Data Collection Instructions for TikTok Video Analysis

Author: Tenzin Kunsang
Engineer: kawsarnewazchowdhury@gmail.com
Date: November 24th, 2025
Github Repo: https://github.com/tenzin-kunsang648/FT-DataCollection.git

The goal is to collect data from TikTok videos at multiple timepoints to track their growth trajectory and extract velocity metrics that indicate viral potential.

Some quick notes:

- Please read the entire instruction before starting anything.
- Document your process. I want to know why an attribute may or may not be available and what your attempts were for each, if you made any assumptions or added filters in your queries for the videos you gathered metrics of, the number of API calls you can make or any other limitations there might be for TikTok API calls, etc
- Note that not all the metrics I need might be directly available through public APIs. Some of these might mean feature engineering, others might mean reference / api call to another source. However, please thoroughly investigate:
        - Inspect browser DevTools Network tab - look for internal API calls
        - Check if any analytics are visible in the HTML (even if hidden)
        - Look for any "insights" or "analytics" sections on public video pages
        - Document what you find vs. what's definitely inaccessible
    - Even if you can't get these now, documenting their location helps us understand what might be accessible in the future or through alternative methods.

1. Time-Series Data Collection **(MOST IMPORTANT)**

    We need to track the SAME videos at multiple points in time to calculate velocity metrics. For each video, collect data at:

    - **Hour 1** after upload
    - **Hour 6** after upload
    - **Hour 24** after upload
    - **Hour 48** after upload

2. Video Selection Strategy

Focus on collecting:

- **Recently uploaded videos** (within the last 1-6 hours)
- From the "trending" or "For You" page
- A mix of performance levels (some with 1K views, some with 100K, some with 1M+)
- A mix of genres/types (game video, educational, makeup, etc)
- Aim for **100-200 videos** to start, tracked over 48 hours

Track these videos over time - this is the dataset foundation.

*It might help to get only videos that have over **some threshold** of views in the first hour, just because our hypothesis is that those videos will be deemed more important by TikTok's algo. A bit of research into that might help. This might not be true necessarily but these just statistically might have more likelihood of being of some importance to us.*

3. Required Data Fields

A. Video Metadata (collect once)

- video_id - unique identifier
- video_url - direct link
- upload_timestamp - when posted (critical for chronological analysis)
- caption / description
- hashtags - list of all hashtags used
- video_duration - length in seconds
- thumbnail_url
- music_id / sound_id - audio track identifier
- sound_title - name of audio/music used

B. Engagement Metrics (collect at each timepoint: 1h, 6h, 24h, 48h)

- collection_timestamp - when YOU scraped this data
- hours_since_upload - calculated: (collection_timestamp - upload_timestamp)
- views - total view count
- likes - total likes (diggCount)
- comments_count - number of comments
- shares - share count
- saves / favorites - bookmark count (collectCount if available)
- duet_count - number of duets (if accessible)
- stitch_count - number of stitches (if accessible)

C. Creator/User Information (collect once per video)

- creator_username - unique username
- creator_display_name
- creator_follower_count - number of followers
- creator_following_count - number they follow
- creator_total_likes - lifetime likes on their account
- creator_total_videos - number of videos posted
- creator_verified - verification status (boolean)
- creator_bio - profile description

D. Comments Data (collect at 48h timepoint only)

For **top 20-50 comments** on each video:

- comment_text - actual comment content
- comment_username - who posted it
- comment_likes - likes on the comment
- comment_timestamp - when posted (if available)

E. Platform-Specific TikTok Features

- for_you_page_appearances - if video appeared on FYP (likely not directly accessible)
- sound_trending - is the audio currently trending? (boolean, may need external check)
- completion_rate / average_watch_time - if available through any means
- completion_rate / video_completion_percentage - what % of viewers watch the full video
- average_watch_time - average seconds watched
- replay_count - how many times people rewatch
- audience_retention_curve - second-by-second view retention
- Video transcript/captions (auto-generated or manual)
- Video file itself (not needed unless doing computer vision analysis)
- Virality Indicators:
    - for_you_page_appearances - times shown on FYP
    - share_destination - where shares are going (Stories, DMs, external)
    - traffic_source - how viewers found the video (FYP, Following, Search, Profile)
- Audio/Trend Signals:
    - sound_trending - is this audio currently trending
    - sound_usage_count - how many videos use this sound

Data Structure Format

Option 1: Time-Series Format (Preferred)

Each row = one video at one timepoint

Option 2: Separate Tables

- **videos** table: static metadata (video_id, creator info, caption, upload_time)
- **video_snapshots** table: time-series metrics (video_id, collection_timestamp, hours_since_upload, views, likes, etc.)
- **comments** table: comment data (video_id, comment_text, comment_user, comment_likes)

Collection Method Guidance

Automation Requirements

Since we need to track videos over 48 hours, you'll need:

1. **Initial scrape**: Identify 100-200 recently uploaded trending videos
2. **Scheduled collection**: Set up automated scraping at 1h, 6h, 24h, 48h marks for the same videos
3. **Storage**: Save to CSV or database with timestamp

Technical Approaches (based on your research)

- Use **Playwright/Selenium** for browser automation to load dynamic content
- Parse JSON data from TikTok's internal API responses when possible
- Handle rate limiting and rotating IPs if necessary
- **Store raw HTML/JSON responses as backup for re-parsing later !!! (I need this for sure to verify some details from time to time)**

Data Quality Checks

- Verify upload_timestamp is accurate
- Verify hours_since_upload is calculated correctly
- Check that view counts are monotonically increasing (views shouldn't decrease over time)
- Flag any missing data fields

What We **DON'T** Need (to save time)

- Video file downloads

- Full comment threads (just top-level comments)
- User's entire video history
- Private account data

Platform Comparison (Nice to Have)

After TikTok data collection is working, document which of these fields are:

- Also available on Instagram Reels
- Also available on YouTube Shorts
- TikTok-exclusive (like duet_count, stitch_count)

This will help us decide if we build a unified model or platform-specific models.

Deliverables

1. **Sample dataset**: 20-30 videos tracked over 48 hours with all fields above
2. **Data dictionary**: Document explaining each field and any collection limitations
3. **Collection script**: Automated tool that can track videos over time
4. **Coverage report**: Which fields you successfully collected vs. which are inaccessible

Timeline

- **Week 1**: Proof of concept - track 10 videos for 48 hours
- **Week 2-3**: Scale to 100-200 videos with full automation
- **Ongoing**: Continue collecting data as pipeline runs

Questions to Investigate (but not limited to these)

1. Can you access upload_timestamp reliably, or do we need to estimate based on when we first see the video?
2. Are duet_count and stitch_count visible in the HTML/JSON, or only the total engagement numbers?
3. Can you identify if a sound/music is "currently trending" through any API or page indicator?
4. How frequently can we scrape without getting blocked? What rate limits exist?

Please prioritize the **time-series collection** aspect - this is the core requirement for building velocity features that predict virality.