University of London

BSc Computer Science

CM3070 Project

Preliminary 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: 16 June 2025

Supervisor: Mr. Kan

Table of Contents

Chapter 1: Introduction

1.1 Project Concept

This project investigates the feasibility of building a data-driven machine learning model to predict trend emergence in short-form video content, using YouTube Shorts as the central case study. The core objective is to determine whether early indicators such as view count, publish time, and engagement metrics can be used to predict whether a video will become a trend within a specific timeframe (e.g., 24–48 hours after publication).

This aligns with **Project Template 1.2 – Predictive Modelling of Social Media Trend Emergence**, which focuses on forecasting when a trend will start based on data patterns observed in early video performance.

1.2 Motivation

Short-form videos have become a dominant mode of digital expression and outreach, especially on platforms like YouTube Shorts. These videos are increasingly used in public health, education, and social campaigns to reach large audiences quickly. However, creators and organisations often struggle to predict which videos will trend. This project aims to address that gap using machine learning techniques, enabling timely content amplification and more efficient targeting of outreach messages.

The ability to forecast trend emergence supports not only creators, but also NGOs and educational institutions that rely on visibility and reach to make an impact.

1.3 Research Question

Can early engagement signals 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 trend emergence in YouTube Shorts videos based on early engagement metrics and metadata.

Objectives:

1. Collect video metadata and public engagement data from YouTube Shorts using the YouTube Data API.
2. Extract structured features from titles, publish times, view counts, and comment/like activity.
3. Define a trend label using a performance threshold (e.g., 100,000 views in under 48 hours).
4. Train classification models (e.g., Logistic Regression, Random Forest) using the processed dataset.
5. Evaluate model performance using F1-score, accuracy, and confusion matrix to assess predictive power.

1.5 Deliverables

- A working Jupyter Notebook containing:
    - API-based data collection
    - Feature engineering and preprocessing
    - Trend prediction modelling
    - Model evaluation and visualisation
- A labelled dataset of YouTube Shorts with early engagement metrics
- A short (3–5 minute) prototype demonstration video
- A full report containing literature, design, and evaluation

---

1.6 Justification

This project offers a technical and ethical approach to trend prediction using only public metadata. It focuses on predictive modelling — in line with the project idea template — and not just descriptive analysis. Using YouTube's official API ensures compliance with platform terms, while the case study format allows an in-depth application of classification techniques to real-world data.

Feedback from project proposal highlighted the need for clear predictive focus, and this version directly responds by demonstrating how the system aims to forecast future trend emergence using early signals, not just analyse historical popularity.

1.7 Scope Adjustment

Originally, this project proposed to include both TikTok and YouTube Shorts. However, TikTok does not offer a public API, and scraping data from the platform would introduce ethical and legal concerns. After discussing with the supervisor, the scope was refined to focus solely on YouTube Shorts, which offers reliable and officially supported access to metadata through the YouTube Data API v3. This allows for clean, consistent data collection, enabling the core prediction objective to be implemented without ethical issues.

Chapter 2: Literature Review

2.1 Overview

This section will review the previous studies and methods relevant to trend prediction on video-sharing platforms, focusing on YouTube Shorts. It explores similar projects, machine learning approaches, the role of metadata, and platform constraints.

2.2 Related Works on Trend Prediction

Several studies have explored the prediction of video popularity using early engagement data. Niture (2021) developed a machine learning model using view count, likes, and metadata to forecast YouTube video trends, achieving an accuracy of 62.5% using Random Forest and Logistic Regression. Their approach followed a standard supervised classification pipeline with manual feature extraction from video metadata. Similarly, Cho et al. (2024) introduced AMPS, a multi-modal model for short-form video popularity prediction across marketing environments, incorporating audio, visual, and text features. Their study showed that multi-modal attention mechanisms can outperform simpler models, but such techniques often require complex preprocessing and computing resources.

Other researchers have approached trend prediction from different angles. For example, Szabó and Huberman (2008) used early view trajectories to predict long-term popularity, proposing that a video's early success is highly indicative of future performance. Their findings are especially relevant when considering the critical time window (e.g., first 24–48 hours) for trend forecasting. Pinto et al. (2013) also reinforced this idea, using early popularity patterns to cluster videos and apply predictive models based on temporal dynamics.

In this project, we adopt a feature-based classification approach focused on lightweight metadata from YouTube Shorts. This allows us to avoid the computational overhead of multi-modal models while still leveraging proven early indicators of virality. By comparing Logistic Regression and Random Forest, we aim to evaluate the trade-off between interpretability and predictive performance.

2.3 Algorithms for Classification in Social Media Analysis

Logistic Regression and Random Forest are commonly used in trend prediction due to their interpretability and performance. Khan et al. (2020) used temporal bipartite networks to model social media trends, showing Random Forest performed better on sparse, noisy engagement signals. Other studies such as He and Li (2024) have applied deep learning, but simpler classifiers were found to be more explainable for decision-making support.

2.4 Metadata and Engagement Metrics as Predictive Features

Metadata like title length, posting time, and initial engagement within 24-48 hours have proven to be strong predictors of future virality (Rodrigues et al, 2021; Figueiredo et al, 2011). Chelaru et al. (2012) showed that early user interactions such as likes and comments are more predictive than content tags or thumbnails. Pinto et al (2013) compared feature weighting models and found early popularity spikes correlated strongly with long-term trending status.

2.5 Ethical and Practical Constraints of Platform APIs

While YouTube provides an official data API (v3), platforms like TikTok lack accessible public APIs. This creates ethical challenges for researchers. As a result, many studies (Li Eng and Zhang, n.d) focus on platforms where data is legally and reliably accessible, ensuring reproducibility and user privacy.

2.6 Summary

Prior work demonstrates the feasibility of predicting video trends using machine learning and early engagement metrics. While deep learning methods like CNNs or attention-based models (e.g., AMPS) have been used in high-resource settings, simpler classifiers such as Random Forest remain effective and interpretable for structured metadata tasks. Importantly, studies consistently show that early popularity indicators—especially within the first 48 hours—are strongly correlated with long-term trend emergence. The decision to use Logistic Regression and Random Forest is supported by prior evidence from Niture (2021), Khan et al. (2020), and Pinto et al. (2013), all of whom demonstrated that well-engineered metadata features can yield robust results without relying on expensive content processing.

This project builds on these insights by focusing on short-form YouTube content, targeting a clearly defined 'trend label' based on performance thresholds (e.g., 100,000 views in 48 hours). Feature engineering will include variables such as publication time, title length, like-to-view ratio, and subscriber count of the channel (if accessible). Model performance will be measured using precision, recall, and F1-score. Future extensions may explore embeddings or API-level comparisons with platforms like TikTok.

Chapter 3: Project Design

3.1 Project Overview

This project aims to develop a machine learning-based system capable of predicting whether a YouTube shorts video will emerge as a trending item within a defined early window, such as 48 hours after upload. This prediction is made using public metadata features obtained through YouTube's official API. This will enable researchers and practitioners to analyse engagement dynamics in short-form video platforms and help content producers optimize timing, and content strategizes based on early performance signals.

3.2 Template

This project follows the Template 1.2- Predictive Modelling of Social Media Trend Emergence. The goal of this project is to use classification models to predict the emergence of trends on social platforms, using structured features such as metadata and engagement metrics. The specific implementation in this case uses YouTube shorts as the sole platform, due to its open access through a public API.

3.3 User and Domain

The proposed system will primarily benefit individuals and organizations involved in content development, creation and digital outreach. Social media creators, digital marketers, non-profit organizations, and educators often rely on timing and virality to maximize their message reach. By providing early predictions of which videos are likely to trend, this project helps inform decisions around promotions, distributions, and optimisation strategies. In addition, data analysts and researchers may use such systems to understand behavioural patterns in short-form media engagement.

The domain of this project lies at the intersection of machine learning, social media analytics, and digital content performance forecasting. It falls under the broader category of predictive analytics within data science.

3.4 Design Choices

A number of design decisions were made to ensure that the project remains both technically feasible and ethically compliant. Initially, both TikTok and YouTube shorts were considered as data sources. However, TikTok's data is not publicly available through a standard API, and attempts to obtain data through scraping would raise ethical and legal concerns. Therefore, YouTube shorts were chosen as the sole platform due to the availability of a well-documented and officially supported API.

Feature selection is grouped into the assumption that early engagement metrics and metadata contains useful signals for forecasting future performance. These include view count, like-to-view ratio, publishing time and day, and text-based metadata such as title length. Some features may also consider binary flags such as whether comments are enabled or ratings are hidden.

A binary label (i.e. trend vs non-trend) will be generated using a predefined threshold for example, a video receiving over 100,000 views within 48 hours maybe considered trending. This threshold is provisional and maybe refined during exploratory data analysis (EDA) based on the actual data distribution.

Two models have been selected for implementation: Logistic Regression and Random Forest. Logistic regression offers interpretability and a baseline for comparison, while Random Forest provides a more robust, ensemble-based method capable of handling non-linear feature interactions. These choices were influenced by similar works in the literature and their applicability to structured metadata.

All data will be retrieved via the YouTube Data API v3, ensuring ethical compliance. The system does not collect user-specific or sensitive data/information, focusing strictly on publicly available metadata.

3.5 Overall Structure of the System

The project is designed around the modular and iterative structure to allow progressive development and evaluation. The major stages of the system are as follows:

1. Data Collection:
   YouTube Shorts metadata will be collected using the YouTube Data API v3. Queries will be constructed to search for recent videos based on categories, keywords, and date filters. The API responses will include engagement statistics, snippet information, and video-level identifiers.

2. Data Cleaning and Feature Engineering:
   Raw data will be pre-processed to ensure consistency. Time-based fields will be transformed to derive publish windows (e.g. weekday, hour of upload), and ratios such as likes per view will be calculated. Features will be extracted into a tabular format suitable for model training.

3. Trend Labelling:
   Each video will be assigned a trend label based on a pre-set threshold (eg. achieving 100,00 views within 48 hours). The label serves as the binary target variable for classification.

4. Model Training and Evaluation:

Two machine learning models — Logistic Regression and Random Forest — will be trained on the labelled dataset. Performance will be evaluated using F1-score, accuracy, and confusion matrix. Cross-validation will be used to assess model stability and generalisation.

5. Prototype Demonstration:
   A Jupyter Notebook will serve as the core interface for demonstrating the system. This includes visualisations, code commentary, and evaluation results, all of which will be recorded as part of the prototype submission.

3.6 Technologies and Tools

The project uses a well-supported Python data science stack that is suitable for prototyping, modelling, and evaluation. The primary tools are:

- **Python**: The main programming language for implementation.
- **Jupyter Notebook**: Interactive environment for development, prototyping, and demonstration.
- **YouTube Data API v3**: Used to fetch real-time and historical metadata on YouTube Shorts.
- **pandas**: For data manipulation and analysis.
- **scikit-learn**: For implementing classification models and performance metrics.
- **matplotlib / seaborn**: Used for data visualisation during EDA and result presentation.
- **Google Cloud Console**: Used to generate and manage the API key, and to monitor usage quotas and access logs.

These tools were selected for now, their reliability, community support, and suitability for small- to medium-scale predictive modelling projects.

3.7 Project Plan

The overall project is structured over a 10-week timeframe, broken into three main phases: preparation, research, and development. The plan ensures that core functionality is completed intime for the feedback and final refinement.

Phase 1: Preparation (Week1-5)

1. Finalised the project topic and scope based on Template 1.2.
2. Selected YouTube Shorts as the sole platform due to API accessibility.
3. Set up Google Cloud Console, generated an API key, and configured credentials.

4. Developed Python code to connect to the YouTube Data API v3.
5. Successfully retrieved metadata and structured the output using pandas.
6. Documented the working prototype in Chapter 4.

Phase 2: Research and Writing (Week6-8)

1. Research the related works for the project.
2. Read through the scholarly sources for the literature review.
3. Complete the literature review for the Preliminary Report.

Phase 3: Final Evaluation and Submission (Week9-10)

1. Finalise the prototype in Jupyter Notebook with output visuals.
2. Record a 3-minute video showcasing the working prototype.
3. Complete and proofread the Report, including updated screenshots and figures.
4. Submit all required deliverables and address for supervisor's feedback.

3.8 Evaluation Strategy

The effectiveness of the proposed system will be assessed using the standard classification performance metrics. Accuracy will be measured the overall correctness of predictions, while F1-score will be used to balance precision and recall, especially if the data is imbalanced (i.e. fewer trending videos than non-trending ones). A confusion matrix will be produced to visualise the distribution of prediction outcomes and identify patterns of false positives and false negatives.

Comparative evaluation between Logistic Regression and Random Forest will be performed, considering both predictive performance and model interpretability. The goal is not only to achieve high accuracy, but also to understand which features are most influential in predicting trend emergence. This evaluation approach ensures that both quantitative and qualitative insights into model behaviour.

Chapter 4: Feature Prototype

4.1 Purpose of the Prototype

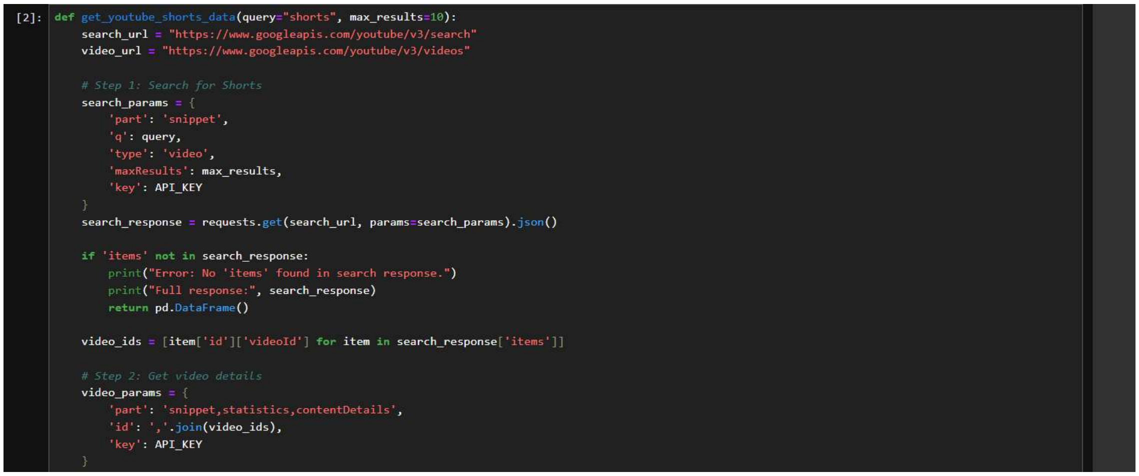
The purpose of this prototype is to demonstrate the core functionality of the proposed trend prediction system. Specifically, it focuses on retrieving metadata from YouTube Shorts using the YouTube Data API v3, processing the data into structured features using pandas, and confirming that this data is suitable for further machine learning development. This early demonstration ensures that the technical foundation for accessing and preparing data is functioning correctly.

4.2 What It Does

The current prototype focuses on the successful integration of the YouTube Data API v3 to collect metadata from YouTube Shorts. It is developed using Python within a Jupyter Notebook and demonstrates the use of the `search` and `videos` API endpoints to query short-form video content based on search terms such as “shorts” and “trending.”

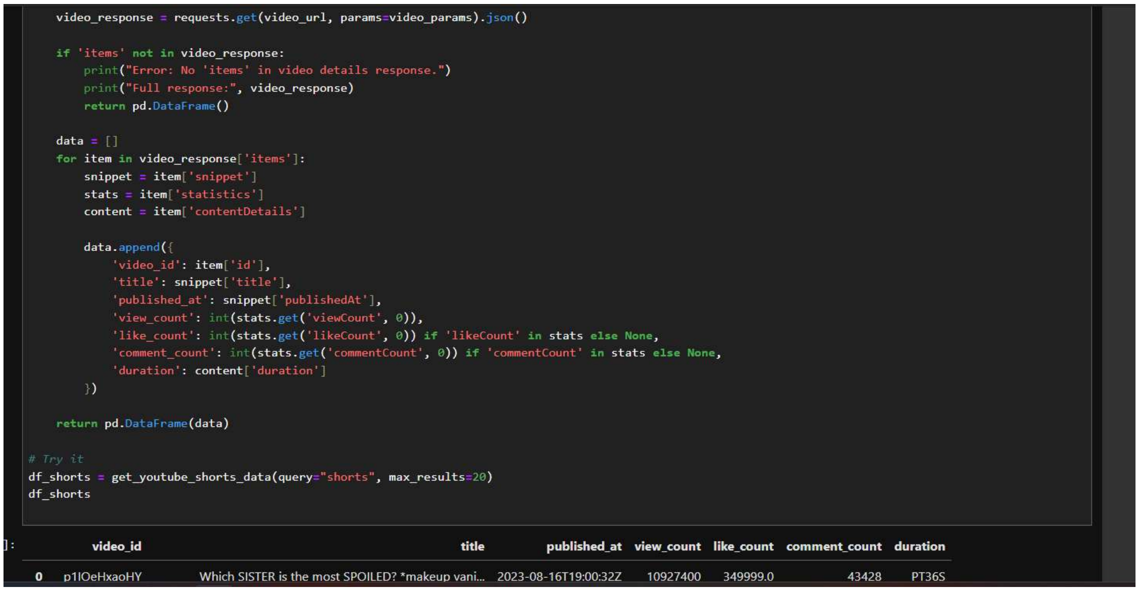
The collected data includes key metadata fields such as video title, publish date, view count, like count, comment count, and duration. These are returned in a structured format using the pandas library and displayed as a Data Frame. This metadata will later be used for feature extraction and trend prediction.

The prototype defines a function that sends a request to YouTube, retrieves video statistics, and processes the data into a usable tabular form. Below are screenshots of the notebook cell that performs this operation:



```
[2]: def get_youtube_shorts_data(query="shorts", max_results=10):
         search_url = "https://www.googleapis.com/youtube/v3/search"
         video_url = "https://www.googleapis.com/youtube/v3/videos"

         # Step 1: Search for Shorts
         search_params = {
             'part': 'snippet',
             'q': query,
             'type': 'video',
             'maxResults': max_results,
             'key': API_KEY
         }
         search_response = requests.get(search_url, params=search_params).json()

         if 'items' not in search_response:
             print("Error: No 'items' found in search response.")
             print("Full response:", search_response)
             return pd.DataFrame()

         video_ids = [item['id']['videoId'] for item in search_response['items']]

         # Step 2: Get video details
         video_params = {
             'part': 'snippet,statistics,contentDetails',
             'id': ','.join(video_ids),
             'key': API_KEY
         }
```

Figure 4.1a: First part of the metadata retrieval function, setting up the API request.



```
    video_response = requests.get(video_url, params=video_params).json()

    if 'items' not in video_response:
        print("Error: No 'items' in video details response.")
        print("Full response:", video_response)
        return pd.DataFrame()

    data = []
    for item in video_response['items']:
        snippet = item['snippet']
        stats = item['statistics']
        content = item['contentDetails']

        data.append({
            'video_id': item['id'],
            'title': snippet['title'],
            'published_at': snippet['publishedAt'],
            'view_count': int(stats.get('viewCount', 0)),
            'like_count': int(stats.get('likeCount', 0)) if 'likeCount' in stats else None,
            'comment_count': int(stats.get('commentCount', 0)) if 'commentCount' in stats else None,
            'duration': content['duration']
        })

    return pd.DataFrame(data)

# Try it
df_shorts = get_youtube_shorts_data(query="shorts", max_results=20)
df_shorts
```

```
]:      video_id                                           title          published_at  view_count  like_count  comment_count  duration
    0   p1IOeHxaoHY   Which SISTER is the most SPOILED? *makeup vani...  2023-08-16T19:00:32Z    10927400    349999.0          43428     PT36S
```

Figure 4.1b: Continuation of the function, showing video metadata extraction and DataFrame creation.

The output of this process is a structured table containing the video metadata, confirming that the system successfully retrieves and processes data for analysis.

[Insert Screenshot of df_shorts.head() here]
Figure 4.2: Output preview showing YouTube Shorts metadata returned in a pandas DataFrame.

This forms the foundation for further development, where feature engineering, labelling, and machine learning will be added in subsequent stages of the project.

4.3 Technologies Used

The following technologies were used in the development of the feature prototype:

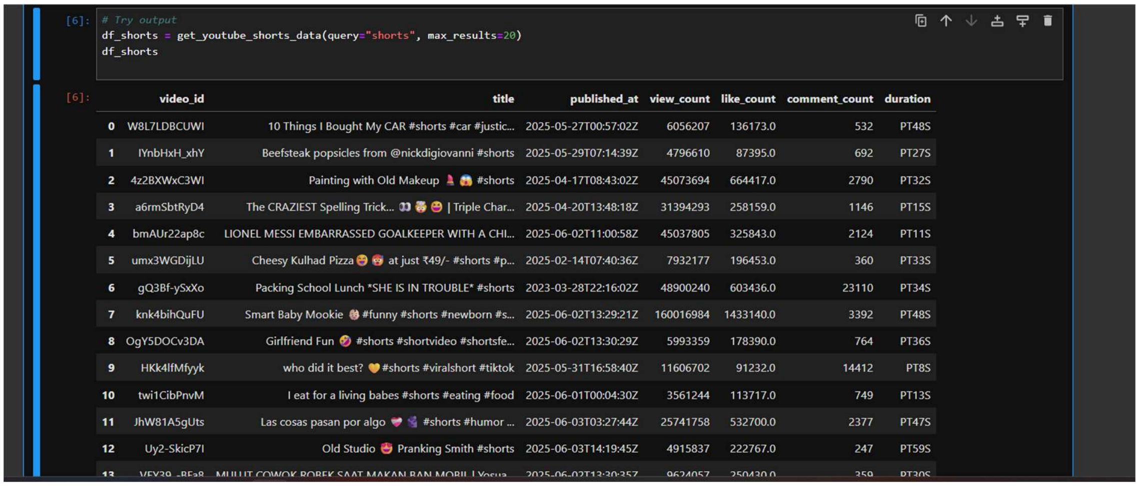
- Python: Programming language used to implement all components.
- Jupyter Notebook: Used for code development, testing, and demonstration.
- YouTube Data API v3: Official API used to retrieve metadata from YouTube Shorts.
- pandas: Used for data manipulation and preprocessing.
- matplotlib / seaborn: Intended for visualization of results in future stages.

4.4 Evaluation or Output Example

In this version of the prototype, an initial classification model was implemented using metadata retrieved from YouTube Shorts. After labelling the dataset using a threshold of view count (> 100), some samples were forcefully marked as non-trending to simulate class variety and allow training. This was done purely for demonstrating purpose due to the imbalance in real API-returned samples.

A logistic Regression model was then trained using **view_count**, **like_count**, and **comment_count** as features. The model was evaluated using a basic train-test split and classification report from scikit-learn.

The classification report showed that the model could successfully identify trending videos (class 1) with high recall and F1-score but performed poorly on the non-trending class (class 0). This reflects the class imbalance and highlights a key area for future improvement, either through more diverse data collection or improved threshold strategies.

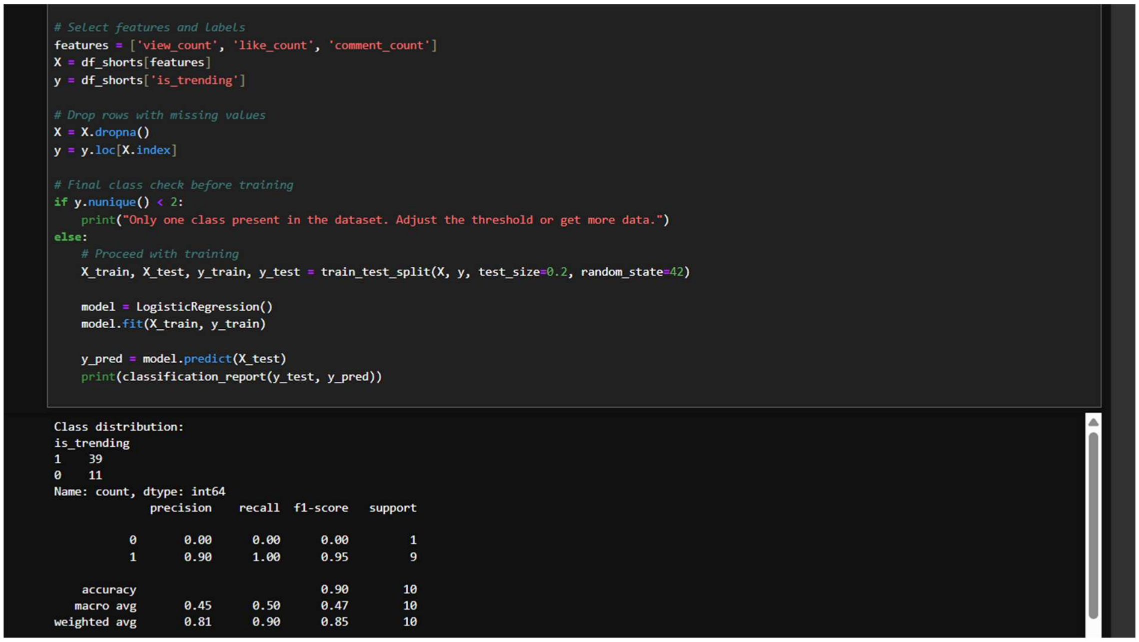


```
[6]: # Try output
     df_shorts = get_youtube_shorts_data(query="shorts", max_results=20)
     df_shorts
```

[6]:

	video_id	title	published_at	view_count	like_count	comment_count	duration
0	W8L7LDBCUWI	10 Things I Bought My CAR #shorts #car #justic...	2025-05-27T00:57:02Z	6056207	136173.0	532	PT48S
1	IYnbHxH_xhY	Beefsteak popsicles from @nickdigiovanni #shorts	2025-05-29T07:14:39Z	4796610	87395.0	692	PT27S
2	4z2BXWxC3WI	Painting with Old Makeup 💄 😱 #shorts	2025-04-17T08:43:02Z	45073694	664417.0	2790	PT32S
3	a6rmSbtRyD4	The CRAZIEST Spelling Trick... 👀 🤯 😄 | Triple Char...	2025-04-20T13:48:18Z	31394293	258159.0	1146	PT15S
4	bmAUr22ap8c	LIONEL MESSI EMBARRASSED GOALKEEPER WITH A CHI...	2025-06-02T11:00:58Z	45037805	325843.0	2124	PT11S
5	umx3WGDijLU	Cheesy Kulhad Pizza 😋 🤯 at just ₹49/- #shorts #p...	2025-02-14T07:40:36Z	7932177	196453.0	360	PT33S
6	gQ3Bf-ySxXo	Packing School Lunch *SHE IS IN TROUBLE* #shorts	2023-03-28T22:16:02Z	48900240	603436.0	23110	PT34S
7	knk4bihQuFU	Smart Baby Mookie 👶 #funny #shorts #newborn #s...	2025-06-02T13:29:21Z	160016984	1433140.0	3392	PT48S
8	OgY5DOCv3DA	Girlfriend Fun 🤣 #shorts #shortvideo #shortsfe...	2025-06-02T13:30:29Z	5993359	178390.0	764	PT36S
9	HKk4lfMfyyk	who did it best? 💛 #shorts #viralshort #tiktok	2025-05-31T16:58:40Z	11606702	91232.0	14412	PT8S
10	twi1CibPnvM	I eat for a living babes #shorts #eating #food	2025-06-01T00:04:30Z	3561244	113717.0	749	PT13S
11	JhW81A5gUts	Las cosas pasan por algo 💝 👾 #shorts #humor ...	2025-06-03T03:27:44Z	25741758	532700.0	2377	PT47S
12	Uy2-SkicP7I	Old Studio 😍 Pranking Smith #shorts	2025-06-03T14:19:45Z	4915837	222767.0	247	PT59S
13	VFY39_-BFa8	MUUUT COWOK ROBEK SAAT MAKAN BAN MOBIL | Yosua...	2025-06-02T13:30:35Z	9624057	250430.0	359	PT30S

Figure 4.3a: Evaluation output showing the DataFrame preview of retrieved metadata.



```
# Select features and labels
features = ['view_count', 'like_count', 'comment_count']
X = df_shorts[features]
y = df_shorts['is_trending']

# Drop rows with missing values
X = X.dropna()
y = y.loc[X.index]

# Final class check before training
if y.nunique() < 2:
    print("Only one class present in the dataset. Adjust the threshold or get more data.")
else:
    # Proceed with training
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

    model = LogisticRegression()
    model.fit(X_train, y_train)

    y_pred = model.predict(X_test)
    print(classification_report(y_test, y_pred))
```

```
Class distribution:
is_trending
1    39
0    11
Name: count, dtype: int64
              precision    recall  f1-score   support

           0       0.00      0.00      0.00         1
           1       0.90      1.00      0.95         9

    accuracy                           0.90        10
   macro avg       0.45      0.50      0.47        10
weighted avg       0.81      0.90      0.85        10
```

Figure 4.3b: Classification report output from Logistic Regression model.

4.5 Limitations

As this is an early-stage prototype, there are several limitations. The dataset is limited in size due to API quota constraints, and trend labels have not yet been implemented. The current implementation focuses on metadata retrieval and does not yet include feature engineering, trend classification, or evaluation using machine learning models. These components will be integrated in the later phases of the project.

Chapter 5: References

Chelaru, S., Orellana-Rodriguez, C. and Altingovde, I.S. (2012) 'Impact of social features on ranking YouTube videos', Proceedings of the IEEE, pp. 32–39.

Cho, M., Jeong, D. and Park, E. (2024) 'AMPS: Predicting popularity of short-form videos using multi-modal attention mechanisms', Journal of Retailing and Consumer Services, 78, p.103778.

Figueiredo, F., Benevenuto, F. and Almeida, J.M. (2011) 'The tube over time: Characterizing popularity growth of YouTube videos', Proceedings of the fourth ACM international conference on Web search and data mining, pp. 745–754.

He, Z. and Li, D. (2024) 'Short Video Popularity Prediction Using IoT and Deep Learning', IEEE Access.

Khan, A. et al. (2020) 'Predicting emerging trends on social media by modelling it as temporal bipartite networks', IEEE Access, 8, pp. 39635–39646.

Li, W., Eng, C. and Zhang, Y. (n.d.) 'Multi-class classification models for YouTube popularity prediction', arXiv preprint.

Niture, A.A. (2021) 'Predictive analysis of YouTube trending videos using machine learning', MSc Thesis, Dublin Business School.

Pinto, H., Almeida, J. and Gonçalves, M. (2013) 'Using early view patterns to predict the popularity of YouTube videos', WSDM ’13: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, pp. 365–374.

Rodrigues, A.P. et al. (2021) 'Real-Time Twitter Trend Analysis Using Big Data Analytics and ML Techniques', Wireless Communications and Mobile Computing, 2021(1), p.3920325.