Week 13: Bring It All Together

Yujia Cao, Wendi Yuan, Yuhan Zhao

Problem Definition

The rapid advancement of technology and widespread internet accessibility have transformed the way people consume content, with platforms like YouTube emerging as integral to digital culture and daily entertainment. As one of the most visited websites globally, YouTube provides a hub for creators, businesses, and audiences to interact, share, and engage. With billions of hours of video consumed daily, understanding the drivers of video popularity is critical for content creators aiming to maximize audience reach and engagement. Unveiling the factors behind a video's success, however, is a complex challenge. This is particularly true in an environment where platform algorithms and user behaviors interact dynamically. While engagement metrics such as likes, dislikes, comments, and shares are visible indicators of viewer interaction, their relationship to view counts remains intricate and multifaceted. This study investigates these connections, leveraging advanced data analysis to decode trends and provide actionable insights. By examining a diverse dataset of trending videos across multiple regions, this research empowers content creators with data-driven strategies for content optimization in an increasingly competitive landscape. Initially, the project aimed to develop a predictive model for ideal content recommendations akin to an AI-driven system. However, after reviewing the dataset's characteristics, the focus shifted to building a model to monitor YouTube video performance. This pivot provided an opportunity to generate meaningful recommendations for both creators and businesses, ensuring actionable and context-specific insights.

Data Collection & Preparation

The dataset used in this study originates from the publicly available YouTube trending video datasets hosted on platforms like Kaggle. This dataset aggregates snapshots of trending videos across various countries, offering a robust view of user engagement and platform dynamics. For this study, we selected data from two English-speaking countries to ensure consistency in language and avoid potential insights being lost due to translation discrepancies. The dataset includes detailed engagement metrics such as views count, likes count, dislikes count, comments, etc. It also contains timestamps for time-series analysis, enabling the exploration of temporal patterns in video popularity. Additional metadata, including video titles, tags, categories, and channel information, enhances the ability to categorize and analyze the

content. The "trending" label ensures that the dataset focuses on high-impact videos, ideal for studying factors contributing to video success.

To ensure the dataset was free from inconsistencies or inaccuracies that could lead to biased or erroneous results, we undertook a comprehensive data preparation process. First, we merged datasets from multiple regional contexts to provide a more holistic view of YouTube video performance. This step allowed us to pool insights across different regions and ensure a comprehensive analysis. Next, we conducted data cleaning, addressing missing values and verifying dataset consistency. Any null entries were either filled using appropriate imputation methods or removed to maintain the dataset's integrity. Outlier detection and removal were also key steps in this process. Videos with abnormally high or low engagement metrics, such as millions of likes but disproportionately few views, were flagged and examined for potential anomalies. Extreme values, which could distort statistical analyses and machine learning models, were handled using statistical methods such as the interquartile range (IQR) and visualization tools like box plots. Depending on the nature of the anomalies, these outliers were either corrected or excluded from the analysis to ensure reliable insights. To streamline the dataset, columns unrelated to the study's objectives—such as video IDs or redundant metadata—were removed. This simplification enhanced computational efficiency and focused the analysis on key variables. Additionally, text data preprocessing was carried out on meaningful columns like video titles and descriptions. This step included cleaning the text, removing irrelevant characters, and preparing the data for sentiment analysis. By addressing these issues, we ensured the dataset was both robust and ready for deeper analysis, enabling a thorough exploration of the factors influencing video performance.

Our research focuses on analyzing numerical data to explore the relationships between the dependent variable (views) and various independent variables. Specifically, we aim to identify key factors that contribute to a video's popularity, providing actionable insights for content creators. By uncovering these factors, we intend to predict effective "keywords" and strategies that creators can leverage to optimize their content and enhance its visibility to audiences. The central objective of our analysis is to use the number of views as the target variable, with all other metrics serving as independent variables, to develop a comprehensive understanding of what drives video performance on YouTube.

Exploratory Data Analysis (EDA)

To uncover meaningful insights and relationships within the data, we conducted a thorough Exploratory Data Analysis (EDA). The process began with an overview of the dataset using descriptive statistics to summarize key metrics, such as views, likes, dislikes, and comments. This initial analysis

provided a foundational understanding of central tendencies, variability, and potential anomalies within the data. It highlighted average engagement levels, identified highly engaged videos, and revealed patterns or irregularities in user interactions, setting the stage for deeper analysis.

The exploratory data analysis revealed several significant insights about the dataset and the relationships between engagement metrics and video view counts.

A correlation heatmap highlighted strong relationships between *views* and engagement metrics such as *likes*, *dislikes*, and *comment_counts. likes* had the highest correlation with views (r = 0.79); *comment_count* also showed a notable positive correlation (r = 0.48); *dislikes* had a moderate positive correlation (r = 0.40). These findings suggest that higher engagement—both positive (*likes*) and constructive (*comment_count*)—generally contributes to higher video view counts. (Appx. 1)

By analysing publishing and trending dates, we revealed patterns in video success over time. We conducted videos published closer to weekends (Friday and Saturday) showed a higher probability of trending, as visualized in histograms of publication and trending days. Trending day analysis indicated that audience engagement peaks during weekends, likely due to increased leisure time.(Appx. 2&3)

By calculating derived metrics such as "likes per view" and "comments per view," we observed a small subset of videos achieved disproportionately high engagement ratios, indicating highly interactive content relative to their reach, and videos with higher comment ratios tended to stay on the trending list longer, highlighting the impact of interactive discussions on maintaining video visibility.(Appx. 4)

Scatter plots of views against likes, dislikes, and comments revealed a few videos were extreme outliers with views exceeding 100 million, often associated with high-profile music or entertainment channels. Outliers also showed disproportionately high or low engagement ratios, emphasizing the variability in content reception.(Appx. 5)

Visualizations played a pivotal role in the EDA. Scatter plots were used to examine correlations between variables, such as the relationship between views and likes or views and comments. Histograms provided insights into the distribution of metrics like views and likes, showcasing the spread of data and revealing any skewness or outliers. Additionally, bar charts were employed to analyze categorical variables, such as the distribution of video categories and trends across days of the week. These visual tools helped identify hidden patterns that were not immediately evident in the raw data.

A key focus of the EDA was analyzing engagement metrics. Ratios such as "likes per view" and "comments per view" were computed to normalize engagement levels across videos with varying view

counts. These derived metrics allowed for fairer comparisons and highlighted the videos with disproportionately high or low audience interaction relative to their popularity. Furthermore, temporal trends were explored by grouping data by upload dates and days of the week to examine how timing affects video performance. For example, videos uploaded on certain days might trend more frequently, providing valuable insights for creators seeking to optimize their upload schedules.

Feature Engineering

After we finish the EDA based on a polished dataset, we do the feature engineering that prepares for the model trial, trying to get a more accurate result that could capture the real-world situation if the model was truly deployed.

What we implemented for the feature engineering steps could be divided into three major parts. We created a new variable *days_since_published* by calculating the number of days between a video's publish date and its trending date and grouped these into bins. Then we try the sentiment analysis, and this step includes derived sentiment polarity for the description and title columns, categorizing them into 'Positive', 'Neutral', and 'Negative' sentiments. After that, we create TF-IDF Features, which are generated textual features for the *description* and *tags* columns, identifying the top features with the highest TF-IDF scores for each video.

For the *days_since_published* variable, we first converted the publish time and trend date columns to a datetime format for analysis. Using these converted columns, we calculated the difference in the number of days to create a new feature, *days_since_published*. This feature was further grouped into meaningful intervals, such as 0–7 days, etc., to analyze trends over time. To understand the relationship between these segments and video popularity, we used box plots to visualize the impact of segments on the number of views, thus clearly presenting the effect of publication time on audience engagement.(Appx. 6) We utilize the TextBlob library in the sentiment analysis to calculate polarity scores for the *description* and *title* bar to measure the emotional tone of the text. Sentiment scores were categorized as positive, neutral, and negative, and their distribution was visualized to highlight sentiment trends in the video metadata. Additionally, we compute and compare the average sentiment scores of descriptions and titles to assess their general sentiment patterns to find how emotional content influences audience engagement. The result we get for the description sentiment score (0.1717) is slightly positive, suggesting that descriptions often include optimistic or encouraging language, and the title sentiment score (0.0478) is nearly neutral, reflecting a focus on being concise and attention-grabbing rather than expressive. (Appx. 7)

We used TF-IDF Vectorizer to extract the top 100 words from the *description* and *tags* columns. A custom function identified the top 5 TF-IDF features per row, which were added as new dataset columns to enhance text analysis and highlight key words for each video. Finally, to reduce dimension, we first removed non-numeric and irrelevant columns to focus on critical features in the analysis. We normalized the remaining numeric features using StandardScaler.

We created the *days_since_published* feature to capture the time difference between posting and popularity, which can affect the number of views and popularity of a video. Staging this feature helps identify patterns of user engagement. Sentiment analysis of *titles* and *descriptions* reveals the emotional tone of the video, which can affect viewer interaction and views. Finally, TF-IDF features highlight the most important words and phrases in descriptions and tags, helping us understand which textual elements increase video popularity and engagement. These feature engineering steps are designed to better analyze and predict the success of videos based on timing, sentiment, and content. On the one hand, this was instrumental in the preparation of the model, and on the other hand, in analyzing the different features, we also more carefully disentangled the video trend factors and combined them with realistic.

Data Splitting & Transformation

To prepare for the model building, we basically transform part of the data and split the data. We dropped irrelevant columns and removed columns such as *thumbnail_link*, *video_id*, *comments_disabled*, etc., that were not contributing to the analysis. Then we cleaned the text in the *title* column by removing stopwords, converting to lowercase, and retaining only alphabetic characters. Since the model framework should be concise, we split the data into train and test to facilitate model training and evaluation. We also did the dimension reduction through the PCA for more efficient modeling.

We removed unnecessary columns from the *merged_df* using the `drop()` method to focus on more relevant features. For data preprocessing, we utilized NLTK's list of stopwords and regular expressions to clean the text, applying a custom function, clean_text, to the *title* column in order to remove stopwords and non-alphabetic characters. To split the data, we employed scikit-learn's `train_test_split()` function, dividing the dataset into a train set (80%) and a test set (20%), ensuring proper training and evaluation of the model.To reduce the complexity of the dataset, we applied Principal Component Analysis (PCA) to transform the data into a low-dimensional space while retaining key information content. To amplify the understanding, we visualize the PCA. (Appx. 8) This step simplified the dataset for subsequent modeling, improving efficiency and interpretability.

We choose to do these solid preparations to make sure of the integrity of the dataset. Dropping irrelevant columns helped focus on more meaningful features. And data preprocessing ensures that only the relevant words are considered in the analysis, improving the quality of features for modeling. After we split the data into training and testing sets, it allows us to train the model on one subset and evaluate its performance on a separate subset, preventing overfitting and ensuring better generalization. Reducing feature dimensions is another way to improve efficiency and minimize the risk of overfitting to focus on the most influential features.

Model Building

After several trials and comparisons between models, we finally used XGBoost to build a predictive model that estimates the number of YouTube video views based on various features. The model was trained using a set of hyperparameters, including learning rate, number of estimators and maximum depth of the tree. To evaluate the performance of the model, key metrics such as Root Mean Square Error (RMSE) and coefficient of determination (R^2) are computed for both the training and test datasets. These metrics help assess the model's ability to generalize and accurately predict unseen data.

To train the model, we used XGBRegressor from the XGBoost library, which is well suited for regression tasks. The model was trained using the training data (x_train and y_train) and predicted on the training and test sets (train_preds and test_preds). To evaluate the performance of the model, we computed the RMSE using scikit-learn's mean square error function, which quantifies the average prediction error on the training and test sets. In addition, we calculated the R^2 score using the R^2_score function, which indicates how well the model explains the variance of the data. In order to optimize the performance of the model, we defined hyperparameter variations, starting with a learning rate of 0.05, 200 estimators, and a maximum depth of 6. These parameters were chosen to strike a balance between model accuracy and complexity, and the results of the different hyperparameter configurations were stored in a DataFrame for further comparison.

We chose XGBoost because of its ability to effectively handle complex feature relationships, especially when faced with a prediction task such as the number of YouTube video views. First, XGBoost is able to automatically recognize interactions between different features; for example, the *title* and *description* of a video may jointly affect the number of views. Through a tree model splitting process, XGBoost captures non-linear relationships and interactions between features. In addition, XGBoost can calculate the importance of features to help us identify which features have the greatest impact on the prediction results, such as the sentiment score of the title, the video type, and the date of release, among

other factors. XGBoost is also good at handling nonlinear relationships, as the relationship between the number of video viewings and some features is often not linear but rather a complex nonlinear pattern. Through the construction of a multilayer decision tree, the XGBoost can accurately capture these relationships. In addition, XGBoost is able to learn more complex feature associations through feature combinations, e.g., the combination of the time of release and the video type may affect the popularity of a video more than the time or type alone. With these advantages, XGBoost helps us gain a deeper understanding of the multiple factors that influence the number of YouTube video views, providing powerful support for video popularity prediction.

Model Evaluation

To ensure the reliability and accuracy of predictions, it is critical to evaluate the performance of the predictive model. Model evaluation helps determine how well the model generalizes to new, unseen data and whether its predictions align with the actual outcomes. In this project, we aimed to build a robust model for predicting YouTube video views, and the evaluation phase played a vital role in understanding the model's effectiveness. Through systematic testing, we ensured that the model met accuracy benchmarks and identified areas for potential improvements.

The evaluation process involved the computation of key performance metrics, specifically Root Mean Square Error (RMSE) and R^2. RMSE was used to measure the average error in prediction, with a lower value indicating higher accuracy. R^2, on the other hand, quantified the proportion of variance in the data explained by the model, reflecting the goodness of fit. These metrics were calculated separately for both the training and testing datasets to assess the model's ability to generalize beyond the training phase. (Appx. 9)

The results demonstrated the strength of the chosen XGBoost model. The model achieved an RMSE of approximately 2.12 million on the training set and 2.91 million on the test set. The R^2 values were 0.98 and 0.96 for training and testing, respectively, showcasing a strong fit with minimal overfitting. These results indicate that the model effectively captures the underlying patterns in the data, making it suitable for predicting YouTube video views with a high degree of accuracy.

Model Deployment and Monitoring

Before we deploy the model, we scratch a monitoring plan in order to make the model fit different kinds of needs. Our YouTube video performance model is best suited for batch processing, leveraging historical data from 2017–2018 for efficient bulk analysis of engagement metrics like

sentiment and interaction trends. Batch processing optimizes resource usage, simplifies implementation, and aligns with our objective of providing periodic insights for content creators to improve engagement strategies, rather than focusing on real-time feedback. To ensure model effectiveness in production, we will monitor key performance metrics such as RMSE and MAE for accuracy and reliability, alongside business metrics like engagement uplift and CTR to evaluate the model's impact on user interaction and platform goals. Operational metrics like latency, throughput, and system uptime will ensure scalable, efficient deployment, maintaining alignment with strategic objectives while delivering actionable and reliable insights.

Deployment is a pivotal stage that bridges the gap between model development and real-world application. By deploying the model, we enable its use in practical scenarios, such as predicting video views for content planning and strategy. Deployment also ensures the scalability and reusability of the model. It’s allowing it to be integrated into various pipelines. This stage includes both model serialization and the establishment of a monitoring framework.

The model was serialized using the Pickle library, a lightweight and efficient way to save and load machine learning models. This serialization ensures that the model can be reused without retraining, significantly reducing computational overhead. To prepare for real-world deployment challenges, a comprehensive monitoring plan was developed. This plan addresses two major issues: data drift, where the distribution of input data changes over time, and concept drift, where the relationship between features and the target variable evolves.

These proactive measures ensure the long-term reliability of the deployed model. Regular monitoring of the model’s predictions against real-world data will allow timely recalibration, ensuring sustained performance. With the deployment strategy in place, the model is operationally ready to provide valuable insights into video view trends and adapt to future changes in YouTube’s dynamic environment. Understanding the predictions of a machine learning model is as important as achieving high accuracy. Model interpretation provides transparency, which is critical for building trust and gaining actionable insights. In this project, interpreting the XGBoost model’s predictions helped us identify the key factors influencing video views, allowing for better content strategy and audience engagement decisions.

Feature importance was extracted from the model to quantify the contribution of each variable to the predictions. The title is the most influential feature, contributing 43.7% to the overall feature importance score (0.437), underscoring its critical role in performance by significantly affecting click-through rates (CTR). Catchy, descriptive titles with relevant keywords enhance search visibility and

drive higher engagement through improved views and likes. The second most important feature, publish_time (0.193), emphasizes the impact of timing, as videos released during peak activity hours or aligned with audience behavior often perform better initially, creating a compounding effect on overall metrics. Engagement metrics, which combine likes, dislikes, and comment counts, emerged as the most significant factors. Timing variables, including the day of the week and the elapsed time between publishing and trending, were also influential. These insights reveal that user interactions and strategic publishing times are crucial for optimizing video performance. (Appx. 10)

The SHAP (SHapley Additive exPlanations) values offered a detailed and granular perspective on how individual features influenced specific predictions. To enhance our understanding of the model's behavior, we employed SHAP, a method grounded in cooperative game theory, to interpret the predictions more effectively. For the randomly selected sample 49053, the most influential features driving the prediction are *dislike*s, *likes, engagement_metrics, comment_count,* and *category_id*, as revealed by their SHAP values. Positive SHAP values indicate features that increase the likelihood of a specific outcome, while negative values have the opposite effect. The SHAP graph provides actionable insights for YouTube content creators by highlighting areas for optimization, such as increasing *dislikes, likes, engagement metrics,* and *category_id,* or *decreasing comment_count*, to shift predictions and improve video performance. SHAP values quantify each feature's contribution to a specific prediction, identifying whether a feature increased or decreased the likelihood of a predicted class. As SHAP is well-suited to tree-based models like XGBoost, we calculated SHAP values for a subset of predictions. For instance, videos with high engagement metrics consistently showed higher predicted views, while publishing during certain time windows significantly boosted viewership. These findings underscore the importance of optimizing both content quality and timing to achieve better performance on YouTube. (Appx. 11 & 12)

Before model development, extensive exploratory data analysis (EDA) was conducted to uncover patterns in the dataset. This analysis revealed valuable insights into video performance trends and their influencing factors. By visualizing key features and relationships, we gained a comprehensive understanding of the data, which informed both the feature engineering and modeling stages.

Visualizations such as scatter plots, word clouds, and box plots further enriched the analysis. Scatter plots illustrated the relationships between views and engagement metrics, while word clouds highlighted frequently used terms in titles, tags, and descriptions. Box plots showed that videos published during specific time windows achieved higher view counts, emphasizing the importance of strategic

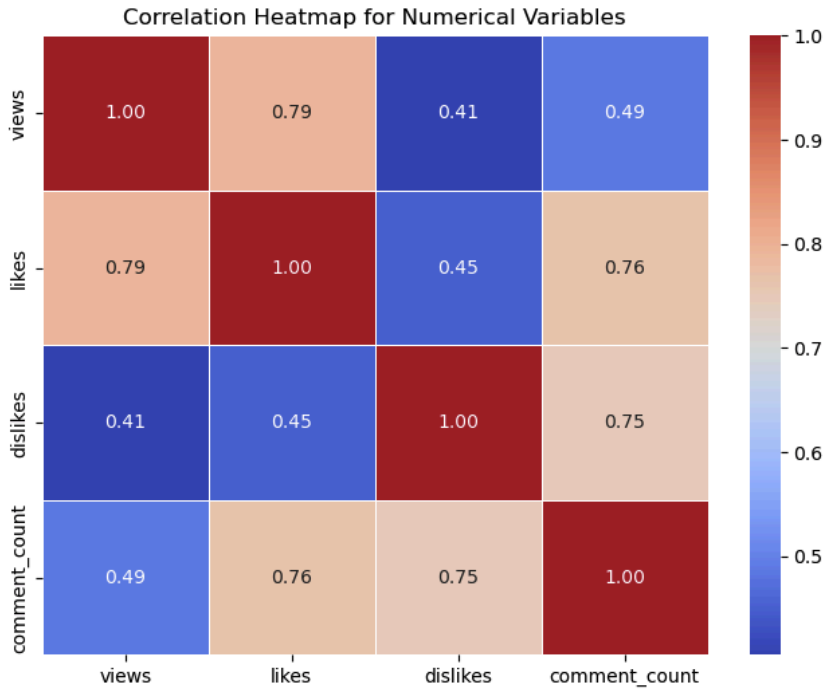
timing in content release. These insights provided a solid foundation for building a predictive model tailored to YouTube's dynamic ecosystem.

Conclusion

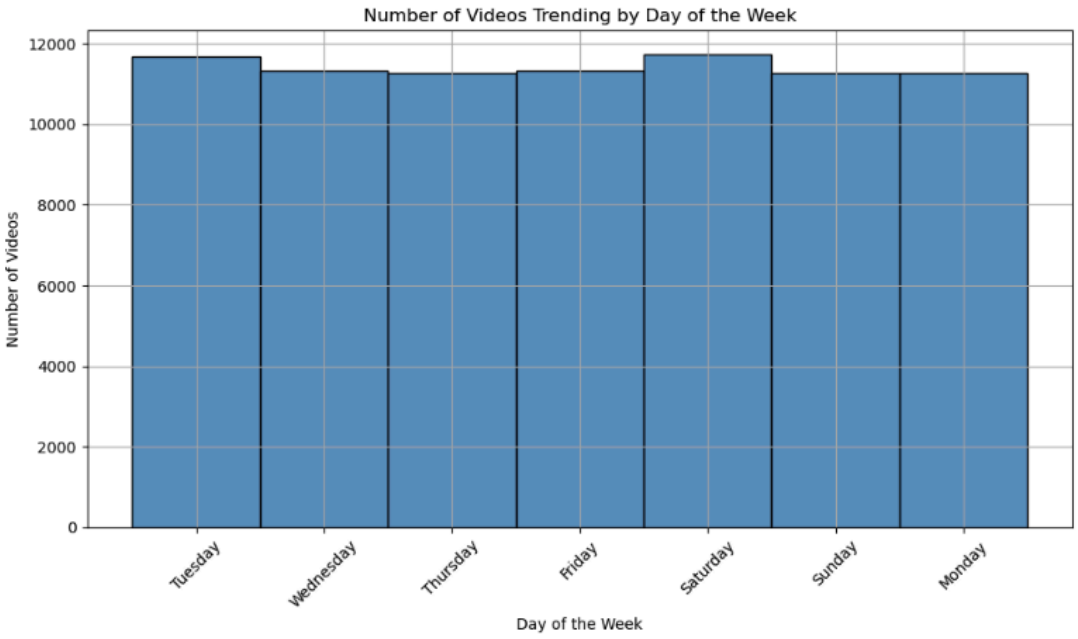
This project successfully demonstrated the application of machine learning to understand the factors driving YouTube video view counts, with a focus on engagement metrics. The XGBoost model performed exceptionally well in predicting video views, offering high accuracy and generalizability. The model's insights provide actionable recommendations for content creators to optimize their videos and strategies based on key engagement metrics, such as likes, comments, and dislikes. Exploratory data analysis and visualization further enhanced our understanding of the dataset, revealing important trends and patterns, such as the impact of publishing timing and engagement ratios on video success.

Future work will involve integrating additional features, such as regional and demographic data, to refine the model’s predictions and further tailor recommendations. Automating the monitoring framework will ensure the model remains adaptable to evolving content trends and audience behaviors. Additionally, incorporating advanced interpretability techniques will allow for deeper insights into the factors influencing video performance, ultimately fostering more informed and effective decision-making for content creators.

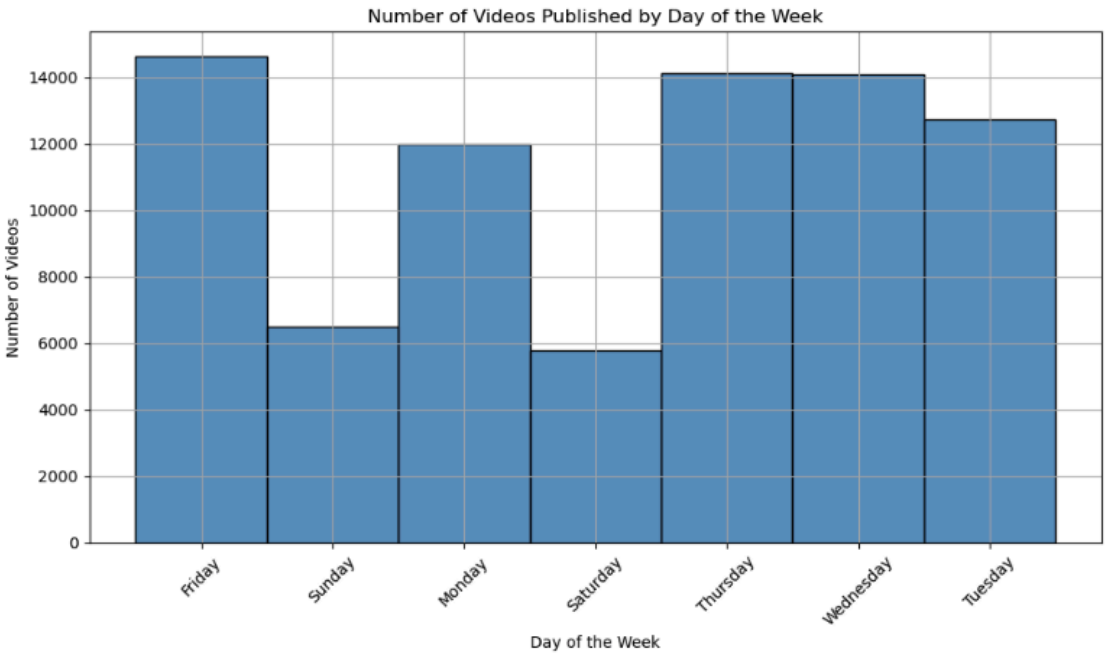
Appendix



Appx. 1 EDA for Correlation Heatmap



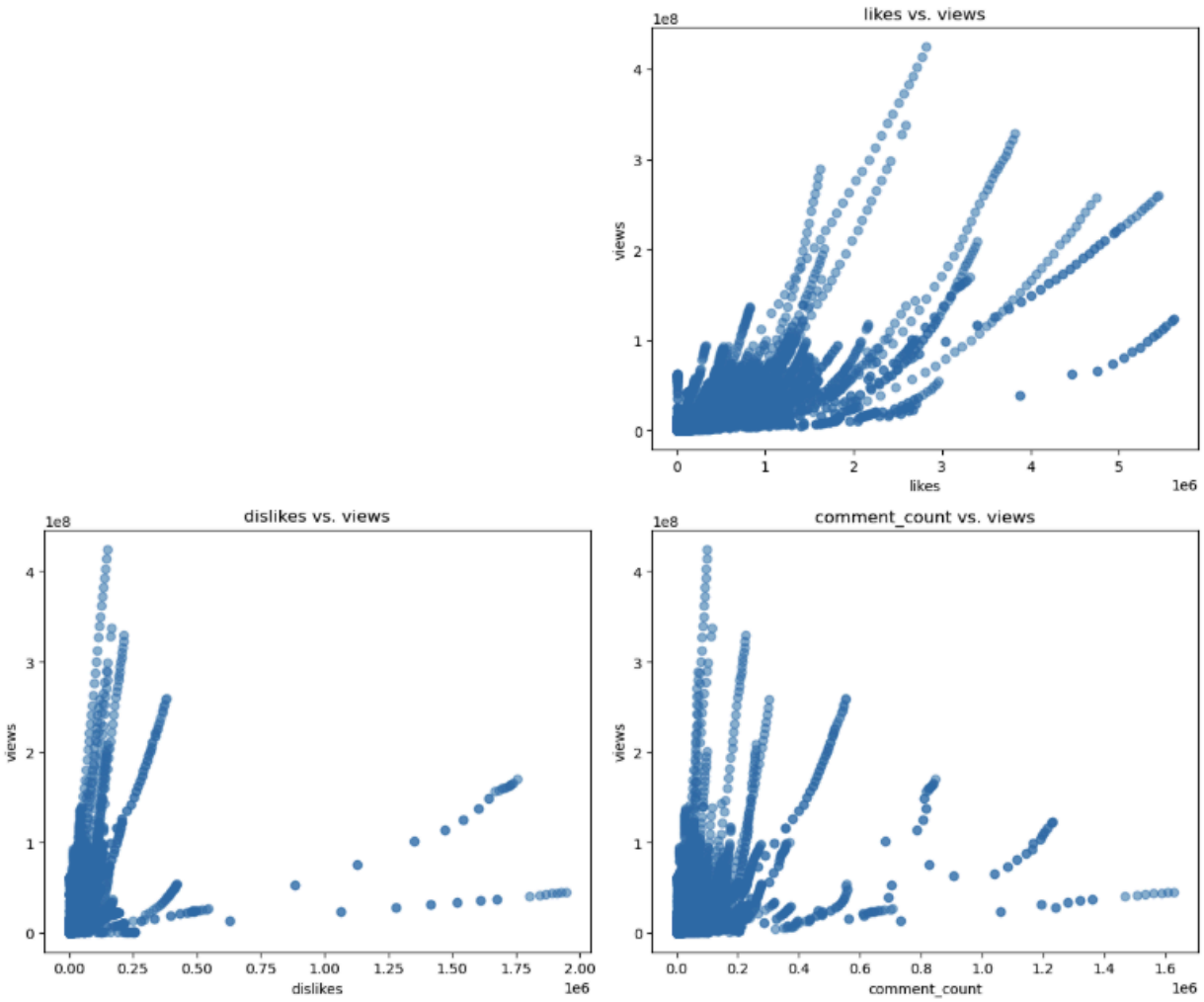
Appx. 2-EDA for Video Trending Date



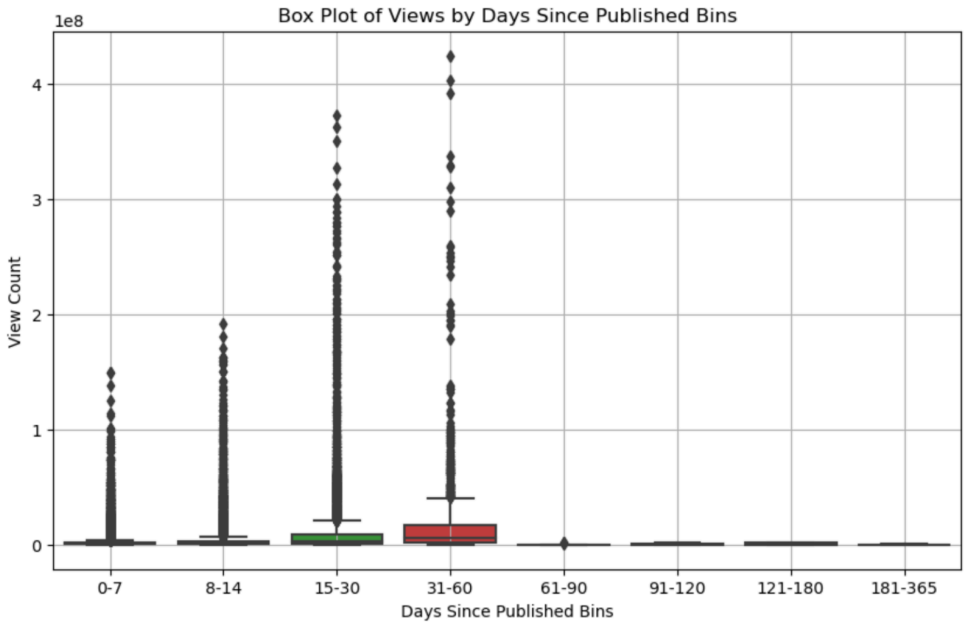
Appx. 3-EDA for Video Publishing Date

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

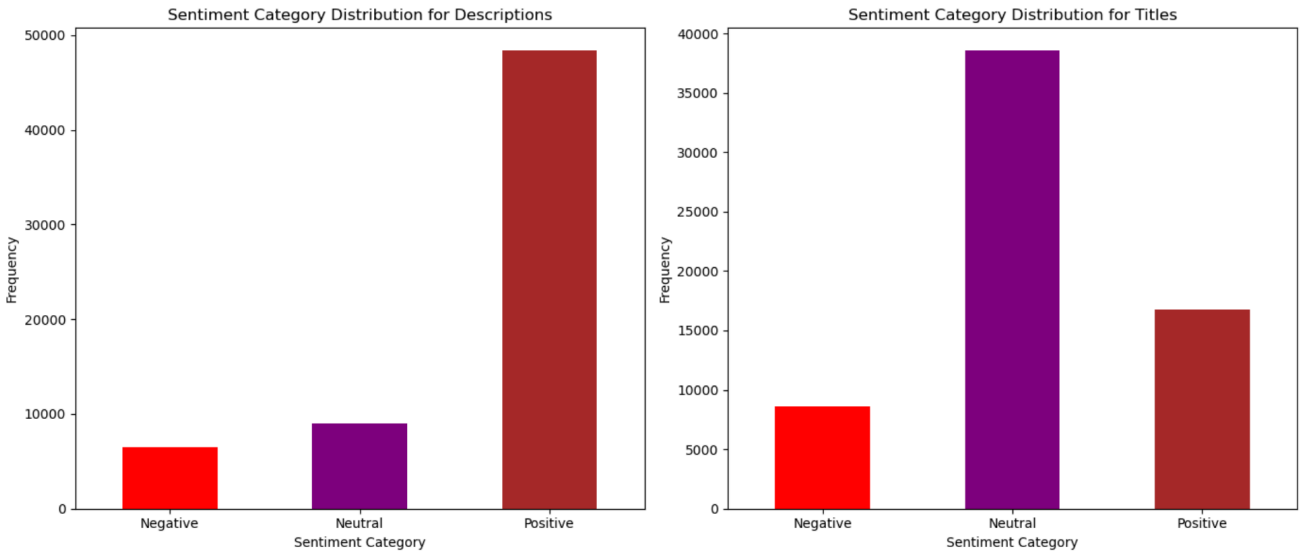
Appx. 4-Assign Weights for Engagement Metrics



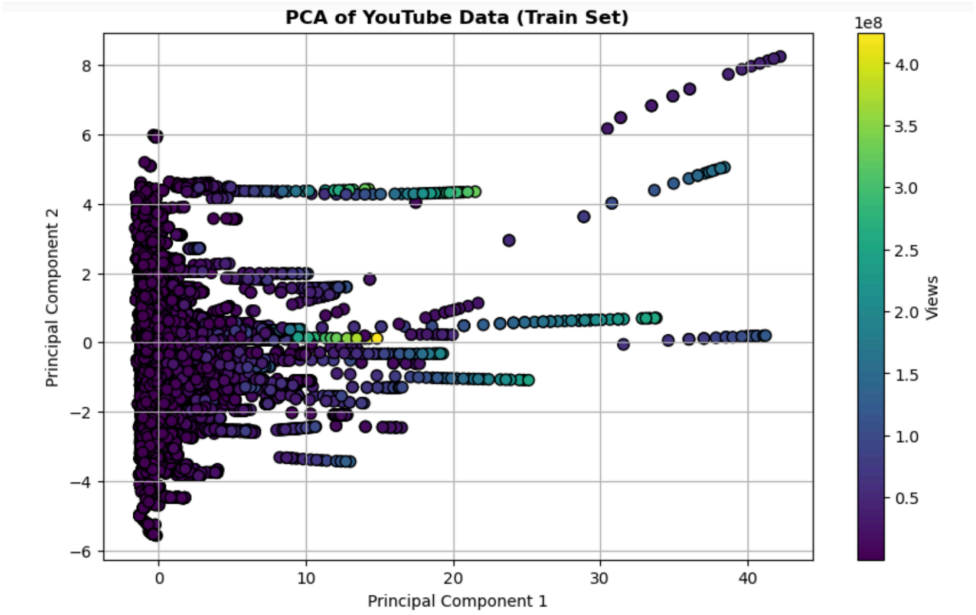
Appx. 5-Scatter Plot of Numerical Variables



Appx.6-Box Plot for *days_since_published*



Appx.7-Sentimental Distribution for *description* and *title*




```
Explained Variance per component:
PC1: 43.35%
PC2: 12.56%
PC3: 11.40%
PC4: 10.22%
PC5: 7.73%
PC6: 7.47%
PC7: 5.90%
```

Appx. 8-PCA Visualization

```
Comparison of XGBoost Model Variations:
     Variation    Train RMSE     Test RMSE  Train R^2  Test R^2
0  Variation 1  2.120971e+06  2.909246e+06   0.978047  0.960791

 Model Variation: 1
Hyperparameters: {'learning_rate': 0.05, 'n_estimators': 200, 'max_depth': 6}
Variation        Variation 1
Train RMSE    2120971.173631
Test RMSE     2909245.576456
Train R^2           0.978047
Test R^2            0.960791
Name: 0, dtype: object
```

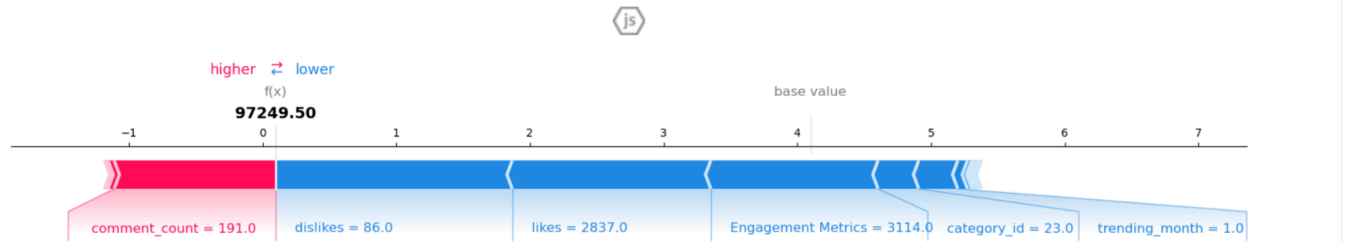
Appx. 9-XGBoost Model Performance Metrics

```
Top 10 Features:
           Feature  Importance
1            title    0.437080
4     publish_time    0.193274
2    channel_title    0.085307
5             tags    0.051353
3      category_id    0.048652
14            rank    0.040345
15        new_text    0.027795
6            likes    0.009272
13           score    0.009106
0    trending_date    0.008889
```

Appx. 10-Feature Importance

	category_id	likes	dislikes	comment_count	Engagement Metrics	score	rank	trending_day_of_week_Monday	trending_day_of_week_Saturday	trending_day_of_v
49053	23	2837	86	191	3114	1341.569024	67271.0	False	False	

1 rows × 22 columns



Appx. 11-SHAP visualization

```
Top contributing features:
                     shap_value
feature
dislikes           -1.770361e+06
likes              -1.485569e+06
Engagement Metrics -1.251085e+06
comment_count       1.207505e+06
category_id        -3.041747e+05
To flip from 0 to 1, consider increasing the values of the most positive SHAP-contributing features.
 - dislikes: Consider a increase of 1770361.00 SHAP impact units to potentially achieve a prediction flip.
 - likes: Consider a increase of 1485569.25 SHAP impact units to potentially achieve a prediction flip.
 - Engagement Metrics: Consider a increase of 1251085.25 SHAP impact units to potentially achieve a prediction fli
p.
 - comment_count: Consider a decrease of 1207504.88 SHAP impact units to potentially achieve a prediction flip.
 - category_id: Consider a increase of 304174.72 SHAP impact units to potentially achieve a prediction flip.
```

Appx. 12-SHAP Value and Analysis