View
caption_ratio
<= 0
> 0
no (237.0/17.0)
caption_ratio
<= 0.13
> 0.13
no (8.0)
yes (5.0)

Fig.4

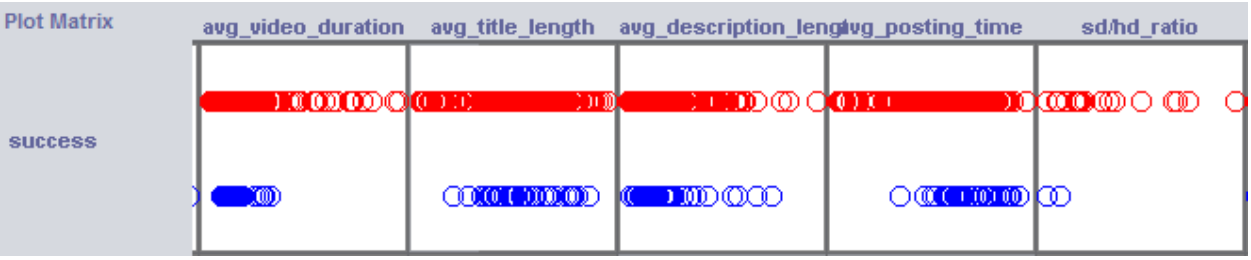
Plot Matrix
avg_video_duration
avg_title_length
avg_description_leng
avg_posting_time
sd/hd_ratio
success

Fig.5

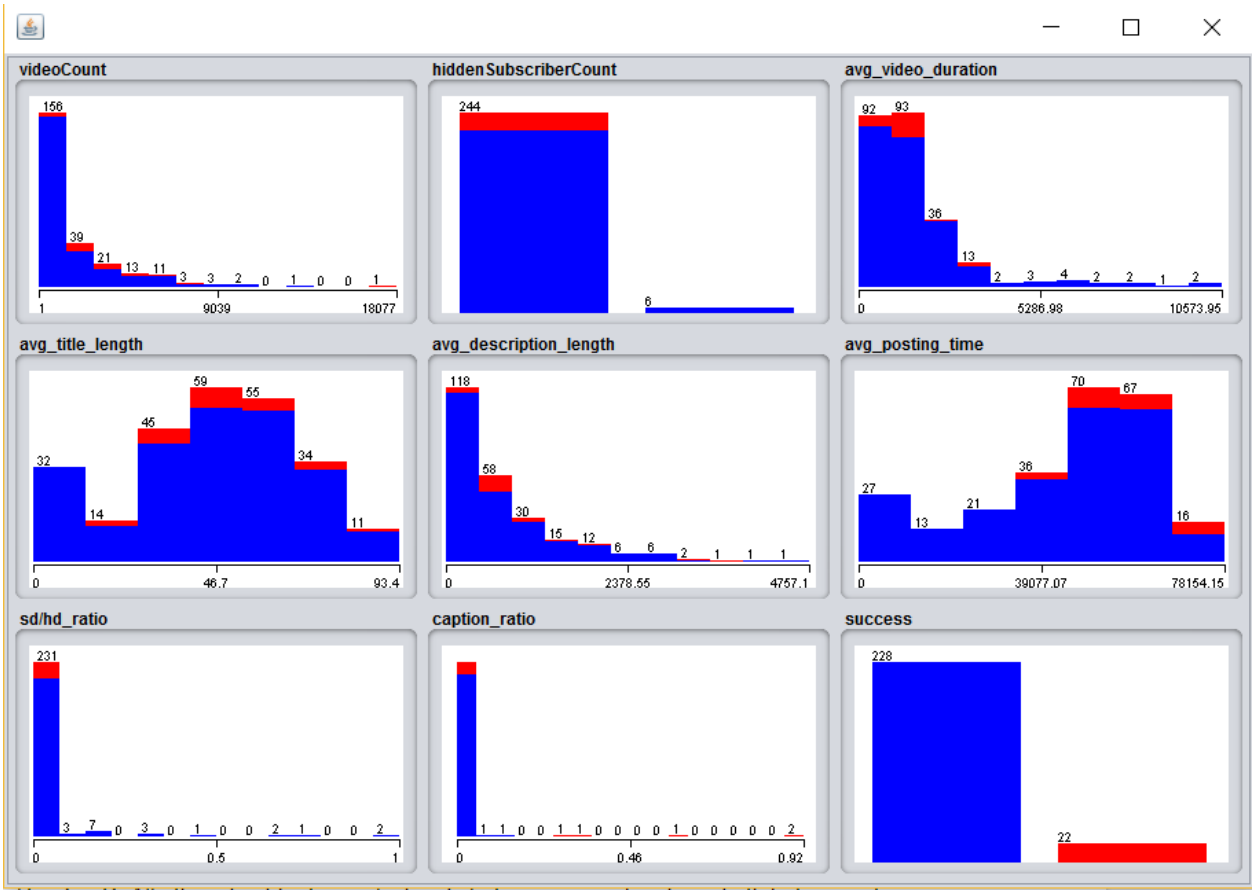

Fig.6

```
Time taken to build model: 0 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0 seconds

=== Summary ===

Correctly Classified Instances          51               82.2581 %
Incorrectly Classified Instances        11               17.7419 %
Kappa statistic                          0
Mean absolute error                      0.3208
Root mean squared error                  0.3846
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances               62

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.823      1.000    0.903      0.000    0.500     0.823     no
                 0.000    0.000    0.000      0.000    0.000      0.000    0.500     0.177     yes
Weighted Avg.    0.823    0.823    0.677      0.823    0.743      0.000    0.500     0.708

=== Confusion Matrix ===

  a  b   <-- classified as
 51  0 |  a = no
 11  0 |  b = yes
```

Fig.7

```
Time taken to build model: 0.01 seconds

=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.03 seconds

=== Summary ===

Correctly Classified Instances          50               80.6452 %
Incorrectly Classified Instances        12               19.3548 %
Kappa statistic                         -0.0305
Mean absolute error                      0.3223
Root mean squared error                  0.4044
Relative absolute error                100.4658 %
Root relative squared error            105.1456 %
Total Number of Instances               62

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.980    1.000    0.820      0.980    0.893      -0.059   0.465     0.814     no
                 0.000    0.020    0.000      0.000    0.000      -0.059   0.465     0.170     yes
Weighted Avg.    0.806    0.826    0.674      0.806    0.734      -0.059   0.465     0.700
```