YouTube Clickbait

Tristan Ainley, Knox Cavitt, Hunter Kitts, Pei Lin, and Shreyank Patel

***Abstract*—Clickbait is a notable problem of digital media. It takes up digital space and wastes a viewer's time. Though clickbait content is subjective in nature, we explore the different data that YouTube stores for each video and analyze if the data can give insight to whether a video is clickbait or not. In this paper, we explore YouTube videos with text analysis, web crawlers, data management, and machine learning, we conclude that the title of the video is still the best determining factor of clickbait content. The dislike to view ratio does help add more analytical detail of a video.**

I. OBJECTIVE

YouTube is a video sharing social media platform that has been growing ever since its launch in 2005. Due to factors like monetization and virality, clickbait videos are becoming more apparent. Oxford dictionary defines clickbait as "content whose main purpose is to attract attention and encourage visitors to click on a link to a particular web page". For our project, we want be more specific and add that clickbait videos on YouTube are often filler content with the main purpose of increasing viewer count that delivers minimum value to the viewer. Experienced users of digital media can spot clickbait with the naked eye, but we would like to analyze it by web scraping data from YouTube and utilize a machine learning algorithm to see if the data collected can give insights on clickbait content. The objective of our project is to determine the defining factors of clickbait videos on YouTube. We looked into many key features, such as the title of a video, like and dislike count, and dislike to view ratio. Clickbait has a negative connotation for many users; however, it is currently unclear how it affects the performance of a video and its online communities. We want to determine if the use of clickbait affects creators and communities positively or negatively. From this, we can determine what categorizes a title as clickbait.

II. MOTIVATION

Social media platforms like this have become more than a place of self expression. Views have the potential of becoming monetized, and with every "thumbs up", comment, and subscription, a business can form. From the point of view of a content creator, having decent exposure or going viral feeds the YouTube algorithm and ensures that their videos are seen. The motivation for this project is to understand what attracts views and encourages viewer engagement. For content creators looking to maximize their income from YouTube, understanding key features of a video that would attract more advertisement revenue would be beneficial. For viewers who want to avoid purely clickbait videos or put a parental filter on certain clickbait contents, an analysis can label and distinguish . Overall, this analysis strives to determine how to navigate through the overcrowded contents of YouTube, either as a content creator or a viewer.

III. DATA

Data collection for this project was achieved with the use of YouTube API calls. Functionality of YouTube's API calls has two categories, one being to add YouTube videos or information to other websites and the other is to search for content on YouTube. Therefore, the amount of API calls that can gather data is limited to two primary options in the search category. The first is to gather video data through a search by keyword or location. The latter is to retrieve data though a video ID or by YouTube's "most popular videos". Hence, our data collection is limited to four options being, search by keyword, search by location, retrieve by id, or retrieve by most popular videos. Considering our primary goal, we decided using retrieve by most popular videos would give us the most amount of data given our limited time. Originally, the idea of using a scraper bot was also present. However, it became apparent the bot would ultimately be limited to YouTube's algorithm suggestions dissimilar to YouTube's most popular videos. YouTube's most popular videos listing is not a chart of videos with the most views. Rather, it is determined by a YouTube algorithm and is updated regularly.

All calls to the API for this project were written in Python, and an example of these calls can be found in Listing 1. Each call to the API is a request. If the request finds a valid video, it returns a response, which contains video resource data within it. The video resource data contains numerous values that range from the video's ID to content ratings in every country [1]. From the data, 12 values were selected that we believed could accurately predict if a video's title is clickbait.

```python
youtubeV3 = build('youtube', 'v3', developerKey=api_key)
        request = youtubeV3.videos().list(
            part="id,statistics,contentDetails,snippet",
            chart="mostPopular",
            pageToken=none
        )
        response = request.execute()
```

Listing 1. list (most popular videos) API call

From the data, 12 values were selected that we believed could accurately predict if a video's title is clickbait. These values include the video ID, video title, channel ID, channel title, description, category the video is in, duration of the video, view count, like count, dislike count, favorite count, and comment count. From October 5, 2021 to November 28, 2021, API calls were made every day to YouTube's most popular videos chart. This span of 55 days accumulated an estimated 3,000 unique videos. Because this number was smaller than we initially intended, we added Mitchell Jolly's data set to our data as well[2]. Jolly's data includes videos from the "most popular videos" chart from several countries' over a six month period. We used only the United States data as we did not want to skew the results we had already accumulated. To make sure all the data would follow the format we had already created, an API call was made to each video ID in Jolly's US data set.

In conclusion, data collection resulted in 6,595 unique videos with 12 values each. Although Jolly's US data set had 30,000 videos, many of them were not unique or had been removed from the platform. In trying to keep our data consistent, we added the available videos to the data set through API calls.

To consolidate the data, some values, such as "comments disabled" were slightly edited. Originally an API call to a video where comments are disabled returns an empty field for comments count. To remove this ambiguity, a Boolean value was set. If the comment count was not empty, then we can assume they were not disabled, and it was set to 1. When comments were disabled, some other values had to be set to 0, such as the comment count. This was achieved as the data was written to the file, not by the API call itself. This problem also occurred in some videos where creators block users from seeing their like and dislike counts. To combat this, another Boolean value was set when reading these values. Once all the initial data was read in, we then created ratios based on some data values. One example of this would be the "like to dislike" ratio. By creating multiple ratios in addition to the values gained from the API calls, we could introduce more features to any data models chosen by the team. Consolidation and slight restructuring of data resulted in another 7 values that could be used.

IV. MODELS

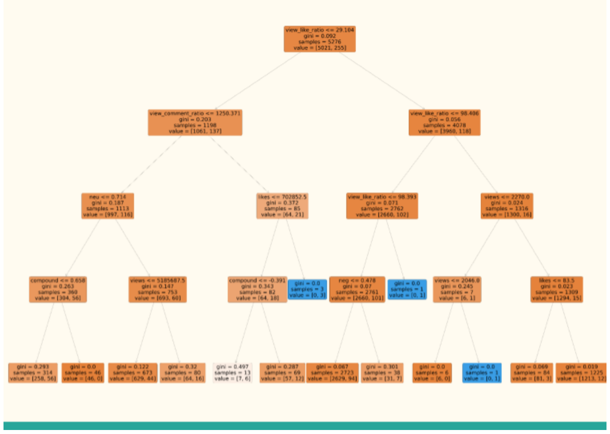
For this project we decided to implement two forms of classifiers, a neural net and a decision tree. Below follows the implementations and design decisions in detail.

A. Neural Network

For our implementation of the Neural Network we used sklearn's MLP classifier. The reason we selected MLP was our familiarity with the classifier. The first step was taking our data set and pulling out the features we wanted to train on. This resulted in us using 16 features. Once the training data was assembled we began by creating a basic model to make sure the data pipeline worked and would allow the training of the model. Once the pipeline was configured properly and we were getting some results we need implemented gridsearch cross validation in order to not only create the best version of the model possible, but to also increase our accuracy. The parameters we tested for were hidden layer sizes, activation function, the learning rate, and the initial learning rate.

B. Decision Tree

The decision tree classification was made using sklearn's Decision Tree classifier. The classifier was trained on 80% of the data and tested on 20% of the data. The decision tree classifier splits the data into two subgroup at each tree node based on whether or not, the data meets the conditional statement of the tree node. Using the sklearn's decision tree classifier, we were able to select the tree depth to maximize the accuracy score of the classifier. We performed hyperparameter optimization on our decision tree classifier. From our hyperparameter optimization, we were able to determine that a tree depth of 4 produced the highest accuracy score for our classifier. The figure below illustrates the decision tree our classifier was able to produced based on the data it was provided.



V. Results

A. Neural Network

From the Neural Network we were able to predict 64.1% of the clickbait videos accurately.

B. Decision Tree

From our decision tree classifier, our team was able to determine that view to like ratio, view to comment ratio and views were the most important data parameters to determine whether or not a video is clickbait or not. Our decision tree classifier was able to correctly classify 59.56% of the data.

VI. PRIMARY CHALLENGES

For data collection, non-Latin characters and emojis became a large hurtle. Many video titles and descriptions include these characters, so both storage and printing of the characters are required. For example, we originally stored our data in a text file. This is because emojis in csv does not directly convert without several added steps. When the team began to analyze the data, we knew some conversion from text to csv would be helpful, but problems arose in conversion. Encoding is the primary culprit. Different data storage techniques can significantly change the data. An example of the title stored in a text file would be: ”DIY TACO PIZZA 🌮 🍕”. In comparison the csv file displayed the text as: ”DIY TACO PIZZA ðŸŒ®ðŸ •”. The raw data is all Unicode values, so as long as the input is correct, it does not affect the output. However, our testing Windows machines could not write the characters to a file. A separate terminal was needed just to write UTF-8 characters [3].

Another challenge we encountered came from the actual YouTube API and our resulting video dataset, or more specifically lack thereof. The YouTube API itself allows users to pull information about 10,000 quota per day of videos, which could give us about 550,000 videos, both unique and non-unique, over 55 days of data collection. As mentioned previously, we pulled video information from YouTube’s most popular section. However, YouTube’s most popular videos section restricts users to accessing information about the 1st through 200th videos. As such, we could not collect information about the 201st video and beyond. This coupled with the fact that videos often maintain a spot in the most popular section over multiple days limited us to pulling information of only about 100 unique videos per day.

This resulted in our initial dataset of about 3,000 videos. We pulled from the most popular section as it was the optimal option to avoid bias as other search options required our input of specific videos. This would have given us more option to gather data, however our input could have easily skewed the data analysis results based on what videos we inserted into the dataset. As such, we decided to incorporated other datasets of YouTube videos. After including Jolly’s data for the United States and filtering for unique videos, our final dataset still unfortunately lacked sufficiency for our data analysis when training with the neural network and decision tree. Ideally, if this project could be repeated, members should collect data over a longer period of time as well as utilize the API’s large quota restriction to pull more videos’ information while still maintaining impartial stance.

VII. CONCLUSION

This paper presents the outcomes of our analysis of YouTube videos for clickbait content. Our conclusion is that the title itself is the best factor that determines if something is clickbait. Word and sentiment analysis avoids factors like the content of the video itself that may cause a higher count of dislike. It is important to understand that clickbait does not always correlate to the meaningfulness of the content itself. Some content creators use clickbait to attract viewers to get ideas across. In those cases, the dislike to view ratio may not reveal unwanted clickbait content. The choice to use clickbait titles boils down to the preference of the content creator. Furthermore, our analysis is a start to looking into the social behaviors that are collected on social media. Each click on digital media quietly collects behaviors that are unsaid and seemingly unseen.

VIII. FUTURE WORK

Unfortunately, some of the future work of this project has been shut down by YouTube’s decision to remove the count of dislikes. This information was essential to the project as it produced ratio values we could measure by, although by our results it has lost some validity. Starting on December 13, 2021, users can no longer see a dislike count. Any new API calls made after this date will not have a dislike count included, thus limiting the amount of new information that can be gained. One feature that can still be implemented, however, is emotionally detection through thumbnails. With this information, videos could be further categorized into facial emotion sections. OpenCV, for example, could detect not only emotion but possibly text in the thumbnail. There are many instances in which text in the thumbnail is not mentioned in the video title or description. This would require a lot of cleanup, as the information would not be clean, however.

Two other possibilities for future work have also been developed from the project. The first is a website or app that allows users to get a rating of how clickbait their title is. This would allow users to either reduce or increase the amount of clickbait in their title. Second is the ability to generate a clickbait title given some keywords. YouTube is already working with auto generating thumbnails so it’s possible this project could help in the generation of titles as well.

IX. ORGANIZATIONAL CHART

API call team consists of Hunter and Shreyank. The data team which performs data scraping and data filtering consists of Tristan. The analysis team consists of Knox and Pei.

09/28 Research
The API team conducted research regarding Google’s API for YouTube. The data team looked into different libraries that could help scrape the data. The analysis team research Youtube’s features.

10/05 Initial Data Collection
The API team ran the first set of Youtube API calls to collect the data. We agreed to collect data from the popular page. The data team looked at the data and determined irrelevant features. The analysis team started to implement a neural network with the data collected.

10/19 Further Data Collection
The API team continued to collect data through API calls. The data team created additional features like the

”like to dislike” ratio. The analysis team selected 16 features to train.

11/02 Data Analysis

The API team fixed major problems with symbols and emojis in the data collection. The data team removed duplicate videos. The analysis team worked on the data pipeline.

11/16 More Data Analysis

The API team consolidated all the data collected. The data team did the final filtering of the data. The analysis team performed a gridsearch cross validation. Shreyank also made a decision tree and conducted sentimental analysis on the video titles.

References

[1] Google, “Youtube videos resource representation,” 2021, last updated 2021-11-18 UTC. [Online]. Available: https://developers.google.com/youtube/v3/docs/videos#resource

[2] M. Jolly, “Trending-youtube-scraper,” 2018, last updated 2018-11-19 UTC. Trending-YouTube-Scraper by Mitchell Jolly (DataSnaek) is licensed under BSD 2-Clause, 2018. [Online]. Available: https://github.com/mitchelljy/Trending-YouTube-Scrape

[3] Microsoft, “Windows terminal,” 2021. [Online]. Available: https://www.microsoft.com/en-us/p/windows-terminal/9n0dx20hk701#activetab=pivot:overviewtab