Predicting Dislike Counts Using Sentiment Analysis on YouTube Video Comments

Hammad A. Siddiqui
CUNY - Hunter College
New York, New York
hammad.siddiqui
@*macaulay.cuny.edu*

Iris Shakya
CUNY - Hunter College
New York, New York
iris.shakya27
@*myhunter.cuny.edu*

Jianfu Jiang
CUNY - Hunter College
ew York, New York
jianfu.jiang16
@*myhunter.cuny.edu*

Abstract

YouTube has emerged as one of the most ubiquitous social media platforms over the past few years. In November 2021, despite its popularity, the company made the controversial decision to remove the dislike count for videos, removing a useful metric many users had come to rely on to assess the quality of content they were about to watch. To address this issue, we propose a new approach that uses the sentiment analysis of comments on a YouTube video among other features to predict the dislike count for that video. To do this, we employed algorithms like LinearRegression, RandomForest, and K-Nearest Neighbor. We used two different datasets each with unique sets of pros and cons. This research presents an innovative approach to understanding the quality of video content using NLP that seeks to enhance user experience.

1 Introduction

The rise of social media and content creation platforms over the past two decades has radically changed the way people consume and share content. Youtube has grown to be one of the most popular and widely used platforms, boasting roughly 2.68 billion monthly active users in 2023. The company plays a critical role for many of its creators and consumers. However, In November 2021, YouTube made the controversial decision to stop publicly displaying the dislike counts for videos, citing concerns for the content creators' mental health among other reasons. Curiously, the dislike count would still be available to the video uploader to see in the backend, ostensibly contradicting the initial reason. Regardless of the rationales provided, this decision removed an important metric that allowed users to quickly figure out whether a video was worth their time and attention–the primary currency of the digital age. The like-to-dislike ratio was an easy way to determine the usefulness of a tutorial or the entertainment value of other videos. Without this metric, viewers essentially go in blind when looking for videos on a certain topic.

In this paper, we propose a novel solution for this issue. By using sentiment analysis on the comments of YouTube videos, we aim to develop an approach that can accurately estimate the number of dislikes a given YouTube video would have using the aggregate sentiment of all the comments for that video as a key factor. We hope to address what kind of insights we can derive regarding a video from the comments and other features.

2 Research

Since our approach is a relatively novel one, there wasn't much relevant previous research available for us to draw from. However, we were able to find three papers, each of which we were able to draw some benefits from:

Classifying YouTube Comments Based on Sentiment and Type of Sentence

This paper analyzes youtube comments and then categorizes them based on sentiment as well as sentence structure into the following categories: positive, negative, interrogative, imperative, corrective, and miscellaneous. One of the key aspects of this paper was the relatively small dataset used: only 10,000 comments. Although easier to handle in terms of processing, storing, and analyzing on top of allowing for a greater focus on quality, smaller datasets face issues like limited generalizability, are less suitable for more complex models which require large amounts of training data and are more difficult to confidently draw statistical conclusions from. In our research, we simplified the categorization into just positive, neutral, and negative and we chose to address the plethora of

issues plaguing smaller datasets by ensuring that ours was relatively larger. [Siersdorfer et al., 2010]

SenTube: A Corpus for Sentiment Analysis on YouTube Social Media

The paper provides a dataset of annotated YouTube comments accounting for information content and sentiment analysis. This paper brings up a novel point we hadn't considered in our initial idea formation which is that on top of the sentiment analysis we were already planning on doing, it also accounts for the relevancy of comments to the content of the video. However, the paper is quite old and was written before the dislike button was removed and the actual dataset the created seems unavailable online. After initially coming across this paper, a solution we proposed was to treat comments that are irrelevant as neutral in our project using potential rudimentary solutions such as enforcing a minimum word count for a comment to be valid. However, when we experimented with the YouTube API, we found that it could automatically order by relevance. Then, when we switched to the Kaggle dataset, it informed us that "The YouTube API is not effective at formatting comments by relevance, although it claims to do so. As a result, the most relevant comments do not align with the top comments at all, they aren't even sorted by likes or replies." [Uryupina et al., 2014]

Measurement and Sentiment Analysis of YouTube Video Comments

The paper strives to analyze the sentiment of the comments on the popular social networking platform. The interesting bit was the usage of ratio of sentiment polarity and like/dislike metric, and views with comment/likes which also inspires our thesis. The paper separates its analysis into topics of videos like Entertainment, Science and Technology and/or Sports and extracts hidden characteristics among them. The sentiment results that can be gleaned from the comments that belong to each genre are distinct i.e. Science/Tech happen to have neutral sentiment which makes it impractical in our prediction model. The key takeaway from this paper is the possibility of predicting the dislike count based on sentiment analysis of the comments. [Xinyu, 2022]

3 Data and Methodology

We initially used a pre-existing dataset created by Mayur Deshmukh[1] that contained 40,949 rows of video ids along with their view counts, comment counts, and dislike counts. When we ran our comment scraper code on this dataset which extracted at max 100 comments per video, we were only able to extract roughly 5000 videos worth of comments because the scraping code was too time consuming, with each 100,000 comments taking roughly an hour. Then, after analyzing the original dataset a bit more, we realized that of the 40,949 video ID's, only 6,351 were unique. When we tried scraping the comments for just the unique video ID's, only 44% were accessible with the API. However, the .csv files that were being generated were ballooning in size with some of them growing to over 78 GB even after the scraper had finished running.

Despite our best efforts to wrangle the plethora of issues, the 78 GB files made further use of this dataset impossible. Which is why we decided to switch to a new dataset created by Mitchell J.[2] that already included over 600,000 scraped comments for 2266 videos with the like and reply counts for each comment along with another csv dedicated to just the videos including the total like count, views, likes, dislikes, comment total for each video. However, since we only had comments for 2266 videos, the rest of the 5,732 videos in the videos csv weren't useful for our sentiment analysis. Using a hashmap, we were able to figure out that the over 600,000 comments were not distributed evenly among the 2266 videos–some videos had comments numbering in the hundreds and others had barely 10. We will address the implications of this ratio later on.

Using the TextBlob package on the text of each comment, we were able to do a spell check before using the VADER package to find the comment's polarity. Although VADER provides polarity metrics in terms of a positive, neutral, negative, and compound score, we opted to only use the compound score values for our models. To aggregate the sentiments for all the comments for a video, we simply calculated the mean by adding up all the compound scores for the video and then dividing by the values in the hash-map mentioned above which were the comment counts per video. The resulting scores were between -1 (indicated a highly neg-

[1] https://github.com/MayurDeshmukh10/youtube_analysis/
[2] https://www.kaggle.com/datasets/datasnaek/youtube

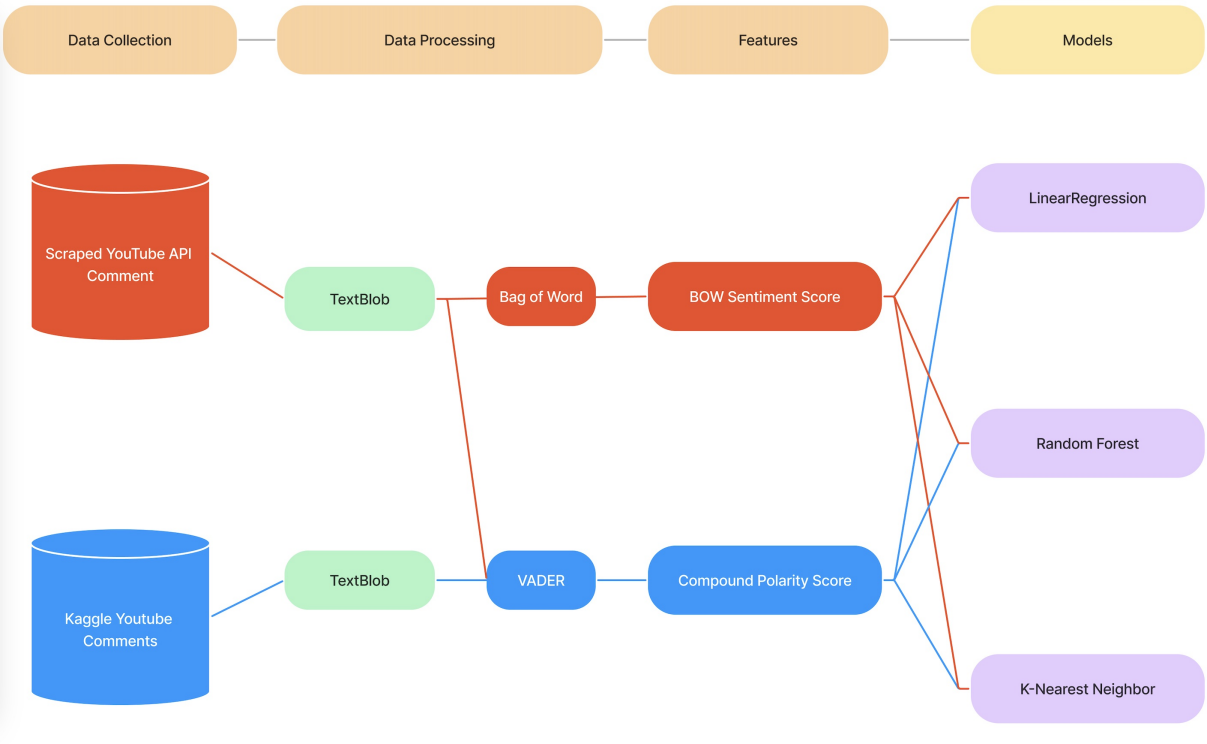

Figure 1: Project Design

ative oval sentiment) and +1 (indicating a highly positive sentiment). This score serves as a valuable metric when seeking a single, unidimensional measure of sentiment for a given video. Referring to it as a "normalized, weighted composite score" is an accurate description.

4 Baseline

Parallel to when we were running into all these issues, we were setting up our models using dummy data based on the initial Deshmukh dataset, only referencing likes, dislikes, views, and comment counts–not sentiment analysis–for videos. When processing this dummy data, we ran into an issue when reading in the .csv file using our read_csv module where it would register the commas contained within the comment strings as column delimiters. However, we were able to solve this problem by only importing specific columns instead of processing the entire file as well as immediately casting the results into a list.

For this scraped data, we used the Bag of Words approach where we created a list of key positive words and key negative words we would search for in our comments and increment or decrement a polarity score for a comment depending on how many of positive or negative words it contained from our two lists, respectively. To aggregate all the polarity scores for a video, we simply calculated the mean.

4.1 Bag of Words Implementation

Bag of Words is a text representation technique widely used in NLP and the industry for text processing. In this project, we are dealing with Youtube comments which may not be grammatical. The Bag of Words approach assumes that the order of words in a sentence or comment is not important, only a word's presence or lack thereof affects the sentiment score. The resulting sentiment scores, which are made up of three columns–positive, negative, and neutral–can be used as the input to machine learning algorithms to train a model and help us predict the dislike count of a Youtube video.

4.2 Machine Learning Algorithms

- **LinearRegression:** a statistical model that analyzes the relationship between two features based on a linear relationship where $y = mx + b$, with y being the dependent and x being the independent variable. It fits a straight line through the data points to estimate the overall growth or decay of the entire dataset. This is for predicting the dislike assuming that the

correlation between the features is linear

- **RandomForest:** A machine learning algorithm that averages the predictions–in regression problems–from multiple decision trees to make a more accurate prediction. Each decision tree is trained on a subset of the training data and aggregates all predictions into one tree.

- **K-Nearest Neighbor:** A feature clustering machine learning algorithm that bases its prediction on a majority vote or the average of the K nearest neighbor data points in the training set. K is the hyper-parameter that controls the number of clusters that a model can have which has a significant impact on the overall performance of the model prediction. This is for predicting the dislike using the grouping technique, assuming there is a pattern in the like, view and dislike counts

4.3 ML with Likes and Views to Predict Dislikes

Predicting without any sentiment score the r2 score on each model is ridiculously low. From 0.18 up to 0.327. This means the model is doing worse than randomly guessing the dislike result. If adding a feature such as the sentiment score can increase the r2 score in this base. That means the feature is helpful/has contributed to building an accurate model.

4.4 ML with sentiment score, like and views to predict dislike

Predicting with the simple bag of word sentiment scores, linear regression models have not changed a lot because the correlation between different features is not linearly related. KNN which is a multidimensional algorithm that creates clusters from the training data also had a bad result which means the datapoints we have in the training set are not grouping into clusters. Random forests have improved significantly on the r2 score; it was over 0.9 on the r2-score. Which means the random forest model can predict the dislike count using the features effectively. Which is a decent score comparing other modeling algorithms.

5 Results and Evaluation

5.1 Extracting Features

To improve from the baseline model, we needed to extract more features or have more data points in the dataset. There are two possible solutions: using the comments we already have from our scraper or adding a new form of sentiment score. We have done both ways to improve the performance of the model.

- **TextBlob: Text Translation**

  Youtube is a worldwide video-sharing platform, comments come in the form of different languages. That the comment data that we have scraped are also in different languages. TextBlob can help us format those comments that are not in English and translate those comments into English. That increases the amount of data points that we have in the colors. Bag of word dictionary only works with English so the sentiment of those neutral with languages not in English will be resolve.

- **TextBlob: Text Correction**

  Youtube comments may come with incorrect spelling words because Youtube comment did not come with auto-correct for users to use. TextBlob can have a significant decrease in neutral comments because the bag of words implementation will not detect those keywords that are spelled incorrectly.

- **VaderSentiment: Polarity Score**

  VaderSentiment is a Python library used for sentiment analysis, providing a simple way to calculate the sentiment polarity score of a given text. The score ranges from -1 to 1, with higher absolute values indicating stronger emotions. It can be used for various applications such as opinion mining, social media monitoring, and customer feedback analysis. In this project VaderSentiment score adds another feature that we can use in ML alrgothim that can improve the accuracy of the prediction.

Unlike typical bag-of-words models, this sentiment analysis approach considers word-order sensitive relationships between terms. It specifically

handles degree modifiers, also known as intensifiers, booster words, or degree adverbs, which can either increase or decrease the intensity of sentiment.

The rule-based sentiment analysis engine is designed to analyze text sentiment based on grammatical and syntactical rules. It follows a set of predefined rules that take into account the structure and composition of sentences to determine sentiment. These rules are established based on grammatical and syntactical patterns commonly found in language.

	sentiment
views	-0.0450
dislikes	-0.0820
likes	0.0437
#comment	-0.0194

Figure 2: We can deduce from above data that we can relate no reasonable correlation between our variable points. The independent variable/ target data is sentiment score and features were the view count, dislike count, like count and comment count. The negative correlation show that when views, dislikes and comment count increases, sentiment increases. This is derived from the Second Dataset.

6 Conclusion

From the first dataset we used, we were only able to work with 5,000 ID's worth of comments with some of the ID's being duplicates. This means that it was very likely for the same video to appear in both the training portion as well as the testing portion of the data leading to a relatively high r^2 value i.e. the data was biased. Additionally, with only a maximum of 100 comments to work with for each video, the sentiment analysis score wasn't as attuned to that video as could have been possible, especially for videos that had thousands of comments. It's also important to take into consideration that the dataset had videos from varying genres and styles. As for the second dataset we used, it had an almost opposite problem: it had 600,000 comments but they corresponded to only 2,266 videos. This is problematic because although

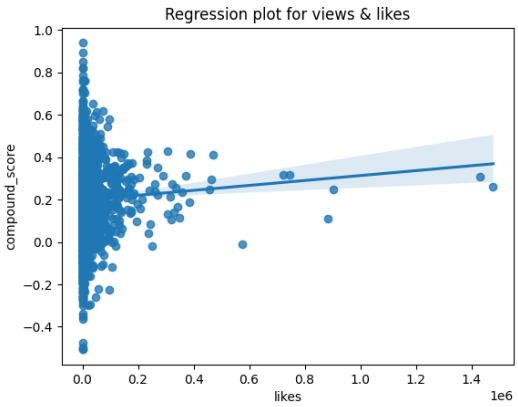

Figure 3: The relationship between number of likes and the compoud_score shows slightly promising relationship compared to other features. Although the data points with high like counts are sparse, a high like count appears appears to correlate to a more positive sentiment score. This is derived from the Second Dataset.

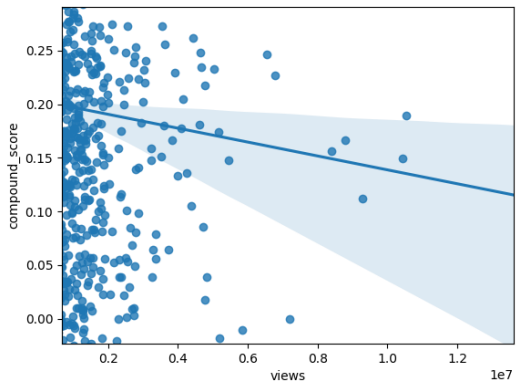

Figure 4: The reg plot of views and sentiment scores show no reasonable relation can be interpreted. The variance increases as the views increases. The blue line drawn show the sentiment approaching to neutral as views increases.

	Linear Regression	Random Forest	K-Nearest Neighbor
MSE	303016296	35132096	1007407
RMSE	17407	5927	1003
r^2	0.628	0.957	0.327

Table 1: Results for First Dataset

	Linear Regression	Random Forest	K-Nearest Neighbor
MSE	1012892	1357846	263484033
RMSE	1006	1165	16232
r^2	0.326	0.093	0.677

Table 2: Results for Second Dataset

there are a lot more comments for each video on average, meaning a more accurate sentiment analysis score for each video, with only 2,266 videos to work with, there was relatively less data correlating a sentiment score–regardless of how well-informed it be–to a dislike count.

If we were to take these results at face value, however, the conclusion we would be forced to draw is that the content of YouTube comments is not a viable source for predicting dislike count. This could be explained by the capricious and subjective nature of YouTube comments in general because it can be said that they vary greatly in tone, style, and other factors, making it difficult to use them for predictive analysis. On the other hand, these extremely disheartening results simply indicate the poor quality of the data used. And as the adage common in computer science and information technology fields goes–"garbage in, garbage out."

7 Future Work

Given that this was a semester long project and we spent much of the time struggling with the data, there is, admittedly, a lot of room for improvement for this research project. Among these potential improvements are the following:

1. Categorize videos into genres e.g. music, science, lifestyle etc. to group together videos that are more similar to each other in terms of content, viewership, and style. As one of our references pointed out, the average sentiment for each genres tends to be distinct. By clustering the data

2. Filter comments by relevance so we can extract discussions that are most the engaging so it better reflects the essence of the forum. As one of our references also mentioned, the YouTube API is not very good at considering relevance.

3. Incorporate the like count for each comment. Since our second dataset included a count for how many likes each comment had, this parameter would be useful in setting up a weighting system for comments i.e. giving more weight to comments with more likes when it comes to aggregating the overall sentiment for a video.

4. Run model on more comments per video as well as more unique videos in general. These two were the major drawbacks of the two datasets we used.

References

Stefan Siersdorfer, Sergiu Chelaru, Wolfgang Nejdl, and Jose San Pedro. How useful are your comments? analyzing and predicting youtube comments and comment ratings. April 2010. URL https://arxiv.org/pdf/2111.01908.pdf.

Olga Uryupina, Barbara Plank, Aliaksei Severyn, Agata Rotondi, and Alessandro Moschitti. Sentube: A corpus for sentiment analysis on youtube social media. Jan 2014. URL http://casa.disi.unitn.it/moschitti/since2013/2014_LREC_Uryupina_CorpusSentimentAnalysis.pdf.

Sui Xinyu. Measurement and sentiment analysis of youtube video comments. 2022. URL https://conservancy.umn.edu/bitstream/handle/11299/252495/Sui_umn_0130M_23905.pdf.

8 Contributions

- **Research:** We each researched independently and each of us was tasked with finding at least

one relevant paper that we could draw from.

- **Data Collection:** Iris took the lead on finding the datasets. He also did the bulk of the work when it came to setting up the scraper for the comments from the first dataset using YouTube API.
- **Data Processing:** Hammad also took the lead on the pre-processing and cleaning the data with Iris helping significantly.
- **Model Development** Jianfu set up the baseline and ML models for the first dataset. Iris and Hammad added to this by including correlation metrics and redoing some of this work for the second dataset.
- **Evaluation** We all worked on this together.
- **Final Paper** We all participated in this with Hammad writing and editing many of the sections, Iris creating majority of the figures, and Jianfu creating the Project Design figure. Hammad and Iris also took care of setting the paper up in LaTeX.