CPT_S 315 Final Report
Washington State University
1st Ethan Villalovoz*, B.S. Computer Science, 11751324
2nd Logan Sutton*, B.S. Applied Mathematics, 11798384
3rd Berkeley Conkling*, B.S Computer Science, 11813150
4th Roy Zabetski*, B.S. Computer Science, 11771891
5th Kyle Hawkins*, B.S. Software Engineering, 11759096
6th Chance Bradford*, B.S. Computer Science, 11720208
7th Matthew Bruggeman*, B.S. Computer Science, 11763228
8th Wenjie Wang*, B.S. Computer Science, 11361265
9th Silvestre Pamatz-Rangel*, B.S. Computer Science, 11731487

Introduction

YouTube is one of the largest online entertainment industries. Anyone with a Google account can upload short and long-form videos to the platform. People who have found success on the platform, often called content creators, post videos and hope to make it on YouTube's trending page. What do all of these trending videos have in common? What statistics make the YouTube algorithm classify a video as trending? Our research will use association rule mining, linear regression, and classification to reveal the commonalities between YouTube's trending videos. As a result, we want to allow content creators to understand what makes a video get on the YouTube trending page.

Each member of the project team made an equal contribution to the project.

Our agenda

We will use three algorithms to analyze YouTube statistics datasets and conclude which attributes of a YouTube video make it popular among the general audience. Additionally, given the data on trending videos, we want to predict whether or not a video has the potential to become trending.

Our methods

The dataset we are using is named "Trending YouTube Video Statistics and Comments" (https://www.kaggle.com/datasets/datasnaek/youtube) [1]. It includes data gathered from videos on YouTube that are within the trending category each day.

We plan on using three data mining algorithms:

1. **Association Rule Mining (Apriori)**: Association rule mining techniques such as the Apriori algorithm [2] can be applied to uncover frequent item sets or patterns among categorical variables like video tags or categories. This helps identify associations between different attributes and understand viewer preferences.
2. **Linear Regression**: Linear regression [3] can be used to model the relationship between numerical variables such as views, likes, and comments. It helps understand how changes in one variable affect another, thus providing insights into factors influencing video popularity.

* Individuals that have contributed equally to the final project

3. **Logistic Regression:** Logistic Regression [4] is a statistical algorithm used for binary classification. It models the probability of a binary outcome by fitting a logistic function to the observed data. We will use this algorithm to predict whether a video will trend based on its features (e.g., likes, dislikes, views, comments, etc.).

There are also several key questions we asked ourselves when deciding which data to mine and collect:

- What are the key factors contributing to a video trending on YouTube?
- Are there any patterns in the characteristics of videos that tend to trend on YouTube (i.e., length, category, language)?
- How does viewer engagement (likes, dislikes, comments) correlate with video performance (views, trending duration)? Can we use this to predict the video's view count?
- Are there any notable differences in viewer engagement across different video categories (e.g., music, gaming, entertainment)?
- Can sentiment analysis of comments help predict the success of a YouTube video?
- Are there any temporal patterns in video trends, such as certain times or days of the week when videos are more likely to trend?
- How does the title (or keywords within the title) impact a video's view count?
- Are there any correlations between viewers' geographic locations and the types of videos that trend in those regions?
- Can we identify influential creators or channels based on their video performance metrics and viewer engagement?
- What emotional conclusions can we draw from performing sentiment analysis in various forms? Do viewers feel happy, amused, or other emotions?

Answering these questions will allow us to collect meaningful data for content creators.

<u>What we expect to find</u>

We expect to find a correlation between datasets that may reveal answers to the questions we set out to solve, such as:

- What aspects distinguish a popular video
- Association rules between different video attributes such as likes, dislikes, comments, and views
- Correlations between specific video categories and user engagement metrics
- Possible patterns in viewer behavior
- Prediction models to estimate the number of views based on stats like (video title, description, number of comments)
- Identifying characteristics that lead to a video receiving a large number of likes, comments, or views
- Impact of video length, creator upload frequency, and content type for user engagement metrics

By utilizing the three data mining algorithms (Apriori, Linear Regression, Logistic Regression), we can gain valuable insights to help content creators, Advertisers, and YouTube data Analysts learn what factors contribute to the success of trending videos. This knowledge can then be utilized to optimize content strategies by these findings.

Github

For additional reference and information about the code we have created, here is the GitHub repo for this project: https://github.com/ethanvillalovoz/CPTS-315-CougCoders

Data Cleaning / Pre-Processing

For this project, we only used the 'UScomments.csv' and 'USvideos.csv' because we believed that these were the most relevant to our audience.

Issues that were in these .csv files were:

- Rows were missing newline separators
- Rows were missing columns
- Rows contained extraneous columns
- Rows were missing closing quotes

We resolved these issues by:

- When it is clear that a single newline and nothing else has been lost, it is reinserted in the correct position.
- When a row is missing columns, the entire row is removed.
- When a single item (video or comment) spans multiple rows, all relevant rows are removed because the item is usually abruptly cut off.
- Any row that contains a CSV header in it is removed because some columns are always cut off.
- Columns in rows that are missing a closing quote are removed because we have to assume that the rest of the text is missing.
- Any extraneous columns in rows will be removed if and only if the column can be extracted cleanly.

Each of our algorithms had to do very little additional data cleaning that was needed to be done on top of this.

Analyzing Relationships

Goals

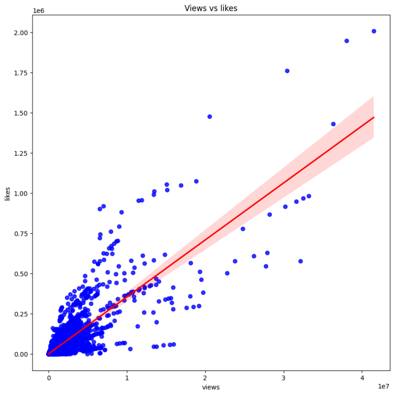
- Find relationships between numerical data categories in terms of strength and direction of the linear relationship between two variables
- Gain valuable insight into dataset relationships to aid in further data mining tasks
- Discover relationships of the video category classifier
- Aid in classification

Process

We utilized correlation and covariance analyses to discover relationships between two datasets or variables. The Pearson Correlation Coefficient served as our primary tool, offering insight into the strength and direction of associations. A coefficient close to 1 or -1 indicates a strong positive or negative relationship, respectively, while values near 0 suggest little to no correlation.

Covariance complemented our analysis by indicating the directional trend between variables. Positive covariance indicates that both variables tend to increase or decrease simultaneously, while negative covariance suggests an inverse relationship, where one variable's increase corresponds with the other's decrease.

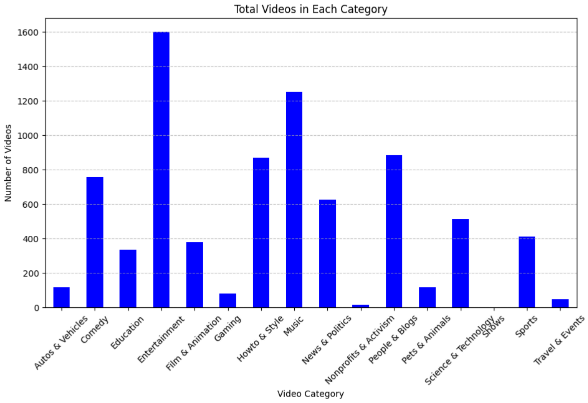
To ensure a consistent analysis, we normalized the dataset values. This allowed for a more accurate comparison. Through these methods, we gained an understanding of how trends are represented in the data.



Results and conclusion

The correlation analysis revealed a strong relationship between view count and likes, which indicates strong user engagement. Similarly, the number of likes on a comment is strongly correlated with the number of replies, as found in our correlation matrix analysis. This finding aligns with our expectations, as the number of views sets a ceiling for potential likes. Furthermore, the linear relationship between the view-to-like ratio further supports this idea, with a linear slope of 0.0354 suggesting a consistent trend.

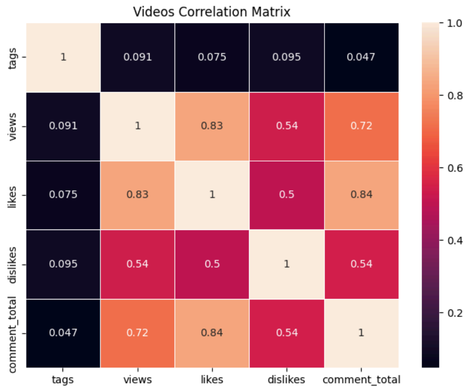
The prevalence of entertainment content is unsurprising given its widespread appeal. However, it's worth noting that creators' categorization may not always accurately reflect content themes. Since the data set does not give any information on the content of the video, we cannot assess how well the tags reflect the actual content within. Conversely, gaming, news, travel, and shows appear infrequently on the trending page. This could be indicative of lower overall user engagement among these categories.

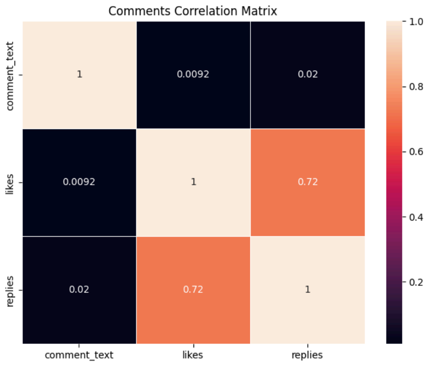


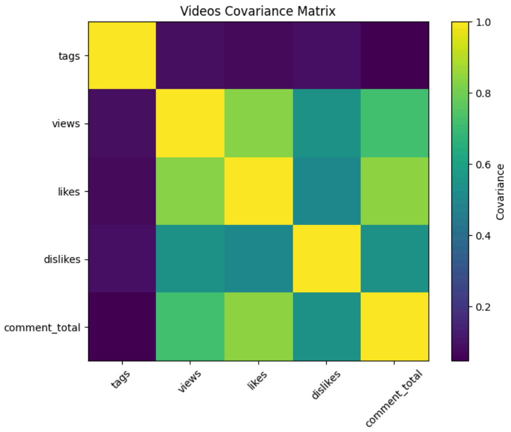
Furthermore, entertainment, music, and comedy categories tended to attract the highest view counts per video, mirroring the distribution of total videos per category. These findings matched our expectations. However, our covariance matrix showed the number of tags on a video does not correlate with a positive trend in video views, as it had a correlation coefficient of 0.091. This was contrary to our expectation that more tags increase the odds of being recommended to a user.

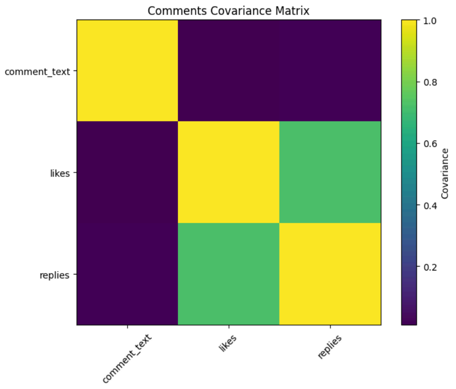
We also found that a higher number of words in a comment does not correlate to higher engagement via likes or comments. We expected longer comments to have more engagement, as longer comments are usually well-thought-out. We could have further investigated this subject by also doing a sentiment analysis on the comments to discover if the contents of the comments would see a trend in likes. However, that is something to note for another project, as time constraints prevented us from doing the analysis.

These results have helped us answer the questions of the differences in user engagement among different video categories and the relationships between different user engagement statistics. However, our research could be improved with information such as the content of the video or the amount of shares. We could also further our findings by conducting a sentiment analysis of the comments to help predict the engagement of a given comment.









Apriori Algorithm

Goals

- Seasonal Content Insights: Investigate the impact of release season on video tags to uncover seasonal trends and preferences, guiding content creation and promotion strategies throughout the year.
- Keyword Optimization: Analyze common keywords in trending video titles to identify effective keywords for maximizing viewer engagement and discoverability, informing content creators' title and description optimization efforts.
- Association Analysis: Explore associations between tags, keywords, and views to gain insights into viewer preferences and behavior, enabling data-driven decisions for content creation and promotion.

- Content Strategy Improvement: Provide actionable insights to content creators for refining their content strategy based on data-driven analysis of trends and patterns in video categorization, tagging, and keyword usage.

Process

We took a four-step approach to analyze the trending video data:

1. We investigated which categories of videos tend to attract higher view counts.
2. We identified frequent patterns of tags that appeared most among the videos using association rules.
3. We explored if the season in which a video is released impacts its tags.
4. We extracted patterns of keywords in trending video titles.

For the technical approach, we started by creating new attributes for seasons and category names. Our strategy was to pair category IDs with actual category names to make more sense. We initially constructed a dictionary mapping video IDs to their respective tags across the dataset. These tags were preprocessed and split using the vertical bar as a delimiter. This resulted in nested lists where each inner list represented tags for a single video. Then, we converted the list of video tags into a binary matrix format suitable for the Apriori algorithm. Each row represented a transaction (video), and each column represented a tag. If a tag was present in a video, the corresponding cell in the matrix was set to one; otherwise, it was set to zero.

We then utilized the TransactionEncoder to create a data frame encapsulating these tag lists. This was then fed into the Apriori algorithm from the mlxtend.frequent_patterns package, yielding a list of frequent itemsets. We refined our analysis by applying association rules to the frequent itemsets. We used a minimum support of 100 and a K_max value of 2 to calculate the following metrics: support, confidence, lift, leverage, conviction, and Zhang's metric. This was done for each antecedent and consequence rule.

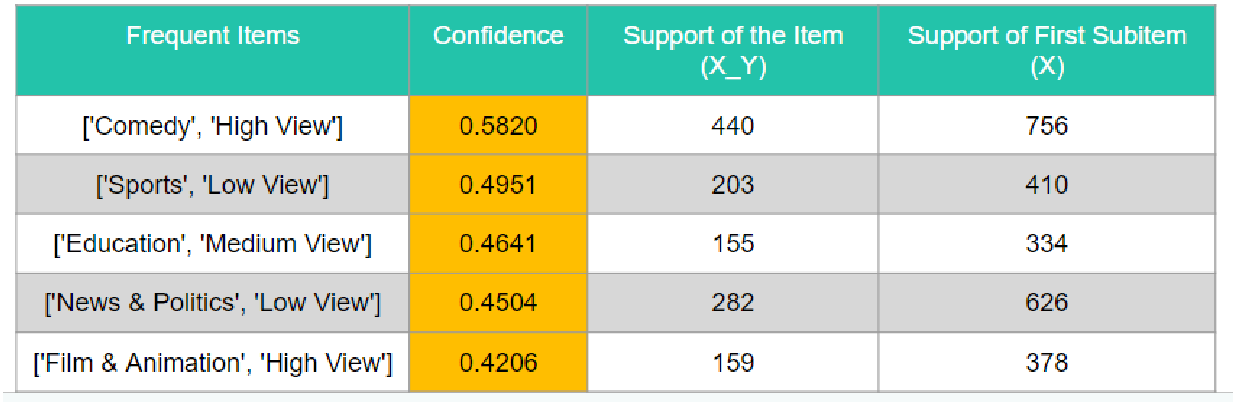
To identify common tags based on season, we separated the lists into subsets based on the season attribute created based on the date. Then, we ran both lists (fall and winter) through the same algorithm, resulting in their respective frequent itemsets (tags).

Lastly, for the identification of common keywords among trending videos, we pre-processed the data to remove filler words such as "of", "the", "a", etc. This approach enabled us to find the associations between tags, keywords, and views, providing valuable insights for a content strategy that content creators could use to improve viewer engagement.

Results and Conclusion

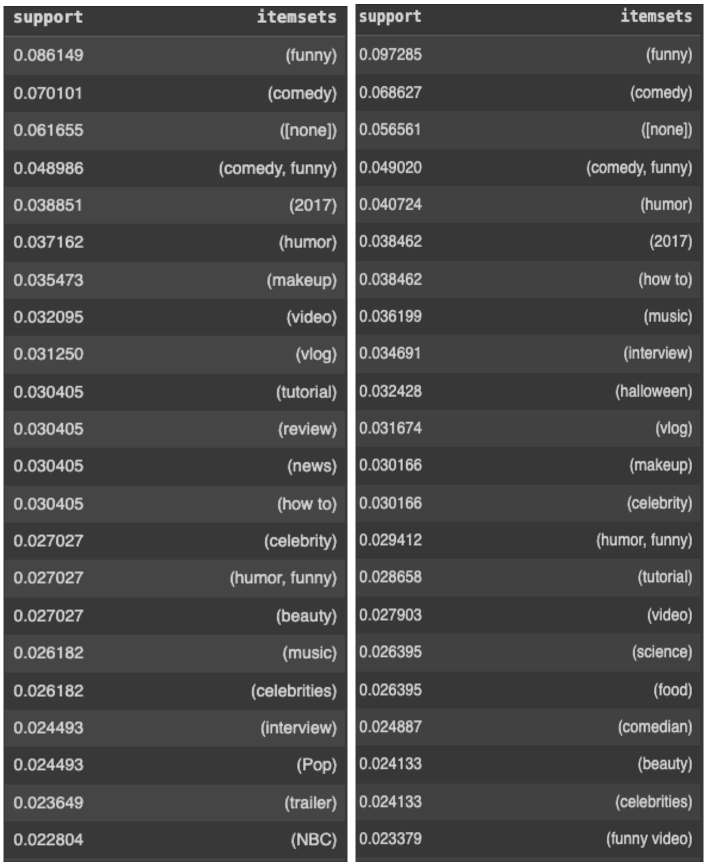
Initially, our research focused on determining which video categories attracted higher view counts. This revealed that the Comedy and Film & Animation categories tend to garner significant attention. Based on this finding, we suggest a content creator aiming for higher viewership should prioritize creating videos within these categories. Additionally, the analysis identified frequent patterns of tags through association rules, revealing overlaps such as

'Funny', 'Jokes', 'Comedy', and 'Humor', which were deemed trivial due to their close association.



Frequent Items	Confidence	Support of the Item (X_Y)	Support of First Subitem (X)
['Comedy', 'High View']	0.5820	440	756
['Sports', 'Low View']	0.4951	203	410
['Education', 'Medium View']	0.4641	155	334
['News & Politics', 'Low View']	0.4504	282	626
['Film & Animation', 'High View']	0.4206	159	378

The process of examining the seasonal impact on video tags was not fruitful. Despite minor variations between fall and winter, there wasn't a significant thematic shift. The data being limited to the small window of September (before Halloween) and January (after Christmas), may have impacted this observation. Content creators should not change their content based on the season as the results suggest no noticeable impact. We were surprised by this result as we expected seasons to influence what consumers were interested in. In the future, we could analyze common item sets based on season, not just common tags. This could enhance our understanding, providing insight into whether seasons impact certain genres more.



support	itemsets
0.086149	(funny)
0.070101	(comedy)
0.061655	([none])
0.048986	(comedy, funny)
0.038851	(2017)
0.037162	(humor)
0.035473	(makeup)
0.032095	(video)
0.031250	(vlog)
0.030405	(tutorial)
0.030405	(review)
0.030405	(news)
0.030405	(how to)
0.027027	(celebrity)
0.027027	(humor, funny)
0.027027	(beauty)
0.026182	(music)
0.026182	(celebrities)
0.024493	(interview)
0.024493	(Pop)
0.023649	(trailer)
0.022804	(NBC)

support	itemsets
0.097285	(funny)
0.068627	(comedy)
0.056561	([none])
0.049020	(comedy, funny)
0.040724	(humor)
0.038462	(2017)
0.038462	(how to)
0.036199	(music)
0.034691	(interview)
0.032428	(halloween)
0.031674	(vlog)
0.030166	(makeup)
0.030166	(celebrity)
0.029412	(humor, funny)
0.028658	(tutorial)
0.027903	(video)
0.026395	(science)
0.026395	(food)
0.024887	(comedian)
0.024133	(beauty)
0.024133	(celebrities)
0.023379	(funny video)

(Left is the Fall, Right is the winter)

Highest lifts:

- Comedic -> clip
- Jokes -> funny Video
- Jokes -> (celebrities, funny video)
- (Celebrities, funny) -> funny

Perfect Conviction (inf):

- Late night -> funny
- (jokes, funny video) -> jokes
- (jokes, celebrities) -> funny video
- (Celebrities, funny video) -> humor

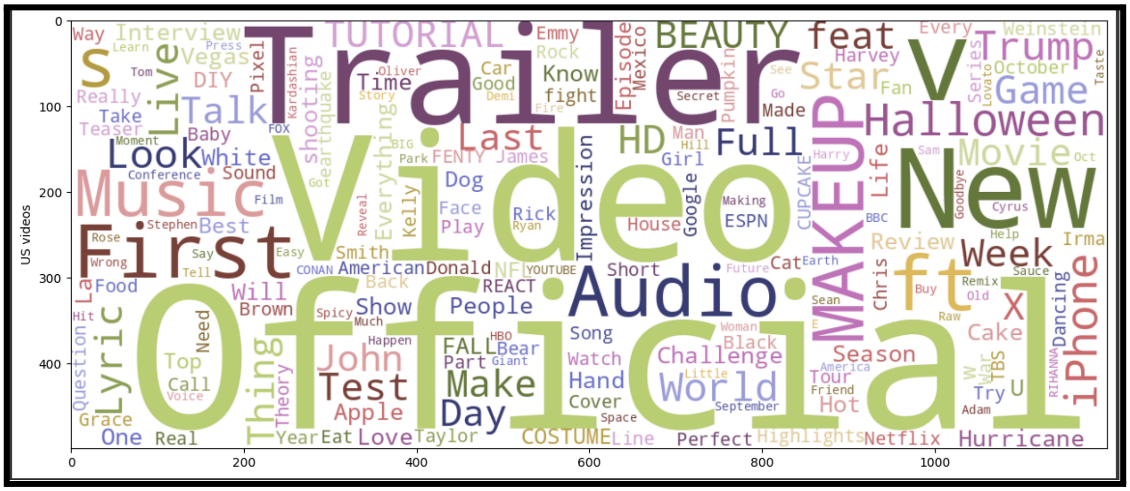
Our technical approach involved:

- Preprocessing video tags
- Converting the processed video tags into a binary matrix format suitable for the Apriori algorithm
- Applying association rules to identify frequent itemsets

Metrics such as support, confidence, lift, leverage, conviction, and Zhang's metric were calculated to assess the significance of rules. However, looking at the highest lifts and perfect convictions (inf), we see that most are trivial. For example, the rule: “Jokes -> Funny video” had the second-highest lift. However, this isn’t surprising as the tag “Jokes” heavily overlaps with “Funny video.” This same relationship follows for the highest conviction values.

We could improve this relationship by reducing overlap between the tags using a clustering algorithm. For example, we could use k-means, which groups similar tags to combat the large diversity of the tags. This would be an improvement over the method of equal-depth binning which we employed since it would handle outliers better than equal-depth.

The most common words obtained from our analysis are shown below:



To conclude, the analysis provided valuable insights for content strategy, emphasizing the importance of video categories such as 'Comedy' and 'Film & Animation' for maximizing viewer engagement. Furthermore, it highlighted the need to improve our methodologies to address overlaps in tags and seasonal variations more effectively.

Regression Analysis

Goal

- Determine what features contribute to creating a viral video.
- Use machine learning models to see another perspective of the relationships between features that provide supporting findings.
- Understand relationships between properties of viral and non-viral videos.

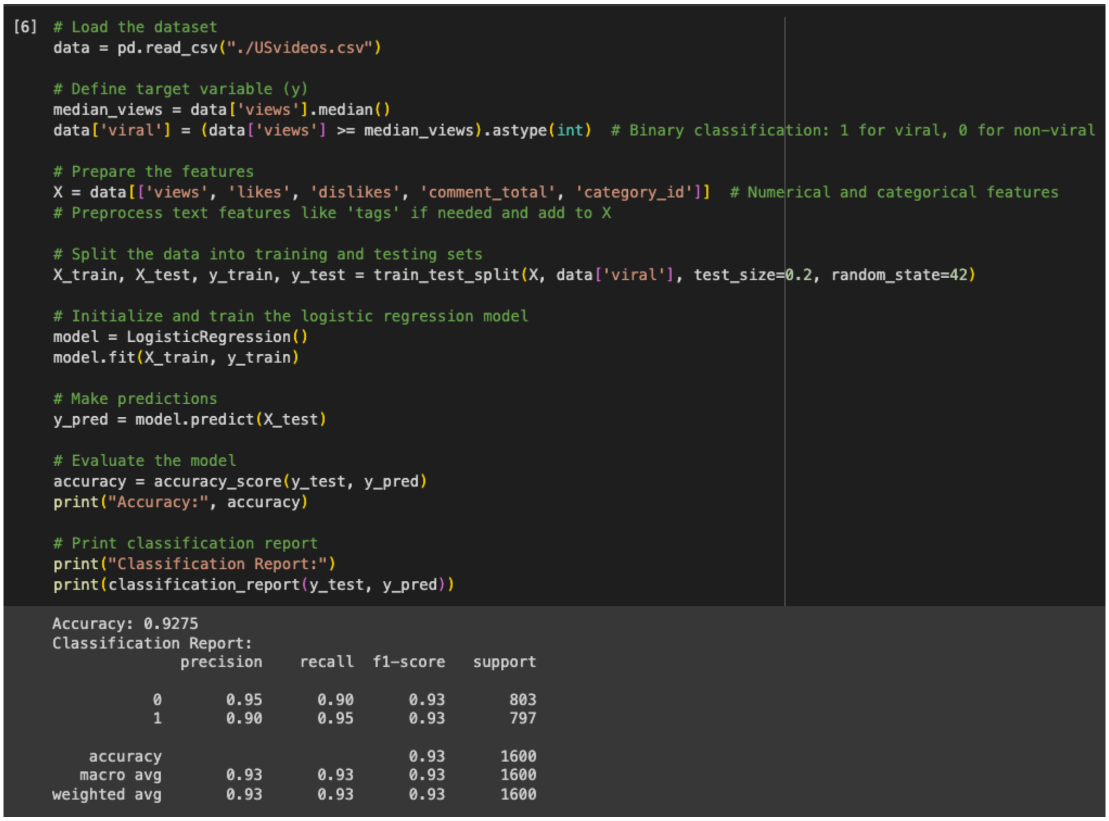
Process

For our regression analysis, we implemented two models to estimate the factors to a video's virality. We applied an 80% training and 20% testing split using the *train_test_split()* method from the scikit-learn library. We also applied a custom normalization to get better data results.

The first approach we took was logistic regression. Using logistic regression, our goal was to find what makes a viral video, given the features in our dataset. The features we used were: 'views', 'likes', 'dislikes', 'comment_total', and 'category_id'. We also made another feature called 'viral' which is our 'y_output' label. After finding the median of all views, we classified an entry based on whether the number of views was greater than the median. An entry was marked with one if larger, and zero if not.

The second approach we used was linear regression. Our goal was to evaluate if there exists a linear relationship between two quantitative values. We did this to predict if a video will be trending. Our dependent variable was ‘views’, and our independent variables were the length of a video title, and the number of video tags. We did not use likes or views as quantitative values as both are a byproduct of a trending video, rather than a precursor.

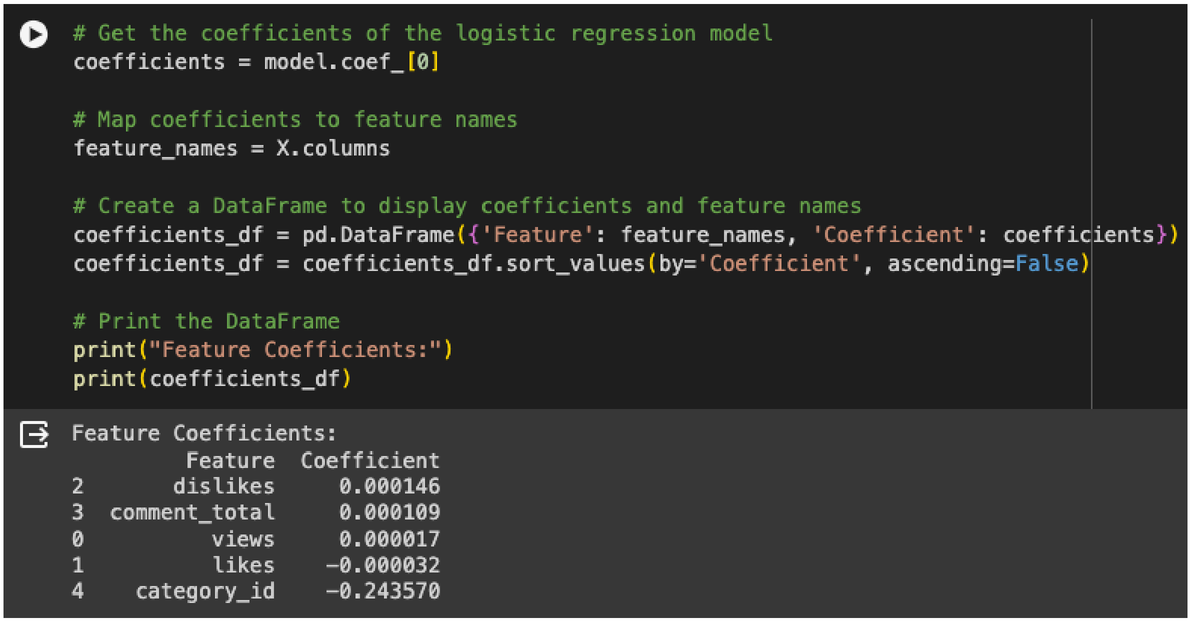
To implement logistic regression, we loaded the data within Google Colab and applied the ‘y_output’ labeling to each row entry. Then, we extracted the required features from the dataset to train. Finally, we tested the features and created the accuracy/classification report.



```
[6] # Load the dataset
    data = pd.read_csv("./USvideos.csv")

    # Define target variable (y)
    median_views = data['views'].median()
    data['viral'] = (data['views'] >= median_views).astype(int)  # Binary classification: 1 for viral, 0 for non-viral

    # Prepare the features
    X = data[['views', 'likes', 'dislikes', 'comment_total', 'category_id']]  # Numerical and categorical features
    # Preprocess text features like 'tags' if needed and add to X

    # Split the data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, data['viral'], test_size=0.2, random_state=42)

    # Initialize and train the logistic regression model
    model = LogisticRegression()
    model.fit(X_train, y_train)

    # Make predictions
    y_pred = model.predict(X_test)

    # Evaluate the model
    accuracy = accuracy_score(y_test, y_pred)
    print("Accuracy:", accuracy)

    # Print classification report
    print("Classification Report:")
    print(classification_report(y_test, y_pred))
```

```
Accuracy: 0.9275
Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.90      0.93       803
           1       0.90      0.95      0.93       797

    accuracy                           0.93      1600
   macro avg       0.93      0.93      0.93      1600
weighted avg       0.93      0.93      0.93      1600
```

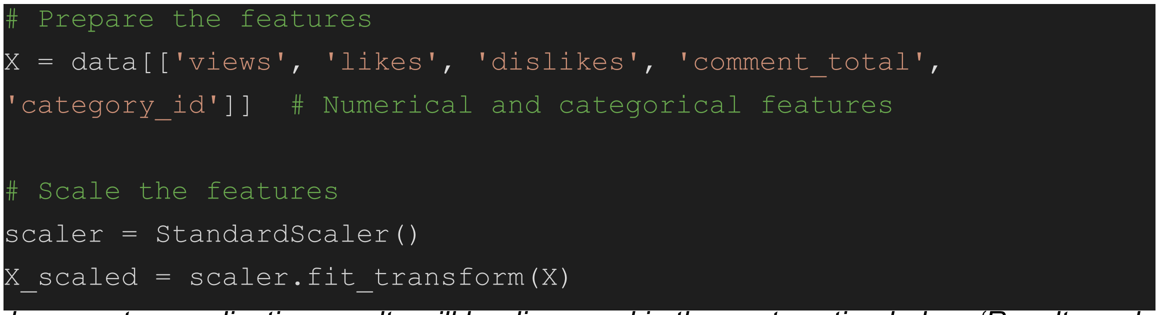
In the above image, we can see the logistic model did well in accurately classifying each entry of videos. Additionally, it accurately predicted whether a video would be viral. This is further supported by analyzing the precision, recall, and others from the classification report. It also justifies the model’s performance. However, we can’t interpret this data and answer the question of what factors make a video viral. We then decided to find the coefficients for each feature, as shown in the code snippet below:



```
# Get the coefficients of the logistic regression model
coefficients = model.coef_[0]

# Map coefficients to feature names
feature_names = X.columns

# Create a DataFrame to display coefficients and feature names
coefficients_df = pd.DataFrame({'Feature': feature_names, 'Coefficient': coefficients})
coefficients_df = coefficients_df.sort_values(by='Coefficient', ascending=False)

# Print the DataFrame
print("Feature Coefficients:")
print(coefficients_df)
```

```
Feature Coefficients:
         Feature  Coefficient
2       dislikes     0.000146
3  comment_total     0.000109
0          views     0.000017
1          likes    -0.000032
4    category_id    -0.243570
```

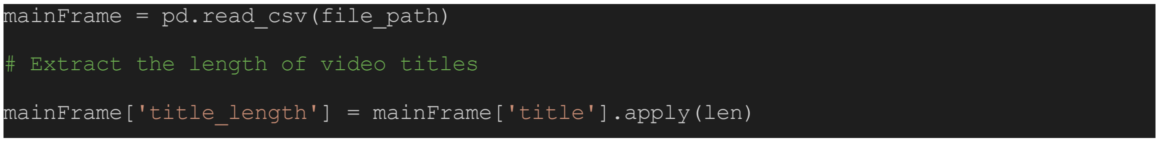
The feature coefficients indicated the impact of each feature on the probability of a YouTube video being viral. The values are small and close to each other, which does not provide any notable results. Next, we decided to try again after normalizing the data to see if we could get better results. We used the *StandardScaler()* method from the scikit-learn library to normalize the data.



```
# Prepare the features
X = data[['views', 'likes', 'dislikes', 'comment_total',
'category_id']]  # Numerical and categorical features

# Scale the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

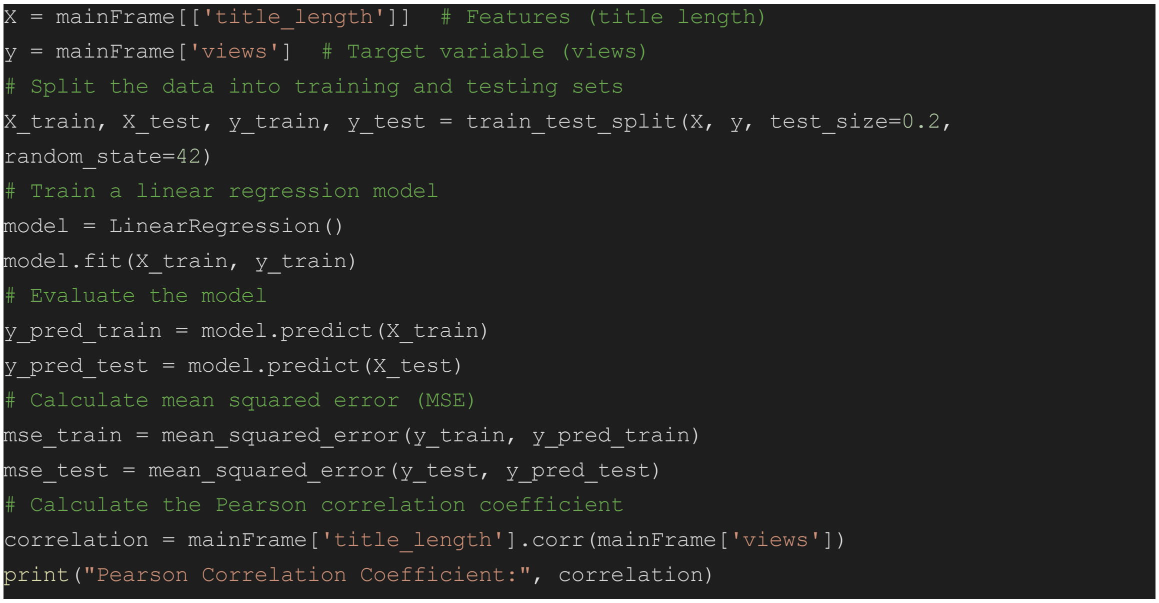
**The subsequent normalization results will be discussed in the next section below, ‘Results and Conclusion’.*

When making our linear regression algorithm, we used Google Colab with Python, importing a ‘.csv’ file containing all relevant features. We then ran a linear regression algorithm on it (see the example below). The output was a scatter plot with a linear curve line of best fit and the Pearson correlation coefficient.



```
mainFrame = pd.read_csv(file_path)

# Extract the length of video titles

mainFrame['title_length'] = mainFrame['title'].apply(len)
```

The dataset used for this algorithm came from a file “USvideos.csv”, a .csv file including the relevant features for this algorithm. The file was then loaded onto a Pandas data frame. The challenge with integrating this input data was that the two independent variables we evaluated, “video title length” and “number of tags”, were not included in the file. So, we created additional code to extract the information and make a new data frame column with it.

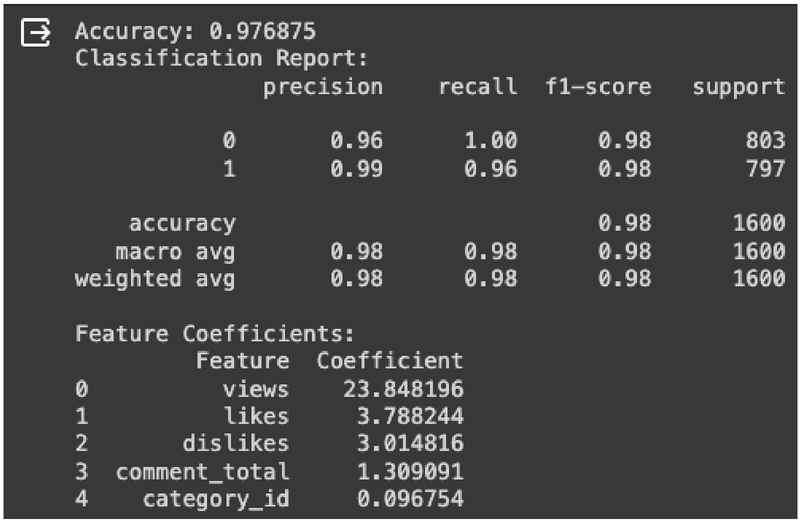


```
X = mainFrame[['title_length']]  # Features (title length)
y = mainFrame['views']  # Target variable (views)
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
# Train a linear regression model
model = LinearRegression()
model.fit(X_train, y_train)
# Evaluate the model
y_pred_train = model.predict(X_train)
y_pred_test = model.predict(X_test)
# Calculate mean squared error (MSE)
mse_train = mean_squared_error(y_train, y_pred_train)
mse_test = mean_squared_error(y_test, y_pred_test)
# Calculate the Pearson correlation coefficient
correlation = mainFrame['title_length'].corr(mainFrame['views'])
print("Pearson Correlation Coefficient:", correlation)
```

For our implementation of linear regression, we first split the data into training and testing using the 80/20 split from before. Next, we trained a linear regression model on “title length” and “number of views”. Then, we evaluated the model and calculated accuracy using Mean Squared Error. Lastly, we calculated the Pearson Correlation Coefficient using the provided library methods.

Results and Conclusion

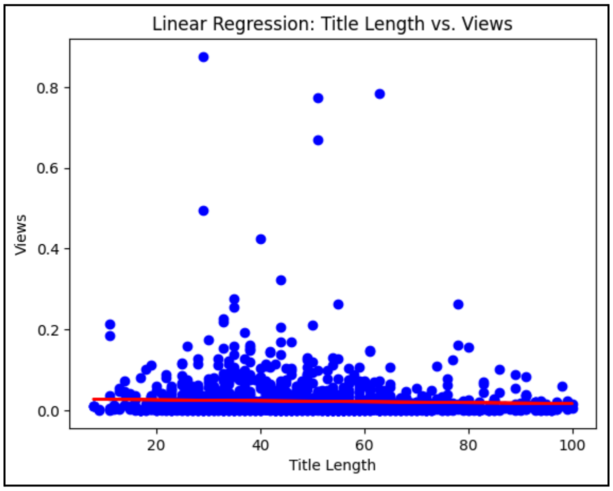
Through our analysis of the data using logistic regression and after normalization, we obtained the following:

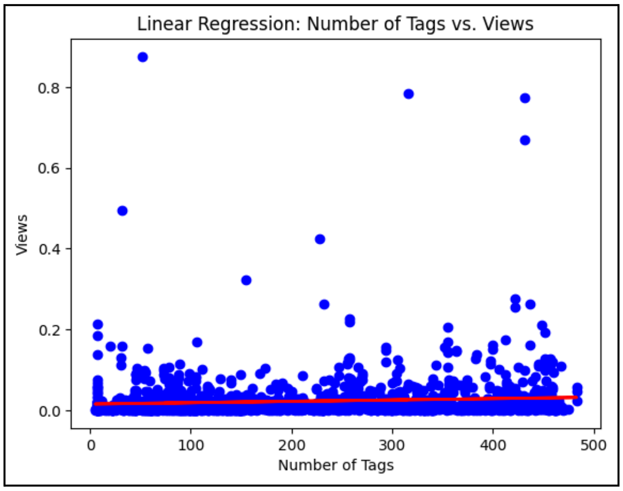


```
Accuracy: 0.976875
Classification Report:
              precision    recall  f1-score   support

           0       0.96      1.00      0.98       803
           1       0.99      0.96      0.98       797

    accuracy                           0.98      1600
   macro avg       0.98      0.98      0.98      1600
weighted avg       0.98      0.98      0.98      1600

Feature Coefficients:
         Feature  Coefficient
0          views    23.848196
1          likes     3.788244
2       dislikes     3.014816
3  comment_total     1.309091
4    category_id     0.096754
```

Compared to our original findings, the accuracy, precision, and recall, all improved after normalization. The values of our feature coefficients also appeared more correct after normalization. The feature that has the largest impact on a viral video is the number of views. We found this result to be trivial. Our next best finding was that likes and dislikes also influence a video becoming viral. Interestingly, the category of a video has little impact on its probability of going viral. We found this surprising, as we originally thought a video category strongly influenced its viral potential.

For future improvements, we would want to apply Principal Component Analysis (PCA) to extract more underlining features that are less trivial and could tell us more about how to create a viral video. Unfortunately, the number of features in this dataset was small and limited the potential of what we could find.





Next, through our analysis and use of linear regression modeling, we found no linear relationship between “video title length” and “number of views”. This is due to the corresponding correlation coefficient being -0.479, which is near zero. Additionally, the correlation coefficient of the comparison between the number of tags and video views was 0.099, which is also near zero. Overall, there was no linear relationship found between these two attributes. Thus, we concluded that the content of the title and tags primarily contribute to a video’s view count, and subsequently virality, rather than their numerical attributes or length.

Final Conclusion

In conclusion, the analysis reveals several main insights. Firstly, there is a clear positive linear relationship between views and likes on YouTube videos. Additionally, common words found in video titles such as "Trailer," "Music," and "Official" suggest potential factors contributing to video popularity. However, the impact of association rules is diminished due to overlapping tags. Notably, comedy videos tend to attract high view counts. Surprisingly, factors like tag and title length do not seem to significantly impact view count. Nevertheless, the dataset's limitations prevent precise prediction of trending videos, suggesting a need for additional numerical and non-numerical data such as share count, subscriber count, thumbnail images, and video content to enhance predictive accuracy.

Class Presentation

For a general outline of this project, please refer to our class presentation here: https://docs.google.com/presentation/d/13ye5v0RlAd7uGb99zyZ5gDUChB2UKltRSzg5-2ggnng/edit?usp=sharing

References:

[1] “Trending YouTube Video Statistics and Comments,” Kaggle, Oct. 25, 2017. https://www.kaggle.com/datasets/datasnaek/youtube

[2] Wei, Dr. Honghao “Mining Frequent Patterns, Association and Correlations: Basic Concepts and Methods.” Class lecture, CPT_S 315 Introduction to Data Mining, Washington State University, Pullman, Washington, February 5, 2024.

[3] GfG, “Linear Regression in Machine learning,” GeeksforGeeks, Mar. 14, 2024. https://www.geeksforgeeks.org/ml-linear-regression/

[4] GfG, “Logistic regression in machine learning,” GeeksforGeeks, Jan. 30, 2024. https://www.geeksforgeeks.org/understanding-logistic-regression/