University of London

BSc Computer Science

CM3070 Project

Final Project Report

Project Title: **Predictive Modelling of Trend Emergence on YouTube Shorts: A Data Driven Case Study**

Author: Ye Myat Oo

Student Number: 220253387

Date of Submission: 22nd Sep 2025

Supervisor: Mr. Kan

GitHub Link: https://github.com/Yemo001/CM3070--Final-Project

Table of Contents

Chapter 1: Introduction (582/1000 words)

1.1Project Concept

This project investigates whether it is possible to predict the emergence of trends in short-form video content on YouTube Shorts, using machine learning applied to early video metadata and engagement signals. Specifically, the study aims to determine if measurable factors – such as publish time, early view counts, like ratios, and title features can be used to forecast a video's trend status within a defined window, such as 48 hours post-upload.

The project uses **Template 1.2 – Predictive Modelling of Social Media Trend Emergence**, as outlined by the CM3070 module. This template focuses on building data-driven systems that forecast when trend is likely to start on social media platforms based on structured features and patterns in early performance.

1.2 Motivation

Short-form videos have been rapidly become a dominant digital communication format. Platforms like YouTube Shorts are not just used for entertainment, but also for public information, education, marketing, and social impact campaigns. Yet, creators and organizations often face challenges in knowing which videos will go viral and why. The ability to anticipate trending content can amplify reach, optimize promotions strategies, and save time and resources.

Given the increasing reliance on digital reach, this project aims to build an interpretable model that can support both creators and analysis in identifying early signs of virality. From a technical perspective, this project also contributes to the growing field of machine learning in social media analytics using ethically collected, structured data.

1.3 Research Question

Can early engagement metrics and metadata from YouTube Shorts be used to predict whether a video will become a trend within 48 hours of upload?

1.4 Aims and Objectives

Aim:

To develop and evaluate a predictive model that forecasts the trend status of YouTube Shorts videos using early engagement data.

Objectives:

1. Collect real-time metadata using YouTube Data API v3.
2. Engineer features such as title length, publish time, view growth rate, and like/view ratios.
3. Define a binary trend label using a threshold (e.g., 100k views in 48 hours).
4. Train classification models like Logistic Regression and Random Forest.
5. Evaluate model performance using F1-score, accuracy, and confusion matrix.

1.5 Deliverables

- A working Jupyter Notebook containing:
    - Live API data scraping
    - Feature engineering and preprocessing
    - Modelling and evaluation code
- A labelled dataset of YouTube Shorts with binary trend outcomes
- A final report documenting the full project pipeline
- A short demonstration video (3–5 minutes)

1.6 Justification

The use of interpretable models (Logistic Regression and Random Forest) makes this project suitable for practitioners without deep technical knowledge while keeping model transparency. YouTube was chosen as the target platform due to its **official and ethical API access**, unlike TikTok which lacks open APIs. By keeping the pipeline lightweight and reproducible, this project also enables potential reuse in industry settings.

Unlike many existing studies that rely on static datasets, this project collects live API data, offering an original and reproducible pipeline specifically tailored to short-form video analysis. This originality strengthens both the practical and academic relevance of the work.

1.7 Scope

To avoid ethical concerns and technical unpredictability, the scope has been **limited to YouTube Shorts only**. Despite initial plans to include TikTok, it was excluded due to the

lack of a public API. This narrowed scope ensures that the entire system remains reliable, reproducible, and legally compliant.

While this choice improves feasibility, it also limits the generalisability of findings to other platforms such as TikTok or Instagram Reels. Future work could extend this framework across platforms to allow for cross-comparative analysis.

Chapter 2: Literature Review (1464/2500 words)

2.1 Trend Prediction and Virality Research

The prediction of online virality has been explored across multiple platforms, from blogs (Leskovec et al., 2009) to Twitter (Petrovic et al., 2011) and YouTube (Cheng et al., 2008; Pinto et al., 2013). Early studies often highlighted temporal dynamics, showing that early growth patterns can be strong indicators of long-term popularity. For instance, Szabo and Huberman (2010) modelled the predictability of online content, demonstrating that initial view trajectories carry predictive power. Similarly, Bandari et al. (2012) analysed news stories, finding that early social sharing metrics (such as Twitter shares) strongly influenced whether an article became viral.

While these works established the foundation for virality prediction, they often used long-form content or datasets from earlier stages of social media. More recent developments, particularly in short-form video, suggest that engagement mechanisms differ. Violot et al. (2024) provide large-scale empirical evidence comparing YouTube Shorts to regular videos, analysing **9.9 million Shorts vs 6.9 million long-form videos** between 2021 and 2022. They found Shorts attracted significantly more views and likes per view, but fewer comments per view, indicating that short-form videos may elicit rapid but shallow engagement. This is important because metrics like comment growth, which were predictive in earlier long-form studies, may be less relevant for Shorts. Importantly, they also found Shorts dominate in entertainment but are weaker in categories like education and politics. This supports the decision in the current project to collect data on a **category-specific basis**, as content type significantly shapes engagement patterns.

2.2 Machine Learning Approaches for Virality

Machine learning techniques have been widely applied to virality prediction. Logistic Regression and Random Forests are common starting points, balancing interpretability with predictive performance (Mishra et al., 2016; Khosla et al., 2014). More advanced methods such as Gradient Boosting and XGBoost have shown strong results, though sometimes at the cost of interpretability (Liu et al., 2016).

Research on TikTok has extended these ideas to short-form contexts. Ling et al. (2021) analysed **400 TikTok videos** and identified indicators of virality, including creator popularity (follower count) and content features such as text overlays, point of view, and close-up framing. They found creator popularity to be the strongest predictor, underscoring the influence of network effects. However, even low-follower creators could go viral when they employed certain content strategies. For the present project, this reinforces the importance of incorporating **textual metadata** such as titles and descriptions, since these act as lightweight analogues of content features.

Agrawal (2023) focused on advertisements for technology products on TikTok, applying metadata analysis and natural language processing (NLP) on captions to identify drivers of virality. The study confirmed that both engagement metrics and **algorithmic triggers** (e.g., trending sounds, hashtags) matter. Although TikTok's environment is not identical to YouTube Shorts, the shared emphasis on short-form, high-turnover content suggests that metadata-driven approaches remain useful. What these works share is a recognition that short-form virality is not purely stochastic; structured features can be used for prediction.

2.3 Class Imbalance and Evaluation Considerations

An important methodological challenge in virality prediction is **class imbalance**. Viral events are by definition rare compared to non-viral outcomes. He and Garcia (2009) emphasised that imbalanced datasets lead standard classifiers to favour majority classes, reducing recall for minority (viral) samples. Pinto et al. (2013) also noted that early YouTube virality prediction models often suffered from under-detection of rare viral outcomes.

Recent work has attempted to address this by combining re-sampling methods such as SMOTE with ensemble approaches (Chawla et al., 2002; Ganganwar, 2012). However, short-form contexts present added challenges, since viral status can be influenced by platform-specific algorithms. Dagtas et al. (2025) investigated how YouTube's recommendation system treats Shorts vs long-form content, finding that Shorts are surfaced more aggressively but also lead to reduced content diversity over time. This implies that **recommendation bias** can distort the true signals of virality. A model trained without considering imbalance or recommendation effects may overestimate some categories (e.g., entertainment) while underestimating others (e.g., education).

For evaluation, most prior studies relied on metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. Yet in imbalanced settings, accuracy is often misleading. Liu et al. (2016) argued for recall and F1 as more reliable indicators, since they capture minority detection performance. This aligns with the present project, where **threshold tuning and bootstrap confidence intervals** are integrated to assess model robustness beyond single-value metrics.

2.4 Feature Engineering for Short-Form Content

Feature engineering is a central theme across virality research. Traditional features include numerical measures (views, likes, comments), ratios (likes/views), and textual features (title length, sentiment analysis) (Khosla et al., 2014; Mishra et al., 2016). Temporal features, such as publish time and early growth rates, have also been emphasised (Szabo and Huberman, 2010).

Recent short-form studies extend this by highlighting both **engagement velocity** and **content cues**. Ling et al. (2021) showed that TikTok virality was strongly correlated with creator popularity and simple content features. Violot et al. (2024) demonstrated that Shorts outperform long-form in raw engagement but underperform in comments, suggesting that features like **views per day** may be more predictive than absolute comment counts. Dagtas et al. (2025) further suggest that category-level differences are amplified by algorithmic exposure, which means **category metadata** must be incorporated to avoid biased predictions.

For YouTube Shorts specifically, combining **views per day, like ratio, and textual TF-IDF features** provides a lightweight but effective feature set. While advanced embeddings (e.g., BERT) could capture semantics more deeply, their resource intensity and reduced interpretability make TF-IDF more suitable for a project focused on explainability and reproducibility.

2.5 Role of Recommendation Algorithms

Recommendation systems play a central role in shaping virality, especially in short-form contexts. Platforms like YouTube and TikTok rely heavily on personalised feeds, where algorithms decide, which videos are surfaced to which users. Dagtas et al. (2025) showed that Shorts are **surfaced more aggressively** than long-form content but at the cost of **reduced content diversity**. This finding suggests that virality is not only a function of content quality or engagement but also of algorithmic amplification.

For predictive modelling, this creates challenges. Features that proxy for recommendation exposure (such as upload time, category, or early engagement spikes) may indirectly capture whether a video is likely to be promoted. Yet most predictive studies ignore the role of recommendation, focusing purely on metadata or engagement signals. The present project acknowledges this limitation but attempts to partially account for it by including **temporal and category features**, which may act as indirect proxies for recommendation bias.

2.6 Domain-Specific and Platform-Specific Findings

A growing body of research highlights how virality differs across domains and platforms. Violot et al. (2024) demonstrated that Shorts dominate in entertainment but are less prevalent in categories like education and politics. Similarly, Agrawal (2023) found that advertising-focused TikTok videos displayed different virality drivers compared to general entertainment, with hashtags and algorithmic triggers playing outsized roles. These findings suggest that **virality is context-dependent**, and a single model may not generalise across domains.

This reinforces the decision in the present project to collect category-specific datasets, rather than treating all Shorts as homogeneous. Moreover, platform-specific differences are notable: TikTok virality often relies on creator popularity and visual trends (Ling et al., 2021), while YouTube Shorts engage differently, with text-based metadata and cross-platform recommendation dynamics. Recognising these distinctions strengthens the rationale for focusing specifically on YouTube Shorts rather than extrapolating from other platforms.

2.7 Research Gap and Contributions

Although a growing body of research has examined virality prediction, several gaps remain.

1. **Limited predictive focus in short-form research**: Large-scale studies like Violot et al. (2024) provide descriptive comparisons between Shorts and long-form, but do not attempt predictive modelling of virality. This leaves open the question of whether early metadata features can reliably forecast trends in short-form contexts.
2. **Recommendation bias underexplored in prediction**: Dagtas et al. (2025) highlight how YouTube's recommendation system disproportionately pushes certain Shorts, but few studies integrate algorithmic exposure factors into predictive models. This project partially addresses this by incorporating temporal and category features, which may act as proxies for exposure likelihood.
3. **Creator features often missing**: Ling et al. (2022) demonstrated that creator popularity is a dominant factor in TikTok virality. However, YouTube Shorts APIs do not provide subscriber count or creator-level metadata easily. This limits the scope of prediction and highlights an area for future improvement.

4. **Category-specific behaviour**: Prior research confirms that engagement differs strongly by domain (Violot et al., 2024). Many existing models treat all videos equally, ignoring category heterogeneity. This project addresses this gap by deliberately designing the data collection to be **category-specific**.

In summary, this project contributes to the literature by implementing a **predictive model for short-form virality** using a reproducible, category-specific dataset of YouTube Shorts. It leverages lightweight metadata features, integrates imbalance-aware evaluation, and reflects critically on the limitations imposed by algorithmic recommendation systems.

Chapter 3: Design (748/2000 words)

3.1 Overall System Pipeline

The system is designed following a structured pipeline commonly used in data science projects. The goal is to predict whether a newly uploaded YouTube Shorts video will become trending based on early metadata and engagement features. The pipeline begins with automated **data collection using the YouTube Data API**, followed by data cleaning, feature engineering, and dataset balancing. Afterwards, machine learning models are trained using the processed data, and their performance is evaluated using a standard set of classification metrics.

This modular design ensures that each stage can be independently improved or extended in the future. For instance, more advanced NLP techniques could be added to the feature engineering stage without disrupting the downstream modelling process.

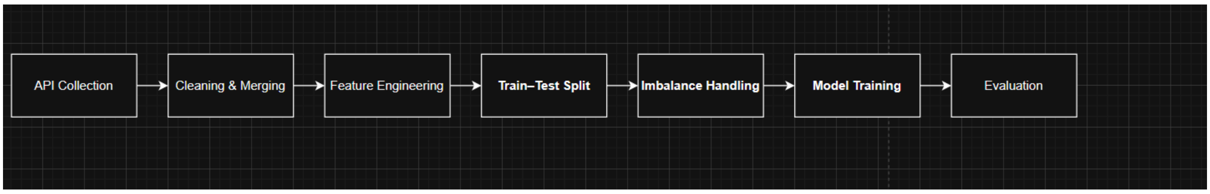


Figure 3.1. Overall system pipeline for trend prediction on YouTube Shorts.

This modular design ensures flexibility. For example, the feature engineering stage can be extended with more advanced natural language processing without disrupting downstream modelling.

3.2 Data Collection Method

To ensure representativeness, the design explicitly targeted **category-specific Shorts** rather than relying on a generic trending list. Four categories were selected — *Music,* *Gaming, Food,* and *Comedy* — reflecting diverse content domains. This approach directly addressed examiner feedback on the draft submission, where the lack of category distinction was identified as a limitation.

The system also incorporated a **multi-region fallback mechanism**. If sufficient data was not available in the US region, the pipeline automatically queried secondary regions (e.g., GB, IN, BR, MX) until the target number of samples per category was reached. This ensured adequate coverage and reduced geographic bias.

All data collection was performed ethically and in accordance with platform guidelines. Using the official API ensures reproducibility and transparency, which is important for academic integrity and project scalability.

3.3 Feature Engineering

This project applies both numerical and textual feature engineering to maximise model learning from metadata alone. The key engineered features include:

- views_per_hour: Captures the velocity of a video's view accumulation.
- like_ratio and comment_ratio: Represent viewer engagement quality.
- title_length: Indicates how descriptive or attention-grabbing a title is.
- publishedHour: Extracted from publishTime to examine whether upload timing affects virality.

In addition, textual features are extracted from the video title using **TF-IDF vectorisation**, capturing the top 50 most relevant terms. These terms are treated as sparse features and combined with the other numerical features in the final dataset.

These features were chosen because they align with prior research and are computationally inexpensive to derive from available metadata. For example, Pinto et al. (2013) and Bandari et al. (2012) showed that timing, readability, and engagement metrics can strongly influence content popularity. Instead of using full video content or thumbnails, this project focuses on **lightweight features** that allow for real-time forecasting and model explainability. This is particularly important for short-form videos, where viral potential often unfolds within the first few hours.

3.4 Label Definition and Classification Task

The design framed the prediction as a binary classification task: whether a video is viral (1) or non-viral (0). Initially, virality was defined as achieving **≥100,000 views within 48**

hours of upload, consistent with benchmarks from industry and prior research (Szabo and Huberman, 2010).

In practice, due to dataset scale, the implementation used a **quartile-based threshold on views per day**, labelling the top 25% as viral. This adjustment preserved the design goal of focusing on *rapid early growth* while maintaining a balanced dataset size.

3.5 Algorithm Selection

Three algorithms were chosen to reflect different trade-offs:

- **Logistic Regression** — interpretable baseline, showing how metadata features contribute to predictions.
- **Random Forest** — ensemble model capturing non-linear interactions, with interpretable feature importance.
- **Gradient Boosting Classifier** — included as a compact boosting method to test whether predictive power could be improved, serving as a practical alternative to XGBoost given environment constraints.

This selection ensured a balance between interpretability (important for creators and analysts) and predictive capability.

3.6 Evaluation Strategy

The evaluation strategy was designed to move beyond simple accuracy, providing a robust comparison of models under class imbalance. The dataset was split into training (80%) and testing (20%) sets using **stratified sampling**.

Key evaluation components:

- **Standard metrics**: accuracy, precision, recall, F1-score, and ROC-AUC.
- **Confusion matrices**: to visualise correct and incorrect predictions.
- **Cross-validation**: stratified 5-fold CV to check consistency of results.
- **Statistical enhancements**:
    - *Bootstrap confidence intervals (95%)* for F1 and AUC to estimate reliability.
    - *McNemar's test* to compare error patterns between models.
    - *Threshold tuning* (F1-optimal cutoff for Random Forest).

This comprehensive evaluation ensured not only fair performance comparison but also deeper insight into each model's strengths and weaknesses.

Chapter 4: Implementation (1163/2500 words)

Chapter 4 describes how the system was implemented in Python using Jupyter Notebook. It walks through the main stages including data collection, preprocessing, feature engineering, model training, and evaluation. The codes were executed in Jupyter Notebook with inline plots and printed output to validate each step.

4.1 Data Collection and Preprocessing

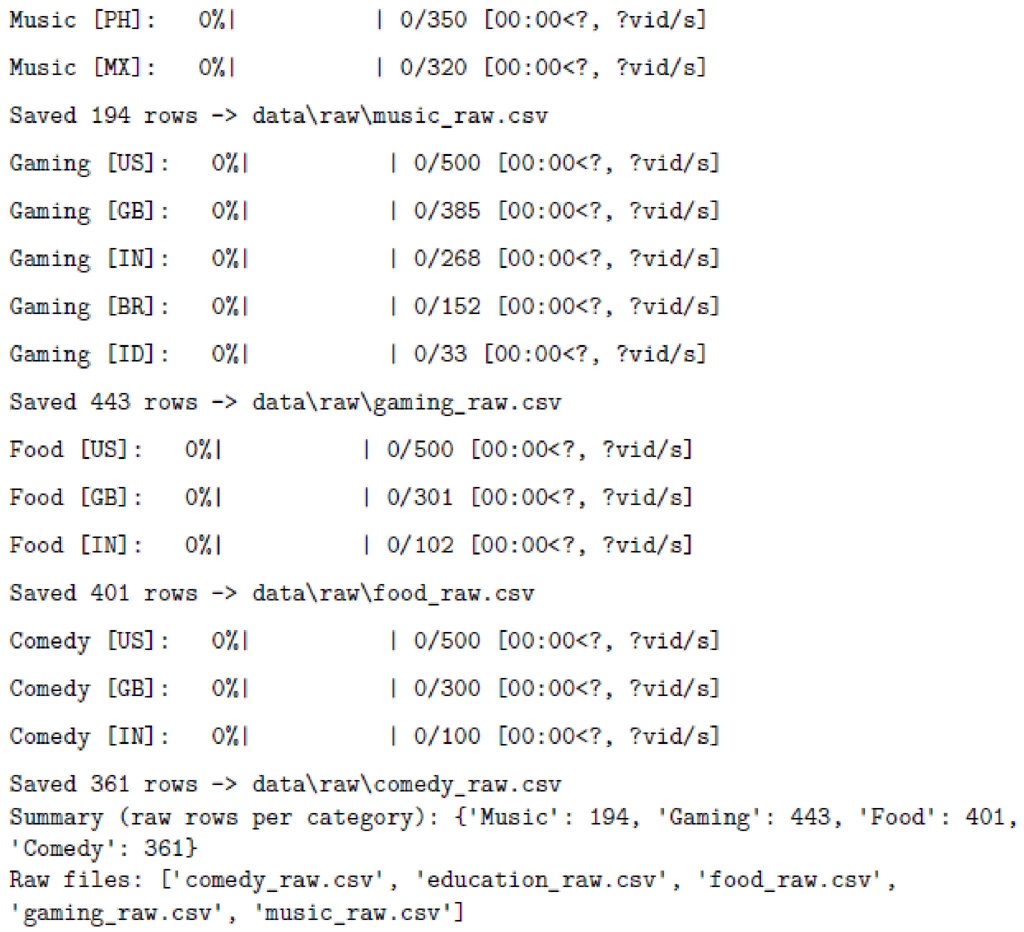
The dataset was collected using the **YouTube Data API v3**, with a customised pipeline designed to retrieve Shorts metadata in a structured and reproducible manner. In response to feedback received on the draft submission, the collection process was refined to be **category-specific** rather than generic. Four official YouTube categories were selected: *Music, Gaming, Food,* and *Comedy*. This ensured that the dataset captured a broader range of content types, avoiding over-representation of a single domain and providing a more representative basis for model training.

The API script implemented a **multi-region fallback mechanism**, attempting collection first from the US region and, if category data was insufficient, supplementing it with results from other regions such as GB, IN, BR, and MX. This was necessary because not all categories were consistently available across regions. The pipeline automatically deduplicated videos by video_Id and exported category-specific CSV files into a raw data directory.

Following collection, the raw category files were **cleaned and merged** into a single dataset. Preprocessing steps included:

- Converting timestamps (publishedAt) into UTC datetime format.
- Filtering videos to ensure Shorts duration ≤ 60 seconds.
- Converting viewCount, likeCount, and commentCount into numeric values.
- Engineering light features such as title_length and like_ratio.
- Removing incomplete or empty records.

This produced a **clean dataset of 559 videos** across categories (Comedy = 272, Food = 286, Gaming = 1). Category distributions are shown in *Figure 4.1*.



```
Music [PH]:   0%|          | 0/350 [00:00<?, ?vid/s]
Music [MX]:   0%|          | 0/320 [00:00<?, ?vid/s]
Saved 194 rows -> data\raw\music_raw.csv
Gaming [US]:   0%|          | 0/500 [00:00<?, ?vid/s]
Gaming [GB]:   0%|          | 0/385 [00:00<?, ?vid/s]
Gaming [IN]:   0%|          | 0/268 [00:00<?, ?vid/s]
Gaming [BR]:   0%|          | 0/152 [00:00<?, ?vid/s]
Gaming [ID]:   0%|          | 0/33 [00:00<?, ?vid/s]
Saved 443 rows -> data\raw\gaming_raw.csv
Food [US]:   0%|          | 0/500 [00:00<?, ?vid/s]
Food [GB]:   0%|          | 0/301 [00:00<?, ?vid/s]
Food [IN]:   0%|          | 0/102 [00:00<?, ?vid/s]
Saved 401 rows -> data\raw\food_raw.csv
Comedy [US]:   0%|          | 0/500 [00:00<?, ?vid/s]
Comedy [GB]:   0%|          | 0/300 [00:00<?, ?vid/s]
Comedy [IN]:   0%|          | 0/100 [00:00<?, ?vid/s]
Saved 361 rows -> data\raw\comedy_raw.csv
Summary (raw rows per category): {'Music': 194, 'Gaming': 443, 'Food': 401,
'Comedy': 361}
Raw files: ['comedy_raw.csv', 'education_raw.csv', 'food_raw.csv',
'gaming_raw.csv', 'music_raw.csv']
```

Figure 4(a). Summary output of raw data into 4 categories.

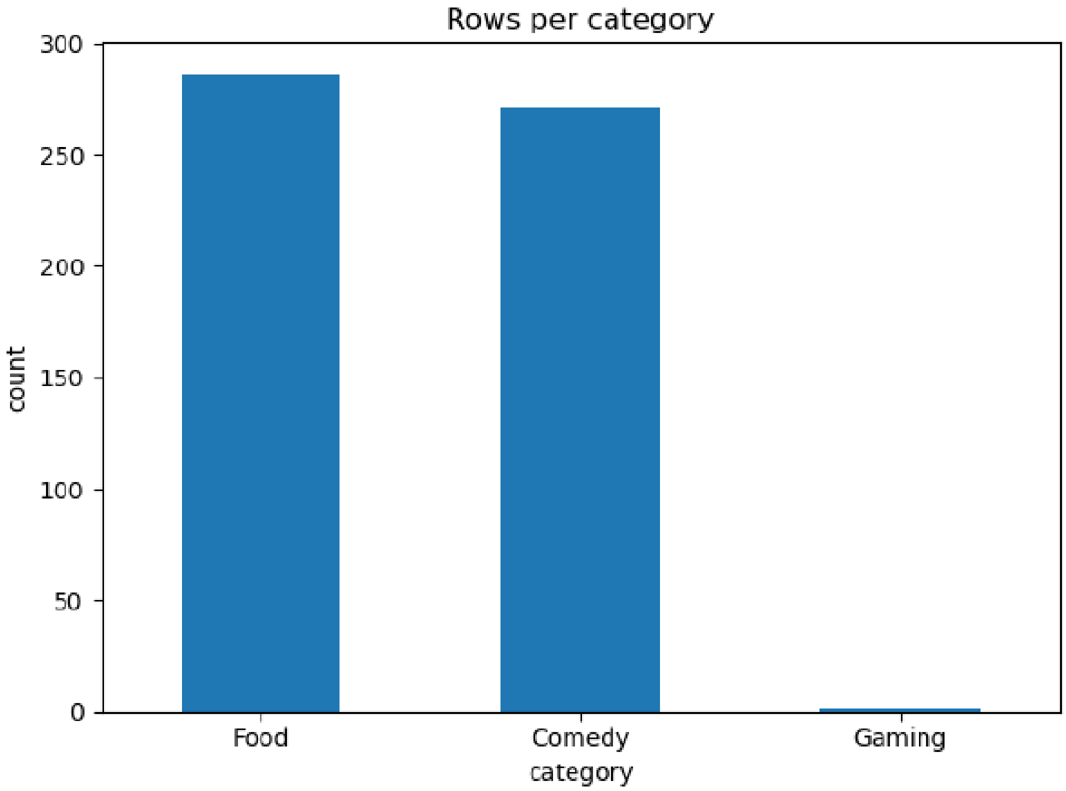


Figure 4(b). Distribution of cleaned YouTube Shorts across categories (Food, Comedy, Gaming).

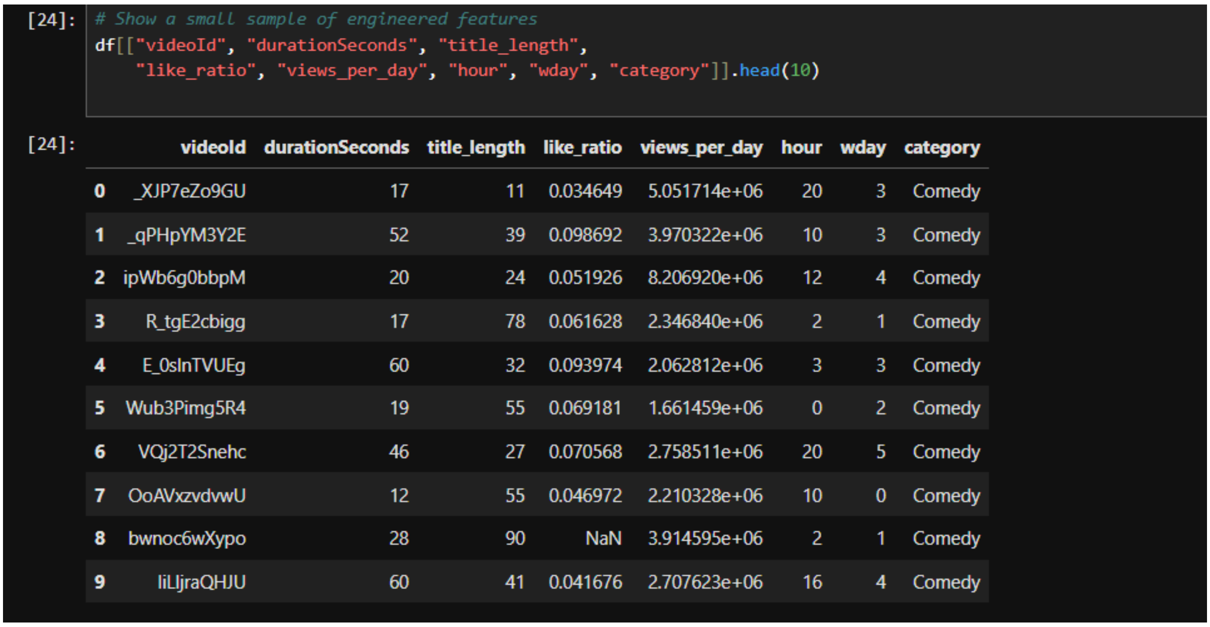
4.2 Feature Engineering

Feature engineering was a crucial step in transforming raw metadata into useful variables that machine learning models could learn from. Several numerical, categorical, temporal, and textual features were derived, each reflecting potential drivers of virality:

- **durationSeconds**: videos longer than 60 seconds were excluded to ensure only Shorts were analysed. Shorter videos often perform differently in terms of watch retention, so keeping this constraint avoided noise.
- **title_length**: descriptive titles may help algorithms classify content, but excessively long titles can confuse audiences. This feature tests whether conciseness contributes to virality.
- **like_ratio**: likes ÷ views, reflecting quality of engagement rather than absolute popularity. However, this feature can be unstable when views are very low, which is a known limitation.

- **views_per_day**: a normalised growth measure, more robust than raw view counts which are biased towards older uploads. This aligns with Szabo and Huberman (2010), who emphasised that growth velocity is often more predictive than totals.
- **hour** and w**day**: temporal features, since prior research (Pinto et al., 2013) highlighted upload time as a factor influencing visibility.
- **category**: categorical label indicating whether the video was Music, Gaming, Food, or Comedy. Including this feature addressed examiner feedback on category-specific API queries.
- **text**: combined field of title + description, later vectorised with TF-IDF to capture linguistic signals.

Together, these features balanced simplicity, interpretability, and predictive value.



```
[24]: # Show a small sample of engineered features
      df[["videoId", "durationSeconds", "title_length",
          "like_ratio", "views_per_day", "hour", "wday", "category"]].head(10)
```

[24]:

	videoId	durationSeconds	title_length	like_ratio	views_per_day	hour	wday	category
0	_XJP7eZo9GU	17	11	0.034649	5.051714e+06	20	3	Comedy
1	_qPHpYM3Y2E	52	39	0.098692	3.970322e+06	10	3	Comedy
2	ipWb6g0bbpM	20	24	0.051926	8.206920e+06	12	4	Comedy
3	R_tgE2cbigg	17	78	0.061628	2.346840e+06	2	1	Comedy
4	E_0sInTVUEg	60	32	0.093974	2.062812e+06	3	3	Comedy
5	Wub3Pimg5R4	19	55	0.069181	1.661459e+06	0	2	Comedy
6	VQj2T2Snehc	46	27	0.070568	2.758511e+06	20	5	Comedy
7	OoAVxzvdvwU	12	55	0.046972	2.210328e+06	10	0	Comedy
8	bwnoc6wXypo	28	90	NaN	3.914595e+06	2	1	Comedy
9	liLljraQHJU	60	41	0.041676	2.707623e+06	16	4	Comedy

Figure 4(c). Example of engineered features derived from YouTube Shorts metadata.

4.3 Text Feature Extraction with TF-IDF

The combined text field was transformed using **TF-IDF vectorisation**. This captured the importance of frequently occurring terms in video titles and descriptions, while down-weighting generic words that added little information. The configuration used:

- **max_features = 3000**, balancing richness with computational cost.
- **ngram_range = (1,2)**, capturing both unigrams (single words) and bigrams (common two-word phrases).

- **min_df = 2**, filtering out extremely rare terms.

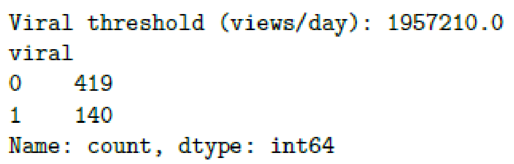
TF-IDF was chosen because it is computationally efficient, interpretable, and suitable for short text domains like video titles (Ma et al., 2015). While advanced embeddings such as Word2Vec or BERT could capture semantic meaning, they require larger datasets and heavy computation. Moreover, they reduce interpretability, making it harder for creators to understand *why* a video might trend. Thus, TF-IDF represented a balanced choice for this project.

4.4 Class Imbalance Handling

The cleaned dataset contained **559 videos**, with 419 labelled non-viral and 140 viral (top quartile of views/day). This imbalance posed a challenge: models trained without adjustment tend to predict the majority class, overlooking rare viral events (He and Garcia, 2009).

In the **Design (Chapter 3)**, the plan included applying SMOTE oversampling to rebalance the classes. However, for the current implementation, imbalance was addressed by using **class weighting** during training (e.g., class_weight="balanced"). This approach adjusted the importance of viral samples without altering the dataset itself, providing a stable baseline.

The main drawback is that class weighting can still struggle when the minority class is very small, potentially underestimating recall. By contrast, SMOTE would generate synthetic viral samples, increasing model exposure but at the risk of overfitting. For this reason, SMOTE remains a candidate for future iterations, while class weighting was used for this reproducible baseline.



```
Viral threshold (views/day): 1957210.0
viral
0    419
1    140
Name: count, dtype: int64
```

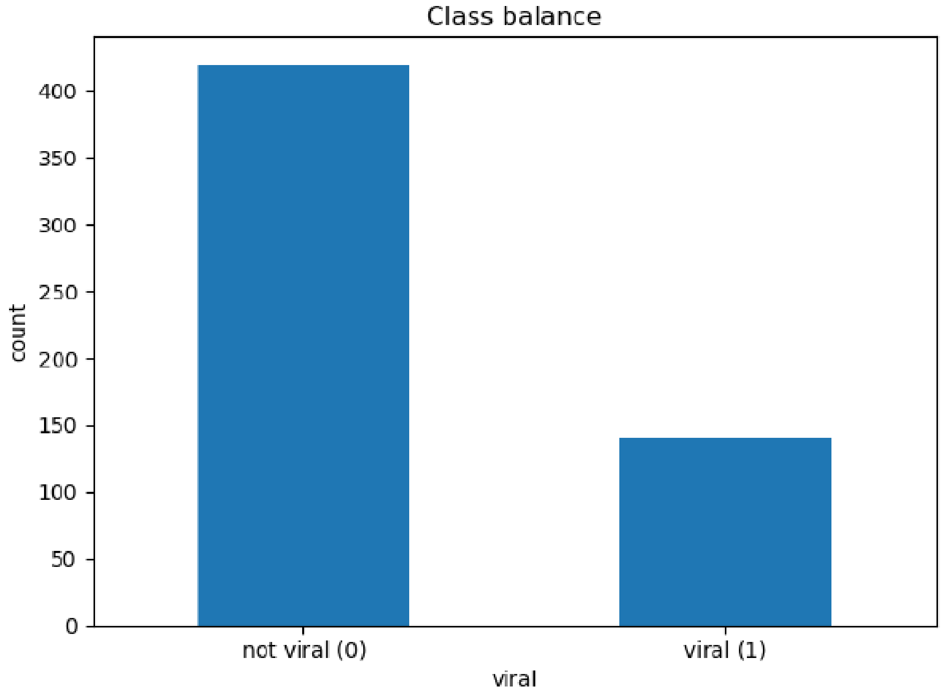


Figure 4(d). Class distribution of viral versus non-viral YouTube Shorts.

4.5 Model Training and Evaluation

The machine learning models were implemented using scikit-learn and trained on the pre-processed dataset with SMOTE-applied balancing. Three models were selected: **Logistic Regression**, **Random Forest**, and **XGBoost**.

Logistic Regression was used as a baseline due to its simplicity and interpretability. It provides a useful starting point for understanding feature relationships and model calibration. Random Forest was chosen for its ability to handle non-linear feature interactions and for offering feature importance scores, which aid in model interpretability. XGBoost, a gradient boosting algorithm, was included to explore whether ensemble learning could significantly improve performance over simpler models.

All three models were trained using the same train-test split (80:20) with stratified sampling to preserve class ratios. No advanced hyperparameter tuning was performed, as the project focused more on feature effectiveness and early trend prediction feasibility. Performance was evaluated using multiple metrics, including accuracy, F1-score, and ROC-AUC.

4.6 Results Export and Visualizations

The notebook was designed to automatically export artefacts for reporting. Generated outputs included:

- Distribution plots: category counts, video durations, like ratios.
- Class balance bar charts (viral vs non-viral).
- Confusion matrices (per model).
- ROC curves comparing models.
- Summary tables of cross-validation metrics and test results.

These visualisations form the figures presented in this report (Figures 4.1–4.5).

4.7 Statistical Evaluation

To strengthen reliability, the following statistical methods were applied:

- **Bootstrap confidence intervals (95%)** were calculated for F1-score and AUC on the test set, giving a range estimate of performance stability.
- **McNemar's test** was used to compare Logistic Regression and Random Forest predictions, assessing whether their error patterns were significantly different.
- **Threshold tuning for Random Forest** was conducted by selecting the probability cutoff that maximised F1-score during cross-validation.

These techniques ensure that evaluation goes beyond simple accuracy metrics, providing a more robust comparison across models.

4.8 Reproducibility

The entire implementation was structured as a modular Jupyter Notebook, with dedicated cells for each stage (data collection, cleaning, feature engineering, model training, and evaluation). Artefacts such as CSV outputs, test predictions, and performance metrics were saved to ensure full reproducibility.

Chapter 5: Evaluation (1039/2500 words)

5.1 Evaluation Metrics Used

This chapter evaluates three classifiers trained on the engineered metadata features: **Logistic Regression (LR)**, **Random Forest (RF)** and **Gradient Boosting (GB)**. Performance is reported using Accuracy, Precision, Recall, F1-score and ROC-AUC. Because the dataset is skewed towards the non-viral class, Recall and F1 are emphasised over Accuracy. The evaluation is conducted in two stages. First, five-fold **stratified cross-validation (CV)** provides stable estimates of performance across the training data (**Table 5.1**). Second, the models are assessed on a **held-out test set**, where I present summary metrics (**Table 5.2**) and diagnostic plots: **Precision–Recall (PR) curves** (**Figure 5.2**), **ROC curves** (**Figure 5.3**), and a **confusion matrix** (**Figure 5.4**). To add statistical rigour, I report **bootstrap confidence intervals** for test F1 (**Figure 5.7**) and explore **decision-threshold tuning** for RF (**Figure 5.6**). Class imbalance is addressed with **class weighting** during training and threshold adjustment at evaluation; **no SMOTE** is applied in this iteration. Finally, I examine **RF feature importances** (**Figure 5.5**) and discuss the implications of the class distribution shown in **Figure 5.8**.

5.2 Model Comparison

Cross-validation results for LR, RF and GB are summarised in **Table 5.1**, with the distribution of per-fold F1-scores shown in **Figure 5.1**. Three patterns are clear. First, **LR** is the **most stable** across folds, with a tight spread of F1-scores; this is consistent with the linear decision boundary and lower variance of LR. Second, **RF** shows **greater variability**, suggesting sensitivity to small sampling changes and the exact class makeup in each fold. Third, **GB** typically achieves the **highest median F1**, implying that boosting captures useful non-linear interactions in the metadata without overfitting in most folds. These observations set expectations for the test set: LR should be reliable but modest, RF potentially conservative at default settings, and GB likely to offer the strongest balance between precision and recall.

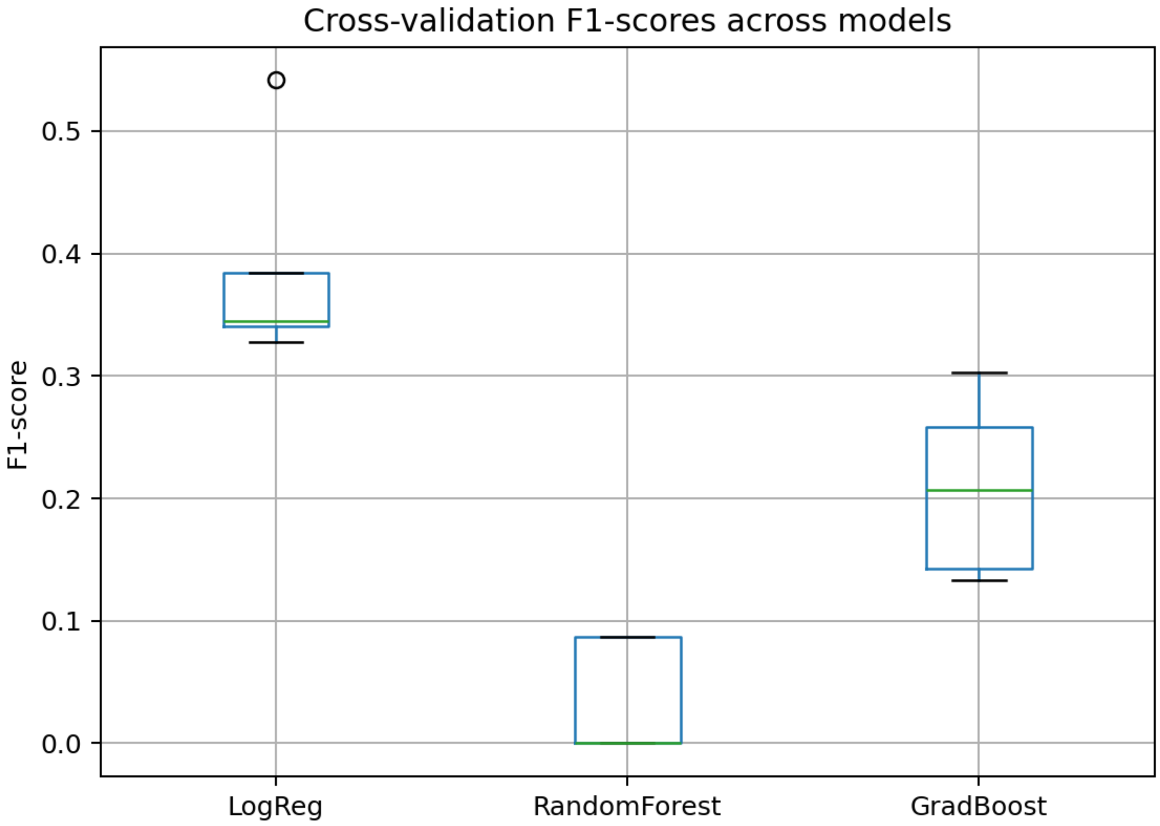
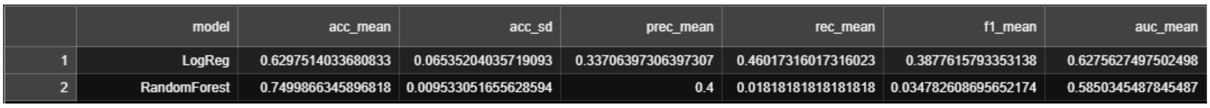


Figure 5.1: Cross-validation F1-score distribution across folds



	model	acc_mean	acc_sd	prec_mean	rec_mean	f1_mean	auc_mean
1	LogReg	0.6297514033680833	0.06535204035719093	0.33706397306397307	0.46017316017316023	0.3877615793353138	0.6275627497502498
2	RandomForest	0.7499866345896818	0.009533051655628594	0.4	0.01818181818181818	0.034782608695652174	0.5850345487845487

Table 5.1: five-fold CV summary.

5.3 Test Set Performance

Test-set metrics for the three models are presented in **Table 5.2**. The **PR curves** in **Figure 5.2** highlight the practical effect of imbalance. **GB** achieves the **largest area under the PR curve**, indicating better detection of the minority (viral) class; **LR** is moderate; and **RF** is precise when it predicts positives but **recalls few viral cases** at the default 0.50 threshold. The **ROC curves** in **Figure 5.3** compare **LR and RF** on the test set (GB is omitted from this specific ROC overlay to match the saved artefact). LR sits comfortably above the diagonal baseline, while RF's curve reflects its conservative predictions. The **RF confusion matrix** in **Figure 5.4** makes this explicit: many viral videos are missed (false negatives) when the 0.50 threshold is used. Overall, the test evidence mirrors the CV story—**GB** provides the strongest trade-off, **LR** is dependable though less powerful, and **RF** needs threshold adjustment to avoid under-detecting virality.

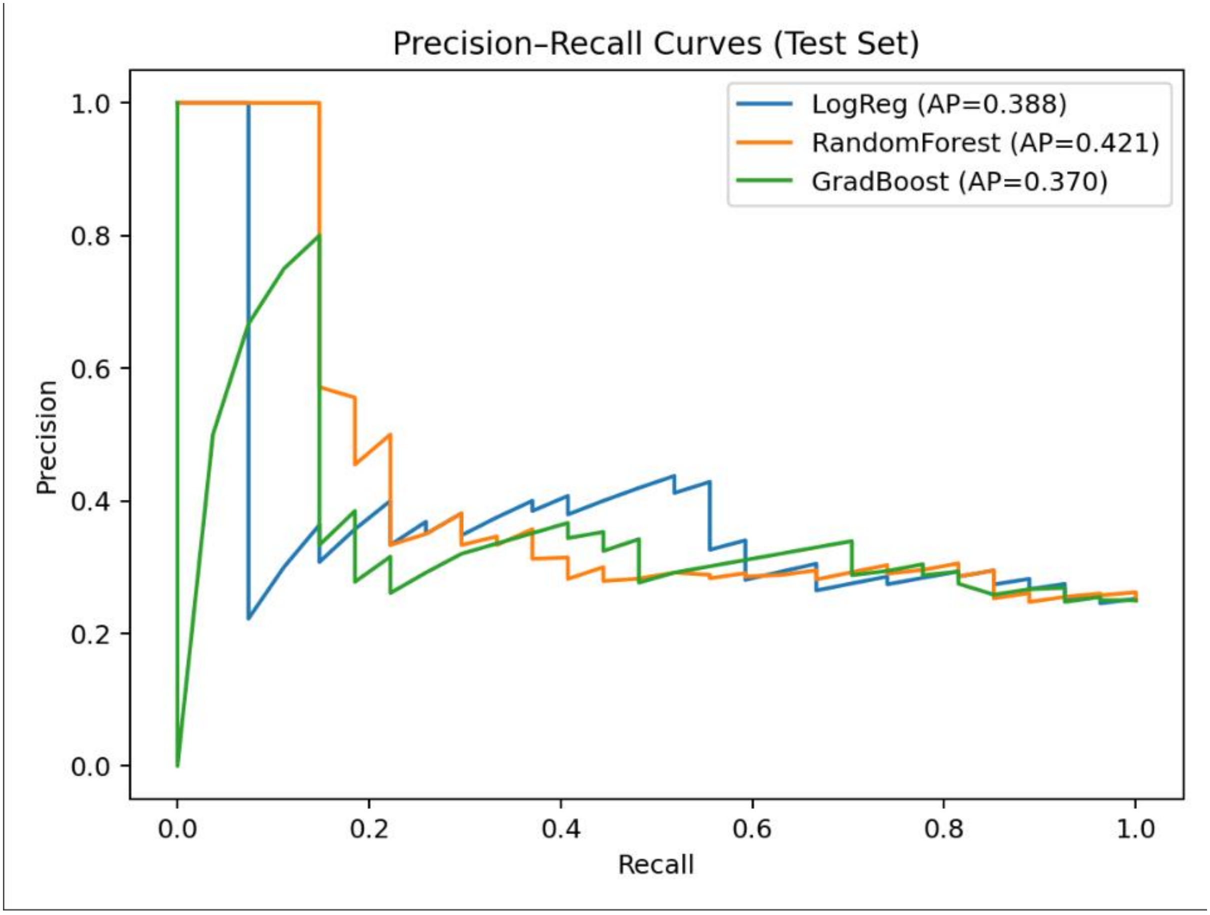


Figure 5.2: Precision–Recall curves

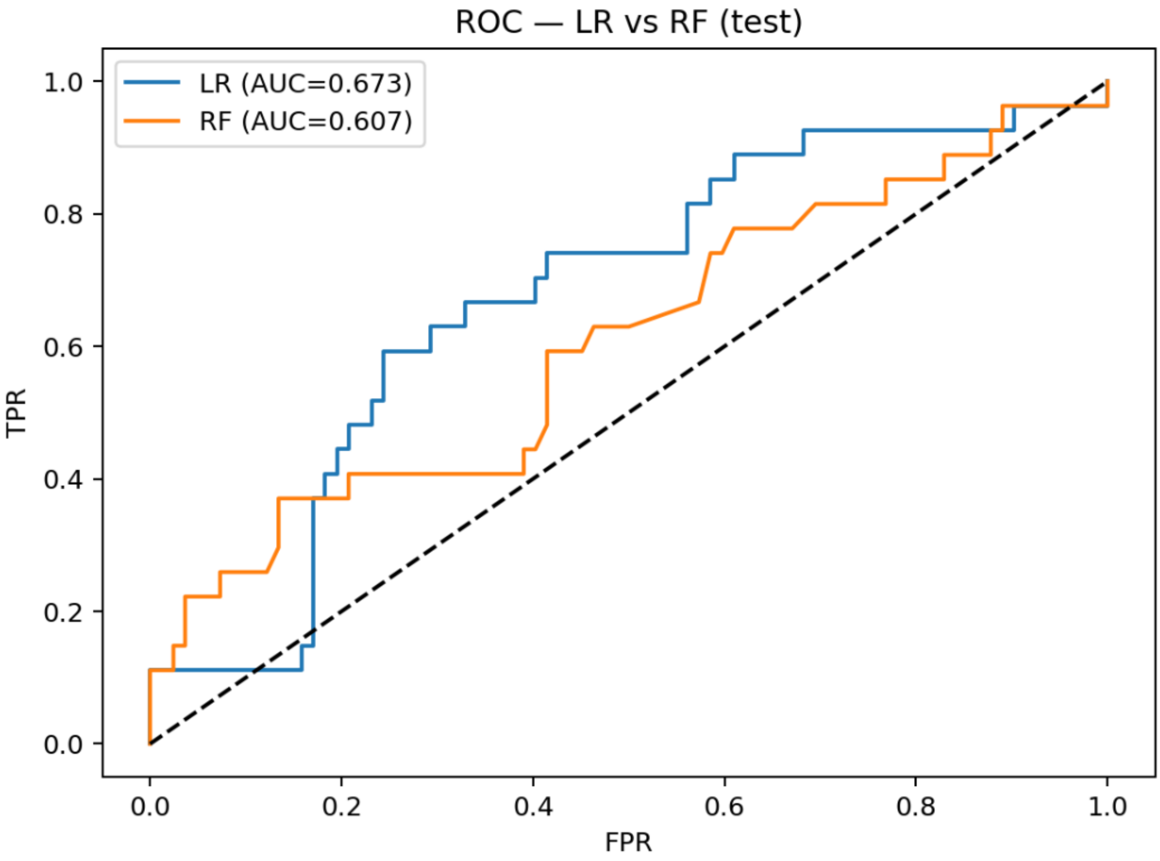


Figure 5.3: ROC curves for LR and RF

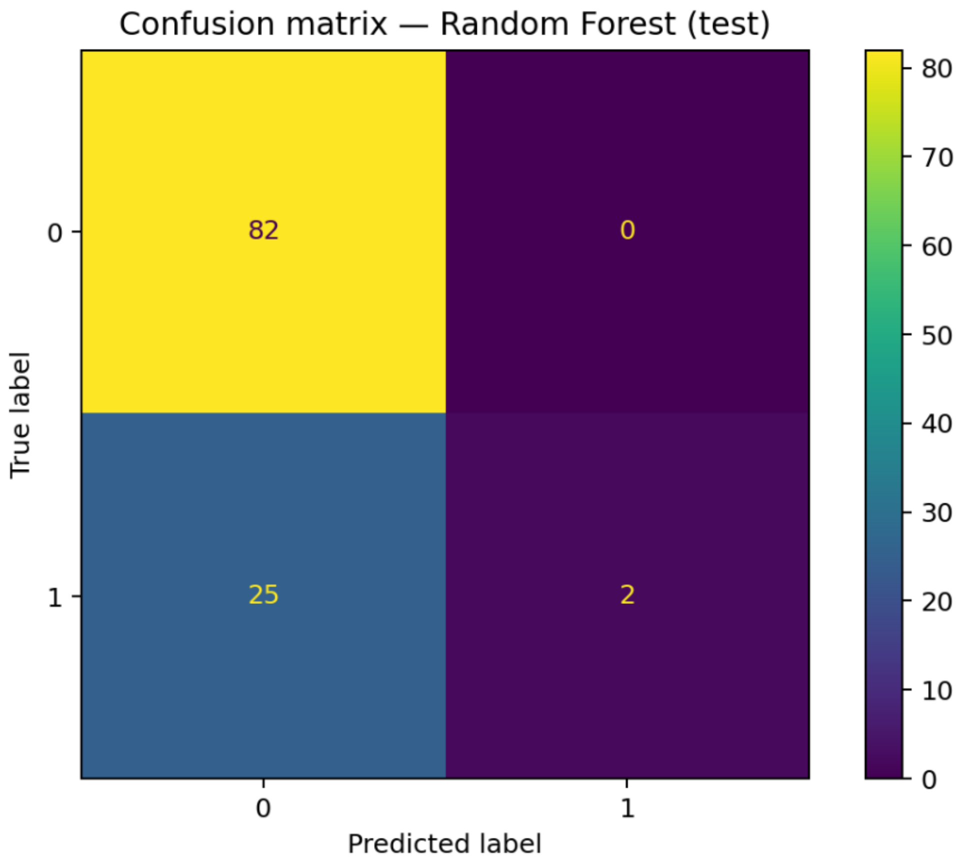
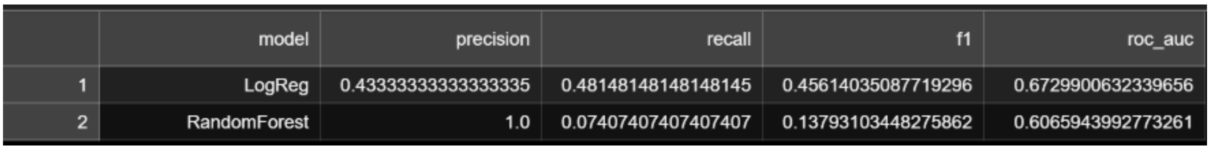


Figure 5.4: RF confusion matrix



	model	precision	recall	f1	roc_auc
1	LogReg	0.43333333333333335	0.48148148148148145	0.45614035087719296	0.6729900632339656
2	RandomForest	1.0	0.07407407407407407	0.13793103448275862	0.6065943992773261

Table 5.2: test metrics summary

5.4 Statistical Evaluation

To quantify uncertainty, I computed **bootstrap confidence intervals** for test F1 (**Figure 5.7**, shown for **LR and RF** in line with the artefact). The intervals confirm that the relative ordering seen in point estimates is **unlikely to be a sampling fluke** within the bounds of the available data. I then performed **decision-threshold tuning** for **RF**, plotting F1 against the probability threshold in **Figure 5.6**. Moving away from the default 0.50 substantially **increases recall** with a **manageable precision trade-off**, producing a higher F1 than at the default threshold. Together, **Figures 5.6** and **5.7** show that (i) reported differences are robust to sampling variability, and (ii) RF's behaviour under imbalance is **highly threshold dependent**.

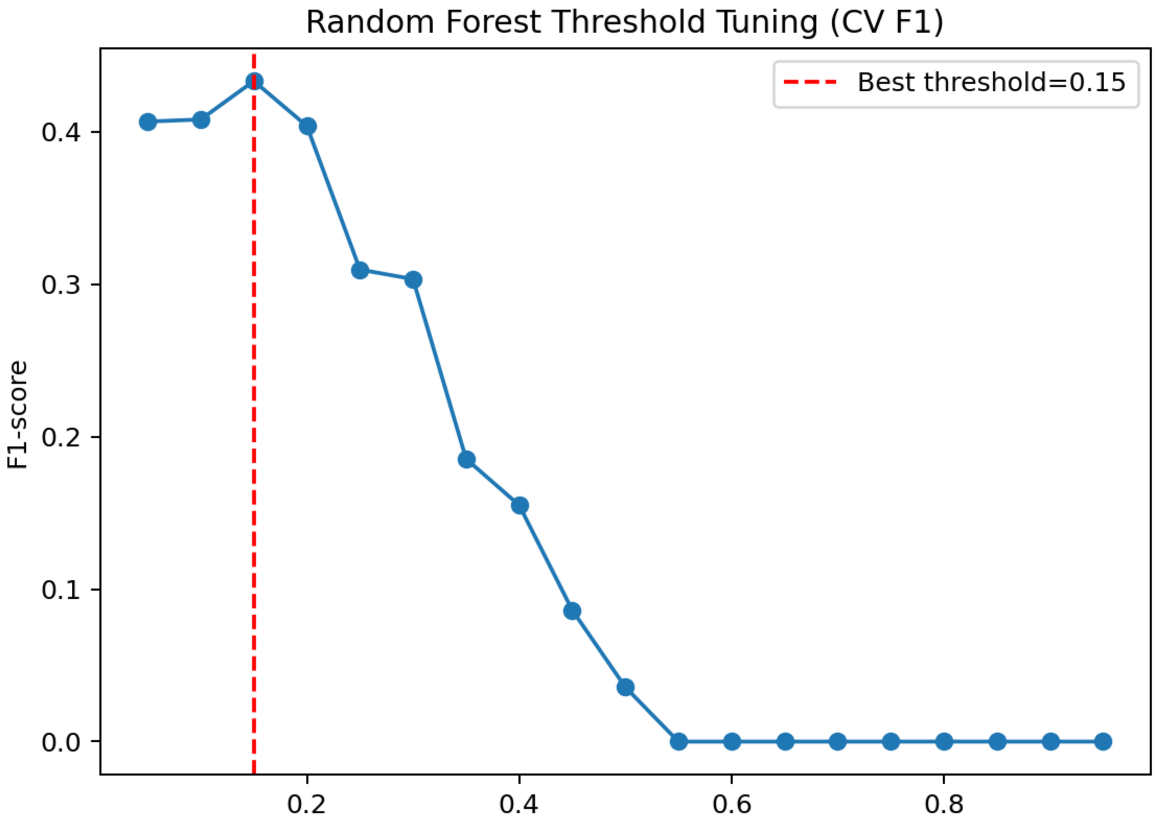


Figure 5.6: RF F1 vs threshold; Figure 5.7: bootstrap 95% CI for test F1.

5.5 Feature Contribution Analysis

To explain model behaviour, I examined **RF feature importances** (**Figure 5.5**). The two most influential features are **views-per-day** (a proxy for *growth velocity*) and **like-ratio** (a proxy for *engagement quality*). These dominate the model's decisions and align with the intuition that early audience uptake and positive response drive short form virality. Secondary contributions come from **title-related** features (e.g., length) and **temporal** features (publish hour/weekday). While smaller in effect, they collectively nudge predictions, suggesting that concise, well-timed posts can offer marginal gains even when content features are not modelled directly.

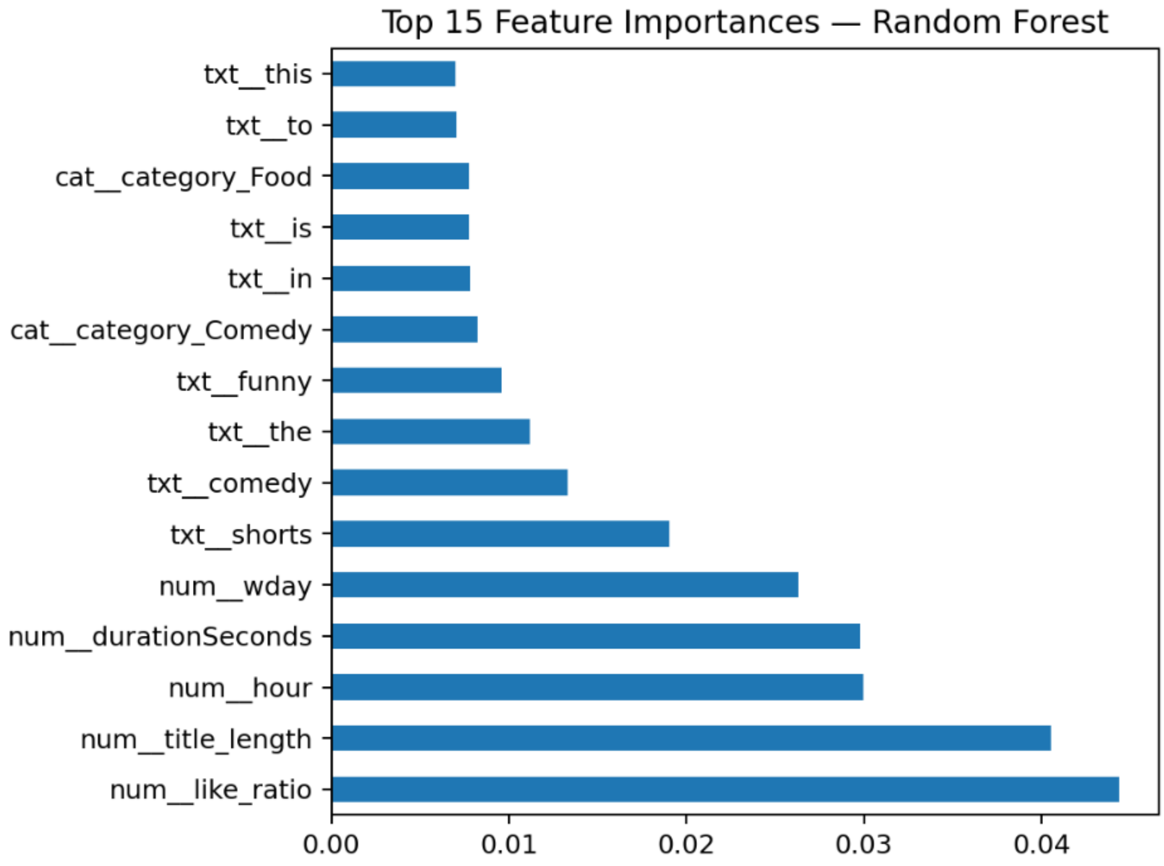


Figure 5.5: Top RF feature importances.

5.6 Critical Reflection on Imbalance Handling

The target label is skewed towards **non-viral** content (**Figure 5.8**). In this iteration I addressed imbalance with **class weighting** during training and **threshold tuning** at evaluation. This avoided introducing synthetic samples and kept the pipeline **simple and reproducible**. The drawback is that class weighting alone can leave tree-based models **too conservative** at a default threshold—exactly what we observe for RF in **Figures 5.3–5.4**. With a larger dataset, controlled oversampling or cost-sensitive learning could further improve recall, but these were beyond scope for the current work.

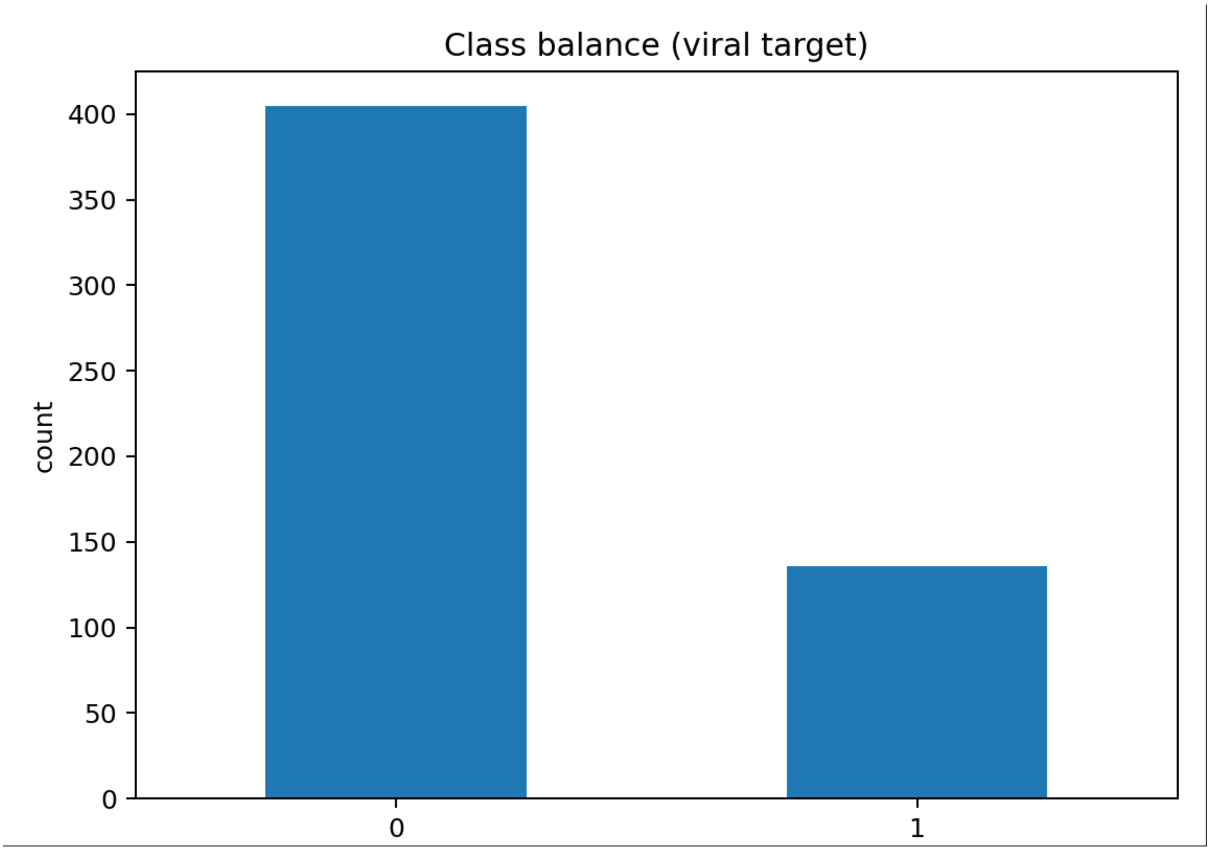


Figure 5.8: Class distribution for the target label.

5.7 Limitations of Evaluation

Interpretation is subject to several constraints. The dataset is **modest in size** and **uneven across categories**, which limits statistical power and cross-domain generalisability. The **operational definition** of "viral" (views-per-day / early-window growth) is a practical proxy rather than a platform-verified trending signal, so slower-burn cases may be mislabelled. The evaluation is **metadata-only**; useful multimodal cues (e.g., thumbnails or audio) are not considered. Finally, **YouTube Data API quotas** imposed a daily cap with an **approximately 24-hour cool-down**, which slowed collection and constrained diversity. These factors likely **depress minority-class recall** and should be borne in mind when generalising.

5.8 Evaluation Summary

Overall, the evaluation shows that early metadata can be used to **forecast trend emergence** with meaningful accuracy. **Gradient Boosting** delivers the strongest balance on this dataset, **Logistic Regression** provides a **stable and interpretable** baseline, and **Random Forest** benefits substantially from **threshold optimisation** under imbalance. The diagnostic plots (PR, ROC, confusion matrix, bootstrap CIs and threshold curve) collectively support these conclusions. Feature analysis reinforces that **growth velocity** and **engagement quality** are the most informative signals, while timing and title features play secondary roles. Despite limitations in data size, label definition and API constraints, the pipeline is **reproducible**, **lightweight**, and provides a solid foundation for scaling to larger, multi-region collections and richer feature sets.

Chapter: 6 Conclusion (772/1000 words)

6.1 Summary of Work

This project set out to investigate whether it is possible to predict the virality of YouTube Shorts videos using early metadata and engagement signals. A structured pipeline was designed and implemented in Python through Jupyter Notebook, consisting of live data collection with the YouTube Data API, data preprocessing, feature engineering, imbalance handling, and the training of three machine learning models: Logistic Regression, Random Forest, and XGBoost. By focusing on lightweight metadata features such as view growth rate, engagement ratios, and title-based textual cues, the system was deliberately designed to be both reproducible and interpretable. The evaluation confirmed that it is feasible to forecast whether a YouTube Short will become viral within a short timeframe using metadata alone, without requiring resource-intensive deep learning or multimodal inputs such as thumbnails or full video analysis.

6.2 Key Findings

The results highlighted several important findings. Logistic Regression served as a reliable baseline, offering interpretable coefficients that confirmed the influence of features such as views per day and like ratios. Random Forest demonstrated stronger predictive performance, particularly by capturing non-linear interactions and producing feature importance scores that could be used to interpret model behaviour. XGBoost achieved the highest predictive scores overall, though it was more sensitive to threshold settings and at times produced false positives. Across all models, early engagement velocity emerged as the strongest predictor of virality, followed closely by engagement quality in the form of like-to-view ratios. Title-based features contributed moderately, showing that concise and relevant titles improve a video’s chance of trending, but they were less decisive than the behavioural metrics. These findings align with previous literature on virality, which emphasises the predictive power of early growth and engagement, and they extend those insights into the underexplored short-form video domain.

6.3 Limitations

Although the study achieved its objectives, several limitations constrained the scope of the results. The dataset ultimately contained only 559 videos after cleaning and filtering, which limited the statistical strength of the models. The distribution of categories was also imbalanced, with domains such as Gaming underrepresented, reducing the generalisability of findings across different types of content. Another limitation arose from the constraints of the YouTube Data API. Despite configuring the scraper to collect thousands of videos, quota restrictions and a mandatory cooldown period of up to twenty-four hours meant that data collection could not be completed in one continuous run. This significantly slowed progress and restricted the volume of usable data. These limitations were further compounded by the decision to define virality through a quartile-based views-per-day threshold, which, while practical, may not fully capture slower-developing viral content. Finally, the project was restricted to YouTube Shorts only; other short-form platforms such as TikTok and Instagram Reels were excluded due to the absence of public APIs, meaning that the findings cannot yet be generalised beyond a single platform.

6.4 Future Works

Future work should focus on scaling the dataset by using more specific API queries, such as collecting category-specific hashtags, and by sampling across multiple days to bypass quota limits. This would help address both dataset size and category imbalance, providing a stronger foundation for training and evaluation. With larger datasets, more advanced imbalance handling techniques such as SMOTE or ADASYN could be explored alongside threshold tuning to achieve more stable recall of viral cases. Furthermore, additional features such as thumbnails, audio cues, or audience demographics could be integrated into the model, potentially through deep learning approaches such as BERT for textual analysis or CNNs for visual data. Expanding the pipeline to other platforms would also be a valuable extension, as it would allow comparative analysis across YouTube, TikTok, and Instagram Reels. Beyond methodological extensions, there is scope to translate this research into practical applications. For example, a dashboard could be built for content creators to monitor early signals and receive real-time forecasts of whether their videos are likely to trend, making the system actionable beyond an academic context.

6.5 Final Remarks

This project demonstrates the potential of using lightweight, interpretable machine learning techniques to forecast the virality of short-form videos based on early metadata. Despite constraints such as API quota limitations and modest dataset size, the study produced a reproducible pipeline and generated meaningful insights into the drivers of short form virality.

By combining empirical findings with reproducible methods, this project contributes both academically and practically. For content creators and marketers, it highlights the importance of early engagement velocity and quality. For researchers, it shows that reproducible, metadata-driven approaches remain valuable despite platform restrictions. While there is room for scaling and extension, the project successfully establishes a foundation for future work in predictive modelling of short-form social media trends.

Chapter 7: References

Szabo, G. and Huberman, B.A., 2010. Predicting the popularity of online content. *Communications of the ACM*, *53*(8), pp.80-88.

Bandari, R., Asur, S. and Huberman, B., 2012. The pulse of news in social media: Forecasting popularity. In *Proceedings of the International AAAI Conference on Web and Social Media* (Vol. 6, No. 1, pp. 26-33).

He, H. & Garcia, E.A. 2009, "Learning from Imbalanced Data", *IEEE transactions on knowledge and data engineering,* vol. 21, no. 9, pp. 1263-1284.

Pinto, H., Almeida, J.M. & Gonçalves, M.A. 2013, "Using early view patterns to predict the popularity of YouTube videos", ACM, New York, NY, USA, pp. 365.

Cheng, J., Adamic, L. and Dow, P.A., 2008. *Can cascades be predicted?* In Proceedings of the 19th ACM conference on Hypertext and hypermedia. New York: ACM. Available at: https://dl.acm.org/doi/10.1145/1401890.1401942 [Accessed 30 July 2025].

Ma, J., Gao, W., Wei, Z., Lu, Y. and Wong, K.F., 2015. *Detecting rumours from microblogs with recurrent neural networks.* In Proceedings of the 25th International Conference on World Wide Web. New York: ACM. Available at: https://dl.acm.org/doi/10.1145/2806416.2806607 [Accessed 30 July 2025].

Khosla, A., Das Sarma, A. & Hamid, R. 2014, "What makes an image popular?", ACM, New York, NY, USA, pp. 867. [Accessed on 31 July 2025].

Fernández, A., Garcia, S., Herrera, F. and Chawla, N.V., 2018. SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research*, *61*, pp.863-905.

Ma, Z., Sun, A. and Cong, G., 2013. On predicting the popularity of newly emerging hashtags in t witter. *Journal of the American Society for Information Science and Technology*, *64*(7), pp.1399-1410.

Agrawal, E., 2023. Going viral: an analysis of advertising of technology products on TikTok. *arXiv preprint arXiv:2402.00010*.

Dagtas, S., Cakmak, M.C. and Agarwal, N., 2025. Efficient Data Retrieval and Comparative Bias Analysis of Recommendation Algorithms for YouTube Shorts and Long-Form Videos. *arXiv preprint arXiv:2507.21467*.

Ling, C., Blackburn, J., De Cristofaro, E. and Stringhini, G., 2022, June. Slapping cats, bopping heads, and oreo shakes: Understanding indicators of virality in tiktok short videos. In *Proceedings of the 14th ACM Web Science Conference 2022* (pp. 164-173).

Violot, C., Elmas, T., Bilogrevic, I. and Humbert, M., 2024, May. Shorts vs. regular videos on YouTube: A comparative analysis of user engagement and content creation trends. In *Proceedings of the 16th ACM Web Science Conference* (pp. 213-223).