Week_3_Exploratory Data Analysis

Yujia Cao, Wendi Yuan, Yuhan Zhao

For our research, we aim to analyze numerical data to explore the relationships between the dependent variable (views) and the independent variables. Specifically, we seek to identify key factors that contribute to a video's popularity, which will provide valuable insights for content creators. By identifying these factors, we aim to predict effective "keywords" and strategies to help creators optimize their content and increase its visibility to audiences. The primary focus of our analysis will be on the number of views as the target variable, with all other factors acting as independent variables.

We divided our dataset in two ways: 60% for training and 40% for testing, as well as an 80/20 split. Ultimately, we chose the 80/20 split because it is a widely accepted approach that balances the size of the training and testing sets. A larger training set ensures sufficient data for model development, while a sufficiently sized test set provides an unbiased evaluation. The split reduces variance in both parameter estimates (from the training data) and performance statistics (from the test data), leading to more robust results.

The dataset used for this analysis contains YouTube video data from five different countries: Canada, Germany, France, Great Britain, and the USA. After loading the data for each country, we merged them into a single data frame, adding a "location" column to identify the country of each video. The final merged dataset was used for all further analyses. To prepare the data for model training and validation, we split the dataset into training (80%) and test (20%) sets. The target variable is 'views', and all other columns are treated as features.

After splitting the dataset, we checked for missing values and data types. The dataset had no missing values in critical columns like *views*, *likes*, *dislikes*, and *comment_count*. The data types were consistent with expectations, including integer and float types for numerical variables and object types for categorical ones. And we have decided to remove some unnecessary variables such as *thumbnail_link*, *comments_disabled*, *ratings_disabled*, and *video_error_or_removed*, they are not directly relevant to our analysis, as they represent technical metadata rather than performance drivers.

We grouped the dataset by *category_id* to explore which content categories are most prevalent in trending videos. The top categories in terms of frequency include:(see Appx.1)

- Entertainment
- Music

- People & Blogs
- Comedy
- News & Politics

These results align with the popularity of entertainment and music content on YouTube, confirming that these categories dominate the platform.

We analyzed the top 50 videos and channels by views. The top channels with the most views were primarily from popular categories like music and entertainment(see Appx.2), and the most viewed videos were heavily concentrated in music and entertainment, with some viral content dominating the top spots(see Appx.3).

We computed the correlation matrix for key numerical variables to assess relationships between them, *likes* and *views* with correlation of 0.78, a strong positive correlation, indicating that videos with more likes tend to have more views. *comment_count* and *view*s with correlation of 0.50, it indicates a moderate positive correlation, and suggesting that videos with more comments generally attract more views. *dislikes* and *views* has a correlation of 0.41, it's a weaker positive correlation, showing that even disliked videos can still have high view counts(see Appx.4).

For better understanding, we know that only get the results of top views of the videos and channels by the view count is not enough, so we evaluate each video's performance by calculating a composite score derived from key engagement metrics: *likes*, *dislikes*, and *comment_counts*. The scores are then used to evaluate individual videos and aggregate them by channel to identify the top-performing channels(see Appx.5). Based on the correlation score we have from above, ensure all metrics contribute proportionally to the score, their absolute values were normalized to sum up to 1. This normalization led to the following weights(see Appx.11):

likes: 46.04%,

dislikes: 24.44%,

and *comment_count*: 29.49%

And here is the formula we use:

```
train['score'] = (
    weights['likes'] * train['likes'] -
    weights['dislikes'] * train['dislikes'] +
    weights['comment_count'] * train['comment_count']
)
```

This formula rewards videos with high engagement in terms of likes and comment counts, while penalizing them for high dislike counts. The scores are then used to rank the videos in descending order, identifying the top-performing videos based on audience engagement.

Finally for this week, we have created word clouds for *video_titles*, *channel_titles*, *tags*, and *descriptions* to visualize the most frequent terms in the dataset. The word clouds showed that common keywords included "official", "music", and "trailer", aligning with the finding that entertainment and music dominate the platform(see Appx. 6,7,8,9)

After we do the EDA for future analysis, we construct new findings that help us solve the problem statement, and also relate to the real-world scenario to come up with solutions. We seek for opportunities that EDA brings to us and also try to tackle challenges.

We have 16 variables in total that are related to the youtube statistics, and in the process of analyzing them, we abandon non-related variables like *thumbnail_links* and increase the values for more related variables that affect the total views: *likes*, *dislikes* and *comment_counts*. Since we rearrange the importance of each variable and design a new criterion to define " what types of video channels get the most scores" , our dataset has been streamlined from a complex one, and it's easier to see what factors are driving the variation and popularity of these youtube videos. After these meaningful variables are reassigned, volume of the views is not the only target variable that we should be looking at, but also the rank (calculated by scores) becomes another integrated and important variable. The concern of the monotony of a single target variable is reduced.

EDA is a crucial part for our future analysis. By calculating the weighted ranking of video channels based on *likes*, *dislikes*, and *comment_counts*, it allows us to identify which channels or videos are currently driving engagement. This serves as a strong foundation for tracking real-time trends and highlights the channels that are getting attention since we truly want to connect our analysis to the real-time data and trends at the end of the project. It's better to capture the trend in 2018 and find real-time data in 2024. Another opportunity is to quickly gather useful text information from the Word Cloud. By examining the frequency of *video_titles*, *descriptions*, and *tags*, we can isolate the most frequently occurring words and phrases that are correlated with high-ranking videos. Even though we are still on the way of connecting the ranking analysis and text popularity analysis, this keyword analysis is a great beginning that can help us. The *likes* and *dislikes* data, combined with user comments, provide a starting point for sentiment analysis. When we begin trying different models on our data and to see how they perform, we consider to refine models by not only identifying popular videos by ranking but also those with positive sentiment. These are all advantages that EDA brings to us and advertisers. Advertisers can

leverage sentimental analysis if they want, and they would like to see predictions for trending topics and channels for next month, next five months, and even next year. It is a chance to optimize ad targeting.

We also face some challenges in solving the problem statement. Firstly, the real-time data capture. Implementing real-time tracking and analysis for this system could be challenging due to the volume of data and the need to frequently refresh engagement metrics. We only did EDA on data that included numerical variables and Word Cloud, we didn't find deeper information on Word Cloud.Moreover, when we did the rankings based on the scores, it's easy to notice that some channels have a monopoly and exceed other channels. Channels with significantly larger audiences could dominate rankings based on the weights metrics, but it skewed the analysis toward only the most popular channels. It might be necessary to normalize the data or apply some weighting mechanism to ensure that newer or smaller creators are not overlooked. The EDA may be not enough to do the sentimental analysis since our criterion is simple: when a video gets more likes, dislikes and comments, scores will be higher. However, some videos are controversial because they have high dislikes and high likes, making it hard to infer whether they are positively or negatively received. We think that's the gap between the real-world scenario and the analysis, and we are eager to conquer these challenges further.

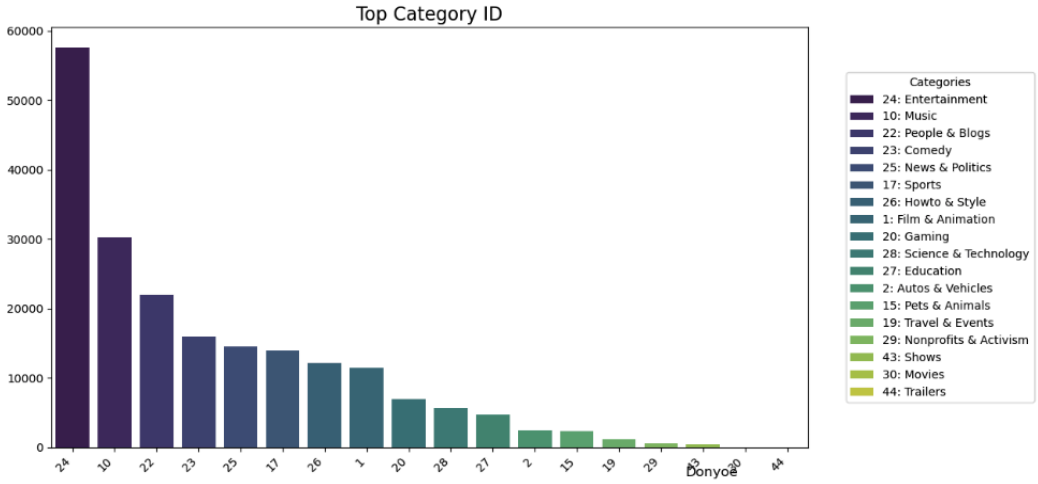
We then ranked the trending videos and assigned a score to each video based on numerical metrics such as *likes*, *dislikes*, and *comment_counts*. We used a correlation matrix to calculate the weight of each video's engagement metrics, and apply the weight to each corresponding value(see Appx.10). By using the score they got, we re-ranked all the videos and formed a new dataframe by the ranking order(see Appx. 5). After completing the numerical analysis, we plan to perform text analysis in the next step. Initially, we generated word clouds to identify frequently occurring words.

The visualizations from the EDA revealed several data issues that need addressing before conducting further analysis. One major problem identified is the skewed distribution of channel scores, with a few dominant channels disproportionately affecting the overall results. This imbalance can skew insights and lead to biased conclusions. Additionally, the word clouds of *video_titles*, *tags*, and *descriptions* highlight repetitive and non-informative content, such as common words ("Official," "Video," "Trailer") and redundant elements like URLs, social media handles, and promotional phrases. The presence of mixed-language content further complicates the analysis, introducing noise that may impact the accuracy of text-based evaluations.

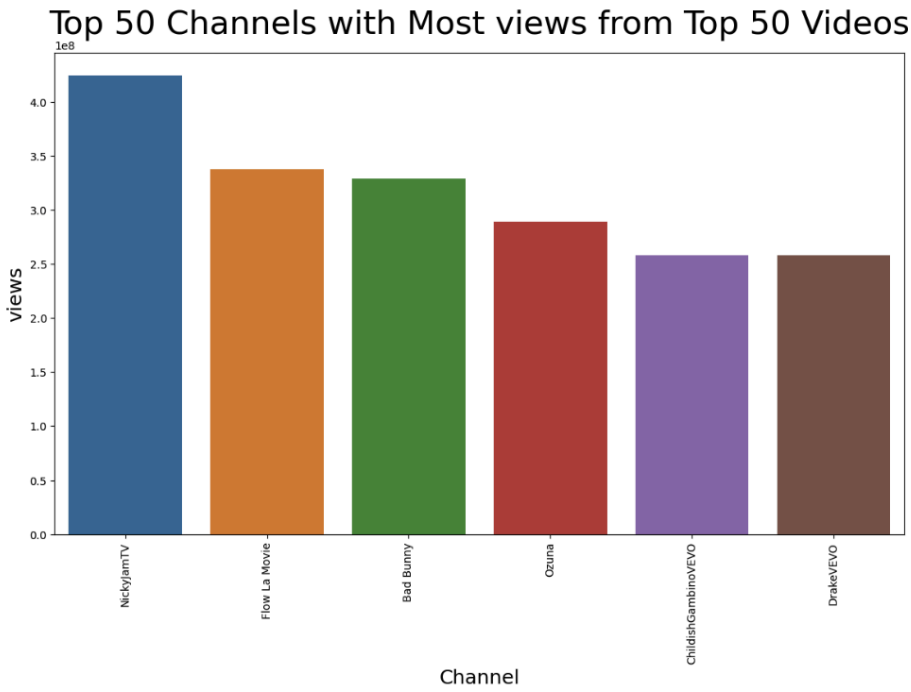
To address these issues, it is recommended to implement data preprocessing steps to clean and standardize the data. For skewed channel distributions, techniques such as normalization or expanding the scope to include a broader range of channels can help balance the analysis. Removing common stop

words, URLs, and social media handles from *titles*, *tags*, and *descriptions* will reduce noise and focus on meaningful content. Language detection tools can be used to filter or segment content by language, ensuring consistency in text analysis. Standardizing text to a uniform case will also help prevent keyword duplication due to case sensitivity differences. These preprocessing actions will create a cleaner dataset, enhancing the reliability of subsequent analysis and insights.

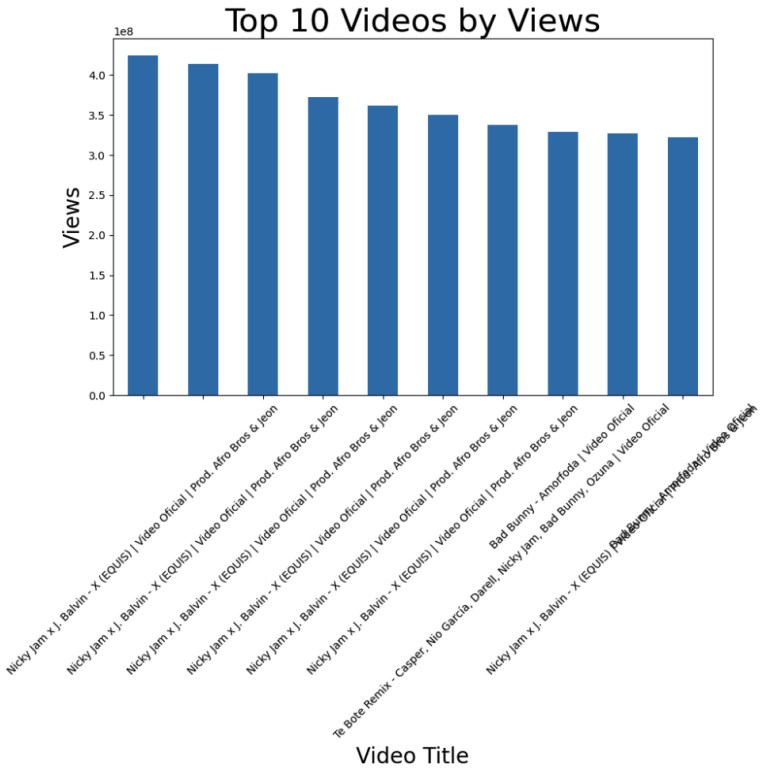
Appendix



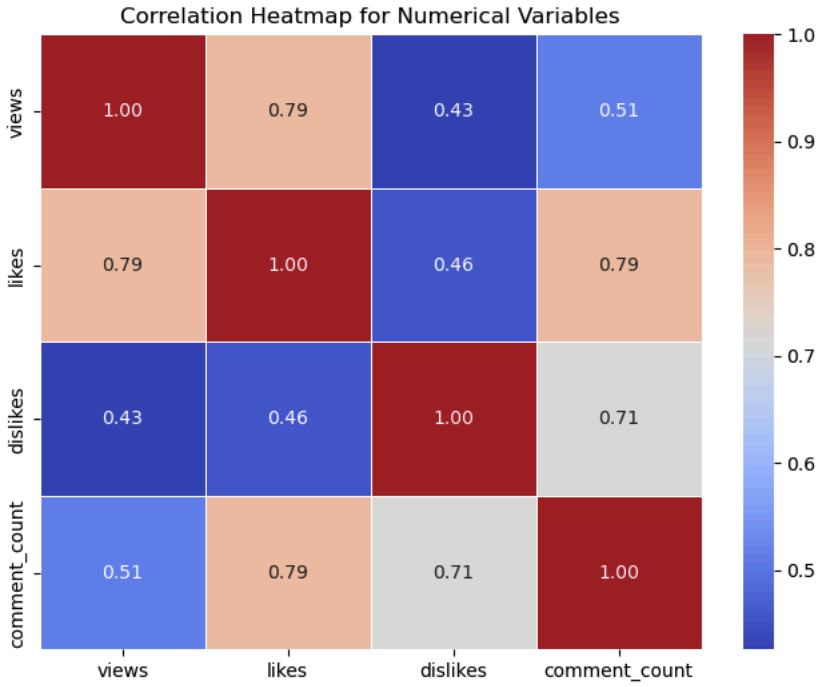
Appx. 1



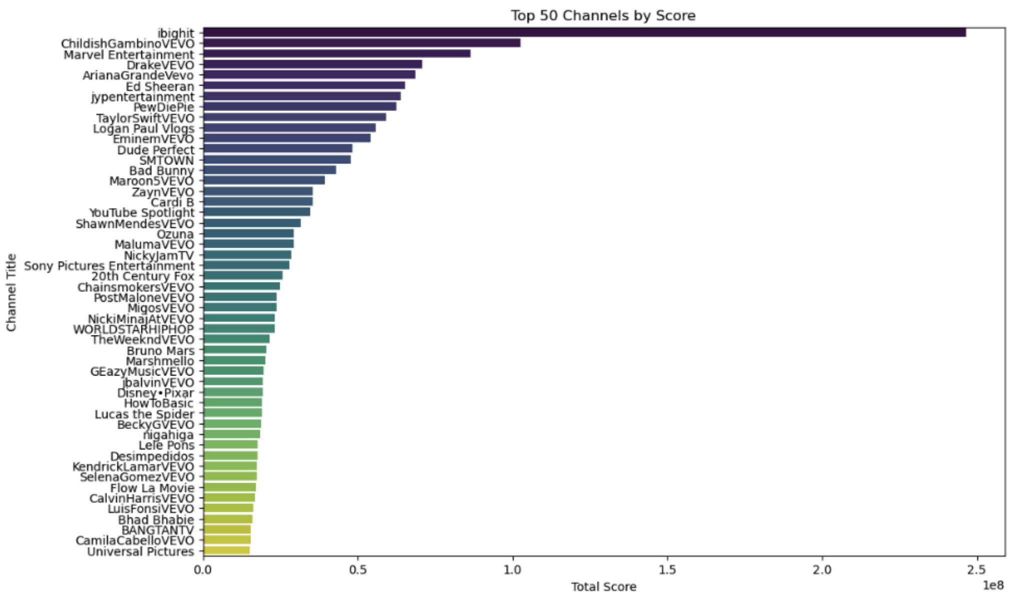
Appx. 2


Top 10 Videos by Views
1e8
Views
Video Title

Appx. 3


Correlation Heatmap for Numerical Variables
views
likes
dislikes
comment_count
1.00
0.79
0.43
0.51
0.79
1.00
0.46
0.79
0.43
0.46
1.00
0.71
0.51
0.79
0.71
1.00

Appx. 4

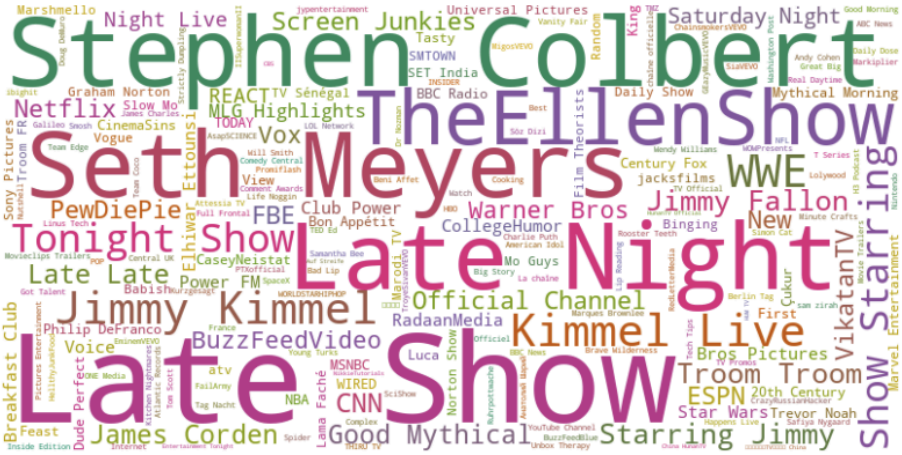

Top 50 Channels by Score
Channel Title
Total Score
1e8
0.0
0.5
1.0
1.5
2.0
2.5
ibighit
ChildishGambinoVEVO
Marvel Entertainment
DrakeVEVO
ArianaGrandeVevo
Ed Sheeran
jypentertainment
PewDiePie
TaylorSwiftVEVO
Logan Paul Vlogs
EminemVEVO
Dude Perfect
SMTOWN
Bad Bunny
Maroon5VEVO
ZaynVEVO
Cardi B
YouTube Spotlight
ShawnMendesVEVO
Ozuna
MalumaVEVO
NickyJamTV
Sony Pictures Entertainment
20th Century Fox
ChainsmokersVEVO
PostMaloneVEVO
MigosVEVO
NickiMinajAtVEVO
WORLDSTARHIPHOP
TheWeekndVEVO
Bruno Mars
Marshmello
GEazyMusicVEVO
jbalvinVEVO
Disney•Pixar
HowToBasic
Lucas the Spider
BeckyGVEVO
nigahiga
Lele Pons
Desimpedidos
KendrickLamarVEVO
SelenaGomezVEVO
Flow La Movie
CalvinHarrisVEVO
LuisFonsiVEVO
Bhad Bhabie
BANGTANTV
CamilaCabelloVEVO
Universal Pictures

Appx. 5


Word Cloud of Video Titles
Official Trailer
Music Video
Official Video
Official Music
Lyric Video
Star Wars
Trailer HD
Clip Officiel
Last Jedi
Super Bowl
Official Audio
Cardi B
Official Teaser
Video ft
Highlights
Logan Paul
Teaser Trailer
Live
Avengers Infinity
Infinity War
Full Episode
Donald Trump
Ed Sheeran
Full Game
Make
Trump
Black Panther
Nicki Minaj
Real Madrid
Goals Highlight
Punjabi Song
Season Episode
Official Lyric
New Year
Johnny Hallyday
Taylor Swift
Honest Trailer
Daily Show

Appx. 6

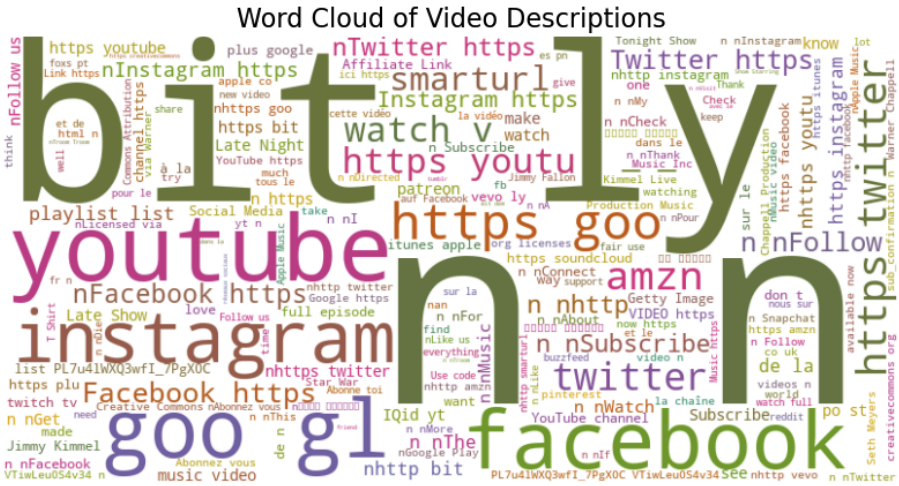
Word Cloud of Channel Titles



Appx. 7

Word Cloud of Tags

Appx. 8



Appx. 9

```
weights = {
    'likes': 0.460435,
    'dislikes': 0.244418,
    'comment_count': 0.294947
}

train['score'] = (
    weights['likes'] * train['likes'] -
    weights['dislikes'] * train['dislikes'] +
    weights['comment_count'] * train['comment_count']
)

train['rank'] = train['score'].rank(ascending=False, method='min')

df_sorted = train.sort_values(by='rank')

print(df_sorted)
```

Appx. 10

```
# Assuming the correlation values are manually entered from the heatmap
correlation_values = {
    'likes': 0.784,          # Correlation of likes with views
    'dislikes': 0.416,       # Correlation of dislikes with views
    'comment_count': 0.502 # Correlation of comment_count with views
}

# Convert the correlation values to absolute values
abs_correlations = {key: abs(value) for key, value in correlation_values.items()}

# Calculate the total sum of absolute correlations
total_correlation = sum(abs_correlations.values())

# Calculate weights by normalizing the absolute correlation values
weights = {key: value / total_correlation for key, value in abs_correlations.items()}

# Convert the weights to a DataFrame for better visualization
weights_df = pd.DataFrame(list(weights.items()), columns=['Variable', 'Weight'])
```

Appx. 11