An Exploration of YouTube's Recommendation Algorithm

- Authors -

Zachary Edwards Downs

Mia Ward

Arturo Mora

Abstract

As online video streaming only continues to grow [3], so does the number of people trying to make YouTube content creation their livelihood. This greater number of people depending on YouTube for their income makes understanding how YouTube's video recommendation algorithm functions an important piece of information for many. In this study, we attempt to gain insight into what kinds of videos YouTube is recommending in their related videos sidebar using datasets gathered from the YouTube Data API. This study analyzes how these related videos compare to the original video in terms of creators, views, likes, comments, and popularity. We observed that above average views, comments, and likes did not greatly affect how often videos were recommended. In fact we found the YouTube algorithm is recommending videos with below average statistics.

Introduction and Study Statement

In this day and age, many of us have gone on to YouTube and spiraled down a hole of mindless video watching not realizing that we keep selecting more videos to watch. One thing people probably have not done is wonder how and why we get the recommended videos that we do when we continue watching videos. For our study, we decided to take that curiosity and collect data from YouTube to get insight on why those certain videos are chosen and how.

Before beginning our study, we wondered how these recommended videos were chosen. Are they based on the videos you have liked/disliked, the videos you have watched or are watching, the channels to which you are subscribed, or even just random suggestions? With these thoughts in mind, we decided to base our study on these questions much like they do in "The YouTube Video Recommendation System" [2]. Davidson et al. [2] created a study that showed how these recommended videos get personalized to each viewer based on some of the same questions that our team has as well as with the slight difference of programs being used.

For our study, we will be accessing the YouTube API and retrieving the top 50 most popular videos for that day as well as the video information like views, comments, likes, etc. From these 50 popular videos, we will also collect the same information for the 15 related (recommended) videos given. After gathering all the information, we will use R to analyze the data and get a better understanding on how and why the recommended videos are selected, much like Tsingalis et al. do in "A Statistical and Clustering Study on Youtube 2D and 3D Video Recommendation Graph" [1].

Past Work

For our first work, we found the research paper "A Statistical and Clustering Study on Youtube 2D and 3D Video Recommendation Graph" by Tsingalis, Pipilis, and Pitas [1]. The data scientist in the paper uses Breadth First Search and Depth First Search to crawl YouTube using the YouTube API. They categorized their data into nodes (video title, views, number of likes and dislikes) and edges (connection between a parent video and the related videos that API returns for a parent video; weight to each edge is assigned according to the order the related videos are returned). Using this data, they plotted the relation of the number of views between a parent video and the number of views of a relevant video. The graphs they came up with gave them the notion that videos are plotted together based on their popularity.

For our next reference paper, we found another research paper "The YouTube Video Recommendation System" [2]. When a user goes to YouTube they go for a reason and giving them the proper personalized video recommendations is the best way to keep them coming back. That is why they have to be updated regularly to reflect the user's recent activity, but also highlight the broad spectrum of content available on the site. By regularly updating the videos, engineers have to consider all challenges and collect data from individual users

about what they are doing (liking, subscribing, etc.) and watching. After collecting data, they have to rank this into sections (clusters) to figure out which videos are relevant to the user. They do not talk about the cluster results, but they do analyze that video recommendations account for about 60% of all video clicks [2010].

The paper "Exploring Sharing Patterns for Video Recommendation on YouTube-like Social Media" by Ma et al [3] is a good resource to understand techniques for personalizing video recommendations. Through studying how people share and watch videos from video streaming sites on non-video social media sites, the authors were able to construct an algorithm for recommending users videos based on their watch history and video sharing habits. Personalization of recommendation is a vital part of studying how streaming sites recommend to users a list of videos. More useful is their findings that 10% of videos account for 80% of views on video streaming platforms.

In "The Impact of YouTube Recommendation System on Video Views" by Zhou, Khemmarat, and Gao [4], they talk about the YouTube video recommendation system, how the system works, and how this affects the views of videos. The authors first collect video data through HTML scraping and the YouTube API which allows them to use this information to find out how videos are being recommended. With the collected data, the authors separated the videos into categories based on views to see if this had any impact on what videos were recommended on the current video beingvwatched. When doing the research, they found a strong connection with the view count of the video the user was watching and the average view count of the top recommended videos. They also discovered that the recommendation system accounts for about 30% of the overall views right behind the YouTube Search.

In the paper "Virality over YouTube: an Empirical Analysis", GoharFeroz Khan and Sokha Vong delved into seeking reasons why a video goes "viral" on YouTube. They modeled their data that they collected from the YouTube API based upon the 100 all-time-most-viewed YouTube videos and their information, such as what they called "virality" of a video -- likes, dislikes, favorite count, view count, and comment count. Their findings made clear that that popularity of videos is not only a function of the YouTube system, but the network dynamics (e.g. in-links and hits counts) and the fan base/fame have played crucial roles in what determines a "viral video".

To understand how YouTube recommendations can be bought, and the prominence of advertisement on the platform, the paper "The Institutionalization of YouTube: From User-Generated Content to Professionally Generated Content" by Jin Kim [6] yields insightful information. It discusses YouTube's transformation from purley user generated content into a mix with professional content produced by companies. It also mentions how YouTube sells feature spots to creators and keywords so that a video appears first when that topic is searched. It comes to the conclusion that YouTube has become an extra source of revenue for professional companies despite their initial misgivings about the new media. One topic it fails to adequately address is the melding of user and professional as users also start to take advantage of these systems to grow and become professional channels not linked to larger corporations.

Methodology and Experimental Design

After much consideration, we have decided that the best way to collect relevant data on YouTube's Recommendation Algorithm is by accessing the YouTube API. By writing a Python script to make requests to the YouTube API, we are able to gain access to various pieces of information that are vital to our stated goal. Below, we cover how we access the API and give a breakdown of how our Python script interacts with the API to obtain and format our data into useful information. We also describe the ways we plan to utilize R to analyze our data and make meaningful discoveries about YouTube's Recommendation Algorithm.

As stated, we would like to explain more about how we gather our data from the YouTube API considering this is one of the most vital steps in our process besides analyzing our data. We developed a fairly straightforward Python script that utilizes an API key to build a link to the YouTube API. This script generates unused file names as a CSV in which to store our data. After everything is set up, we are able to make requests to the YouTube API

where we request the 50 most popular videos and their related videos. Among these videos, we are storing the most important information about them, such as video ID, title of the video, channel ID, as well as how many views, likes, and comments each video currently has. Since we would like to see the relationship between videos, we store the video ID of 15 related videos that correspond to the 50 most popular videos that we initially gathered. Finally, we follow the same information gathering process as we did with the 50 videos, but this time on the 15 related videos. After the Python script has completed, we have a CSV file consisting of 50 videos with their corresponding 15 related videos, including information about each.

To help understand how the like system works for recommended videos, we would like to see which videos, popular or related, have the most likes among them. This can be beneficial to our research because the "like system" is one of the main factors that determines a satisfactory video from an excellent video. To find out which videos, popular or related, are the ones that get the most likes, we will make a simple R script that will check a popular video against it's related videos and, if the popular video has more likes, then it gets a tally; otherwise, the related will get a tally. At the end of the loop, we will have a clear number how many popular videos or related videos have more likes.

We will be testing if videos recommended by the YouTube algorithm clearly prioritize videos with above average likes, views, and comments. This test will reveal how often YouTube is recommending extremely popular videos in comparison to less popular videos. This will be done by gathering how many times a video is recommended by the algorithm as well as by calculating and assigning category scores for every video.

Another way we decided to use our data is to get the mean for the subtopics in popular videos and see if that is an aspect as to why the popular videos are chosen. To get the mean for each subtopic, what we did was get the median for the views, comments, likes, and dislikes on each video and then with the median results from each video we got the mean.

We will also be testing our whole dataset to see which videos are the most popular among all the videos. To be able to figure this out, we will look at how many occurrences each video has related to how they are recommended from YouTube. To first figure out the occurances, we will write an R script that will loop through all the videos and keep a tally of every time a video occurs. Once this loop has finished, we will have a list of videos and how many times they occur in our data. For the second part, we will loop through this list of videos and if a video has more than two occurrences, then it will be added to a list of the most popular videos. Finally, we will have a list of video IDs that are the most popular videos among all the videos.

The final way we will experiment with our data that we collected is by constructing a decision tree and running a confusion matrix to determine how many of the related videos in our data also qualify as "popular" videos. To build the decision tree, we would be averaging the attributes (views and likes) on the already known popular videos and with this information, construct the decision tree around these attributes. We will also be making a confusion matrix to see how split our attributes really are and how effective our decision tree is in determining if a video is popular.

Experimental Results

Related Video Likes

Through a series of trial and error collecting data, we were successfully able to find out the relationship of likes per video between popular videos and their corresponding related videos. From our R script, we were able to get a look at exactly how many videos, popular or related, experience more likes. We have tested four data sets, four different days of data from YouTube, to get these percentages. For our first day of gathering data, we found that 52% of liked videos were from popular videos while only 48% came from related videos. On our second day of collecting data, we ran our script again. This time we found that only 39.3% of likes came from popular videos while an astounding 60.7% came from related. Looking to find a definite conclusion to our experiment, we ran our script on two more days of data. For day three, we found popular videos made up 46% of the liked videos while related made up 54%. Finally, on the fourth day, we found that, again, popular videos made up a smaller percentage of the total liked videos at 38% while related videos dominated with 62%. With more time, we would like to run even more data sets with our script to see if this stays consistent.

More Often Recommended

Next we attempted to find determinant factors for videos that were recommended more than once by the algorithm. To do this, we compiled the number of times a video occured in our dataset and kept the ones with values greater than two. Then scores were given to each video to determine how above average they were in terms of views, comments, likes, and dislikes. Means were calculated for each of the previously mentioned categories, and then every video was compared against it and given a score in each category. A 0 was given if the video fell below the average in a category, and a 1 if it was above it. This was different for dislikes where a -1 was given for higher than average dislikes and a 0 otherwise. All four categories of views, comments, likes, and dislikes were summed to give a total score with a maximum of 3 and minimum of -1. Finally each of these videos was given a true or false value on whether it was also a popular video. For the first data set, 3.41% of related videos appeared more than once. Of this set 25% were also popular videos. The majority with 60% had an above average total of 0, 20% with a rating of 1, and 20% with a rating of 2. Only 15% were recommended more than twice, and this percentage was also popular videos as well. One had a score of 0, and and the remaining had a score of 2. The other data sets will be explained via a table using the formatting above.

Videos Recommended More Than Once

Percent More Than Once	Percent Popular	Percent Score For -1, 0, 1, 2, 3	Percent More Than Twice	Percent Popular	More Than Two Scores
3.41%	25%	0%, 60%, 20%, 20%, 0%	15%	100%	0, 2, 2
6.19%	24.3%	2.7%, 89.2%, 5.4%, 2.7%, 0%	27%	50%	0, 0, 0, 0, 0, 0, 0, 0, 0, 0
8.48%	33.3%	1.9%, 79.6%, 5.6%, 13%, 0%	30%	43.8%	-1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1
8.35%	31.4%	2%, 84.3%, 11.8%, 2%, 0%	37%	52.6%	-1, 0, 0, 0, 0, 0 ,0 ,0 ,0 , 0, 0, 0, 0, 0, 0, 0, 0, 0 , 2

Mean of Popular Data Subtopics

After a couple of tries, we finally got our data to successfully collect the median for each subtopic in the popular data. The issue we kept having was trying to remove the empty/Null rows in the data. Once this was done, we were able to successfully get the results needed. Once we retrieved the median for each video, we then took the results of each subtopic for each video and obtained the mean of each one (views, comments, likes, dislikes). There really was not a prominent pattern in the data for each subtopic besides the fact that if the topic started at a certain range (hundreds, thousands, millions), then it would stay the same range throughout each video collected. The mean for each one was views: 1467147, comments: 3073.4, likes: 28377.4, and dislikes: 1087.4. From these results we can see that views has the higher mean and dislike has the lower mean which may imply that most likely YouTube uses the view count as a way to decide what video is shown in the popular videos.

Most Popular Videos

After we had run our algorithm on all four days of datasets, we had a clear result as to what videos were the most popular among all the videos that we had gathered. On day 1, we found there to be 3 videos. Continuing on, we found 10 videos on day 2 and 16 videos on day 3. On the final day, we found there to be 19 videos that had more than 2 occurrences in our data.

Decision Tree

For our decision tree based on views, comments, likes, and dislikes we were able to achieve between 93-95% accuracy depending on the data set. However, it achieved this accuracy only because it guessed non-popular correctly often, and the number of non-popular videos was higher. The error percentage almost exclusively comes from incorrectly labeling non-popular videos as popular. The decision tree did not remain constant either as if run on the same dataset, the layout would continuously change. Using k-fold cross-validation in an attempt to remedy this did not give better results.

Discussion

Related Video Likes

After collecting our data and testing each one, we have concluded, for the most part, the related videos will usually have a higher count of video likes than the popular video with which the user started [5]. The only time our conclusion was proven incorrect was on the first data set, which had more likes on the popular video, but just by 4%. After that, each video tested proved our conclusion. With the given information, we can clearly see that the YouTube Recommendation Algorithm allows the users to connect to content they are more likely to enjoy. This, in return, will allow them to give more likes to the related video than the original video on which they started.

More Often Recommended

Based on our results, an average of 6.61% of related videos is recommended more than once on the most popular YouTube videos. We believe this provides evidence that YouTube attempts to recommend a wide array of videos to entice the user rather than repeatedly pushing the same content to eventually get clicks. An average of 28.5% are also popular videos themselves. This low percentage shows that the algorithm prioritizes non-popular videos more than popular videos when recommending. It is possible that recommending non-popular videos on popular ones is how YouTube tries to promote smaller creators. The data also shows that on average 78.28% of videos recommended more than once have a score of 0, meaning their views, comments, likes, and dislikes are all below average. This data in conjunction with non-popular videos being pushed further strengthens the evidence that the YouTube algorithm is attempting to advertise less popular videos.

In the case of videos that were recommended more than twice, they make up on average 2% of all related videos on popular videos, and 27.25% of videos related more than once. We believe this shows that YouTube pushes a relatively small amount of videos a lot to their users. On average, 61.5% of these videos were popular which is around twice as high as videos recommended more than once. This indicates that videos with which YouTube repeatedly tries to grab viewer attention are more likely to be popular videos. The scores again show a tendency toward 0 with an average of 76%, and an even higher 90% average if the outlier of the first data set with a low sample size is removed. The data shows a high overall lean towards videos that do not receive above average views, comments, likes, or dislikes. This shows that YouTube is recommending videos heavily on categories not tested here, such as video tags, titles, or channel.

Mean of Popular Data Subtopics

From the medians and means, we could see that there really is no pattern in the data collected because some days the subtopics would be high, but another day it would be low. One thing that did not change though was that whatever the range in which it started, it stayed the same throughout. With the mean results being views: 1467147, comments: 3073.4, likes: 28377.4, and dislikes: 1087.4, we can see that views has the higher mean than the rest of the topics which could show that YouTube uses views as a way of placing a video in the popular section.

Most Popular Videos

Since we were given quite a few video IDs each day of the most popular videos from our dataset, we were able to look at each video associated with their ID and identify what kinds of videos were being suggested more than twice by the YouTube Recommendation Algorithm. After searching most video IDs, we found that the most recommended video, or the most popular videos, were those of sportscast highlights, such as *"THUNDER vs TRAIL BLAZERS | MUST-SEE Finish That Will Leave You SPEECHLESS! | Game 5"* and *"Everton v. Man United | PREMIER LEAGUE EXTENDED HIGHLIGHTS | 4/21/19 | NBC Sports"*. Although we personally did not find much use from these videos given we are not sports fans, we did find some familiar creators, such as The Try Guys, Good Mythical Morning, James Charles, BBC, and Lil Dicky. These familiar faces allowed us to gather that although most of the videos among the most popular videos from our dataset were sports related, we think everyone would be able to pick out a few creators that they recognize. This in turn allows us to infer that the YouTube Recommendation Algorithm allows for a variety of videos among their top ranks and mostly accommodates to all their users in one way or another.

Decision Tree

The decision trees we obtained corroborate our findings that views, comments, likes, and dislikes are not major influences in how YouTube recommends their videos or how videos get popular on the site. The inability to get a stable predictive tree on these attributes alone shows that there are attributes on which YouTube is recommending videos, such as thumbnails, video length, etc, that are most important. Future attempts to create predictive models for what videos will become popular on YouTube will need to include the video attributes we left out to increase their accuracy.

Future Research

The most significant way to expand upon this paper's research would be to include video thumbnails, titles, video tags, and video length into the research and see if any defining attributes emerge. Our paper showed the numerical attributes of YouTube videos do not play a significant role in what videos get recommended, nor how often. Thus, it is likely that videos are recommended on their non-numerical attributes. Future research would do well to take

these attributes into account when studying the algorithm. In addition, using the YouTube API to select videos would be integral to getting a deeper understanding of how the algorithm personalizes to a user based on the videos they watch. The topic could also be expanded by seeing how sharing of YouTube videos on social media sites, such as Facebook and Twitter, affect a video's growth.

References

1. Tsingalis, I. Pipilis, and I. Pitas, "A statistical and clustering study on Youtube 2D and 3D video recommendation graph," *2014 6th International Symposium on Communications, Control and Signal Processing (ISCCSP)*, 21-23 May 2014. pp. 1-4.,
   DOI: 10.1109/ISCCSP.2014.6877872.

2. J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, and D. Sampath, "The YouTube video recommendation system," *Proceedings of the fourth ACM conference on Recommender systems*, pp. 293–296, Sep. 2010.

3. Ma, Xiaoqiang, et al. "Exploring Sharing Patterns for Video Recommendation on YouTube-like Social Media." *Multimedia Systems*, vol. 20, no. 6, 2013, pp. 675–691., doi:10.1007/s00530-013-0309-1.

4. R. Zhou, S. Khemmarat, and L. Gao, "The impact of YouTube recommendation system on video views," *Proceedings of the 10th annual conference on Internet measurement - IMC 10*, pp. 404–410, 2010.

5. Gohar Feroz Khan, Sokha Vong, (2014) "Virality over YouTube: an empirical analysis", *Internet Research*, Vol. 24 Issue: 5, pp.629-647,
   DOI: https://doi.org/10.1108/IntR-05-2013-0085

6. Kim, Jin. "The Institutionalization of YouTube: From User-Generated Content to Professionally Generated Content." *Media, Culture & Society*, vol. 34, no. 1, 31 Jan. 2012, pp. 53–67.,
   doi:10.1177/0163443711427199.