YouTube Video Popularity Prediction and Engagement Analysis

CMPSC 445 Section 001: Machine Learning Data Science

Professor Yang

Author: Jomiloju Odumosu

1. Description of the Project

This project predicts YouTube video view counts using two datasets: scraped video metadata and API-provided engagement metrics. Regression models are trained to evaluate which features best influence video popularity, and feature importance techniques identify key attributes. The project provides insights into video performance trends and demonstrates the differences between scraped versus API-based predictions.

2. How to Use

Training:

- Load either the scraped (`Youtube_Data.csv`) or API (`youtube_videos.csv`) dataset.
- Preprocess the data by cleaning, scaling, and encoding features.
- Split data into training (80%) and testing (20%) sets.
- Train regression models (Linear Regression with RFE, Random Forest, or XGBoost) on the processed features.

Inferencing:

- For new videos, input the same set of features as used in training.
- Preprocess these features similarly (scaling, encoding, etc.).
- Apply the trained model to predict view counts or engagement metrics.
- Visualize results using feature importance charts and actual vs. predicted plots.

3. Data Collection

Used Tools

- **Python**
- **Pandas, NumPy** for data manipulation
- **Selenium** for scraping (`Youtube_Data.csv`)
- **YouTube Data API** for API dataset (`youtube_videos.csv`)

Collected Attributes

Scraped Data (`Youtube_Data.csv`):

- `Title`, `URL`, `Views`, `UploadDate`, `Comments`, `FirstTag`, `Category`

API Data (`youtube_videos.csv`):

- `video_id`, `title`, `duration_seconds`, `days_since_upload`, `views`, `likes`, `comments`, `channel_title`, `subscribers`

Number of Data Samples

- Scraped: 266
- API: 1254

API Usage

- YouTube Data API v3 for video statistics and metadata, including views, likes, comments, and subscriber count.

Sample Data After Preprocessing

Scraped Data Example:

Title	Views	UploadDate	Comments	FirstTag	Category
BEST 20 Perfect 10/10 Switch Games	472000	N/A	0	no_tag	no_category
Top 10 BEST Nintendo Switch Games	1200000	N/A	0	no_tag	no_category

API Data Example:

video_id	title	duration	days_since	views	likes	comments	channel_title	subscribers
S6p123T0Fbc	Marvel Rivals Vs Overwatch	673	81	97777	2913	764	CR0W	4910

2iTJB8qaeqM	Overwatch Players Try Marvel Rivals	3011	293	244748	6183	481	Salty Phish	806000

4. Data Preprocessing

- Before training the regression models, the raw datasets were processed to ensure data quality, reduce noise, and create meaningful features for the models. The following steps were applied:
- **Missing Value Handling (Imputation):**
  - Target variable views rows with missing values were dropped.
  - Numerical features such as comments, likes, subscribers, duration, and days_since_upload were imputed with the **median** value to replace missing entries, reducing bias from outliers.
- **Outlier Treatment (Clipping):**
  - Extreme values in views were clipped to the **2nd and 98th percentiles** to prevent them from disproportionately influencing the regression model.
- **Skewness Correction (Log Transformation):**
  - Skewed numerical features (views, comments, likes, subscribers) were **log-transformed using log1p** to reduce skewness and stabilize variance for better model performance.
- **Feature Engineering:**
  - **Text Features (Scraped Data):** TF-IDF vectorization was applied to Title and FirstTag to convert text into numerical features capturing term importance.
  - **Length Features:** Computed title_length and title_word_count (number of characters and words in video titles) for both datasets; first_tag_length for scraped data; channel_name_length for API data.
- **Categorical Encoding (One-Hot Encoding):**
  - For scraped data, the Category column was converted to **one-hot encoded dummy variables** to allow numerical model input.
- **Normalization/Scaling:**

    - Numerical features were **MinMax scaled** to range [0, 1] to ensure equal contribution of each feature to the regression model.
- **Train-Test Split:**
    - The dataset was split into **80% training** and **20% testing** sets to evaluate model generalization performance.
-

5. Feature Engineering

Feature engineering transforms raw data into meaningful input features for machine learning models. This step is critical to improve model performance by providing informative, normalized, and predictive features.

Scraped Data (`Youtube_Data.csv`)

1. Basic Text Features

- `title_length`: The number of characters in the video title. Longer titles may contain more descriptive information, which can affect engagement.
- `title_word_count`: Counts the number of words in the title. Titles with more words can be more descriptive or click-worthy, potentially affecting views.
- `first_tag_length`: Length of the first tag, providing a rough measure of the specificity of content labeling.

2. TF-IDF Vectors

- `Title` and `FirstTag` are converted into **TF-IDF (Term Frequency-Inverse Document Frequency)** vectors.
    - TF-IDF transforms textual data into numerical features that capture **how important a word is to a particular video relative to all videos**.
    - This allows the model to understand which terms in titles or tags are associated with higher engagement.
    - Example: If “Nintendo” appears frequently in high-view videos, TF-IDF assigns a higher weight to this term.

3. One-Hot Encoding for Categories

- The `Category` column is converted into dummy variables (0/1).
- This allows the model to consider the impact of the video category (e.g., Gaming, Education) without imposing a numerical hierarchy.

4. Combined Feature Set

- After engineering, the scraped dataset features include:
  - Numeric features: `uploaddate`, `comments`, `title_length`, `title_word_count`, `first_tag_length`
  - TF-IDF features from titles and first tags
  - Category one-hot encoded features

API Data (`youtube_videos.csv`)

1. Basic Numeric Features

- `title_length` and `title_word_count`: Similar to scraped data, capturing descriptive richness of titles.
- `channel_name_length`: Number of characters in the channel name, which can sometimes correlate with brand recognition or professionalism.
- `duration`: Video length in seconds. Viewer engagement often correlates with video duration.
- `days_since_upload`: Age of the video in days, useful to normalize views over time.
- `likes`, `comments`, `subscribers`: Direct engagement metrics and channel popularity indicators.

2. Combined Feature Set

- The final API feature set includes:
  - Video characteristics: `duration`, `days_since`, `title_length`, `title_word_count`, `channel_name_length`
  - Engagement metrics: `likes`, `comments`, `subscribers`

Scaling and Normalization

- Both datasets use **MinMaxScaler** to scale features between 0 and 1.

- Scaling ensures that features with different magnitudes (e.g., `comments` in thousands vs `title_length` in tens) do not bias the model training.
- This is particularly important for regression models like Linear Regression and tree-based models such as Random Forest or XGBoost.

Why These Features Matter

- Scraped data relies heavily on textual content (titles, tags) to infer engagement, since explicit engagement metrics are incomplete or missing.
- API data contains explicit engagement features (`likes`, `comments`, `subscribers`) that provide strong predictive signals.
- Feature engineering ensures that the model captures both **intrinsic video characteristics** (title, tags, duration) and **extrinsic engagement signals** (likes, comments, subscribers) to predict view counts more accurately.

6. Model Development and Evaluation

Train and Test Partition

- Both datasets were split into **training (80%) and testing (20%) sets** to evaluate the predictive performance of the models.
- This ensures that the model is tested on unseen data, providing a realistic estimate of generalization performance.
- Scrambled sampling with `random_state=42` was used to ensure reproducibility.

Model-1: Scraped Data

Machine Learning Model:

- **Linear Regression** was used due to its simplicity and interpretability, along with **Recursive Feature Elimination (RFE)** to select the most predictive subset of features.
- RFE ranks features by importance and iteratively removes the least significant features, resulting in the **top 25 features** used for model training.
- TF-IDF features, text lengths, comment counts, and category dummies were included as predictors.

Input:

- Preprocessed features from the scraped dataset including numerical features (`uploaddate`, `comments`, `title_length`, `title_word_count`, `first_tag_length`), TF-IDF vectors for titles and first tags, and one-hot encoded categories.

Size of Train Data:

- 212 samples (80% of 266).

Selected Attributes:

- Top 25 features based on RFE.

Performance:

- Training MSE: ~4.64
- Training R^2: ~0.155
- Test MSE: ~7.16
- Test R^2: ~-0.126

Evaluation Notes:

- The low R^2 indicates that **text-based features alone are insufficient** to explain variations in view counts for scraped data.
- SHAP analysis highlights which TF-IDF terms and categorical features contributed most to predictions.
- This model serves as a baseline to understand the predictive power of scraped metadata.

Model-2: API Data

Machine Learning Model:

- **Linear Regression** with **RFE**, leveraging richer engagement metrics available via the API.
- Explicit engagement features (`likes`, `comments`, `subscribers`) provide stronger signals than scraped metadata.

Input:

- Preprocessed API features including `duration`, `days_since`, `likes`, `comments`, `subscribers`, `title_length`, `title_word_count`, and `channel_name_length`.

Size of Train Data:

- 1003 samples (80% of 1254).

Selected Attributes:

- `duration`, `likes`, `comments`, `subscribers`, `channel_name_length`

Performance:

- Training MSE: ~0.48
- Training R^2: ~0.93
- Test MSE: ~0.58
- Test R^2: ~0.91

Evaluation Notes:

- High R^2 and low MSE indicate that API features **capture most of the variance** in video views.
- Feature importance ranking shows that **likes and subscribers** are the most influential predictors, followed by `comments`, `duration`, and `channel_name_length`.
- The model demonstrates that API-provided engagement metrics dramatically improve predictive performance compared to scraped data alone.

Feature Importance Analysis

Scraped Data:

- Determined using **RFE** (which selects the top features) and **SHAP values** for detailed contributions of TF-IDF terms and categorical variables.
- Provides insights into which textual or categorical features drive view counts.

API Data:

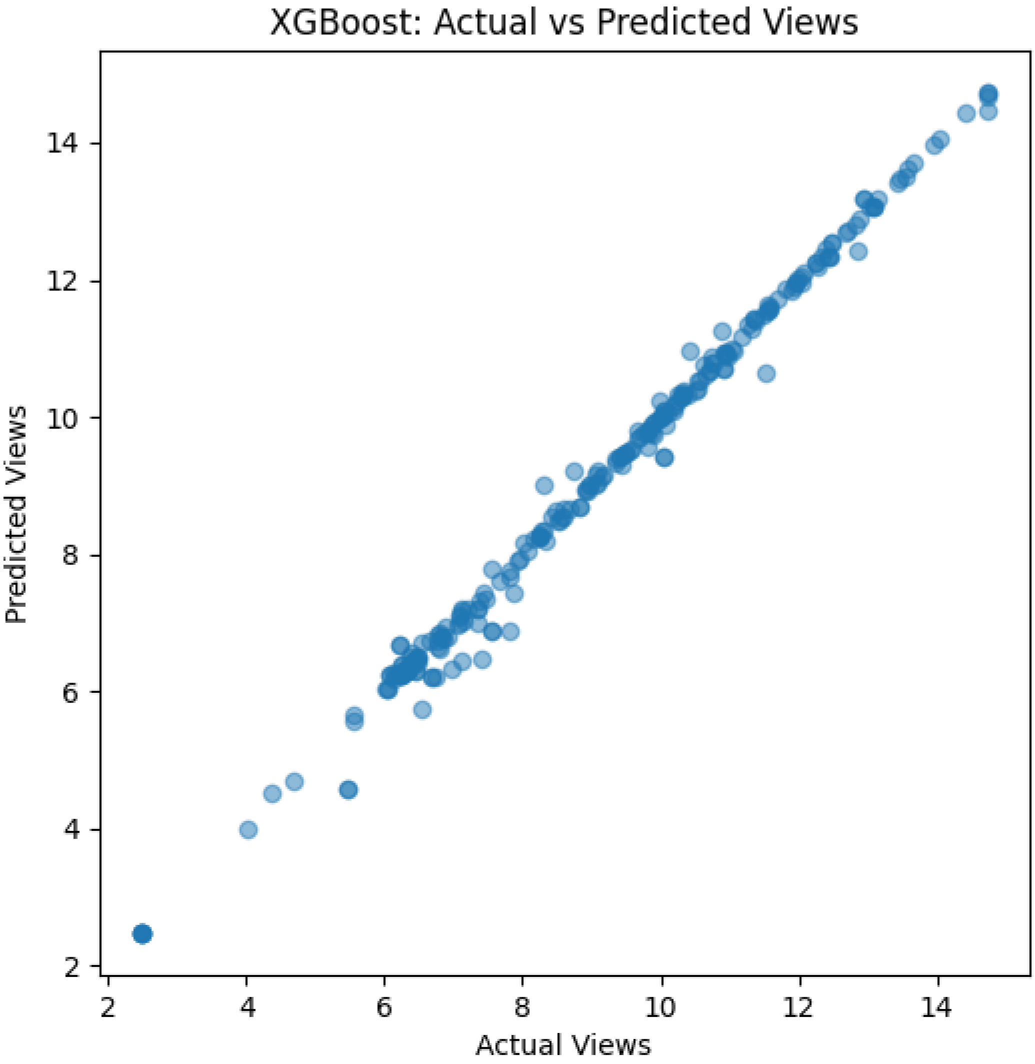
- Determined using **RFE feature ranking** in scikit-learn.
- Indicates that **engagement metrics** (likes, comments, subscribers) dominate importance, while video characteristics (duration, channel name) are secondary.

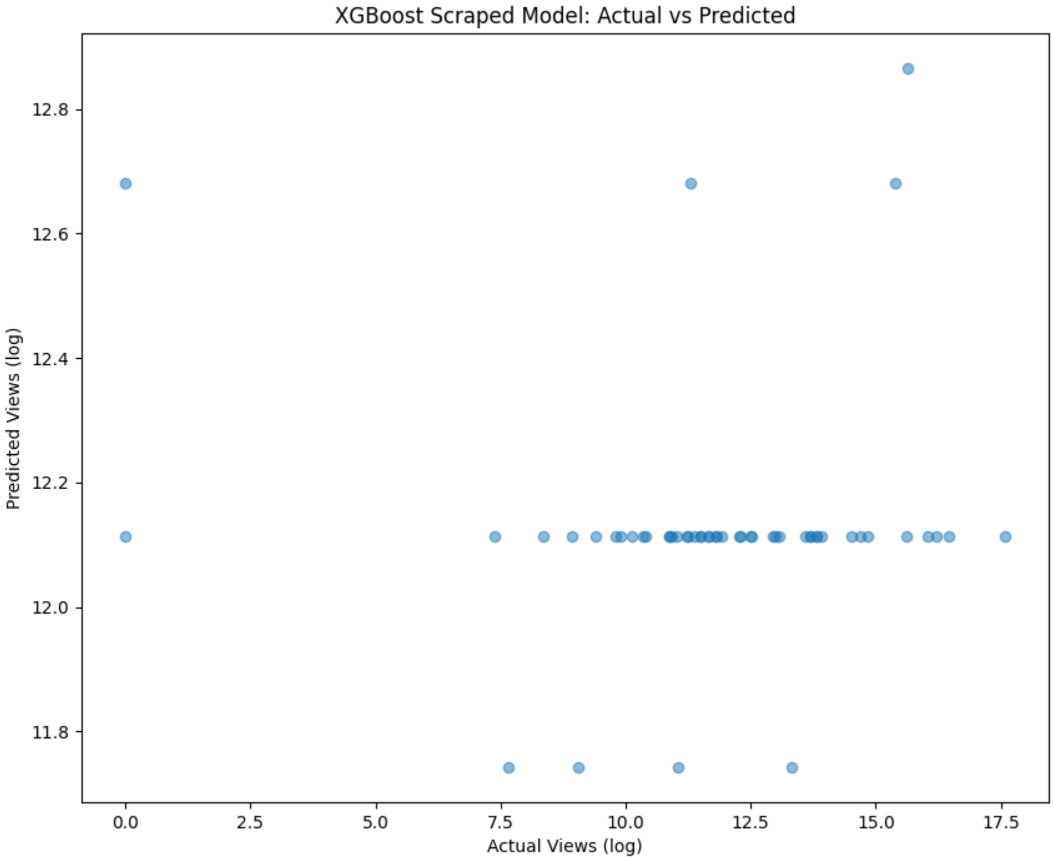
Interpretation:

- Comparing feature importance between models highlights how **rich, structured engagement metrics** yield far more predictive power than scraped metadata.
- SHAP visualizations for scraped data can reveal subtle patterns, e.g., certain title words or categories that slightly increase predicted views.

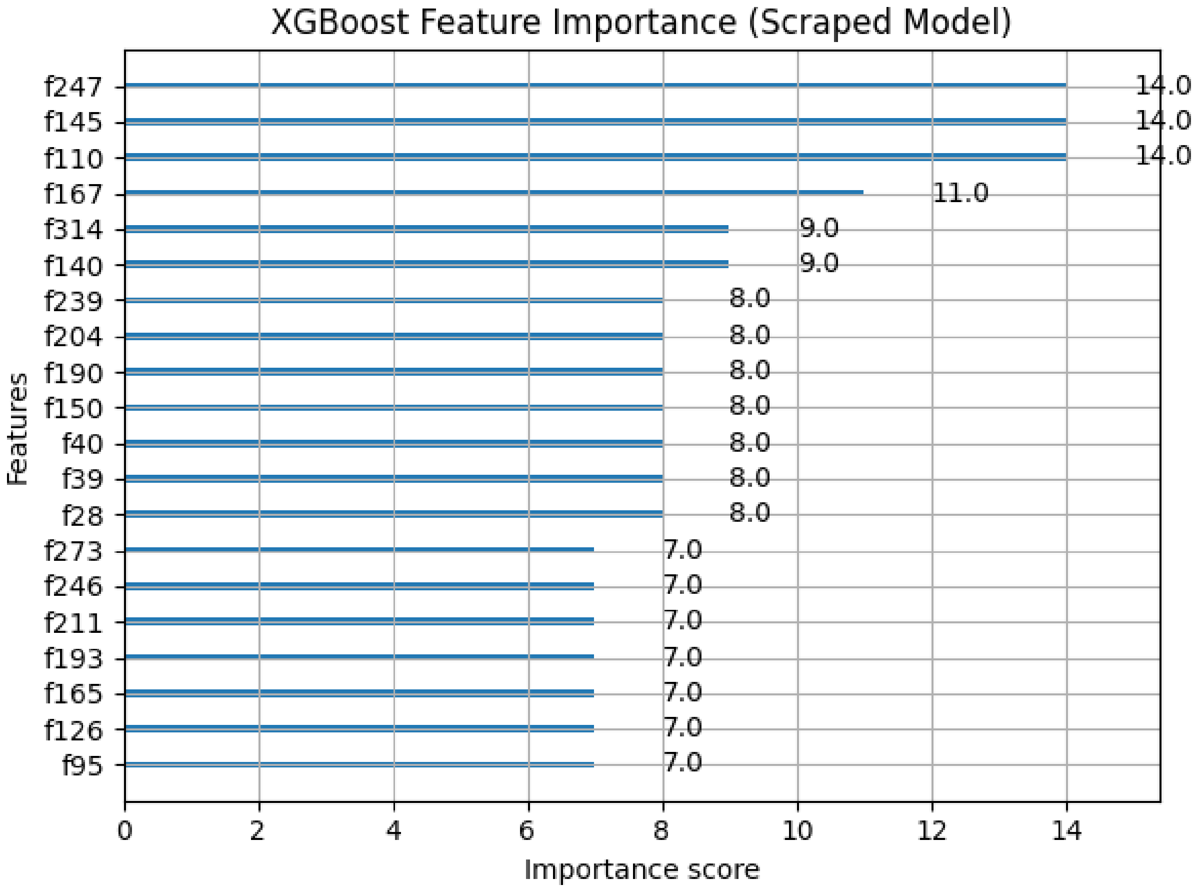
7. Visualization

1. **Prediction Accuracy Comparison**

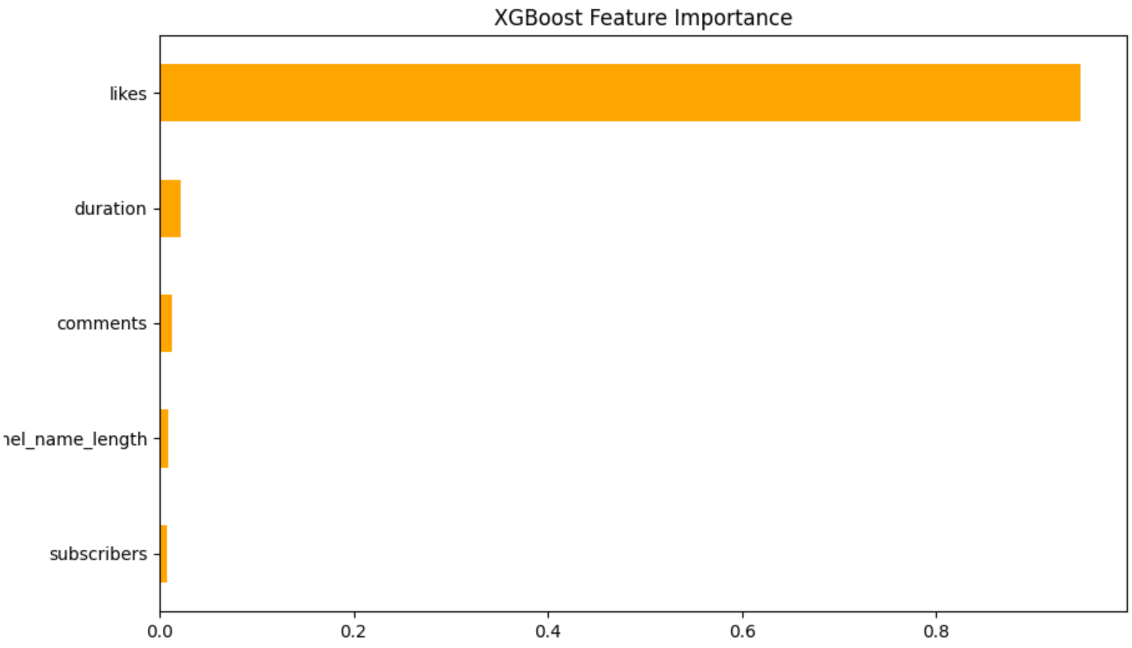

XGBoost: Actual vs Predicted Views
Predicted Views
Actual Views
2
4
6
8
10
12
14
2
4
6
8
10
12
14



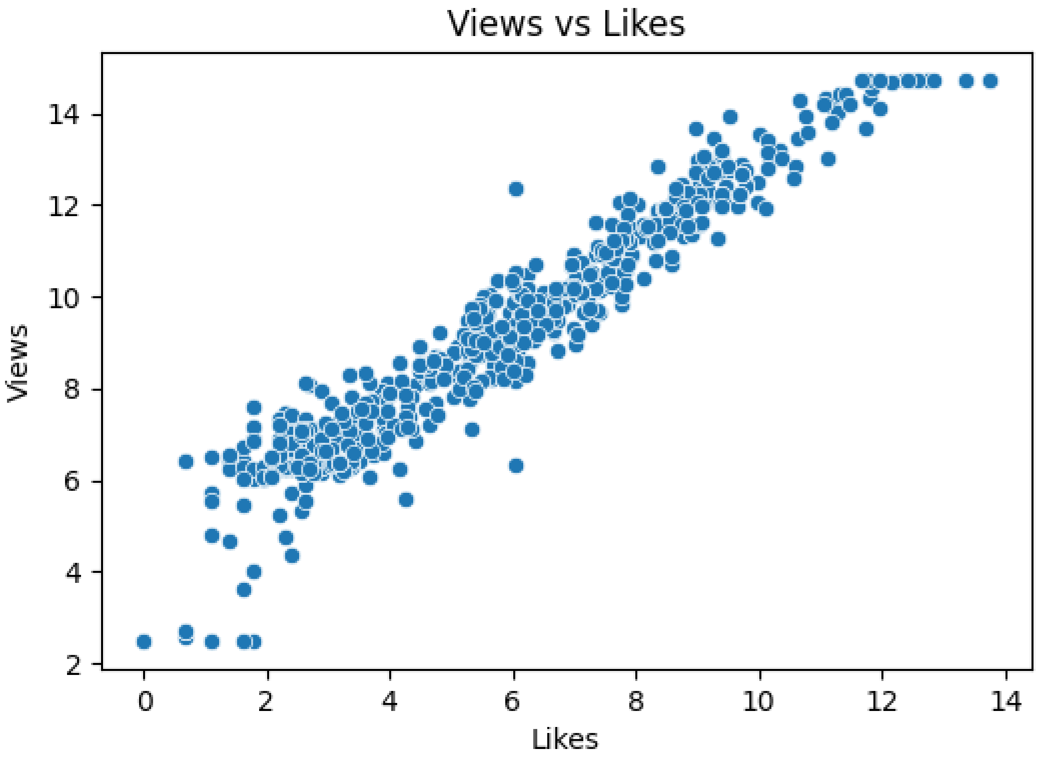
2. **Feature Importance**
    a. Scraped model:


XGBoost Feature Importance (Scraped Model)
Features
Importance score
f247 14.0
f145 14.0
f110 14.0
f167 11.0
f314 9.0
f140 9.0
f239 8.0
f204 8.0
f190 8.0
f150 8.0
f40 8.0
f39 8.0
f28 8.0
f273 7.0
f246 7.0
f211 7.0
f193 7.0
f165 7.0
f126 7.0
f95 7.0
0
2
4
6
8
10
12
14

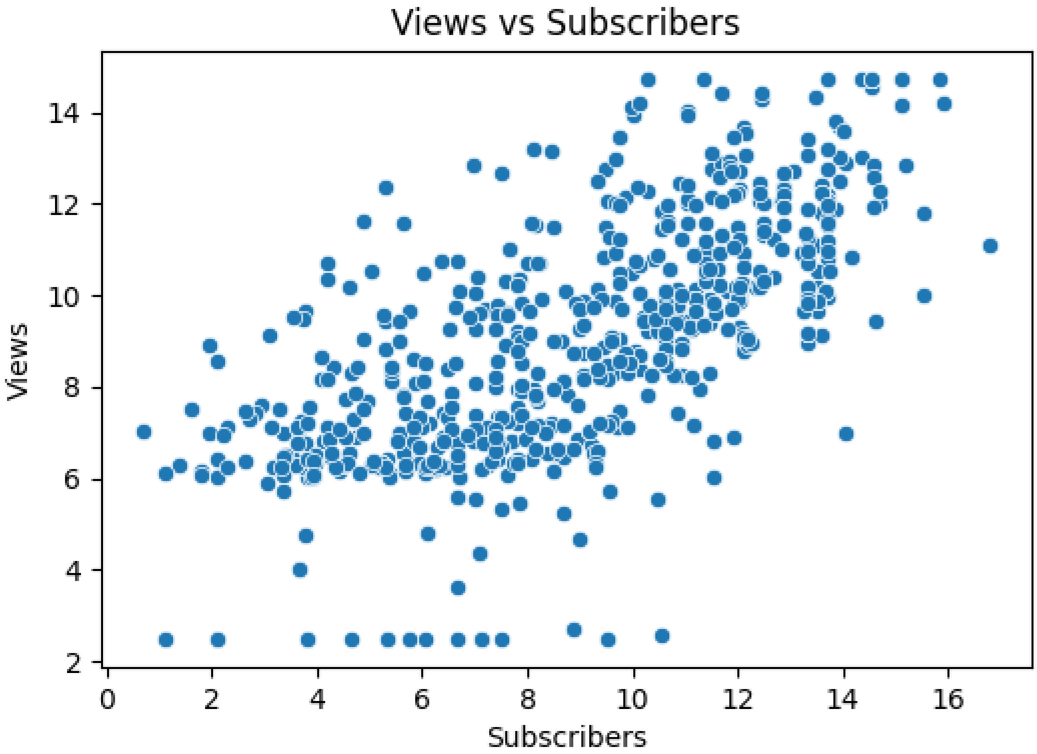
b. API model:



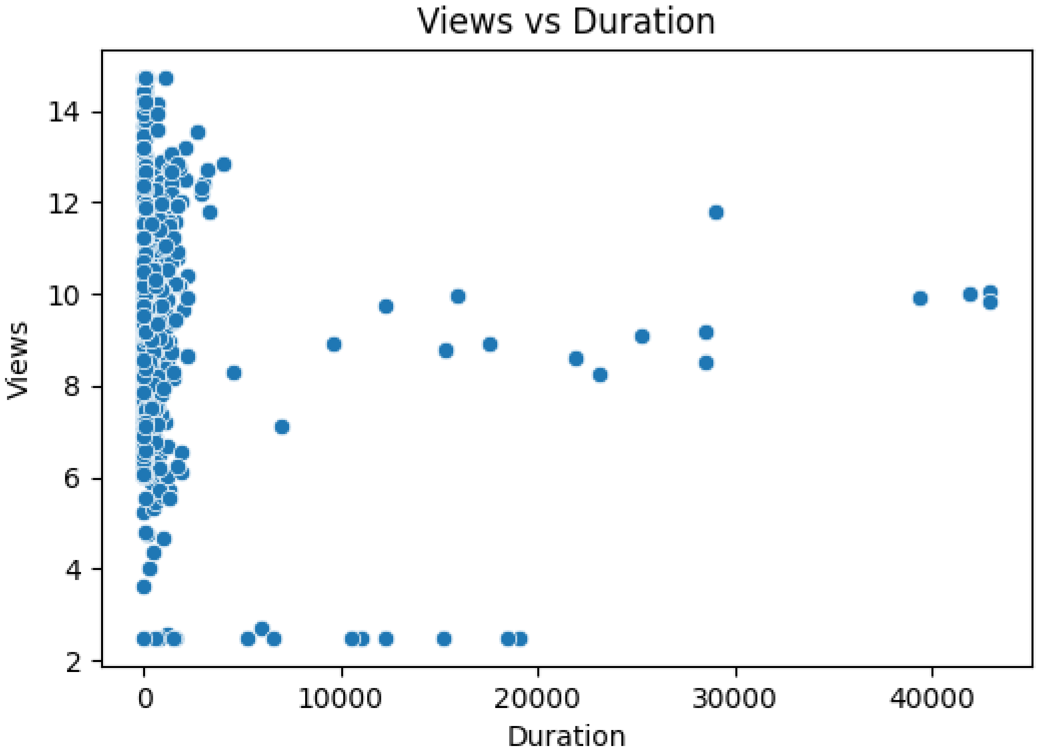
3. **Engagement Trends (API Data)**
    a. Views vs Likes:



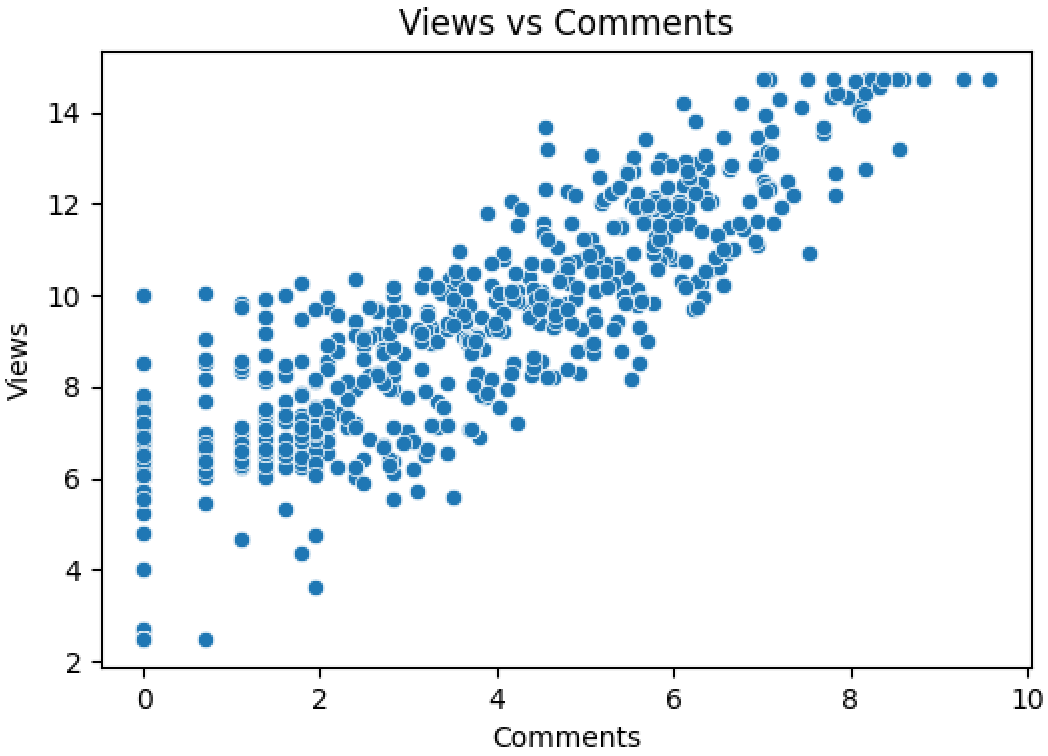
    b. Views vs Subscribers:



c. Views vs Duration:



d. Views vs Comments:



8. Discussion and Conclusions

Project Findings:

- The analysis demonstrates that **API-provided engagement metrics** (likes, comments, subscribers) are significantly more predictive of video views than scraped metadata alone.
- Scraped features such as title length, title word count, first tag length, and TF-IDF vectors capture some information, but their predictive power is limited.
- Linear Regression models trained on API data achieved **$R^2 > 0.9$**, whereas scraped data models achieved only **$R^2 \approx 0.15$**, illustrating the substantial value of structured engagement metrics.
- Feature importance analysis highlights that **user engagement metrics** (likes, comments, subscribers) are the primary drivers of video popularity. Textual metadata contributes minimally but may still offer insights for niche or content-specific predictions.

Challenges Encountered:

- **Scraped data limitations:** Many scraped fields were missing, inconsistent, or required normalization (e.g., "views" in “K” or “M” format, missing comments).
- **Small dataset for scraped data:** Only 266 samples were available, which limits model generalization.
- **Text preprocessing:** Converting titles and tags into TF-IDF vectors required careful handling to avoid empty vocabulary errors.
- **Outliers and skewed data:** View counts and engagement metrics were highly skewed, requiring clipping and log transformation to stabilize model training.

Ethical and Legal Considerations:

- Scraping data from public sources must respect **platform terms of service** and **privacy laws**. No personal user information beyond public engagement metrics was collected.
- API usage provided structured, permissioned access to public data, which is more reliable and compliant.

Recommendations for Improving Model Performance:

- **Increase scraped data volume:** More samples could improve predictive power of models based on metadata.
- **Incorporate additional features:** Including video descriptions, tags, thumbnails, or content categories could improve scraped data models.
- **Use advanced models:** Tree-based models (Random Forest, XGBoost) could capture non-linear relationships and interactions between features.
- **Leverage temporal trends:** Engagement over time (views per day, recent activity) could enhance prediction accuracy.
- **Cross-dataset validation:** Comparing predictions between API and scraped datasets can help identify systematic biases in scraped data.

Conclusion:

The analysis demonstrates that API-provided engagement metrics, such as likes, comments, and subscriber counts, are significantly more predictive of YouTube video views than scraped metadata alone. Models trained on API data achieved much higher performance metrics, with R^2 values exceeding 0.9, compared to R^2 around 0.15 for models trained on scraped data. This indicates that while textual and categorical features extracted from scraped data, such as title length, word count, and tags, provide some insight, they are insufficient to accurately predict video popularity on their own. Feature importance analysis confirms that user engagement metrics are the primary drivers of view counts. Improving model performance could involve incorporating more comprehensive content features, increasing data volume, or using advanced regression techniques. Overall, API-based data offers a more reliable and robust foundation for predictive modeling of YouTube video performance, whereas scraped data is better suited for exploratory analysis or supplementing structured engagement metrics.