# Modelo predictivo para viralidad en videos

**Estudiantes:**

**Kinan El Halabi Jaber & Joel Sebastián Montenegro González**

**Universidad de La Sabana — Teoría de la Generalización**

**Docente:**

**Daniel Garzón Rodríguez**

**Fecha:**

**12 de Noviembre**

## Resumen

Este proyecto propone un modelo para predecir la viralidad de videos en TikTok, combinando datos estadísticos con análisis automatizado del contenido visual y auditivo. Para ello se emplearon los modelos *LLaVA-NeXT-Video-7B* (Zhang et al., 2024) para describir el contenido visual del video, y *Audio Flamingo 3* (NVIDIA, 2025) exclusivamente para determinar si el audio contiene música.

A partir de estas variables y métricas como número de seguidores, *likes*, comentarios y compartidos, se calculó el *engagement rate*, utilizado como medida principal de rendimiento. En lugar de emplear un umbral fijo, la clasificación de viralidad se definió utilizando el **percentil 75 del engagement rate**. Esto significa que el **25 % de los videos con mayor interacción** fueron considerados como virales.

Este enfoque permite adaptar la definición de viralidad a la distribución real de los datos y construir un modelo predictivo más coherente con el comportamiento observado en

la base.

**Palabras clave:** viralidad; redes sociales; modelos multimodales; TikTok.

# 1. Introducción

Las redes sociales se han convertido en uno de los principales medios de comunicación, consumo cultural y entretenimiento. En plataformas como TikTok, donde predominan videos cortos de alta rotación, la viralidad depende de múltiples factores relacionados tanto con el creador como con el contenido.

En este proyecto se propone un modelo predictivo multimodal, combinando datos estadísticos con información visual y auditiva. Para ello se utilizó *LLaVA-NeXT-Video-7B* para generar descripciones automáticas de los videos, y *Audio Flamingo 3* para identificar si el contenido incluye música. La transcripción completa del audio no se empleó en esta versión del proyecto y se deja para trabajos futuros.

El objetivo es determinar si un video pertenece al *25 % con mayor engagement rate*, utilizando el percentil 75 como umbral de referencia. Esta clasificación sirve para estudiar el comportamiento del algoritmo de viralización y explorar qué características del contenido visual y auditivo están asociadas con un mejor desempeño.

# 2. Metodología

Se usaron dos fuentes de datos:

- **TikTok Trending Videos** (van de Ven, 2020).
  - **Cantidad:** 1000 videos de TikTok.

**Base de datos generada artificialmente.**

- **Descripción:** Datos creados para complementar el análisis audiovisual.
- **Variables:**

| | |
|---|---|
| *ID del usuario* | *Número de seguidores* |
| *Número de cuentas seguidas* | *Cantidad de likes* |
| *Número de comentarios* | *Número de compartidos* |
| *Fecha del video* | *Fecha de creación de la cuenta* |
| *Categoría del contenido* | *Confianza de la categoría* |
| *Likes por seguidor* | *Comentarios por seguidor* |
| *Compartidos por seguidor* | |

Además, se añadieron dos nuevas variables:

1. **Contenido visual**:

   El modelo *LLaVA-NeXT-Video-7B* interpreta escenas del video y las transcribe en base al siguiente prompt:

   **Describe brevemente qué sucede en este video,**
   **incluyendo personas, objetos, acciones y entorno.**

   Esto para poder tener una descripción de los eventos que pasan en cada video para luego categorizar el tipo de contenido al que pertenece y identificar si el tipo de contenido tiene relación con la viralización del video.

**Ejemplo**

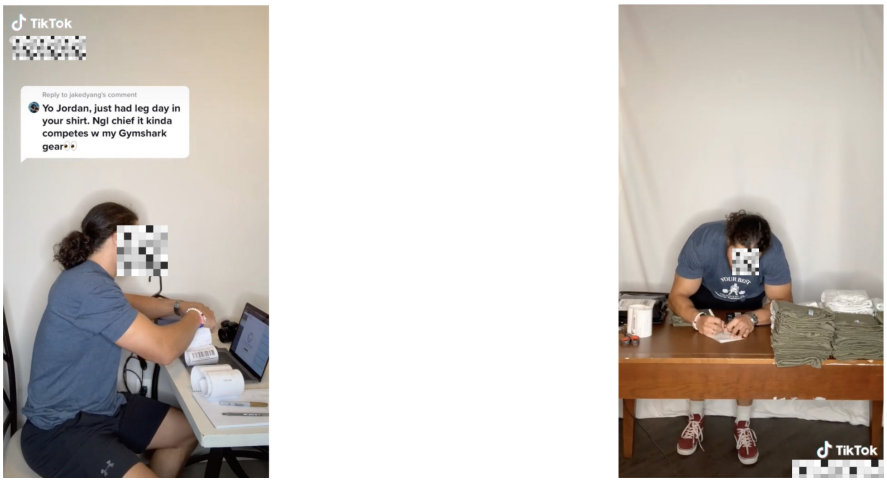


Figura 1: Imágenes extraídas por el modelo

**Descripción que brindó el modelo**
Vemos a un hombre trabajando en su escritorio. Él está vestido con una camisa de correa y tiene una larga melena. En el frente de su mesa, hay varias camisetas blancas y un ordenador. El hombre está escribiendo en una hoja de papel y parece estar organizando o revisando las camisetas. También hay varias bolsas de trabajo y un reloj en la mesa.

## Contenido auditivo

En esta versión del proyecto, el análisis auditivo se limitó a detectar si el video contiene música. Para ello se aplicó el modelo *Audio Flamingo 3* utilizando el siguiente prompt:

**Does this audio contain a song, singing, or background music? Answer 'yes' or 'no'.**

La transcripción completa del audio no se incluyó en esta versión del modelo, pero se documenta como trabajo futuro.

## Cálculo del *engagement rate* y definición de viralidad

El nivel de interacción de cada video se calculó con la fórmula:

$$ER = \frac{(\text{likes} + \text{comentarios} + \text{compartidos})}{\text{seguidores}} \times 100 \tag{1}$$

Esta medida compara la cantidad de interacciones con el número de seguidores, permitiendo evaluar cuentas de distinto tamaño.

Se definieron como virales los videos cuyo *engagement rate* se encuentra en el 35 % superior de la distribución, utilizando el percentil 65 como umbral. Esta decisión permite reducir el desbalanceo entre clases y facilitar el entrenamiento del modelo.

$$\text{viral} = \begin{cases} 1, & \text{si } ER \geq P_{75} \\ 0, & \text{si } ER < P_{75} \end{cases}$$

donde $P_{65}$ corresponde al valor del engagement rate en el percentil 65 de la base de datos.

Como trabajo futuro, el proyecto tiene como meta recolectar datos oficiales de videos reales de TikTok. Esto permitirá validar el umbral de 25 % con una distribución empírica y ajustar la definición de viralidad a partir de datos reales.

## Traducción de la descripción

Se implemento del modelo facebook/nllb-200-distilled-1.3B (Véase et al., 2022). Se buscaba traducir con certeza la variable de descripción ya que estaba en Ingles, con terminos ruidosos que pueden afectar el rendimiento del modelo, por eso se tradujo en ingles, se uso este modelo ya que no queremos traducir de manera literal los detalles que pasaron en cada video perdiendo contexto.

## Categorización de la descripción

Se implemento del modelo MoritzLaurer/deberta-v3-large-zeroshot-v2.0 (Véase Laurer et al., 2023)

Se uso este modelo para categorizar cada descripción del modelo, al tema principal, teniendo estas posibles categorias : (comedy, dance, sports, gaming, educational, vehicles, beauty, food, pets, technology, music, unidentified, other).

Con el proposito de reducir el ruido que generaban terminos conectores, ya con esto, el modelo muestra las categorias seleccionadas por video y su respectiva probabilidad de relacion con la descripcion.

## Modelo predictivo

Se empleó un modelo de red neuronal simple (*Multilayer Perceptron, MLP*) para predecir si un video es viral o no. El modelo utiliza como entrada las variables numéricas del dataset y las categorías codificadas.

# 3. Resultados

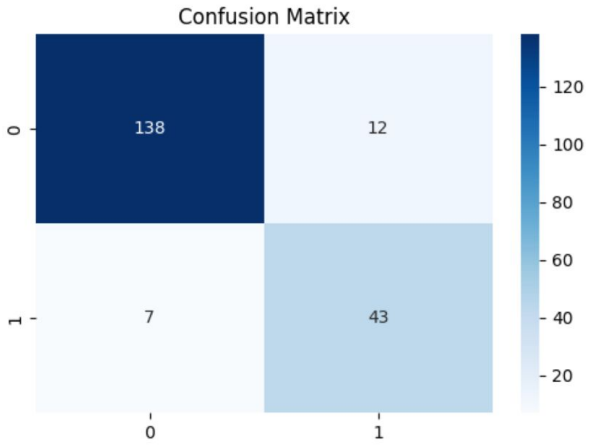


Figura 2: (a) Matriz de confusión

| **Métrica** | **Valor** |
|---|---|
| Accuracy | 0.90 |
| Precision | 0.7419 |
| Recall | 0.92 |
| F1-score | 0.8214 |
| AUC | 0.976 |

Cuadro 1: Resultados de desempeño del modelo.

El uso de modelos multimodales como *LLaVA-NeXT-Video-7B* y *Audio Flamingo 3* es una opción prometedora para analizar contenido audiovisual. El cálculo del *engagement rate* y el uso de percentiles para definir la viralidad permiten medir el éxito de manera clara y comparable.

Los resultados obtenidos muestran que el modelo propuesto es capaz de predecir la viralidad de videos con un desempeño sólido, reflejado en métricas como un AUC de 0.976 y un F1 de 0.82. Esto sugiere que las variables seleccionadas, combinadas con las descripciones generadas mediante modelos multimodales, capturan información relevante para distinguir

entre contenido viral y no viral. No obstante, dado que el dataset utilizado proviene de un subconjunto limitado de videos y los datos son artificiales, deja duda si sirve en la practica.

Como trabajo futuro, se plantea ampliar el estudio utilizando datos reales y modelos más avanzados, con el fin de obtener una representación más precisa de la distribución de la viralidad y mejorar la capacidad predictiva del sistema.

## Código y Modelos Entrenados

Para acceder al código completo del proyecto, así como a los modelos entrenados, visite el siguiente enlace al repositorio de GitHub:

**https://github.com/Kinanel07/PROYECTO-TEORIA-predecir-viralidad/tree/main**

## Bibliografía

et al., N. T. (2022). NLLB-200's distilled 1.3B variant.

Laurer, M., van Atteveldt, W., Casas, A., & Welbers, K. (2023). Building Efficient Universal Classifiers with Natural Language Inference.

NVIDIA. (2025). Audio Flamingo 3: Advancing Audio Intelligence with Fully Open Large Audio-Language Models.

van de Ven, E. (2020). TikTok Trending Videos (December 2020).

Zhang, Y., Li, B., Liu, h., Lee, Y. j., Gui, L., Fu, D., Feng, J., Liu, Z., & Li, C. (2024, abril). LLaVA-NeXT: A Strong Zero-shot Video Understanding Model. https://llava-vl.github.io/blog/2024-04-30-llava-next-video/

# A. Anexos

## A.1. Construcción de la Base de Datos

Para este proyecto se usó una base inicial de videos tomada de Kaggle. Estos videos solo incluían el archivo del video de TikTok, sin métricas como *likes*, comentarios o compartidos. Únicamente unos 36 videos tenían información adicional, pero como eran muy pocos, se decidió generar todas las variables faltantes mediante un código de simulación.

El prompt utilizado para crear estos datos está disponible en el repositorio de GitHub del proyecto.

A partir del código se generaron las siguientes variables:

- ID del autor del video.
- Número de seguidores y seguidos del autor.
- Fecha de creación de la cuenta del autor.
- Fecha de publicación del video.
- *Likes*, comentarios y compartidos.
- Descripción del video y variable que indica si el audio contiene música o no (generadas por modelos de lenguaje, explicados en otra sección).
- Categoría asignada al contenido del video (generada mediante clasificación por texto).
- Probabilidad o nivel de confianza asociado a la categoría asignada.

La idea principal fue crear un conjunto de datos que se pareciera a lo que normalmente se ve en TikTok y que mantuviera coherencia entre las variables.

## A.2. Parámetros Utilizados en la Simulación

Durante la simulación se tomaron varias decisiones sobre los rangos y las distribuciones de cada variable. A continuación se explican los parámetros más importantes:

- **Seguidores**: Se usó un rango entre 100 y 1 000 000. Este rango se escogió porque es común que la mayoría de usuarios tengan pocos seguidores y solo algunos lleguen a cifras altas. Se usó una distribución log-normal, que refleja este comportamiento.
- **Seguidos**: Se definió un rango entre 50 y 10 000. TikTok permite seguir hasta 10 000 cuentas, por ello se usó este umbral.
- **Fecha de creación de la cuenta**: Se generó entre 2017 y la fecha actual, ya que TikTok empezó a crecer desde ese año. Para simular diferentes antigüedades se usó una distribución exponencial.
- **Engagement rate**: Los valores van entre 0.4 % y 8 %. Este rango permite incluir cuentas con muy baja interacción y otras con un rendimiento más alto.
- **Likes, comentarios y compartidos**: Estas métricas no se generaron al azar, sino que dependen del número de seguidores y del engagement del autor:
  - los **likes** se calculan a partir de los seguidores y un nivel de engagement,
  - los **comentarios** representan entre el 3 % y el 15 % de los likes,
  - los **compartidos** están entre el 1 % y el 7 % de los likes.
- **Métricas normalizadas por seguidor**: Además de las variables absolutas, se incluyeron indicadores relativos para capturar la intensidad de interacción ajustada por el tamaño de la audiencia:

- **likes por seguidor**: permite medir qué tan efectiva es la interacción independientemente del número total de seguidores,
- **comentarios por seguidor**: refleja el nivel de participación activa proporcional a la audiencia,
- **compartidos por seguidor**: mide la propensión del contenido a ser difundido más allá de los seguidores directos.

Estas métricas reducen el sesgo que producen las cuentas muy grandes y permiten comparar usuarios con tamaños de audiencia distintos.

- **Fecha del video**: Siempre se genera después de la creación de la cuenta del autor, para mantener coherencia.

Estos parámetros fueron definidos buscando que los datos generados fueran razonables, consistentes y útiles para el análisis.

## A.3. Engagement Rate: Uso y Significado

El *engagement rate* (ER) es una de las métricas más utilizadas en redes sociales para medir qué tan bien interactúa la audiencia con un contenido. A diferencia de métricas como los *likes* o los comentarios que se obtienen después de subir el vídeo, el ER permite comparar cuentas de distintos tamaños, ya que toma en cuenta el número de seguidores al calcular la proporción.

En redes sociales, esta métrica se usa porque ayuda a responder una pregunta clave: **¿qué tan atractivo es el contenido para su audiencia?** Dos videos pueden tener la misma cantidad de interacciones, pero su impacto no es el mismo si uno pertenece a una

cuenta pequeña y el otro a una cuenta con cientos de miles de seguidores. El engagement rate corrige estas diferencias y permite una comparación justa para definir la viralidad.

En el área de marketing digital, el ER se usa para evaluar:

- Qué tan activos son los seguidores de una cuenta,
- Si un video tuvo un rendimiento mayor o menor de lo esperado,
- Qué tipo de contenido genera mejores resultados,
- Y si una cuenta tiene una audiencia real o inflada (por ejemplo, seguidores inactivos).

Además, esta métrica es útil para modelos predictivos porque resume varias interacciones en un solo valor. En este proyecto se usó tanto para analizar el desempeño de los videos como para definir qué se considera un contenido viral.

En resumen, el engagement rate es una métrica clave porque:

permite comparar cuentas de diferentes tamaños de forma justa; combina varias señales de interacción en un solo indicador; es ampliamente utilizada en estudios sobre redes sociales y marketing digital; ayuda a identificar el rendimiento de los vídeos.

Su cálculo se explica en el documento principal, y aquí se detalla por qué es una métrica fundamental para entender el comportamiento del contenido en TikTok y otras plataformas sociales.

# A3. Profundización en los modelos de análisis visual y auditivo

## A3.1 Modelo de visión: LLaVA-NeXT-Video-7B

LLaVA-NeXT-Video-7B es un modelo de inteligencia artificial que combina visión y lenguaje. Esto le permite analizar imágenes o videos y producir descripciones en texto. El modelo fue entrenado con grandes cantidades de videos e imágenes acompañados de textos descriptivos, lo que hace posible que relacione lo que ve con palabras y frases completas.

En este proyecto se usó para obtener descripciones automáticas del contenido visual de cada video. El modelo identifica elementos como personas, objetos, acciones, escenarios y actividades. La salida generada se usa luego como texto para crear categorías y variables que ayudan a estudiar si el tipo de contenido visual influye en la viralidad.

De forma sencilla, el modelo funciona combinando un componente visual que extrae información de los fotogramas del video y un componente de lenguaje que transforma esas características en una descripción coherente. No solo detecta objetos, sino que también interpreta la escena, por ejemplo: “una persona cocinando en una cocina” o “dos personas bailando en un escenario”.

Este análisis visual permite estudiar los videos sin revisarlos manualmente y facilita la clasificación de su contenido en temas como comedia, baile, tutoriales, actividades diarias, entre otros.

## A3.2 Modelo de audio: Audio Flamingo 3

Audio Flamingo 3 es un modelo especializado en análisis de audio, capaz de transformar sonido en texto y detectar elementos relevantes dentro del audio de un video. Está

entrenado en una amplia colección de idiomas y estilos auditivos, lo que lo hace adecuado para estudiar contenido generado en redes sociales como TikTok.

En este trabajo el modelo se utiliza únicamente para generar la siguiente variable:

1. **Detección de música**: el modelo identifica si el video contiene música, canto o algún tipo de pista sonora característica, devolviendo únicamente un resultado binario ("sí" o "no").

La transcripción completa del audio no se utilizará en esta versión del proyecto. Aunque el modelo es capaz de hacerlo, esta funcionalidad se reservará para futuras actualizaciones, ya que actualmente limita el flujo del modelado.

Audio Flamingo 3 reconoce más de 100 idiomas. Entre los más frecuentes en redes sociales que maneja con buena precisión se incluyen:

- Inglés
- Español
- Árabe
- Francés
- Mandarín

El modelo también puede manejar mezclas de idiomas, algo común en plataformas como TikTok.

La inclusión de esta variable auditiva permite estudiar si la presencia de música —especialmente música asociada a tendencias o retos virales— influye en el rendimiento del video. Esto aporta una capa adicional de información que complementa las señales visuales y textuales.

## A3.3 Modelo de traducción: NLLB-200 Distilled 1.3B

NLLB-200 (No Language Left Behind) es un modelo de traducción desarrollado por Meta. Está diseñado para traducir contenido entre más de 200 idiomas, manteniendo la mayor parte del significado original. Este modelo es útil cuando se trabaja con audio o texto que no está en inglés, ya que muchos modelos de análisis funcionan mejor cuando la entrada está en este idioma.

En este proyecto, NLLB-200 se utilizó para traducir las transcripciones generadas por el modelo de audio. Esto significa que si un video tiene audio en un idioma diferente al inglés, primero se transcribe con Audio Flamingo y luego se traduce al inglés con NLLB-200. Esto permite que los modelos posteriores trabajen con un texto más claro.

El modelo es especialmente útil porque TikTok contiene videos en muchos idiomas. La traducción ayuda a que las descripciones y variables generadas sean comparables entre sí, sin importar el idioma original del audio del video.

En resumen, NLLB-200 funciona como un paso intermedio que mejora la comprensión del contenido auditivo y permite que todo el análisis del proyecto use un formato estándar en inglés.

## A3.4 Modelo de clasificación por cero entrenamiento: DeBERTa-v3-large-zeroshot-v2.0

El modelo DeBERTa-v3-large-zeroshot-v2.0 es una herramienta de clasificación de texto que no requiere entrenamiento adicional (zero-shot). Esto significa que puede asignar categorías a textos sin necesidad de haber sido entrenado específicamente para cada categoría.

En este proyecto, planteamos usar este modelo para clasificar la descripción generada por el modelo de visión y agrupar los videos por tipo de contenido (por ejemplo: baile, humor,

tutorial, reto). Aunque no fue diseñado originalmente para videos ni contenido multimedia, su capacidad de clasificación de texto lo hace una opción útil para organizar los resultados.

Algunas claves de su funcionamiento:

- Se transforma la descripción del video en un texto de entrada.
- Se define un conjunto de clases (“baile”, “comedia”, “tutorial”, etc.).
- El modelo evalúa qué tan probable es que el texto pertenezca a cada clase, usando un formato de inferencia tipo "¿Este texto trata sobre clase?".
- Finalmente, se asigna la clase con mayor probabilidad.

Limitaciones importantes:

- Este modelo trabaja únicamente con texto (no procesa video ni audio directamente).
- La precisión puede bajar si la descripción no es clara o tiene errores.
- Al clasificar muchas clases muy específicas, puede confundirse o generar categorías inadecuadas.

En resumen, DeBERTa-v3-large-zeroshot-v2.0 se incorpora como una fase experimental para organizar las descripciones de contenido. Los resultados sirven como apoyo, pero no son definitivos, y pueden mejorarse con modelos diseñados específicamente para vídeo.

## A3.5 Modelo de predicción MLP (Multilayer Perceptron)

## Arquitectura

- Entrada: todas las características numéricas y categóricas.

- Capa oculta 1: 128 neuronas (ReLU).
- Capa oculta 2: 64 neuronas (ReLU).
- Dropout: 0.3 y 0.2 para reducir sobreajuste.
- Salida: 1 neurona con función sigmoide.

## Entrenamiento

- Función de pérdida: `BCEWithLogitsLoss`.
- Optimizador: AdamW (lr = 0.001).
- Épocas: 10.

## Predicción

La salida del modelo es una probabilidad entre 0 y 1. Se clasifica como viral si:

$$\hat{y} \geq \tau,$$

donde $\tau$ es el umbral seleccionado.

**Early Stopping**: Es una técnica usada durante el entrenamiento de modelos de machine learning para evitar el sobreajuste. Consiste en detener el entrenamiento cuando el modelo deja de mejorar en el conjunto de validación durante un número determinado de épocas.

Según Prechelt (1998), detener el entrenamiento antes de que el error de validación aumente ayuda a mejorar la generalización del modelo.

**pos_weight fuerte**: En problemas desbalanceados, como clasificación binaria donde la clase positiva es rara, el parámetro `pos_weight` (usado en funciones como `BCEWithLogitsLoss`) permite .ªumentar"la importancia de los ejemplos positivos.

Un *pos_ weight fuerte* significa asignar un valor alto a esta ponderación, de manera que los errores en la clase minoritaria tengan más impacto en la pérdida. Esto ayuda al modelo a no ignorar la clase minoritaria, especialmente cuando está muy subrepresentada.

**Threshold Automático**: En un modelo de clasificación binaria, normalmente se asigna una clase positiva cuando la probabilidad es mayor a 0.5. Sin embargo, cuando el dataset está desbalanceado o el proyecto necesita optimizar una métrica específica, este umbral puede ajustarse automáticamente.

Un *threshold automático* selecciona el punto de corte que maximiza una métrica como F1, precisión, sensibilidad o Youden's J (tasa de aciertos del AUC-ROC).

Biemann y Riedl (2013) explican que ajustar el umbral mejora el rendimiento en escenarios desbalanceados y puede adaptarse según el objetivo del modelo.

**Nota sobre privacidad.** Las imágenes del video fueron censuradas ocultando el rostro y el nombre de usuario con el fin de proteger la identidad de las personas que aparecen en el contenido original. Esto se hace para evitar cualquier exposición no autorizada de datos personales y cumplir con buenas prácticas de uso responsable de material obtenido de redes sociales.