Ryan Ball, 11686775

CptS 315

Project Report

Introduction

YouTube is one of the largest and most unique social media platforms currently. The cultural impact the platform has can be seen everywhere, and one of the most interesting indicators of culture is YouTube's analytics. For my project, I used exploratory data analysis and insights generation on YouTube's video statistics to determine which attributes in the dataset are most closely correlated to video popularity. For this project, I set out to answer one key using my data from YouTube. I wanted to see if I could determine which video attributes (likes, dislikes, comment total, etc.) are most closely related the popularity of a video on YouTube. I want to find a metric for measuring popularity, and use that to find correlations between popularity and certain video attributes. These questions are difficult to answer and there will be challenges throughout this project. One issue is that I can only work on the current dataset of videos, at one point in time. I will have no way of determining if the outcomes of my statistics analysis are applicable on any video analytics from different periods in time. Another challenge will be finding a way to measure the correlation between popularity and video attributes that are not quantitative, like the content of comments. One of the most major issues is finding a metric or multiple metrics for popularity. I chose this topic because I use YouTube every day, and the algorithm YouTube uses for recommendations has helped me find many of my favorite creators on the platform. Understanding what makes a video popular (and therefore marketable for YouTube) is very interesting to me as it may lead to further insight into how YouTube recommends videos to users. As a result of this project, I was able to determine a solid metric for video popularity and was able to test and calculate the strength of correlations between several video attributes. I was able to develop many insights, as well as learn new methods for data mining and gaining insights from non-quantitative values.

Data Mining Task

To solve the questions asked in my report, the first step is to examine the input data. I will be using the trending YouTube video statistics dataset provided by Mitchell J. on the website Kaggle. The dataset features statistics for roughly 7000 of the most trending videos on YouTube from a given day in 2017. additionally. Each video has 11 columns of data that detail certain traits the video has. These traits include: video ID, video title, channel name, category ID, video tags, views, likes, dislikes, total comments, date, and a link to the video thumbnail. The dataset also features roughly 70,000 comments from the videos in the other part of the dataset.

With this input, I will answer my main research question: what video traits most strongly correlate to video popularity? To do this I have some smaller questions I want to answer. Firstly, what is a good metric for video popularity? Second, how would I determine correlation for non-numerical values such as comment text.

From my data mining approach, I am attempting to output a sorted list of video traits and rank them by correlation to video popularity. The dataset I am using only has eleven traits to examine, and one of them is the video ID, which should not have any correlation to popularity.

This means I am left with ten different video traits to rank by their correlation to video popularity. There are several key challenges to approach before solving the task. The first is selecting a metric for video popularity that is simple to do correlation calculations with. Another challenge is determining which video attributes I should test for correlation with video popularity. Measuring correlation with likes, dislikes, and comment total should be simple because of their numerical values, but I also want to find a way to use the comments dataset to find some correlation between comments and video popularity.

Technical Approach

In This section, I will outline the major steps I took in my algorithmic approach.

The first step I took was reading the input data into dataframes. For the purposes of this project, I used only the US comments and US video statistics as my datasets. With these two data sets read, I had decided I first wanted to attempt to address one of my challenges and evaluate a non-numerical value. I realized the only way to calculate the correlation for a non-numerical data sample like comment text would be to transform it into numerical data that could be calculated for correlation. With this in mind, I researched methods and found that using a python library like TextBlob would allow me to turn keywords in the comment text into numerical values representing the degree of polarity in attitude and sentiment for each comment. With this method decided on, my next step was to clean the data and get it ready to solve the data task. This step required multiple subtasks, as explained below:

1. In order to analyze any meaningful data for the video comments, I first had to clean the data. I went through several stages of cleaning to remove any characters or elements that would make sentiment analysis difficult later in the project. I started by removing punctuation, then digits, than emoji, and finally I removed any short words that were less than four characters long.
2. The purpose of cleaning the comments was so I could more reliably calculate the sentiment of each comment. I used the TextBlob library to calculate the sentiment of each comment. This allowed me to record the polarity and sentiment of each comment based on keywords identified in the cleaned dataset. I then aggregated the sentiments by video ID, so a video's overall comment sentiment is the mean of the all comment sentiments for a given video ID.
3. With sentiment calculated for each video, I decided to start testing whether sentiment has any correlation with video popularity. To do this, I first had to solve one of my challenges by selecting a metric for video popularity. I ended up deciding to use video views as the metric, because it made calculations simple and views are already used as a metric for video success and popularity on YouTube by content creators and the service itself.

Pseudo-code for this step can be seen below:

Read dataset into dataframe

Change comment text to lower case characters

Remove punctuation from comment text

Remove digits from comment text

Remove emojis from comment text

For each cleaned comment in the comment text data:

Calculate the sentiment polarity of the comment text using TextBlob

Aggregate the sentiments of comments with the same video ID

Aggregate video views with the same video ID

Store the aggregated views and sentiments into a new dataframe

Plot the relationship between aggregate views and sentiments on a Seaborn graph

Remove rows with NaN values from the dataframe of aggregated views and sentiments

Calculate the Pearson Coefficient and P-value using the pearsonr() function

This was the most code and runtime intensive task in the project. The remainder of the calculations were relatively simple. For all of my correlation coefficients, I used the function pearsonr() from scipy.stats to calculate the Pearson coefficient and p-value for each of the relationships. I later learned that p-value calculations are unreliable on large data sets, but I decided to keep them in the code just for the sake of context for my later explanations. My next calculation required me to make a new dataframe that stores the like-to-dislike ratio I calculated for each video. I then used pearsonr() to calculate the correlation coefficient and p-value for this relationship and plotted my results with a Seaborn graph.

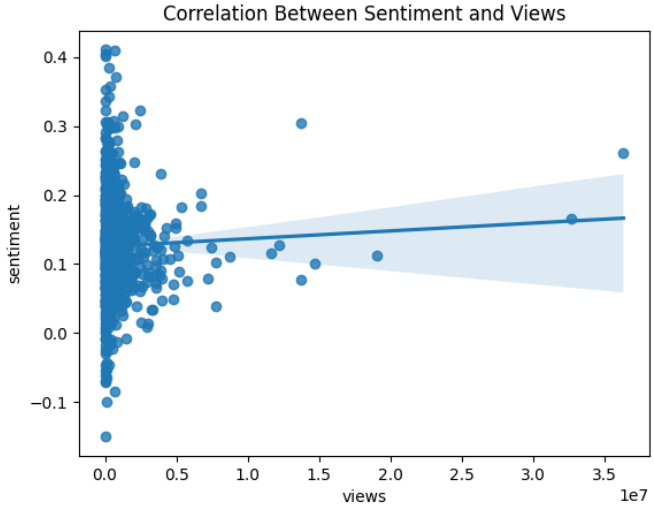
The next three correlation tests I did were practically the same, calculating the correlation between video views and a column of numerical data in the dataset. All of these used the pearsonr() function to calculate the correlation coefficient and Seaborn to plot a graph of the relationship. For these three correlation tests, I tested the correlation between likes, dislikes, and total comments, each in relation to video views.

For my final test, I wanted to do something with category ID, but there is no reliable way to test correlation for multiple categories. Instead, I just chose to plot a bar chart with the categories ranked in ascending order from most viewed to least viewed category. There is no statistical correlation with this step, but it does show the most viewed category, and while this is not a statistical correlation, it does show which categories are more popular based on views. To create this graph, I first grouped the category IDs by views using the sum() method to get the total views for each category. I then changed the category ID to the category name using a dictionary I created with the category IDs as keys and their actual category names as values.

Finally, I decided to sort the correlation coefficients I calculated from strongest correlation to lowest. I printed the original outputs of my calculations, then stored the correlation coefficients in a dictionary which I then sorted by value. I then simply printed the 5 attributes in order of highest correlation to lowest.

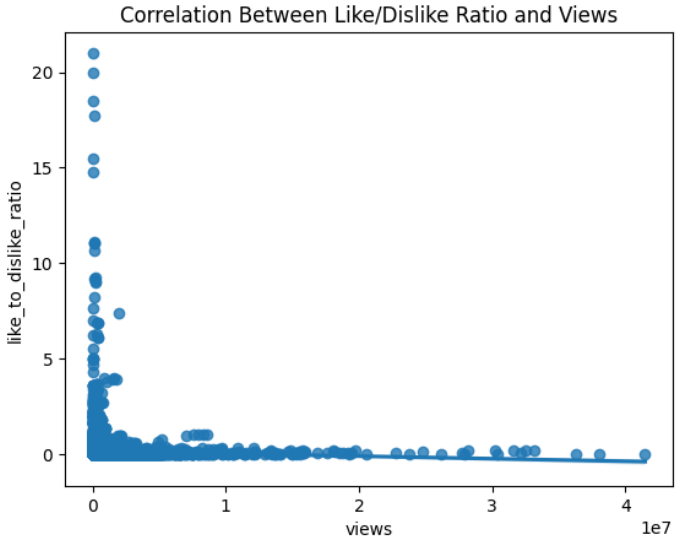
Results and Discussion

With the data ready for analysis, my first task in analysis was determining whether comment sentiment has any significant correlation with video popularity. For much of this report, I have chosen to use view count as a simple metric for video popularity. With this in mind, I plotted a graph that shows comment sentiment relative to video views:



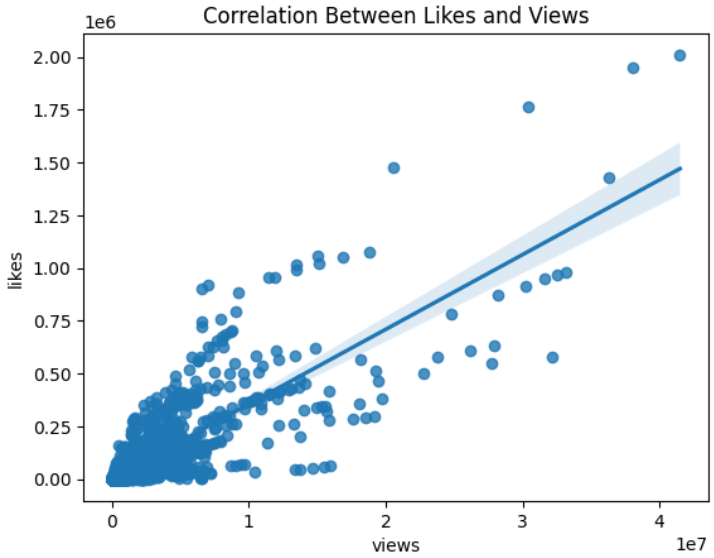
To statistically determine whether there is any significant correlation between comment sentiment and views, I calculated the Pearson correlation coefficient and the p-value between views and sentiment. The Pearson correlation coefficient I obtained was 0.036 and the p-value I obtained was 0.308. Since a P-Value of around 0.05 or lower is necessary to consider a correlation statistically significant, I am unable to conclude that comment sentiment has any correlation with views, which in turn allows me to infer that comment sentiment does not correlate with video popularity. The Pearson coefficient is not very high either, further indicating weak correlation. It was important for me to calculate the P-value for this analysis, as the Pearson coefficient is only capable of quantifying a correlation, and has no ability to assess whether the correlation in real or statistically significant.

With the first test not showing any correlation, I then decided to test another indicator of sentiment for a video: the like to dislike ratio. I hypothesized that a high like to dislike ratio would have a positive correlation to views, which in turn would point to a correlation with popularity. Once again, I plotted a graph that shows the like to dislike ratio relative to the view count:



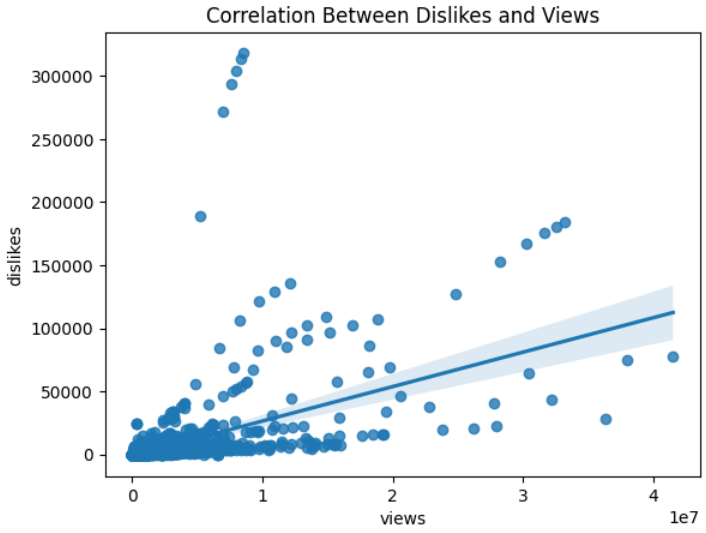
This time, the p-value was well below 0.05, with the actual value being 0.0000000685. This indicate that there is a statistically significant correlation between the like to dislike ratio and views; unfortunately, the Pearson Coefficient calculated was -0.0611, which not only shows a very weak correlation, but also shows an inverse correlation between the ratio and view count. This means a high ratio actually would correlate with a lower view count, not a higher one. Since the correlation coefficient is inverse, I must conclude that the like-to-dislike ratio does not correlate with video popularity.

With like and dislike ratios not correlating to video popularity, I chose to simplify my approach to find a positive correlation by testing the correlation between views and likes. The like-to-dislike ratio may have had issues in calculations due to the fact that I had to remove video entries with zero likes or dislikes to avoid division by zero errors. For likes I plotted the graph in relation to views:



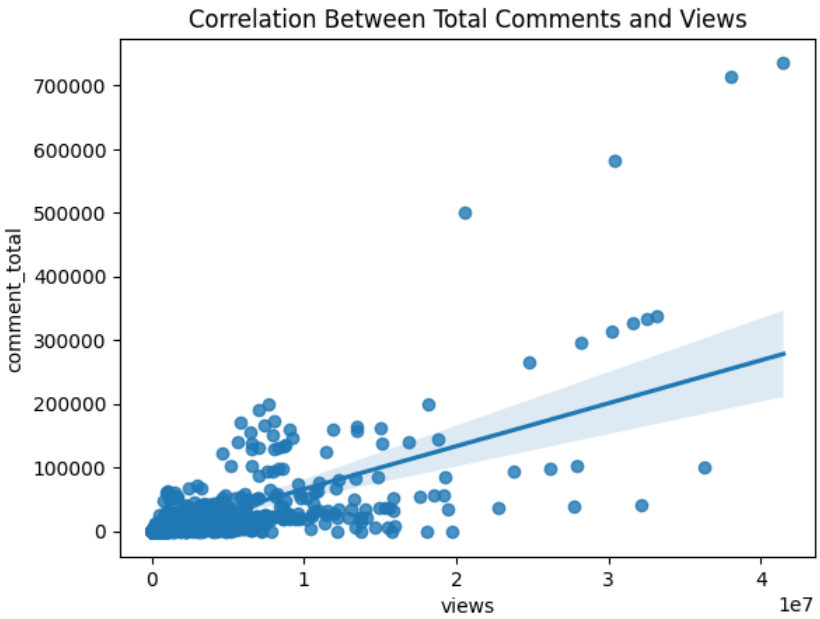
This time, the p-value I obtained during calculations was 0.0. It was at this point in testing where I realized that p-value can be very unreliable on large datasets, which will be discussed more in a later section. The Pearson Coefficient was a 0.83, which is the largest correlation coefficient calculated thus far. This is the first attribute calculated that demonstrates a strong correlation with views and video popularity.

Because the correlation for video likes and views was relatively easy to calculate, I also calculated the correlation between views and dislikes and graphed the results:



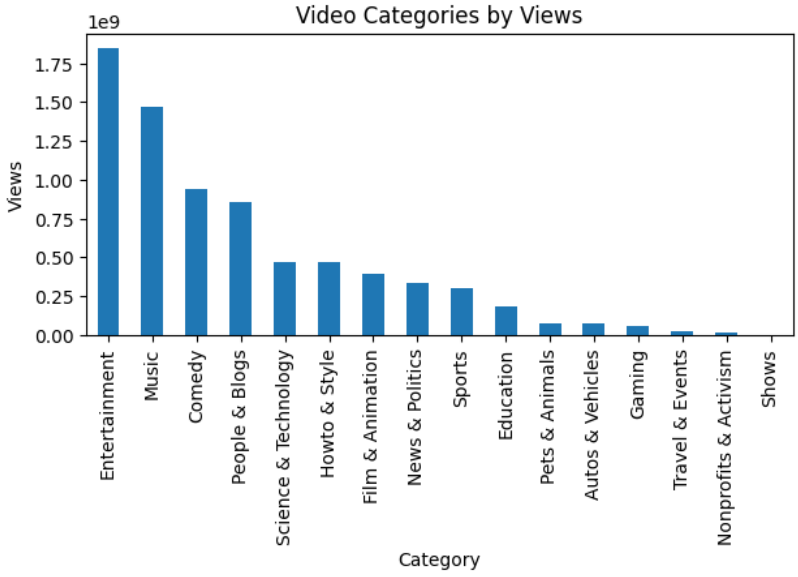
Again, a p-value of zero was obtained, but the correlation coefficient calculated was 0.542, which demonstrates a moderate correlation between dislikes and video views.

The final numerical attribute from the dataset to test was total comments. As with the other tests, the relationship between view count and comment total was plotted and graphed:



Another 0.0 value was calculated for the p-value, but the Pearson Coefficient calculated was 0.716, indicating a moderately strong correlation between views and total number of comments for a video.

Before looking at the ranked correlation coefficients, I want to first discuss the bar graph that shows the category IDs ranked by view count:



This graph shows that categories like Entertainment and Music receive the most views, while other categories receive considerably less. This indicates that to at least some degree, certain categories are more popular than others. This does not account for the number of videos in each category, so it is not viable to concretely say there is a direct correlation, but the differences in view count does indicate some degree of influence on video popularity.

Finally, we have five Pearson coefficient calculated, so the next step is to rank them into a list from ascending order. Below is the list of sorted coefficients printed from my program:

Views-Likes: 0.8328442631592106

Views-Total Comments: 0.7155565333359567

Views-Dislikes: 0.5419550396950303

Views-Sentiment: 0.03591662291773122

Views-Like/Dislike Ratio: -0.061124867785269946

As we can see, likes, total comments, and dislikes were the most strongly correlated attributes with views, and thus video popularity. In contrast, the values I transformed using the data like sentiment and the like/dislike ratio showed extremely low correlation coefficients, leading me to believe there really is not much of a correlation between video popularity and these two data samples.

It is clear that numerical values already included in the dataset ended up with higher correlation coefficients. I had hoped sentiments and like/dislike ratios would yield so insightful correlations, but based on my calculations no such correlations existed. Despite this, I still think my calculations for comment sentiment worked very well, which further confuses me about the results. Another aspect that did not work in my calculations was the p-value. After multiple strange and unhelpful values, I did some research and found that p-values are much less applicable to larger datasets. In large datasets, even small or practically insignificant correlations can be statistically significant. This is because the p-value depends on the sample size, and with a larger sample size, the power of the statistical test increases. This means that it becomes more likely to detect even tiny effects or relationships. So, when dealing with large datasets, it is possible to obtain a statistically significant p-value for a correlation that might not be meaningful or practically significant. This means the p-values I collected are unreliable, as the dataset is too large for the p-value to indicate that a correlation is statistically significant.

Lessons Learned

This project was a good lesson in the principle of Occam's Razor. For much of my project, I was overthinking my model and approach to answering the questions I had, which only ended up confusing me more. To start, I struggled a lot at first starting this project because I kept trying to come up with complex metrics for popularity that were out of touch with the goal of my project. I had explicitly stated that my question was to determine what video attributes most closely correlate to video popularity. I realized that instead of coming up with a complex metric for video popularity that would be time consuming and difficult, my dataset already featured a metric that people have been using in real-world situations to measure video popularity since the service was launched: view count. With view count as a metric, the principle of Occam's Razor was clearly demonstrated to me. View count is a better metric for popularity than any other metric I tried because it is simple, easy to understand, and is more applicable to what real world channels use to measure their video's success and popularity. The only other metric I thought would also be potentially viable is trending data like the length a video was trending or its highest rank on trending, but this data is not available in the dataset I used. These metrics also have the same issues as others: needless complexity. View count as a metric is already a metric both YouTube and its Creators use to determine video popularity and engagement, so using a metric that is less prominent in the industry would only result in a project that is less applicable to real world applications.

I tried multiple methods at first like content-based filtering, TF-IDF (Term Frequency * Inverse Doc Frequency), and even some research into machine learning approaches. I learned that while these methods had very powerful uses, their complexity was just not necessary for answering my research questions. With a metric chosen, I wanted to see if there was any correlation between another video attribute in my dataset and the view count of the video. Since my main objective was to find correlation between view count and other video attributes, I realized that in its most basic form, my project mainly involved finding the strength of correlations between two datasets. This is when I once again realized that a simpler solution is often more applicable. Calculating the Pearson Coefficient

between views and another video attribute was a simple and effective way of determining the correlation strength between video attributes and my metric for popularity. I learned that a simple model can often lead to more applicable results than a model that uses more subjective metrics and complex processes.

I also learned about the limitations of some statistical models, like the Pearson p-value. I would have saved a lot of time on this project if I had realized p-values are less applicable for larger datasets, and I regret that I did not have enough time to find another metric to determine if a correlation is statistically significant.

In Hindsight, there are many things I would have done differently to improve the project. Firstly, I would have tried to find or calculate more attributes that I could test for correlation with video popularity. I found that a lot of the attributes I spent more time processing or calculating myself yielded low calculations for correlation, and I would have liked to have calculated a metric that did show some positive correlation with view count. Considering the processing to calculate comment sentiment is the most runtime intensive step in my program, I wish I had been able to come up with additional attributes that applied more data manipulation principles to test for correlation with popularity. Calculating the comment sentiment polarity was already a very new data mining concept for me, so I wish I had more time to find other attributes that would allow me to use more data mining principles. If I had more time, I also would have moved all my transformed data and other data used in calculations to a single new dataframe for consistency, rather than pulling from multiple different dataframes (the calculations should be the same, but it would make the data appear more uniform and consistent in my code). Overall, though, my project showed me that I was able to analyze a given dataset and calculate multiple strong correlations between video attributes and popularity. Analysis methods were simple, I believe the analysis is sound and the outcome is very easy to interpret and apply. I was able to answer the main question I investigated and identified several video attributes that correlate with video popularity, including likes, dislikes, and total comments. I gained a lot of experience working on my own project and developing my own approach, which I hope is reflected in this report.

Acknowledgements

Jakasaniya, S. (2022, September 30). *EDA_youtube*. Kaggle. Retrieved May 4, 2023, from https://www.kaggle.com/code/sachinjakasaniya/eda-youtube

Explanation: This user operating on the same YouTube dataset was also doing exploratory data analysis. I was confused on how to clean comment text before calculating sentiment, so I i implemented his methods to clean the data like remove punctuation, emojis, digits, and setting text to lowercase. Nothing else from his project was used by me, we took radically different analysis approaches by the end.

Lin, M., & Lucas, H. C. (2013, December). *Too Big to Fail: Large Samples and the p-Value Problem*. ResearchGate. Retrieved May 4, 2023, from https://www.researchgate.net/publication/270504262_Too_Big_to_Fail_Large_Samples_and_the_p-Value_Problem

Explanation: This paper explained how p-value is less reliable on large datasets, it was important for my project as it helped me understand why the p-value was not viable to use on my dataset due to its size.

Chatzopoulou, G., Sheng, C., & Faloutsos, M. (2010). A First Step Towards Understanding Popularity in YouTube. *2010 INFOCOM IEEE Conference on Computer Communications Workshops*, 1-6.

Explanation: This paper gave me several possible metrics for video popularity. I decided to stick with using video likes due to their simplicity and ease to apply to real-world scenarios