Youtube Censorship in Culture and Politics

IS 517 Final Project

Prithvi Prakash

Saahil Hiranandani

Zak Kilhoffer

May 08, 2022

Introduction

Project Overview

YouTube is the most popular video platform online. In the United States, YouTube is a significant source for news; over a quarter of U.S. adults get their news from YouTube videos, watching established news channels and independent content creators in about equal measure.[1]

With such a wide assortment of community created content, moderating YouTube content is a herculean task. Given the complex considerations for censoring political content, and sheer amount of videos, YouTube uses blackbox Machine Learning models to remove objectionable content. While these models tend to perform well, innocuous content is also taken down or demonetised (Kurdi, Albadi, and Mishra 2020, 2021). This has led to accusations that YouTube is biased against certain political leanings (Wu and Resnick 2021) and spawned ethics discussions about using AI to curate and censor political speech.

Problem Statement

We aim to understand what factors influence the removal of political videos on Youtube, and to what degree can we predict whether a video will be taken down. We further attempt to predict the time YouTube took to remove these videos.

- **RQ1**: How accurately can we predict which political videos will be removed by YouTube's algorithm using video metadata?
- **RQ2**: How accurately can we predict how long it takes for a political video to be removed based on its metadata?

Related Work

At present, there is relatively little literature on YouTube's video removal behavior. While content moderation in online communities has been studied a good deal, content moderation on Youtube has been significantly understudied. Part of the difficulty in the research comes from the lack of publicly-available data on removed videos, making it very difficult to understand YouTube's black box removal algorithm (Clark and Zaitsev 2020).

However there exists a substantial body of literature for studying video popularity (Trzciński and Rokita 2017) and videos promoting extreme ideologies (Ottoni et al. 2018). Also of note, independent researchers published[2] a list of words that, when present in a video title, seem to cause YouTube to

[1] Pew Research (2020), Many Americans Get News on YouTube Where News Organizations and Independent Producers Thrive Side by Side, retrieved here.

[2] See the published Google Doc (YouTube Demonetization Words).

demonetize it (Dodgson 2019). Among these words are: lesbian, Dick Cheney, Brazil, Idaho, and jew (Fabrizio Bulleri 2019). The puzzling inclusion of words like these further increased concern for how YouTube removed videos and potentially censors political speech.

Our Proposal

Given the lack of research on political content removal on Youtube, we contribute to the literature by assessing a different time frame than Kurdi M. (2021), as we consider videos removed during 2021. In 2021, the COVID-19 context may have led Youtube to change its policies on content and monetization. Additionally, we assess the length of time YouTube takes to remove content with RQ2.

Analysis

Data Description

Our data mainly comes from two sources: Transparency Tube[3] and Youtube Data Tools.[4] Transparency Tube categorizes and analyzes over 7,300 of the largest English language YouTube channels actively discussing political and cultural issues, scraping these channels semi-regularly. When videos are no longer available, they are added to a dataset of removed videos. We retrieved this data for videos uploaded in 2021.

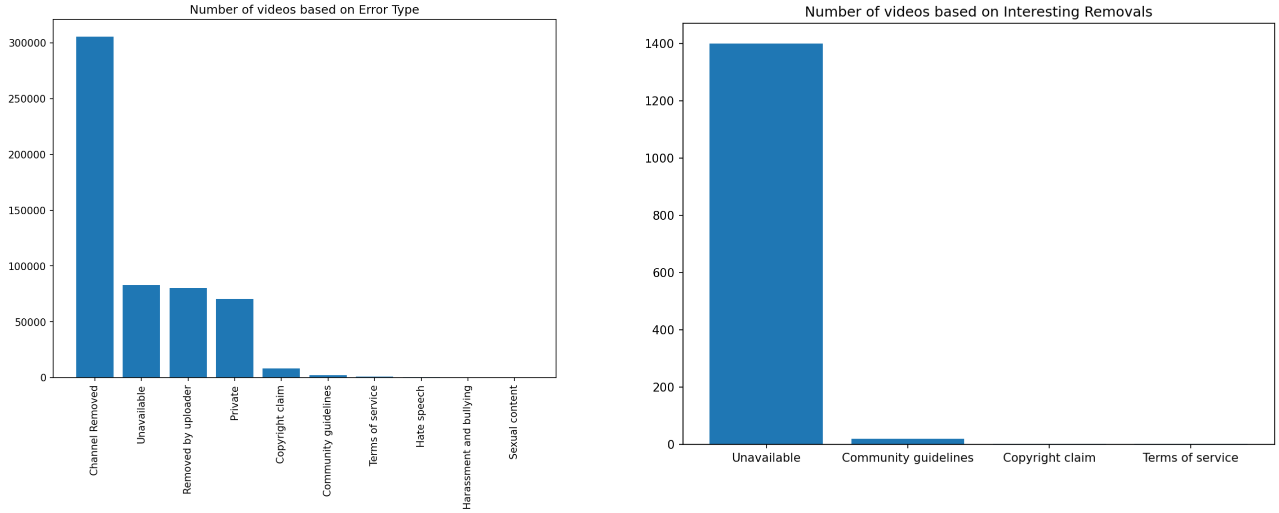


Figure 1: Video removal counts

Transparency Tube makes it relatively easy to download a list of all videos that they have found to be removed. For our prediction task, however, we benefit from assessing videos that are and aren't removed. We therefore needed to gather additional data on unremoved videos. It was not feasible to retrieve all of a channel's videos programmatically, so we made a selection of channels to retrieve manually. We randomly selected three channels for each of the 18 political categories[5] (see Figure 2) to retrieve using Youtube Data Tools, as we consider political affiliation of a channel to be an interesting factor for removal. We retrieved this data, then filtered for videos uploaded in 2021.

[3] See page here: transparency.tube
[4] See page here: Youtube Data Tools
[5] For details on the labeling process, see here: https://github.com/markledwich2/Recfluence.

Transparency Tube also has a dataset where each row is one of the ~7,300 political channels, and the columns are channel information (text description, subscription counts, etc.). For our working dataset, we combined these three datasets: removed videos, unremoved videos, and channel data. It is from this dataset that we sampled and ran models.

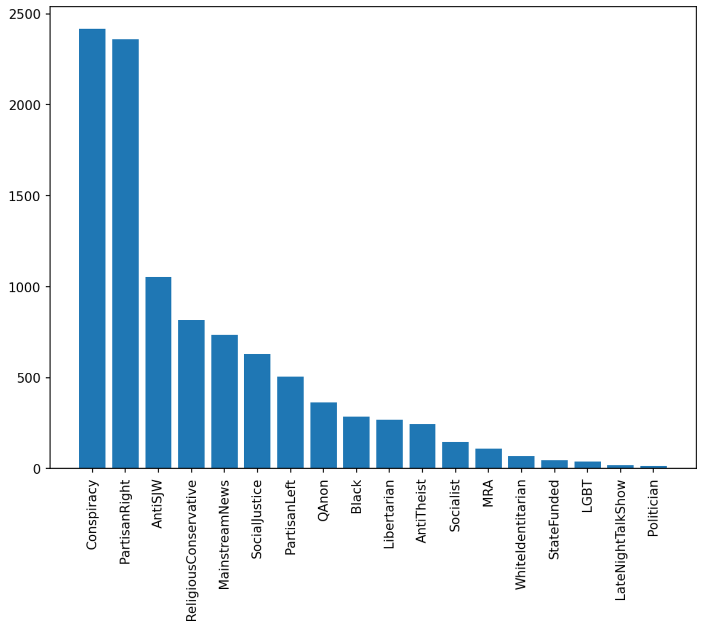


Figure 2: Number of channels based on Channel Tags extracted from Removed Videos Dataset

Importantly, there are different types of video removals, which are contained in the categorical variable "errorType", and not all of them are of interest. We aim to predict when YouTube removes the video, as opposed to the uploader. Therefore, we do not consider videos removed or made private by the uploader as "removed". Removed videos are defined by the following errorType categories: "Community guidelines", "Copyright claim", "Terms of service", or "Unavailable".

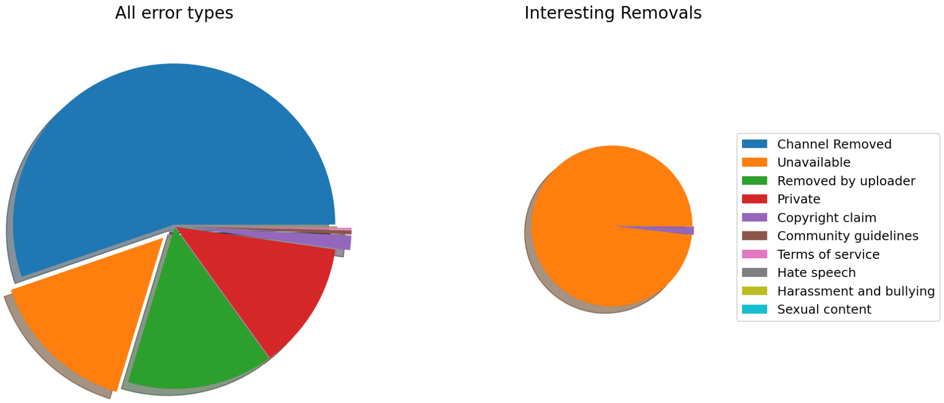


Figure 3: Number of videos based on Error Type vs Interesting Removals from the Combined dataset of removed and unremoved videos.

Most of these are quite specific and self-explanatory reasons for video removal, but "Unavailable" seems to be a very broad category for when YouTube removes a video without providing explanation. In testing, we concluded that "Unavailable" could relate to objectionable content, country-specific copyright concerns, or even software bugs, among other reasons.

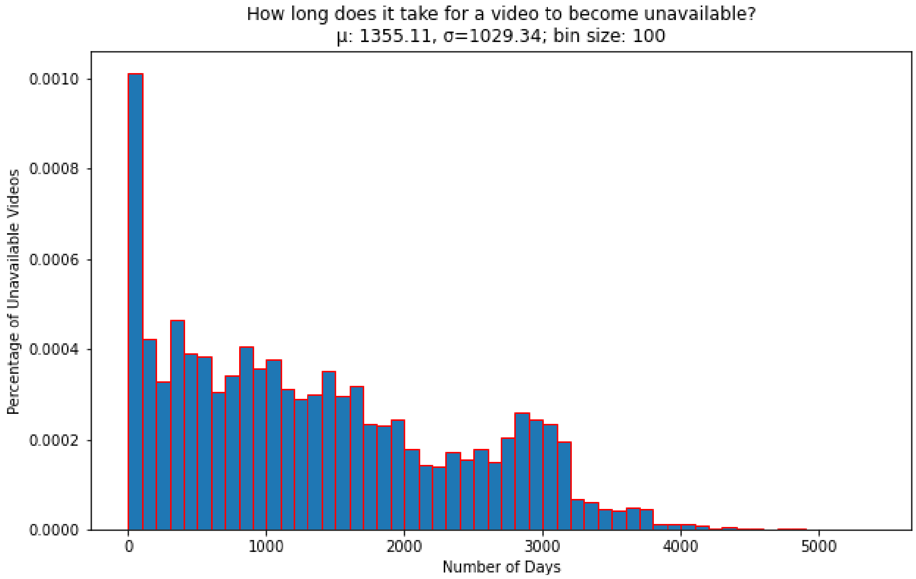


Figure 4: Video Removal Time Histogram

To create the response variable for RQ2, we calculated the difference between "upload date" and the "last seen" columns and generated an additional column, "removeTime". The figure above shows a histogram of the distribution of this newly created column.

Feature Engineering

Our dataset contains a variety of variable-types ranging from numeric, categorical, textual to binary. While numeric, categorical and binary variables are easy to incorporate in modeling, textual data needs some encoding. Therefore, we use $\text{BERT}_{\text{BASE}}$ (Bidirectional Encoder Representations from Transformers), a transformer based machine learning technique for NLP.[6] which essentially converts textual data into a variable number of encoder layers. The title of a video is considered by far the most important piece of metadata.[7] It is crucial to include keywords in the title for SEO. Considering their importance in our dataset, we used bert-base-nli-mean-tokens to convert video titles into 768 encoder layers. Since generating these layers is computationally expensive, we used Google Colab.[8]

Furthermore, we were interested in the profanity level of these titles. We implemented two processes to check for profanity. The first made use of Youtube's list of demonetized words to check if any title contains those and output True if so. The second used "profanity-check",[9] a third party library which uses SVM to classify titles as profane and outputs a probability metric.

Beyond these features, we rely on video and channel metadata. Most importantly, we rely on political leaning of a channel (left, right, center), and which of 18 political categories the channel matches.[10]

[6] https://en.wikipedia.org/wiki/BERT_(language_model)

[7] Deansays:, B. (2017, February 28). *We analyzed 1.3 million Youtube videos. here's what we learned about YouTube Seo*. Backlinko. Retrieved May 8, 2022, from https://backlinko.com/youtube-ranking-factors

[8] https://colab.research.google.com/

[9] https://github.com/vzhou842/profanity-check

[10] For details on the labeling process, see here: https://github.com/markledwich2/Recfluence.

Algorithms and Techniques:

Our problem statement can be broken down into two broad tasks: classification – to predict if YouTube will remove a video, and regression – to predict the time YouTube is likely to take to remove the video. For our choice in algorithms, we bear two key parameters in mind: quantitative accuracy and qualitative interpretability. These goals often entail a tradeoff, so we selected models to achieve both goals. For the goal of most accurate models, we chose random forest and a three-layered feedforward neural network (with one hidden layer). For the goal of most interpretable models, with acceptable accuracy, we chose logistic regression and decision tree.

Methodology

Data Preprocessing

For RQ1, data cleaning to create our working dataset is detailed in the data cleaning file. Beginning from this working dataset, we needed to create samples for analysis. The removals we are interested in predicting are rare: about 4% of 35,463 observations. We therefore sampled in two ways: with removed to unremoved in proportion to the actual dataset, and with removed to unremoved in even proportion to allow our models to perform better. For the latter, given that we have 1,425 removed observations, we randomly selected 1,425 unremoved observations, for a total n=2,850. For each of these, we sampled 10%, 50%, and 100% of the dataset. For each sample, we used an 80-20 train-test split.

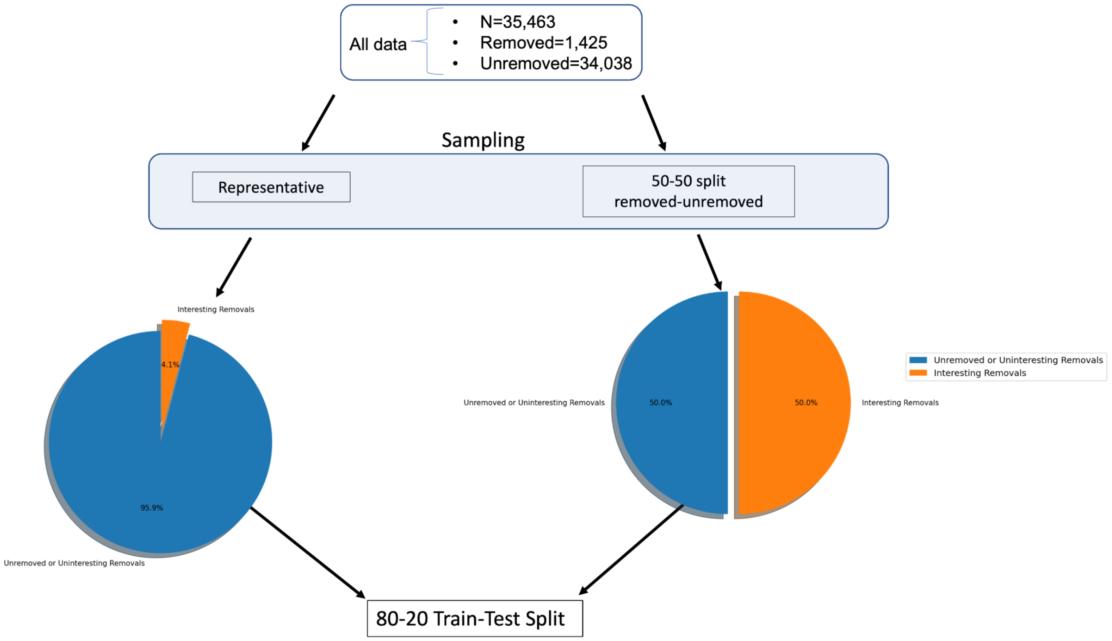


Figure 5: Sampling and Splitting Strategy

For RQ2, the dataset only contains the removed videos. Sampling was not necessary since there was no concern over any imbalance within the dataset.

Implementation

For RQ1, we run a series of models roughly going from least to most complicated: logistic regression, random forest, decision tree, SVM, and neural network. Each model was run six times: for the data with removed and unremoved videos in equal proportions, at 10%, 50%, and 100% sample size; for the data with removed and unremoved videos proportional to the real world data, at 10%, 50%, and 100%

sample size. With six runs each of five different model types, a total of 30 models were attempted. However, due to computational restraints, only 26 ran successfully on R.

For RQ2, we ran a total of three Linear Regression models to assess the impact each predictor had towards making an accurate prediction. For model 1, we made use of only video metadata that was publicly available to make predictions; model 2 included the associated political leaning of the video as an additional predictor; lastly, model 3 included the *channel_id* as an additional predictor.

Results

Model Evaluation

RQ1

These results are shown in detail in the Appendix. Starting with the GLM model, using sampling with a 50-50 split of removed and unremoved videos, we achieved respectable accuracy in both the train (85%) and test (87%) data. Using the sampling with a representative split, however, R was unable to finish computations using over 10% of the data. The random forest models proceeded similarly in its best iteration with the train (86%) and test data (87%).

In the representative split, GLM and Random Forest crashed R for the 50% and 100% samples. Looking only to the 10% samples, GLM and Random Forest both achieved about 96% accuracy for train and test data. However, the high accuracy is because the models practically predict that a video will never be removed, which is correct 96% of the time. Therefore, precision and recall are better metrics to evaluate model performance. Precision tells us, when we predict a video will be removed, what percent of the time are we correct? Recall tells us the inverse. The test and train precision and recall are 0 for both GLM and Random Forest, indicating a failed model with the representative 10% samples.

Moving to decision tree models, these did manage to compute. Results are quite accurate using the test (86%) and train (87%) datasets with a 50-50 split. When we move to the representative datasets, the decision tree moves up to about 96% accuracy, but with very low precision for both train and test datasets. This again shows a tendency to predict “all zeros” when the predicted phenomena is so rare. Again, the precision and recall are better measures of accuracy, and these indicate a prediction that is much worse than a coin flip when looking at representative split data.

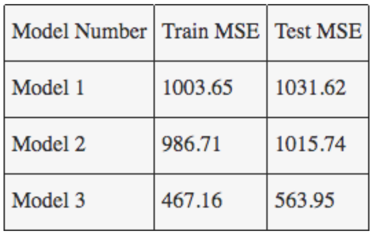
The SVM model slightly underperforms when using the datasets with a 50-50 split, with the best train accuracy being 84% and test accuracy being 86%. The accuracy again improves to around 96% with the representative data, but the precision and recall show that the models are actually just predicting all zeros again.

Finally, the neural network is a different story. The best performance with the 50-50 split data is 99% accuracy for train and 82% for test, indicating significant overfitting. The overfitting is even more pronounced looking at the performance for the representative data. The train accuracy is 99% and test accuracy is 96%, but the precision and recall drop from 99% in the train data to around 50% in the test data.

Still, the neural network performed the best of all models when considering the representative data. This is unsurprising, as the neural network is the only model where we included the BERT encodings, which added a significant amount of data (768 floating point number columns). Looking at the 50-50 split data, the neural network performs worse than the GLM, random forest, decision tree, and SVM in terms of precision in test data sets. Once again, this is primarily an effect of the neural network overfitting.

RQ2

For our regression task, we observe that successive inclusion of additional information as predictors brings down the mean squared error (MSE) down. Starting with Model 1, which made use of the basic metadata associated with a video which proves to be useful information towards predicting the time taken for a video to be removed.



Model Number	Train MSE	Test MSE
Model 1	1003.65	1031.62
Model 2	986.71	1015.74
Model 3	467.16	563.95

Table 1: RQ2 Model Summaries

The addition of political leaning shows a minor but evident drop in MSE. Specifically, in Model 2 (shown in the Appendix), we see that if a channel is left-leaning, there is no significant effect on the time for a video to be removed. If a channel is right-leaning, there is a highly significant and moderately negative effect on the time for a video to be removed. In other words, compared to the reference point of a center-leaning channel, all else being equal, a right-leaning channel's video will be removed faster. However, the low adjusted R-squared shows that the predictors we use only explain about 6% of the variance in removal time.

Lastly, the addition of *channel_id* information drops the mean squared error dramatically as seen in the predicted vs. actual plot, as seen in Figure 6. This means that the specific channel, rather than the political affiliation of a channel, is more important to predicting the time it takes for a video to be removed.

Interpretation

RQ1

For the 50-50 split data, the highest test accuracy we achieved was 87%. Interestingly here, the simpler GLM, random forest, and decision tree models surpassed the SVM and neural network in terms of peak test accuracy. Looking to test precision, even with the addition of BERT data, the neural network performed worse than all other models.

A few takeaways are apparent. First, the 50-50 split data avoids the problems with the representative data, wherein most models tend to simply predict all zeroes, i.e. that the videos won't be removed. This confirms the importance of careful sampling when the scenarios we are trying to predict are rare. Second, the more complex models do not necessarily outperform the simpler models, even with the addition of more data (the BERT embeddings with the neural network). Third, when looking to the representative data samples, no model performs better than a coinflip in both precision and recall. The reason for this could simply be that we do not have an adequately sized dataset to achieve good results, or some more subtle issue in our sampling and modeling strategy.

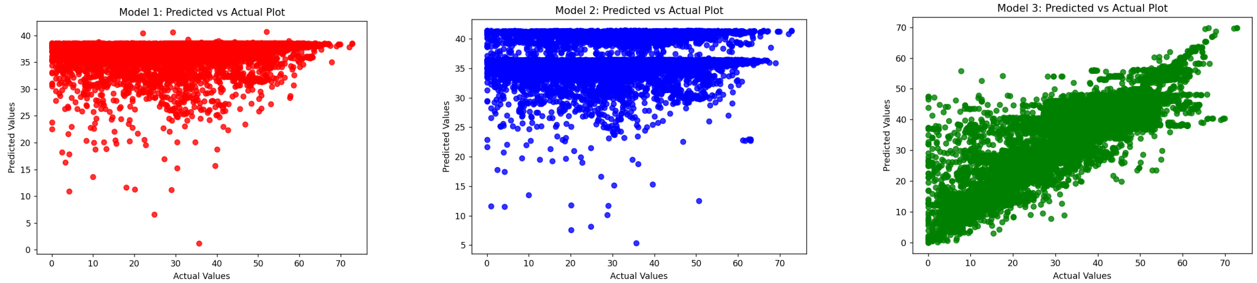


Figure 6: Predicted vs Actual Plot for Regression Models

RQ2

We see that political leaning and channel_id are key predictors towards accurately predicting the time YouTube takes towards removing a video. This could be interpreted in one of several ways: YouTube is faster to act in removing videos of content creators who are politically right leaning due to political bias; or some commonality shared by right leaning content creators contributes to videos being removed faster. For example, it could be that objectionable videos uploaded by politically right leaning content creators are easier to identify and are therefore removed faster. This finding merits further research as it could indicate political bias of YouTube, which acts as a key gatekeeper to information online.

Conclusion

We have found that we can predict if a political video will be removed with reasonable accuracy (~87%) with a relatively simple decision tree. However, the models tend to fail if we use a representative dataset, where only around 4% of videos are removed. In these instances, precision and recall are very poor.

In terms of explainability, the political leaning and channel videos come from having significant explanatory power. This makes intuitive sense, and also suggests that there is reasonable concern that YouTube's algorithm is biased towards certain political leanings. For example, the decision tree shows that right-leaning political videos are indicative of higher removal likelihood (RQ1), while the regression (RQ2) shows that right-leaning political videos are taken down quicker.

Future studies should attempt to achieve better results, especially by using larger datasets, and perhaps a variety of different sampling techniques. It would also be very helpful if YouTube would make data on removed and unremoved videos more readily available. In the absence of such transparency, regulation of some sort may be required to ensure equitable distribution of and access to political information.

Bibliography

Clark, Sam, and Anna Zaitsev. 2020. "Understanding YouTube Communities via Subscription-Based Channel Embeddings." *arXiv:2010.09892 [cs]*. http://arxiv.org/abs/2010.09892 (March 21, 2022).

Dodgson, Lindsay. 2019. *YouTubers Have Identified a Long List of Words That Immediately Get Videos Demonetized, and They Include "gay" and "Lesbian" but Not "Straight" or "Heterosexual."* https://www.insider.com/youtubers-identify-title-words-that-get-videos-demonetized-experiment-2019-10 (May 8, 2022).

Fabrizio Bulleri. 2019. *These Are the Words That Get Creators Demonetized on YouTube*. https://reclaimthenet.org/youtube-demonetization-words-blacklist/ (May 8, 2022).

Kurdi, Maram, Nuha Albadi, and Shivakant Mishra. 2020. "'Video Unavailable': Analysis and Prediction of Deleted and Moderated YouTube Videos." In *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, , 166–73.

———. 2021. "'Think before You Upload': An in-Depth Analysis of Unavailable Videos on YouTube." *Social Network Analysis and Mining* 11(1): 48.

Ottoni, Raphael et al. 2018. "Analyzing Right-Wing Youtube Channels: Hate, Violence and Discrimination." In *Proceedings of the 10th ACM Conference on Web Science*, , 323–32.

Trzciński, Tomasz, and Przemys\law Rokita. 2017. "Predicting Popularity of Online Videos Using Support Vector Regression." *IEEE Transactions on Multimedia* 19(11): 2561–70.

Wu, Siqi, and Paul Resnick. 2021. "Cross-Partisan Discussions on YouTube: Conservatives Talk to Liberals but Liberals Don't Talk to Conservatives." In *Proceedings of the Fifteenth International AAAI Conference on Web and Social Media*,.

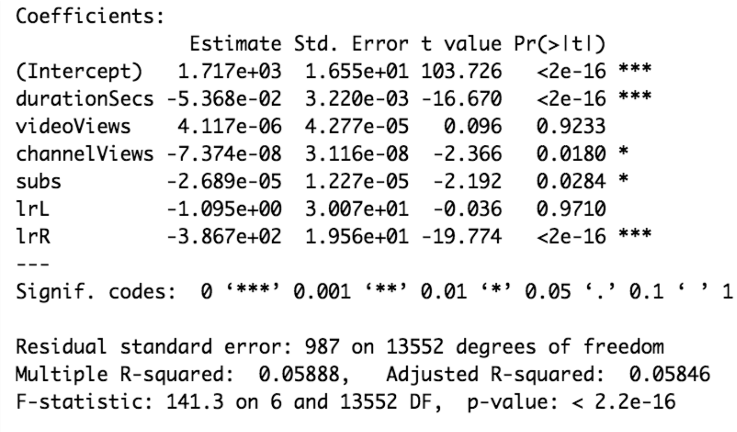
Appendix

#	Platform	Model	Split Type	Split Size	Train Accuracy	Test Accuracy	Train Precision	Test Precision	Train Recall	Test Recall
1	R	GLM	Even (50-50)	10%	0.864	0.81	0.966	0.997	0.807	0.737
2	R	GLM	Even (50-50)	50%	0.853	0.87	0.954	0.965	0.793	0.811
3	R	GLM	Even (50-50)	100%	0.845	0.844	0.954	0.948	0.783	0.78
4	R	GLM	Representative	10%	0.956	0.956	0	0	0	0
5	R	GLM	Representative	50%						
6	R	GLM	Representative	100%						
7	R	Random Forest	Even (50-50)	10%	0.873	0.81	0.956	0.966	0.82	0.737
8	R	Random Forest	Even (50-50)	50%	0.859	0.874	0.963	0.972	0.797	0.812
9	R	Random Forest	Even (50-50)	100%	0.853	0.846	0.966	0.968	0.788	0.778
10	R	Random Forest	Representative	10%	0.957	0.956	0.016	0	1	0
11	R	Random Forest	Representative	50%						
12	R	Random Forest	Representative	100%						
13	R	Decision Tree	Even (50-50)	10%	0.86	0.776	0.86	0.793	0.86	0.767
14	R	Decision Tree	Even (50-50)	50%	0.855	0.874	0.972	0.979	0.788	0.808
15	R	Decision Tree	Even (50-50)	100%	0.838	0.832	0.971	0.975	0.767	0.758
16	R	Decision Tree	Representative	10%	0.958	0.959	0.177	0.226	0.579	0.583
17	R	Decision Tree	Representative	50%	0.964	0.969	0.148	0.173	0.796	0.846
18	R	Decision Tree	Representative	100%	0.965	0.967	0.314	0.299	0.652	0.625
19	Python	SVM	Even (50-50)	10%	0.829	0.793	0.904	0.966	0.786	0.718

20	Python	SVM	Even (50-50)	50%	0.841	0.86	0.954	0.972	0.778	0.793
21	Python	SVM	Even (50-50)	100%	0.837	0.832	0.952	0.958	0.774	0.765
22	Python	SVM	Representative	10%	0.956	0.956	0	0	0	0
23	Python	SVM	Representative	50%	0.959	0.964	0	0	0	0
24	Python	SVM	Representative	100%	0.959	0.962	0	0	0	0
25	Python	Neural Network	Even (50-50)	10%	1	0.69	1	0.759	1	0.667
26	Python	Neural Network	Even (50-50)	50%	1	0.814	1	0.824	1	0.807
27	Python	Neural Network	Even (50-50)	100%	0.999	0.819	1	0.825	0.998	0.816
28	Python	Neural Network	Representative	10%	1	0.956	0.992	0.355	1	0.5
29	Python	Neural Network	Representative	50%	1	0.963	0.997	0.433	0.997	0.474
30	Python	Neural Network	Representative	100%	0.999	0.964	0.994	0.493	0.992	0.528

Appendix Table 1: RQ1 Model Summaries

To address RQ2, 30 models were attempted, however the computational limitations and inefficiencies within R, caused some models to crash. This problem was also encountered when training an SVM in R, therefore the same was implemented in Python instead.



```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)   1.717e+03  1.655e+01 103.726   <2e-16 ***
durationSecs -5.368e-02  3.220e-03 -16.670   <2e-16 ***
videoViews    4.117e-06  4.277e-05   0.096   0.9233    
channelViews -7.374e-08  3.116e-08  -2.366   0.0180 *  
subs         -2.689e-05  1.227e-05  -2.192   0.0284 *  
lrL          -1.095e+00  3.007e+01  -0.036   0.9710    
lrR          -3.867e+02  1.956e+01 -19.774   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 987 on 13552 degrees of freedom
Multiple R-squared:  0.05888,	Adjusted R-squared:  0.05846 
F-statistic: 141.3 on 6 and 13552 DF,  p-value: < 2.2e-16
```

Appendix Table 2: RQ2 Model 2 Summary

Note that Model 3 is not shown as the number of features makes viewing difficult.