Universidad Peruana de Ciencias Aplicadas

FACULTAD DE INGENIERÍA

Ciclo: Quinto
Curso: Fundamento de Data Science
Sección: 258
Docente: Nérida Isabel Manrique Tunque

INFORME DEL TB2
"Trending Youtube Videos Statics"

Integrantes:

Cahuana López, Leicy Cristell (U20231E777)
Huamán Cortez, Anabella Karina (U202216171)
Mercado De La Rosa, Luis Marcelo (U20211B656)
Montenegro López, Valentina Étoile (U202312021)

Contenido

1. INTRODUCCIÓN

   En la actualidad, el análisis de datos desempeña un papel fundamental en la toma de decisiones estratégicas en múltiples industrias. Plataformas digitales como YouTube generan volúmenes masivos de datos diariamente, los cuales contienen información valiosa sobre las preferencias, intereses y comportamientos de los usuarios. En este contexto, el presente proyecto tiene como propósito aplicar la metodología CRISP-DM para analizar los datos de videos en tendencia en Canadá, con el objetivo de responder a los requerimientos de información que nos pide una consultoría internacional, con sede en lima, y descubrir los patrones relevantes y generar conocimiento útil para la toma de decisiones.

2. INTEGRANTES DEL GRUPO Y LOS ROLES

El grupo está conformado por cuatro integrantes, quienes asumieron roles específicos dentro del marco del proyecto de ciencia de datos, siguiendo la metodología CRISP-DM. La asignación de roles se realizó en base a las fortalezas personales, afinidad por ciertas herramientas y áreas de interés profesional. A continuación, se detalla cada uno de los roles, responsabilidades y etapas del proyecto en las que participan:

- **Marcelo – Business Project Sponsor**
  Encargado de definir los objetivos del proyecto y alinear el análisis con las metas del negocio. Participa en la Comprensión del Negocio, redacción de los objetivos, validación de los entregables finales y elaboración de las Conclusiones.

- **Valentina – Data Scientist**
  Responsable del modelado de datos, incluyendo la selección de variables, entrenamiento de modelos (como regresión) y evaluación de métricas de desempeño. Participa principalmente en las etapas de Preparación de los datos y Modelado.

- **Leicy – Data Engineer**
  Se encarga del procesamiento de datos, limpieza, transformación y validación del dataset. Participa en la fase de Comprensión de los datos y en la Preparación de los datos, asegurando su calidad para el análisis posterior.
- **Anabella – Data Analyst**
  Su responsabilidad es realizar el análisis exploratorio de los datos, generar

visualizaciones significativas y extraer insights que respondan a las preguntas del negocio. Participa en la Comprensión de los datos y en la generación de dashboards y visuales para la presentación final.

3. METODOLOGÍA CRISP-DM

1. COMPRENSIÓN DEL NEGOCIO

Esta fase tiene como propósito definir claramente los objetivos del proyecto desde una perspectiva de negocio, entendiendo qué se busca responder o resolver mediante el análisis de datos. En nuestro caso, el negocio gira en torno a la plataforma YouTube y los factores que influyen en la popularidad de los videos en Canadá.

Objetivos del proyecto

- El objetivo principal del análisis es comprender qué factores influyen en que un video sea tendencia en YouTube en el contexto canadiense. Para ello, se plantea:

- Identificar qué categorías y características están más asociadas al éxito de un video.
- Detectar patrones en la interacción del público (likes, dislikes, comentarios) en los videos más vistos.
- Estimar o predecir el número de likes a partir de variables relacionadas con las vistas, cantidad de comentarios y canales.
- Extraer insights accionables que puedan ser utilizados por creadores de contenido, agencias de marketing y anunciantes.

Objetivos de Data Science

- Desde el punto de vista de ciencia de datos, este proyecto busca traducir las preguntas del negocio en tareas de análisis cuantitativo. Para ello, se propone:

- Aplicar árbol de decisión de regresión para predecir el número de me gustas (likes) como variable objetivo.
  Usar variables independientes como: views, comment_count y channel_title.

- Analizar los datos, limpiarlo y pre-procesar los datos.
- Realizar análisis de correlación y modelado para identificar las variables con mayor capacidad predictiva.
- Evaluar el desempeño del modelo utilizando métricas de regresión como MAE (Mean Absolute Error), RMSE (Root Mean Squared Error) y R^2 (coeficiente de determinación)
- Responder los requerimientos de la consultoría internacional.

2. COMPRENSIÓN DE LOS DATOS

Descripción de la estructura de los datos:

El dataset cargado contiene información sobre videos en tendencia en la plataforma YouTube en Canadá.

Cada fila representa un video en tendencia, mientras que las columnas describen distintos aspectos del video, como el título, el canal que lo publicó, la fecha en que se volvió tendencia, las métricas de interacción (vistas, likes, dislikes, comentarios), la ubicación geográfica y otros metadatos relacionados.

```
# Número de filas y columnas
print("Filas y columnas:", df.shape)

# Columnas disponibles
print("\nColumnas:")
print(df.columns.tolist())

# Tipos de datos por columna
print("\nTipos de datos:")
print(df.dtypes)
```

El dataset contiene un total de 40,881 registros (filas) y 20 variables (columnas).

Columnas:
['video_id', 'trending_date', 'title', 'channel_title', 'category_id', 'publish_time', 'tags', 'views', 'likes', 'dislikes', 'comment_count', 'thumbnail_link', 'comments_disabled', 'ratings_disabled', 'video_error_or_removed', 'description', 'state', 'lat', 'lon', 'geometry']

A continuación, se presentan los tipos de datos identificados para cada columna del dataset

```
Tipos de datos:
video_id                  object
trending_date             object
title                     object
channel_title             object
category_id                int64
publish_time              object
tags                      object
views                      int64
likes                      int64
dislikes                   int64
comment_count              int64
thumbnail_link            object
comments_disabled           bool
ratings_disabled            bool
video_error_or_removed      bool
description               object
state                     object
lat                      float64
lon                      float64
geometry                  object
dtype: object
```

A continuación, se muestra un resumen del conjunto de datos, incluyendo el tipo de cada columna y la cantidad de valores no nulos.

#	Column	Non-Null Count	Dtype
0	video_id	40881 non-null	object
1	trending_date	40881 non-null	object
2	title	40881 non-null	object
3	channel_title	40881 non-null	object
4	category_id	40881 non-null	int64
5	publish_time	40881 non-null	object
6	tags	40881 non-null	object
7	views	40881 non-null	int64
8	likes	40881 non-null	int64
9	dislikes	40881 non-null	int64
10	comment_count	40881 non-null	int64
11	thumbnail_link	40881 non-null	object
12	comments_disabled	40881 non-null	bool
13	ratings_disabled	40881 non-null	bool

```
14  video_error_or_removed      40881 non-null    bool
15  description                 39585 non-null    object
16  state                       40881 non-null    object
17  lat                         40881 non-null    float64
18  lon                         40881 non-null    float64
19  geometry                    40881 non-null    object
dtypes: bool(3), float64(2), int64(5), object(10)
```

Descripción de los datos:

Columna	Tipo	Descripción
video_id	object (Categórica)	ID único del video de YouTube
trending_date	datetime	Fecha en que el video fue tendencia
title	object (Categórica)	Título del video
channel_title	object (Categórica)	Nombre del canal de YouTube
category_id	object(Categórica)	Id de la categoría del contenido
publish_time	datetime	Fecha y hora de publicación
tags	object (Categórica)	Etiquetas relacionadas al video
views	int (Numérica)	Número de veces que el video ha sido reproducido.
likes	int (Numérica)	Número de “me gusta” que recibió el video
dislikes	int (Numérica)	Número de “no me gusta” que recibió el video
comment_count	int (Numérica)	Total de comentarios que recibió el video
thumbnail_link	object (Categórica)	URL de la miniatura (imagen de portada) del video
comments_disabled	bool (Categórica binaria)	Indica si los comentarios están desactivados (True o False).
ratings_disabled	bool (Categórica	Indica si los “me gusta” y “no

	binaria)	me gusta" están ocultos (True o False).
video_error_or_removed	bool (Categórica binaria)	Indica si el video fue eliminado o tiene error (True si el video ya no está disponible).
description	object (Categórica)	Descripción del video publicada por el canal
state	object (Categórica)	nombre del Estado perteneciente al país
lat	float (Numérica)	latitud geográfica de ubicación del Estado
lon	float (Numérica)	longitud geográfica de ubicación del Estado
geometry	object (Categórica)	Coordenadas geográficas en formato espacial (POINT(lon lat)) que indican la ubicación del video.

Exploración de los datos:

La estructura general del dataset muestra que cada fila representa un video en tendencia en Canadá, con variables que incluyen el título, canal, métricas de interacción (vistas, likes, dislikes, comentarios), ubicación geográfica y fecha de publicación. Las variables están distribuidas entre numéricas, categóricas y booleanas. El dataset también contiene coordenadas (lat, lon) y geometry para análisis espaciales.

CARGAR LOS DATOS

```
import pandas as pd

# Cargar el archivo CSV
df = pd.read_csv('CAvideos_cc50_202101.csv')
```

INSPECCIONAR LOS DATOS

```
# Ver las primeras filas
df.head()
```

	video_id	trending_date	title	channel_title	category_id	publish_time	tags	views	likes	dislikes	comment_count	thumbnail_link
0	n1WpP7iowLc	17.14.11	Eminem - Walk On Water (Audio) ft. Beyoncé	EminemVEVO	10	2017-11-10T17:00:03.000Z	Eminem|"Walk"|"On"|"Water"|"Aftermath/Shady/In...	17158579	787425	43420	125882	https://i.ytimg.com/vi/n1WpP7iowLc/default.jpg
1	0dBIkQ4Mz1M	17.14.11	PLUSH - Bad Unboxing Fan Mail	iDubbbzTV	23	2017-11-13T17:00:00.000Z	plush|"bad unboxing"|"unboxing"|"fan mail"|"id...	1014651	127794	1688	13030	https://i.ytimg.com/vi/0dBIkQ4Mz1M/default.jpg
2	5qpjK5DgCt4	17.14.11	Racist Superman | Rudy Mancuso, King Bach & Le...	Rudy Mancuso	23	2017-11-12T19:05:24.000Z	racist superman|"rudy"|"mancuso"|"king"|"bach"...	3191434	146035	5339	8181	https://i.ytimg.com/vi/5qpjK5DgCt4/default.jpg
3	d380meD0W0M	17.14.11	I Dare You: GOING	nigahiga	24	2017-11-12T18:01:41.000Z	ryan|"higa"|"higatv"|"nigahiga"|"i dare you"|"...	2095828	132239	1989	17518	https://i.ytimg.com/vi/d380meD0W0M/default.jpg

```
# Ver cantidad de filas duplicadas
duplicados = df.duplicated().sum()
print(f"Total de filas duplicadas: {duplicados}")
```

```
Total de filas duplicadas: 0
```

```
# Convertir fecha de formato '17.14.11' a 'YYYY-MM-DD'
df['trending_date'] = pd.to_datetime(
    df['trending_date'].astype(str).apply(
        lambda x: f"20{x.split('.')[0]}-{x.split('.')[2]}-{x.split('.')[1]}" if
isinstance(x, str) and len(x.split('.')) == 3 else None
    ),
    errors='coerce'
)
print(df['trending_date'].head(10))
```

```
2017-11-14
2017-11-14
2017-11-14
2017-11-14
2017-11-14
2017-11-14
2017-11-14
2017-11-14
2017-11-14
2017-11-14
```

```
# Tabla de frecuencia para videos por canal
df['channel_title'].value_counts().head(10)
```

channel_title	count
SET India	192
MSNBC	189
FBE	188
The Young Turks	186
REACT	183
CNN	182
VikatanTV	182
The Late Show with Stephen Colbert	172
ARY Digital	168
RadaanMedia	168

```
#Frecuencia de provincias
df['state'].value_counts().head(10)
```

state	count
Quebec	3247
Alberta	3209
British Columbia	3196
Yukon	3159
Ontario	3156
Prince Edward Island	3141
Northwest Territories	3140
Newfoundland And Labrador	3136
New Brunswick	3131
Nunavut	3123

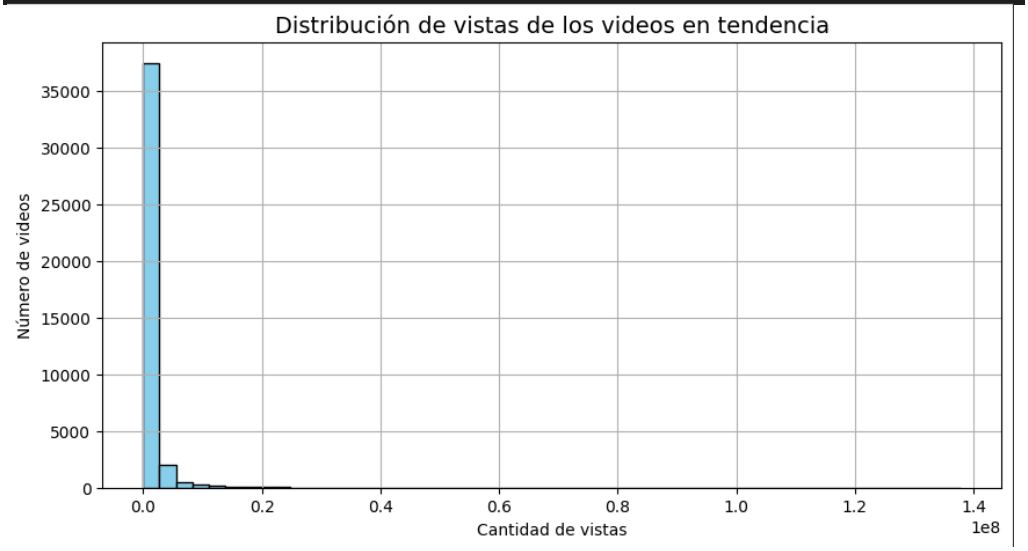
```
#Frecuencia de comentarios deshabilitados
df['comments_disabled'].value_counts()
```

comments_disabled	count
False	40298
True	583

VISUALIZAR LOS DATOS

```
#Distribución de vistas de los videos
plt.figure(figsize=(10,5))
# Generar un histograma con 50 intervalos (bins)
# Muestra la distribución de vistas ('views') en los videos
df['views'].hist(bins=50, color='skyblue', edgecolor='black')
#etiquetas
plt.title('Distribución de vistas de los videos en tendencia', fontsize=14)
plt.xlabel('Cantidad de vistas')
plt.ylabel('Número de videos')
plt.grid(True) #cuadrícula

plt.show()
```

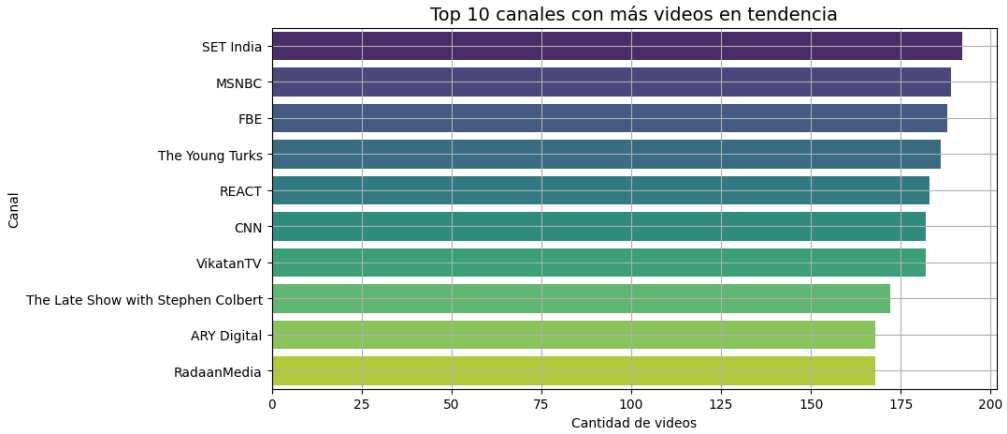


El gráfico muestra que la mayoría de videos en tendencia acumula pocas vistas, mientras que unos pocos superan ampliamente los valores promedio. Esta visualización permite detectar posibles valores atípicos y comprender mejor el comportamiento general de la variable views, lo que orienta futuras decisiones sobre su análisis.

```
#Top 10 canales con más videos en tendencia
```

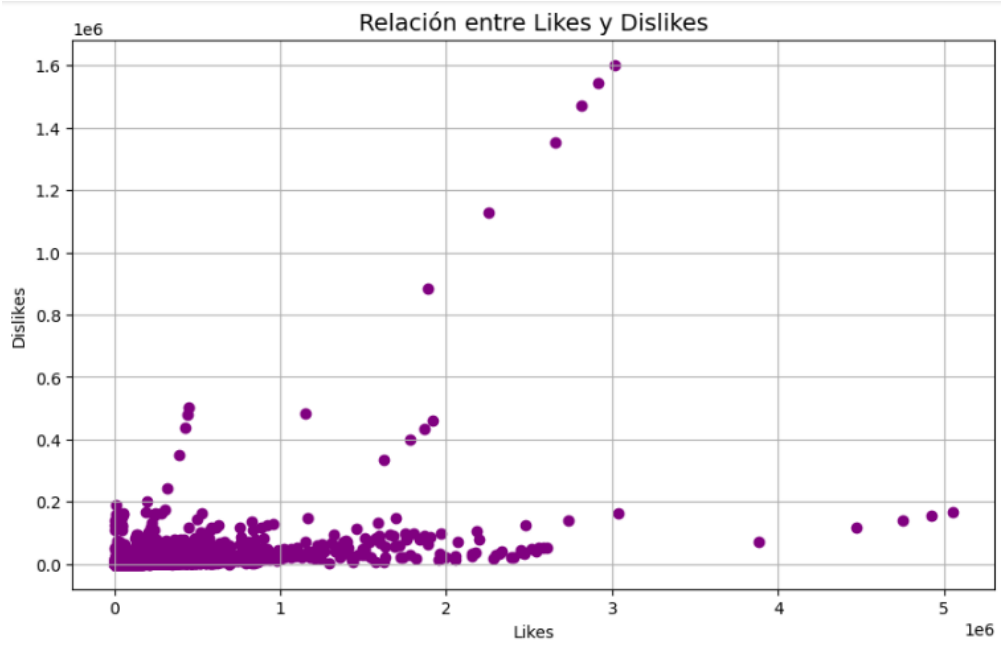
```
# Obtener los 10 canales con mayor cantidad de videos
top_canales = df['channel_title'].value_counts().head(10)

plt.figure(figsize=(10,5))
sns.barplot(x=top_canales.values,  y=top_canales.index, palette='viridis')
#etiquetas
plt.title('Top 10 canales con más videos en tendencia', fontsize=14)
plt.xlabel('Cantidad de videos')
plt.ylabel('Canal')
plt.grid(True)
plt.show()
```



Se identifican los canales con mayor presencia en tendencias, lo que permite observar patrones de repetición y concentración de popularidad. Esta visualización facilita reconocer qué creadores dominan el contenido destacado y puede orientar análisis posteriores sobre frecuencia, temática o estrategia de publicación.

```
#Correlación entre Likes y Dislikes
plt.figure(figsize=(10,6))
plt.scatter(df['likes'], df['dislikes'], color='purple')
plt.title('Relación entre Likes y Dislikes', fontsize=14)
#etiquetas
plt.xlabel('Likes')
plt.ylabel('Dislikes')
plt.grid(True)
plt.show()
```



Visualizar la relación entre likes y dislikes ayuda a entender el comportamiento de la audiencia frente al contenido. Permite identificar si los videos con mayor alcance generan reacciones mixtas, o si existe una correlación entre popularidad y polarización (es decir, cuando un video recibe tanto muchos likes como muchos dislikes, mostrando opiniones divididas del público). También puede ayudar a detectar casos atípicos, como videos con muchos dislikes en proporción a sus likes.

Verificar la calidad de los datos

- Se determina la consistencia de los valores individuales de los campos, verificar la cantidad y distribución de los valores nulos, y encontrar valores fuera de rango, los cuales logran constituirse en ruido para el proceso. La idea una vez llegados a este punto es poder garantizar la completitud y corrección de los datos.

- Consistencia de los valores individuales:

	category_id	views	likes	dislikes	comment_count	lat	lon
count	40881.000000	4.088100e+04	4.088100e+04	4.088100e+04	4.088100e+04	40881.000000	40881.000000
mean	20.795553	1.147036e+06	3.958269e+04	2.009195e+03	5.042975e+03	52.025876	-88.817702
std	6.775054	3.390913e+06	1.326895e+05	1.900837e+04	2.157902e+04	7.213076	25.119498
min	1.000000	7.330000e+02	0.000000e+00	0.000000e+00	0.000000e+00	44.566645	-139.000002
25%	20.000000	1.439020e+05	2.191000e+03	9.900000e+01	4.170000e+02	46.249282	-110.733329
50%	24.000000	3.712040e+05	8.780000e+03	3.030000e+02	1.301000e+03	49.822578	-81.236083
75%	24.000000	9.633020e+05	2.871700e+04	9.500000e+02	3.713000e+03	53.016698	-64.347995
max	43.000000	1.378431e+08	5.053338e+06	1.602383e+06	1.114800e+06	68.767467	-57.426919

En el cuadro podemos visualizar que category_id está como numérico pero el id no puede ser procesado como un numérico por lo que lo pondremos como un categórico. Además, hay valores como '0' en likes, dislikes y comment_count, aun que muchos pueden ser porque ratings_disabled esta desactivado.

	video_id	trending_date	title	channel_title	publish_time	tags	thumbnail_link	description	state	geometry
count	40881	40881	40881	40881	40881	40881	40881	39585	40881	40881
unique	24427	205	24573	5076	23613	20157	24422	22345	13	13
top	6ZfuNTqbHE8	17.14.11	Most Popular Violin Covers of Popular Songs 20...	SET India	2017-12-20T23:00:00.000Z	[none]	https://i.ytimg.com/vi/VYOjWnS4cMY/default.jpg	Subscribers Link: http://bit.ly/2qb69dZ\n\nCon...	Quebec	POINT (-64.34799504 49.82257774)
freq	8	200	15	192	11	2385	8	130	3247	3247

Del cuadro, visualizamos que el video_id se esta comportando como un numerico pero al ser id no debe ser un numerico, por lo que cambiaremos eso. También, notamos que hay canales que se repiten varias veces porque de los 40 mil datos solo 5 mil son diferentes así que sería mejor ponerlo como categorico para ahorrar el espacio. Igual con tags, además para el análisis la descripción no será necesaria ya que no podriamos analizar nada con él.

- Cantidad y distribución de los valores nulos.

```
video_id                    0
trending_date               0
title                       0
channel_title               0
category_id                 0
publish_time                0
tags                        0
views                       0
likes                       0
dislikes                    0
comment_count               0
thumbnail_link              0
comments_disabled           0
ratings_disabled            0
video_error_or_removed      0
description              1296
state                       0
lat                         0
lon                         0
geometry                    0
```

Notamos que solo la descripción tiene valores nulos, aun que hay valores de 0 en likes, dislikes y comment_count porque en ratings_disabled y comments_disabled es 0.

Además, hay valores en tags que son [none] y son 2385 por lo que tendremos que analizar qué hacer con eso.

```
Tags como none: 2385
Comments Disabled: 583
Ratings Disabled: 279
Likes en 0: 284
Dislikes en 0: 393
Comment_count en 0: 646
```

Los valores son pocos por lo que no podría afectar si las vistas son mínimos, pero si son muchas sí puede afectar en algo al promedio por lo que tendremos que limpiarlo o simplemente no contar con esos datos.

- Inconsistencias:

```
Inconsistencias flags disabled:
Comments_disabled=True con comentarios >0: 0
Ratings_disabled=True con likes >0: 0
```

Por la imagen no vemos inconsistencias en los comentarios, en otras palabras sí está desactivado los comentarios no hay comentarios, así que cumple.

Estadísticas resumen:

	trending_date	views	likes
count	40881	4.088100e+04	4.088100e+04
mean	2018-02-27 05:29:22.882512640	1.147036e+06	3.958269e+04
min	2017-11-14 00:00:00	7.330000e+02	0.000000e+00
25%	2018-01-04 00:00:00	1.439020e+05	2.191000e+03
50%	2018-02-26 00:00:00	3.712040e+05	8.780000e+03
75%	2018-04-24 00:00:00	9.633020e+05	2.871700e+04
max	2018-06-14 00:00:00	1.378431e+08	5.053338e+06
std	NaN	3.390913e+06	1.326895e+05

	dislikes	comment_count	lat	lon
count	4.088100e+04	4.088100e+04	40881.000000	40881.000000
mean	2.009195e+03	5.042975e+03	52.025876	-88.817702
min	0.000000e+00	0.000000e+00	44.566645	-139.000002
25%	9.900000e+01	4.170000e+02	46.249282	-110.733329
50%	3.030000e+02	1.301000e+03	49.822578	-81.236083
75%	9.500000e+02	3.713000e+03	53.016698	-64.347995
max	1.602383e+06	1.114800e+06	68.767467	-57.426919
std	1.900837e+04	2.157902e+04	7.213076	25.119498

Notamos que la mediana y media de views, likes, dislikes y comment_count tienen una gran diferencia. Por lo que sé sabe que hay un gran ruido.

PREPARACIÓN DE LOS DATOS

Limpiar los datos

- Esta tarea es una de las que más tiempo y esfuerzo consume debido a la diversidad de técnicas que pueden aplicarse para optimizar la calidad de los datos.

De lo que hemos hablado anteriormente en el punto “Verificar la calidad de los datos”, hemos encontrados los posibles valores nulos. Vamos a eliminar la columna de Stage porque hay demasiados valores [none], también con respecto a los valores nulos de ‘likes’, ‘dislikes’ y ‘comment_count’ vemos que hay pocos valores con ‘0’ así que no vamos a cambiar esos valores pero

gracias a lo mean y mediana notamos que hay una gran diferencia y la mayoría es una desviación estándar hacia la izquierda, así que vamos a normalizarlo. Además, valores como la descripción no aportan mucho al ser cadenas de texto muy variados por lo que vamos a eliminarlo igual que tags que tiene varios valores none y es muy variado.

- Valores incoherentes

```
views: valores < 0 = 0
likes: valores < 0 = 0
dislikes: valores < 0 = 0
comment_count: valores < 0 = 0
```

No encontramos por ahora valores incoherentes, pero sí hay muchos valores atípicos, pero ese punto lo exploraremos mejor en el preprocesamiento de los datos.

PRE-PROCESAR LOS DATOS

- Verificar datos faltantes: Hemos decidido eliminar la columna de "descripción" porque por más que hagamos el proceso de reemplazar esos datos faltantes no nos ayudarían en el análisis, y también eliminamos tags porque son más de 20 mil tags diferentes y hay como 5 mil valores faltantes por lo que decidimos eliminarlo porque reemplazarlo con un valor sería difícil. Después de eso no hemos encontrado otros valores faltantes, aun que hay valores que son '0'.

```
Tags como none: 2385
Comments Disabled: 583
Ratings Disabled: 279
Likes en 0: 284
Dislikes en 0: 393
Comment_count en 0: 646
```

Por lo que, vamos a reemplazar los valores '0' de comment_count, likes y dislikes según si ratings_disabled y comments_disabled son True o False, ya que significa si los comentarios o reacciones de deshabilitaron.

```
Comments Disabled: 583
Ratings Disabled: 279
Likes en 0: 279
Dislikes en 0: 279
Comment_count en 0: 583
```

Al final nos queda como la imagen de arriba y significa que sé equilibró con ratings_disabled y comments_disabled, porque es raro que un video en

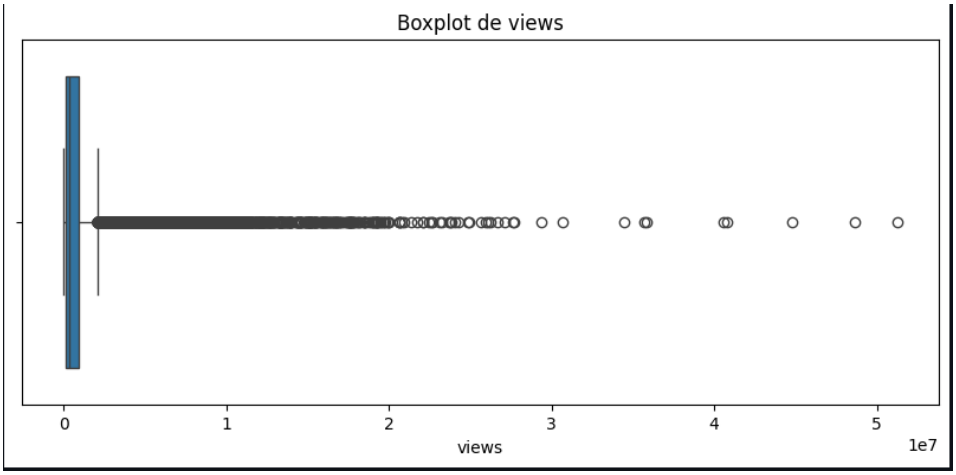
tendencia no tenga racciones si ratings_disabled y comments_disabled están activados.

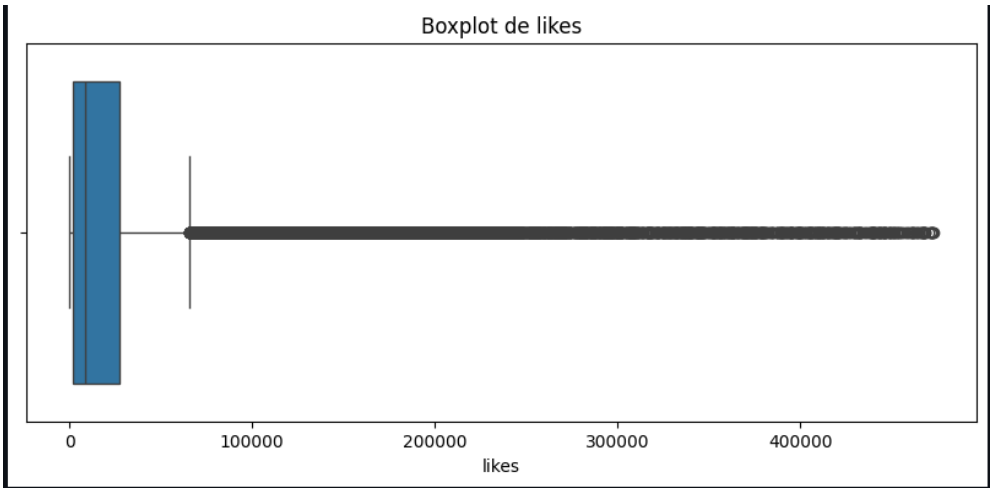
También, eliminamos columnas innecesarias.

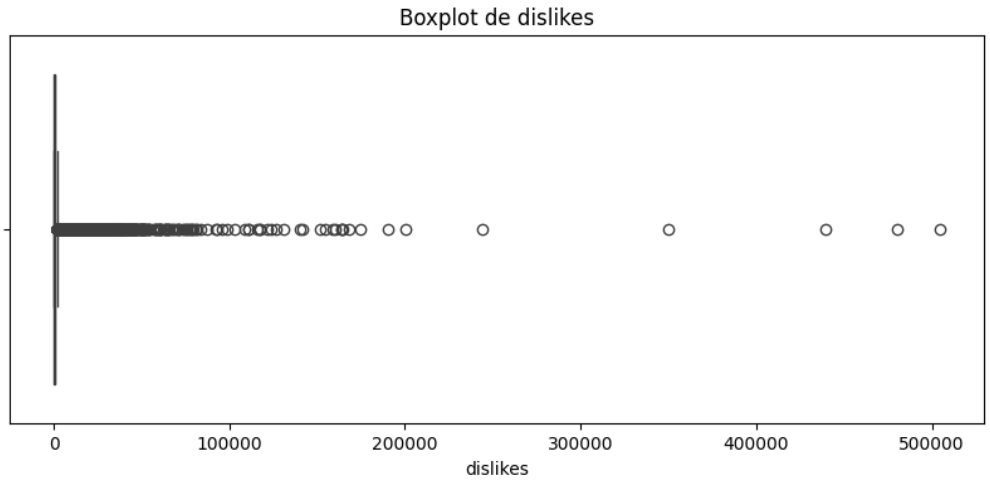
```
# Eliminas columnas innecesarias
df.drop(['description', 'tags'], axis=1, inplace=True)
```

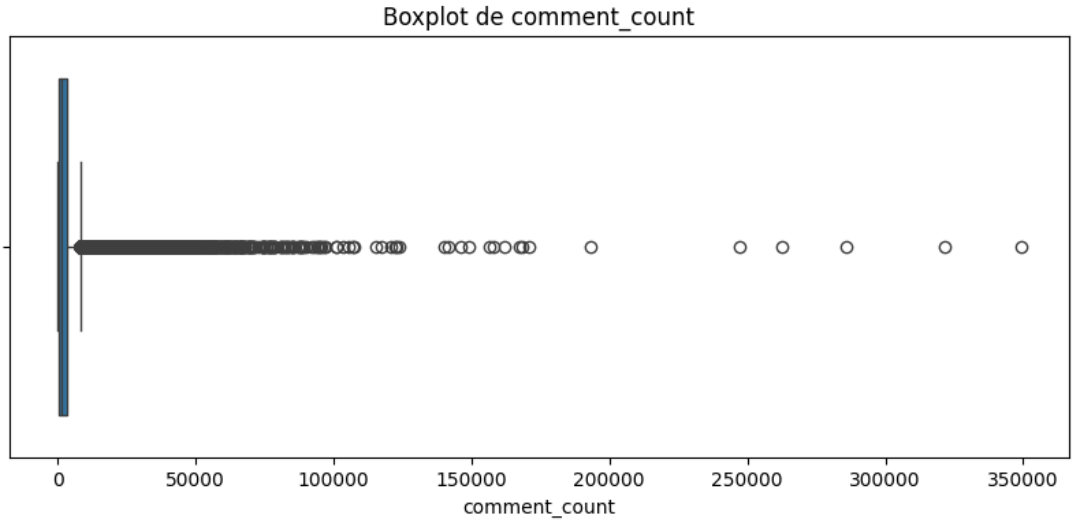
-Identificación de los datos atípicos u outliers:

- Identificación de los valores atípicos.









	Variable	Q1	Q3	IQR	Lower Limit	Upper Limit	\
0	views	142372.75	933691.75	791319.00	-1044605.750	2120670.250	
1	likes	2159.00	27630.25	25471.25	-36047.875	65837.125	
2	dislikes	97.00	915.00	818.00	-1130.000	2142.000	
3	comment_count	411.00	3597.25	3186.25	-4368.375	8376.625	

	Outliers Below	Outliers Above	Total Outliers
0	0	4225	4225
1	0	4810	4810
2	0	5107	5107
3	0	4594	4594

Para datos reales y sabiendo que son videos virales de youtube en Canada, podemos ver que no hay casi valores atipicos, tal vez podamos limpiar un poco los dislikes ya que hay valores atípicos muy separados de los demás y números muy grandes, así que vamos a aplicar un método de tratar con valores atípicos y se transformará en una nueva columna.

- Capping / Winsorization:

Se aplicó Winsorización para suavizar valores extremos en las métricas de interacción (likes, dislikes y comentarios). Los percentiles intercuartílicos se mantuvieron, garantizando la representatividad de la muestra, mientras que se redujo la dispersión extrema que podría distorsionar el modelado.

Y quedamos con:

```
=== Likes ===
Original:
count    4.088100e+04
mean     3.958280e+04
std      1.326895e+05
min      0.000000e+00
25%      2.192000e+03
50%      8.780000e+03
75%      2.871700e+04
max      5.053338e+06
Name: likes, dtype: float64

Sin atípicos:
count    4.088100e+04
mean     3.826275e+04
std      1.172519e+05
min      0.000000e+00
25%      2.192000e+03
50%      8.780000e+03
75%      2.871700e+04
max      4.796235e+06
Name: likes_sin_atipico, dtype: float64

=== Dislikes ===
Original:
count    4.088100e+04
mean     2.009228e+03
std      1.900837e+04
```

```
min       0.000000e+00
25%       9.900000e+01
50%       3.030000e+02
75%       9.500000e+02
max       1.602383e+06
Name: dislikes, dtype: float64

Sin atípicos:
count     40881.000000
mean       1568.206575
std        5547.812661
min           0.000000
25%          99.000000
50%         303.000000
75%         950.000000
max      243458.000000
Name: dislikes_sin_atipico, dtype: float64

=== Comment Count ===
Original:
count    4.088100e+04
mean     5.043967e+03
std      2.157888e+04
min      0.000000e+00
25%      4.180000e+02
50%      1.302000e+03
75%      3.714000e+03
max      1.114800e+06
Name: comment_count, dtype: float64

Sin atípicos:
count     40881.000000
mean       4637.924072
std       13894.922735
min           0.000000
25%         418.000000
50%        1302.000000
```

```
75%       3713.000000
max     615889.000000
```

	Column	Unique Values	Rare Values (# freq = 1)	Rare Values (%)	Most Frequent Value	Top Value Freq
0	video_id	24427	14516	59.43	6ZfuNTqbHE8	8
1	title	24573	14758	60.06	Most Popular Violin Covers of Popular Songs 20...	15
2	channel_title	5076	1694	33.37	SET India	192
3	category_id	17	0	0.00	24	13451
4	thumbnail_link	24422	14507	59.40	https://i.ytimg.com/vi/VYOjWnS4cMY/default.jpg	8
5	state	13	0	0.00	Quebec	3247
6	geometry	13	0	0.00	POINT (-64.34799504 49.82257774)	3247

Con respecto a estos valores no habría valores atípicos, es muy importante quedarnos con esos canales que solo aparecen una vez para nuestro modelo de predicción.

Construir nuevos datos

- Creamos nuevas columnas y eliminamos otra.
    - Medicion_interaccion: Esta columna puede ser la relación de los likes/dislikes

```
# Crear la nueva columna
df['likes_dislikes_ratio'] = df['likes'] / df['dislikes']
```

    - Eliminamos los thumbnail_link porque no es necesario.

```
df = df.drop('thumbnail_link', axis=1)
```

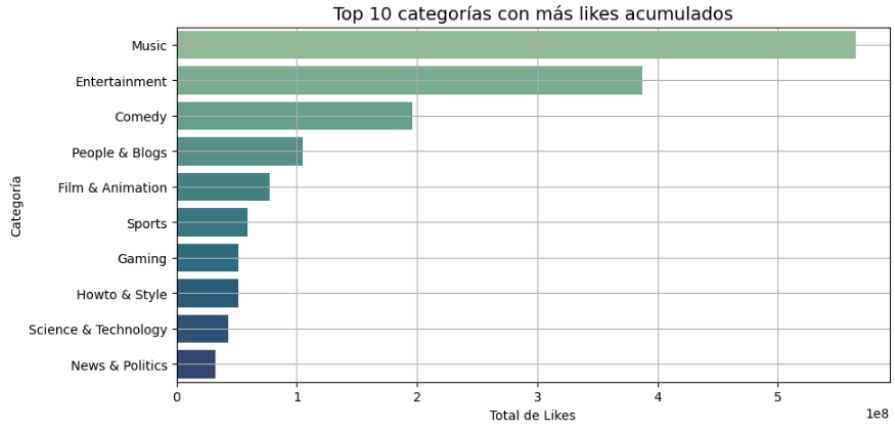
REQUERIMIENTOS

▪ Por Categoría de Videos

1. ¿Qué categorías de videos son las de mayor tendencia?

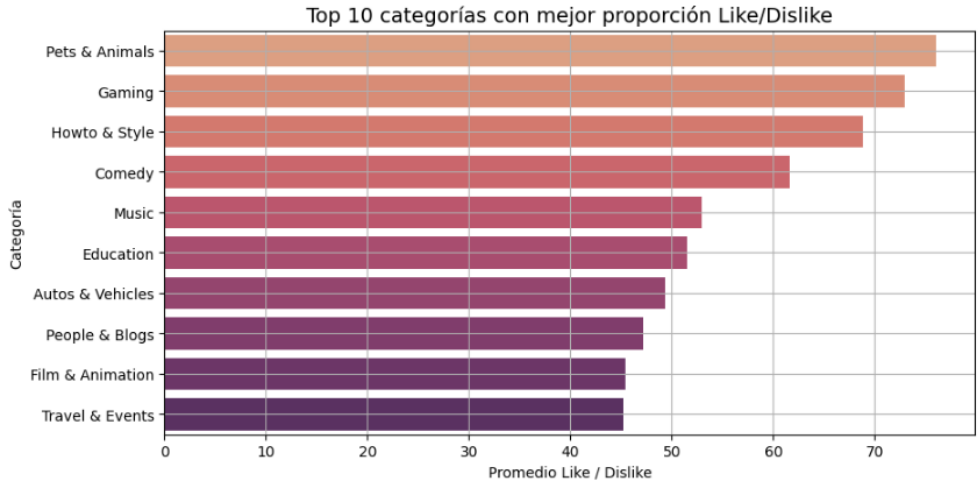
Se observa que la categoría "Entertainment" es la que cuenta con mayor cantidad de videos en tendencia, seguida por "News & Politics", "People & Blogs" y "Comedy". Esto refleja una alta preferencia del público por contenidos orientados al entretenimiento, la actualidad y experiencias personales, que tienden a captar mayor atención y generar interacción.

2. ¿Qué categorías de videos son los que más gustan? ¿Y las que menos gustan?



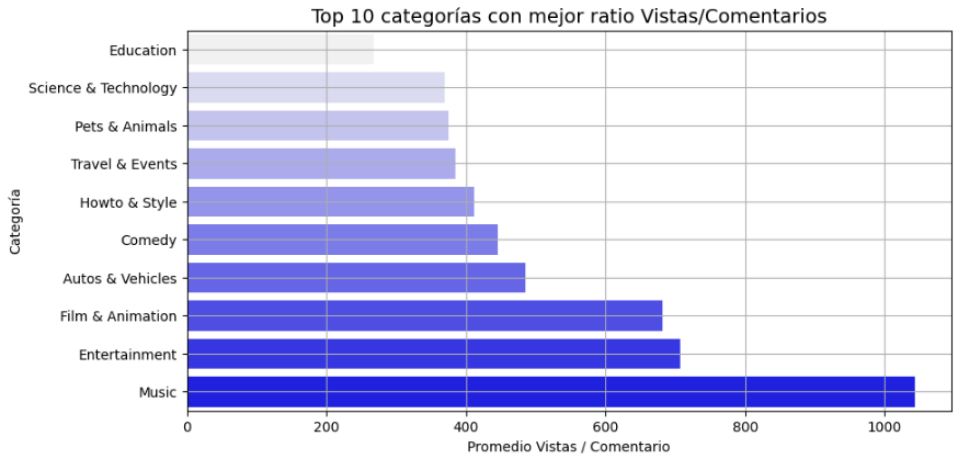
La categoría “Music” es la que acumula mayor cantidad de likes, seguida por “Entertainment” y “Comedy”. Esto indica que los contenidos musicales y de entretenimiento generan una alta respuesta positiva del público. En contraste, categorías como “Science & Technology” y “News & Politics” muestran un menor total de likes.

3. ¿Qué categorías de videos tienen la mejor proporción (ratio) de “Me gusta” / “No me gusta”?



Las categorías con mejor proporción de "Me gusta" / "No me gusta" son "Pets & Animals", "Gaming" y "Howto & Style". Esto indica que, en promedio, los videos de estas categorías generan una respuesta mayoritariamente positiva por parte del público. Un ratio alto sugiere que los espectadores valoran este tipo de contenido y lo reciben con poca crítica (menos dislikes).

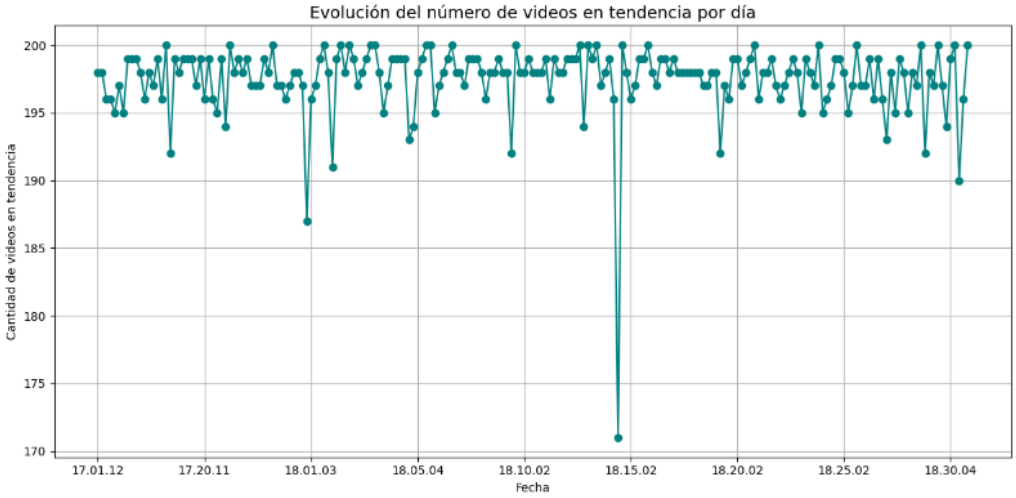
4. ¿Qué categorías de videos tienen la mejor proporción (ratio) de "Vistas" / "Comentarios"?



En este gráfico se observa que las categorías "Music", "Entertainment" y "Film & Animation" tienen un mayor promedio de vistas por comentario. Esto quiere decir que, aunque estos videos son muy vistos, generan menos participación en forma de comentarios. En cambio, categorías como "Education" o "Science & Technology" tienen una relación más baja, lo que podría reflejar que su audiencia tiende a interactuar más con el contenido.

▪ Por el tiempo transcurrido

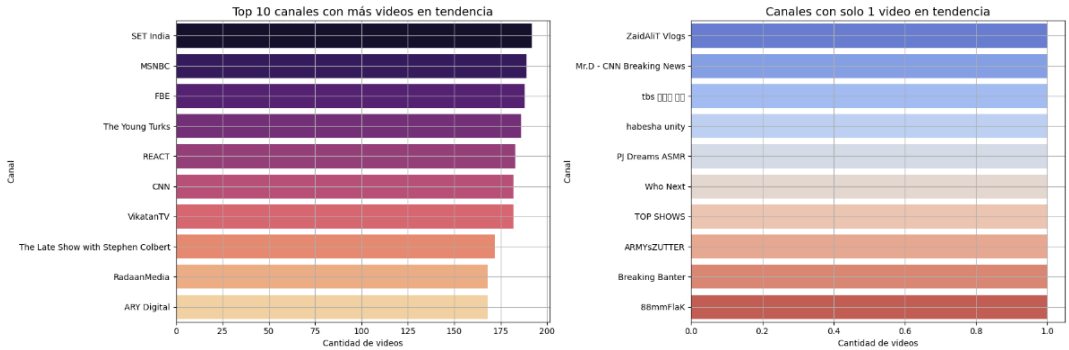
5. ¿Cómo ha cambiado el volumen de los videos en tendencia a lo largo del tiempo?



El gráfico muestra que el número de videos en tendencia por día se ha mantenido relativamente estable a lo largo del tiempo, con ligeras variaciones. En la mayoría de fechas se registra un volumen cercano a los 200 videos, aunque también se observan caídas puntuales. En general, no se identifican cambios drásticos en la cantidad diaria, lo que sugiere un comportamiento bastante constante en la plataforma durante el periodo analizado.

▪ Por Canales de YouTube

6. ¿Qué canales de YouTube son tendencia más frecuentemente? ¿Y cuáles con menos frecuencia?

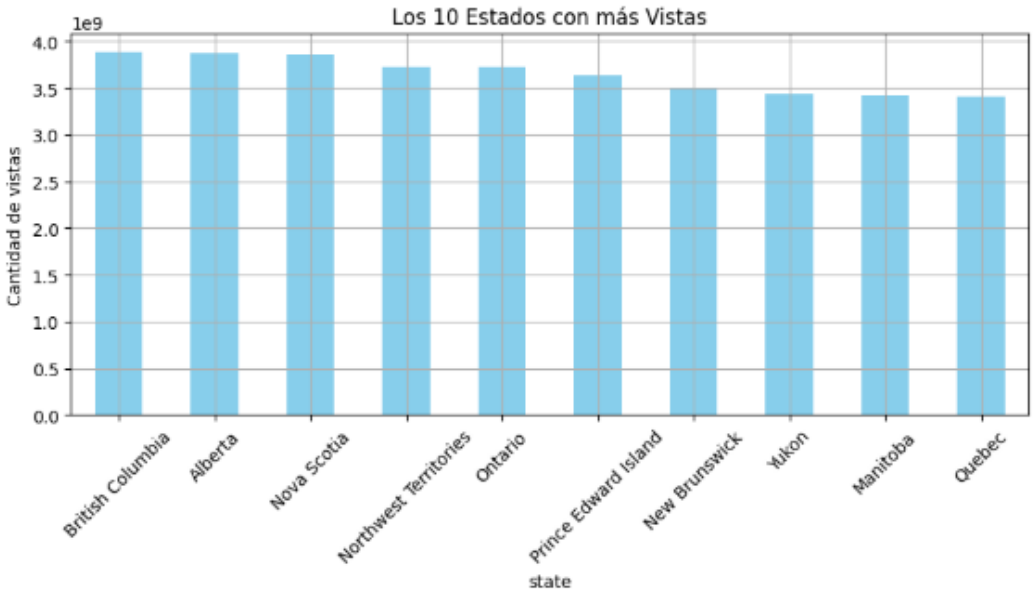


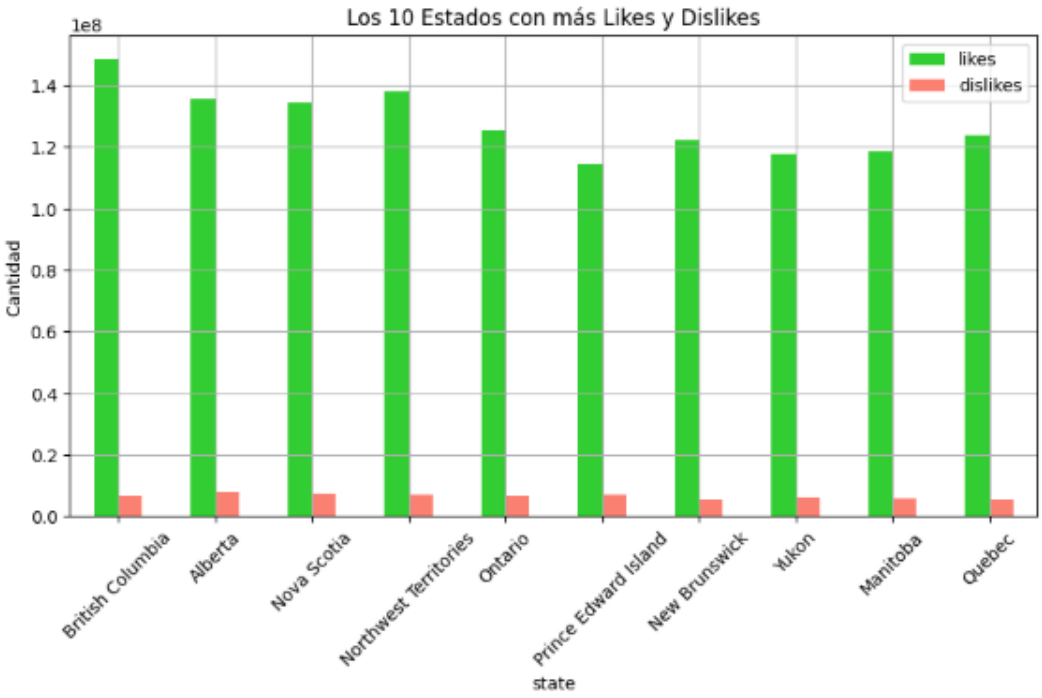
El gráfico de la izquierda muestra que canales como “SET India”, “MSNBC” y “FBE” aparecen con mayor frecuencia en las tendencias, lo que refleja una producción constante de contenido exitoso o una fuerte presencia en la plataforma. En cambio, el gráfico de la derecha muestra canales con solo una

aparición en tendencia, como “Jaziel Vlogs”, “Mr.D - DnD Healing items”, “TED-Ed EDU”, “Alvaro Carrillo” y “Elemental”.

▪ Por la geografía del país

7. ¿En qué Estados se presenta el mayor número de “Vistas”, “Me gusta” y “No me gusta”?



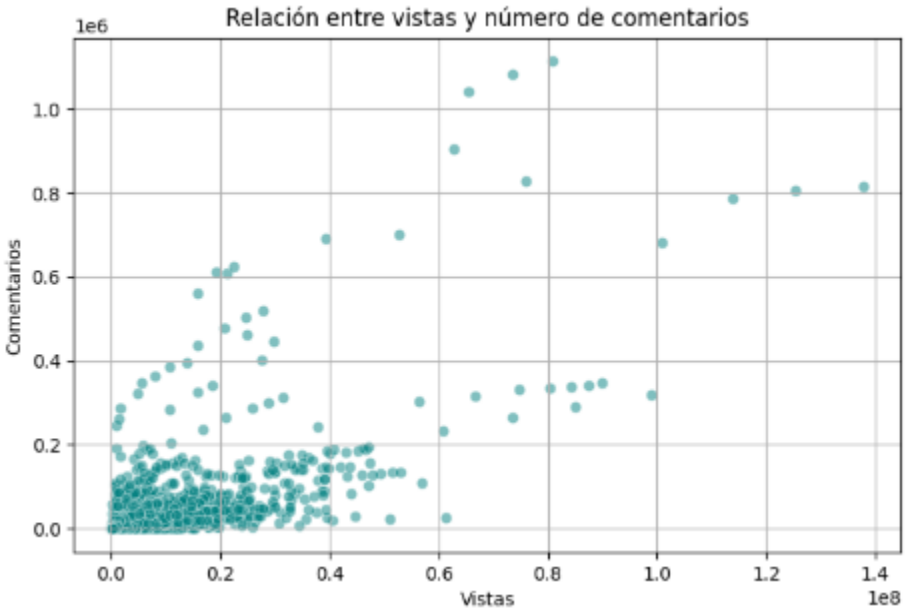


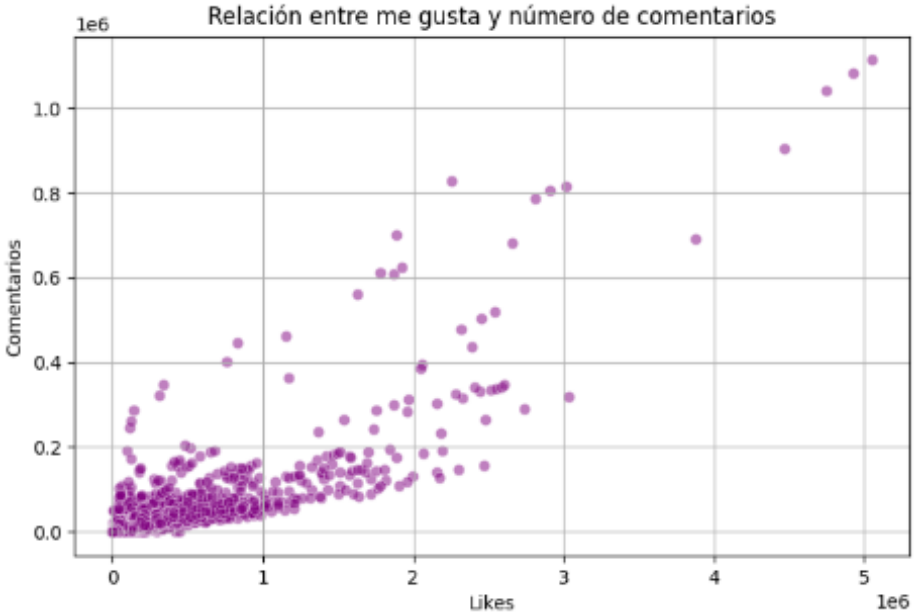
Los gráficos muestran que British Columbia, Alberta y Nova Scotia son los estados con mayor número de vistas, likes y dislikes acumulados. En particular,

British Columbia destaca como el estado con la mayor cantidad de vistas y también lidera en interacciones positivas (likes) y negativas (dislikes). Esto indica una fuerte presencia y consumo de contenido en esas regiones. En cambio, estados como Québec, Manitoba y Yukon se ubican en los últimos lugares del top 10, con cifras significativamente menores, aunque aún representativas dentro del volumen total del país.

▪Adicionalmente, al cliente le gustaría conocer si:

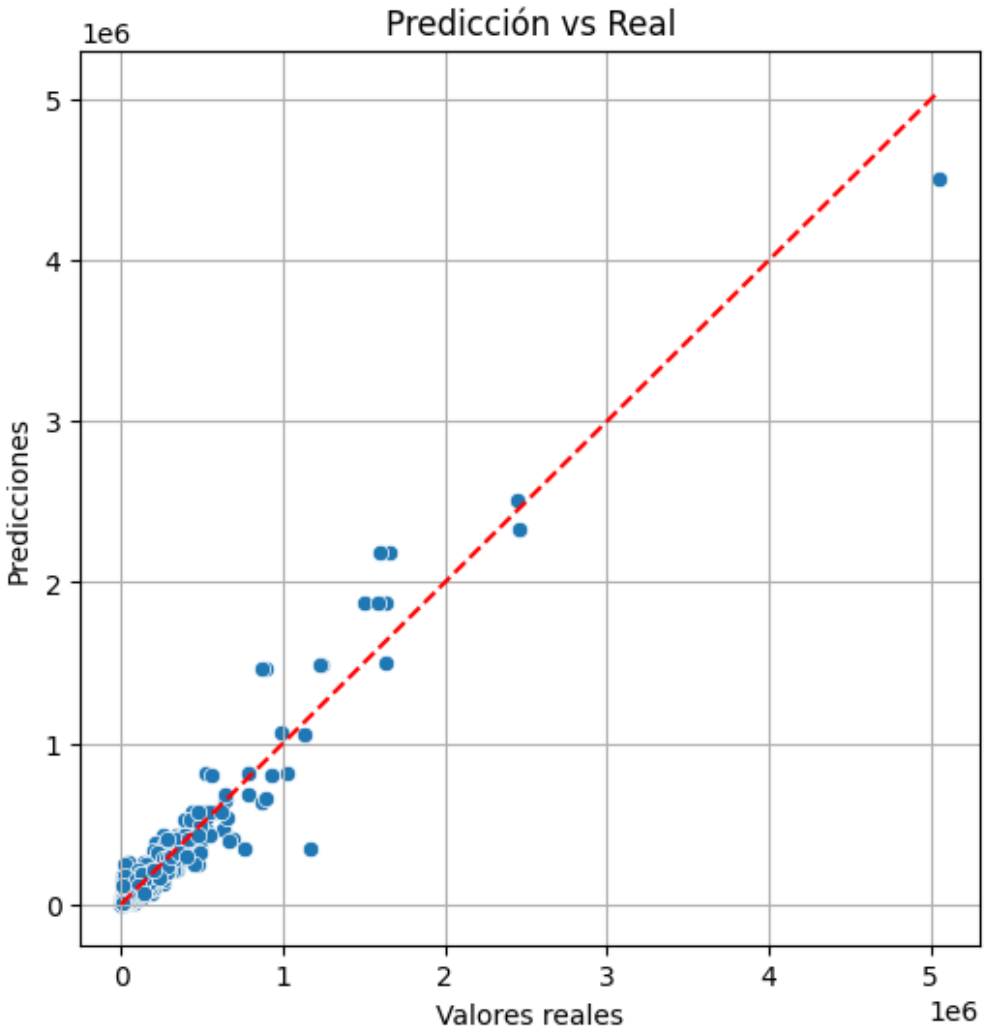
8. ¿Los videos en tendencia son los que mayor cantidad de comentarios positivos reciben?





Se realizó un análisis de correlación entre las variables comment_count, views y likes. Se observó que los videos con mayor número de vistas y “me gusta” tienden a tener también mayor cantidad de comentarios, lo cual permite inferir que los videos en tendencia suelen recibir una mayor cantidad de comentarios positivos.

9. ¿Es factible predecir el número de “Vistas” o “Me gusta” o “No me gusta”?



Este gráfico muestra que el modelo predice bastante bien la cantidad de “me gusta” que recibirá un video. La mayoría de los puntos están cerca de la línea roja, lo que significa que los valores reales y los predichos son parecidos. Esto indica que el modelo funciona bien, sobre todo en videos con pocas o medianas cantidades de likes. Además, al analizar junto con otras variables como vistas y comentarios, se confirma que cuando un video tiene más vistas,

también suele tener más likes y comentarios, lo que refleja un mayor interés del público.

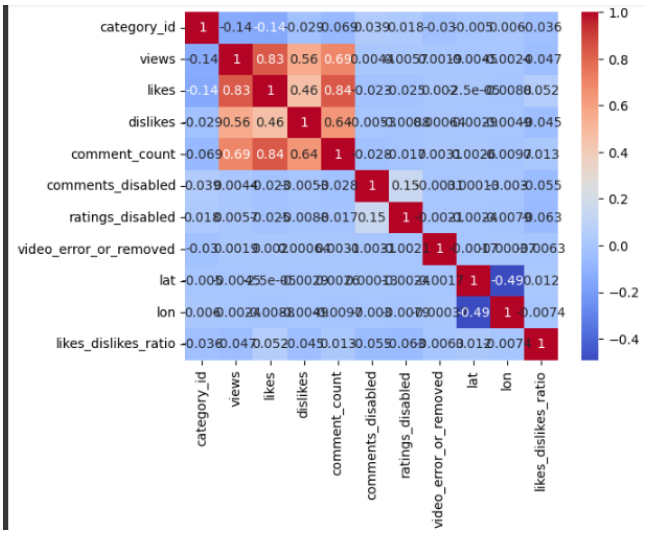
MODELIZAR Y EVALUAR LOS RESULTADOS

Nuestro objetivo es predecir la cantidad de likes que tendrá un vídeo, así que usaremos como técnica de Data Mining el árbol de regresión múltiple.

Construir el modelo:

Hacemos un análisis de correlación:

```
df_num = df.select_dtypes(include=['float64', 'int64', 'bool'])
sns.heatmap(df_num.corr(), cmap='coolwarm', annot=True)
```



Definimos las variables:

```
scaler = MinMaxScaler()
df['views_scaled'] = scaler.fit_transform(df[['views']])
df['comment_count_scaled'] = scaler.fit_transform(df[['comment_count']])

X = df[['views_scaled', 'comment_count_scaled', 'channel_title']]
y = df['likes']
```

Calculamos el target encoding solo con X_train:

```
channel_mean_views =
X_train.join(y_train).groupby('channel_title')['likes'].mean()
Aplicamos el encoding:
```

```
X_train['channel_title_encoded'] =
X_train['channel_title'].map(channel_mean_views)
X_test['channel_title_encoded'] =
X_test['channel_title'].map(channel_mean_views)
Rellenamos los valores desconocidos con el valor promedio de likes global de
X_train:
X_test['channel_title_encoded'].fillna(y_train.mean(), inplace=True)
Escalamos y utilizamos la versión escalada:
scaler_channel = MinMaxScaler()
X_train['channel_title_encoded_scaled'] = scaler_channel.fit_transform(
    X_train[['channel_title_encoded']]
)
X_test['channel_title_encoded_scaled'] = scaler_channel.transform(
    X_test[['channel_title_encoded']]
)
X_train = X_train[['views_scaled', 'comment_count_scaled',
'channel_title_encoded_scaled']]
X_test = X_test[['views_scaled', 'comment_count_scaled',
'channel_title_encoded_scaled']]
```

Entrenamos el modelo:

```
tree_model = DecisionTreeRegressor(random_state=42, max_depth=10,
min_samples_split=10)
tree_model.fit(X_train, y_train)
```

Una vez hecho todo eso, pasamos al modelado de la predicción para calcular la altura del arbol para obtener una buena predicción.

```
param_grid = {
    'max_depth': [4, 6, 8, 10, 12, 15,20],
    'min_samples_split': [2, 5, 10, 20]
}
# Define el modelo base
tree = DecisionTreeRegressor(random_state=42)

# Define GridSearchCV
grid_search = GridSearchCV(estimator=tree,
                           param_grid=param_grid,
                           cv=5,              # 5-fold cross-validation
                           scoring='r2',      # Métrica que optimizas
                           n_jobs=-1)         # Usa todos los núcleos disponibles
```

```
# Ajusta con tus datos de entrenamiento
grid_search.fit(X_train, y_train)
```

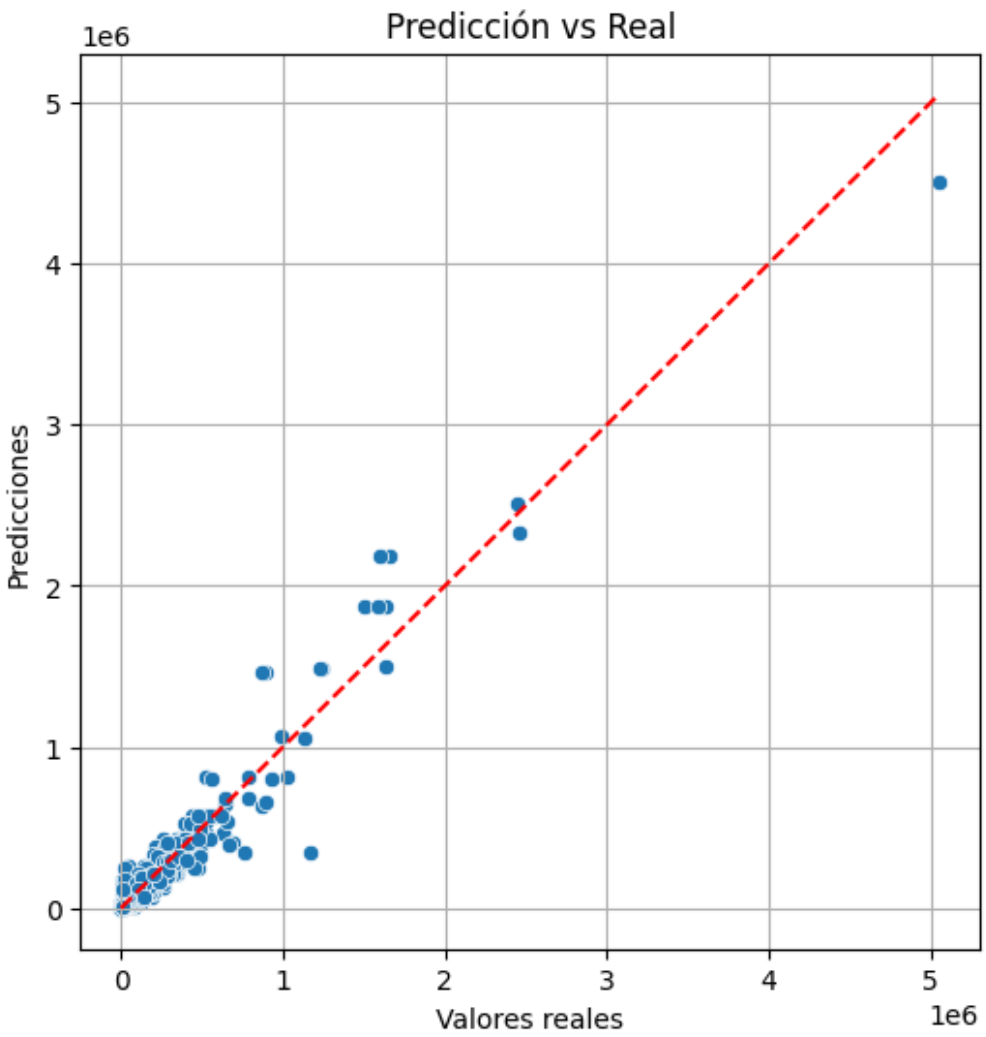
```
plt.figure(figsize=(6,6))
sns.scatterplot(x=y_test, y=y_pred)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')  # Línea ideal
plt.xlabel('Valores reales')
plt.ylabel('Predicciones')
plt.title('Predicción vs Real')
plt.grid(True)
plt.show()
```

```
Mejores parámetros encontrados: {'max_depth': 10, 'min_samples_split': 5}
Mejor score R² promedio: 0.9468484234542558
R²: 0.9629666735615383
MAE: 10086.454606282461
RMSE: 27677.208957715797
```

El cuadro de predicción es la siguiente:

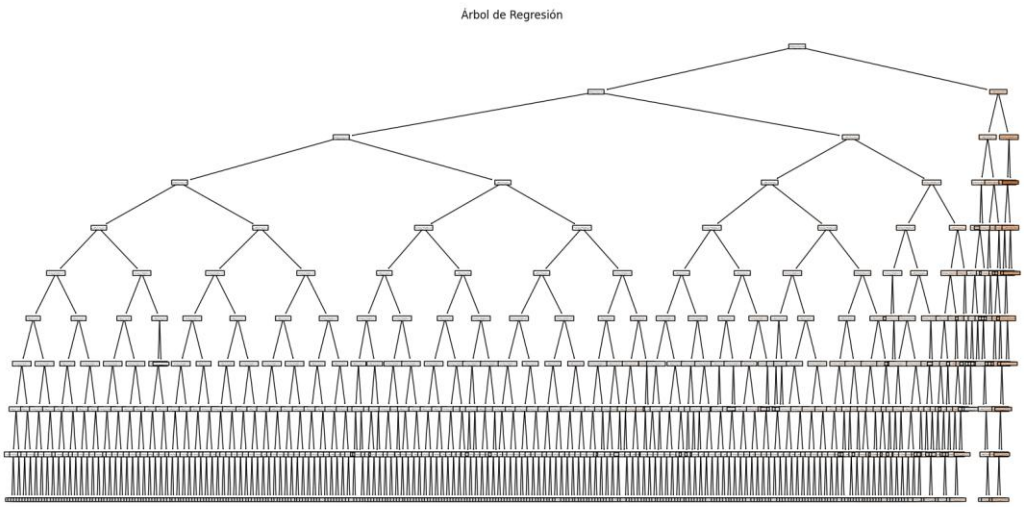
	views_scaled	comment_count_scaled	channel_title_encoded_scaled	Likes	Pred
3719	0.011772	0.000741	0.015911	17553	29461.932203
36535	0.000529	0.000070	0.000052	111	199.190899
15981	0.003007	0.017329	0.009727	35691	23321.194245
11810	0.005832	0.001252	0.013984	17493	30995.655556
15278	0.001717	0.001105	0.001833	5536	5043.962433
18210	0.005287	0.000667	0.002756	6257	5043.962433
15503	0.000149	0.000722	0.000492	2453	861.160823
13702	0.000606	0.000282	0.015667	263	4085.587838
40626	0.001821	0.001409	0.000979	3352	2238.843278
3102	0.000370	0.000059	0.000260	311	861.160823
11527	0.000808	0.000240	0.001995	5583	2968.590348
25862	0.000219	0.000152	0.000353	265	861.160823
4508	0.000640	0.003246	0.011432	25599	23321.194245
6366	0.000075	0.000000	0.000169	29	432.810573
16637	0.000116	0.000061	0.003022	1005	1894.856061
16133	0.000743	0.001196	0.004974	7544	10531.587112

Grafico de predicción vs Real:



Y a la visualización del árbol

```
plt.figure(figsize=(20, 10))
plot_tree(tree_model, feature_names=X_test.columns, filled=True,
rounded=True)
plt.title("Árbol de Regresión")
plt.show()
```



Evaluación del modelo:

```
# calcular estas métricas
print('MAE:', metrics.mean_absolute_error(y_test, y_pred))
print('MSE:', metrics.mean_squared_error(y_test, y_pred))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

```
MAE: 10058.805111412368
MSE: 754259437.636193
RMSE: 27463.784109918157
```

Los resultados nos idican que el modelo tiene un rendimiento razonable, aunque los errores pueden ser grandes a comparación con la cantidad de likes donde puede ser hasta 100 mil, sigue siendo buena para tener una aproximación.

```
# Precisiòn del Modelo (Coeficiente de determinación)
print("Precisión del Modelo: ",metrics.r2_score(y_test,y_pred))
```

```
Precisión del Modelo:  0.963535615177375
```

Como observamos, la precisión es buena, ya que este valor esta cercano a la unidad.

Según el resultado del análisis existe una tendencia general donde los videos con mayor cantidad de visualizaciones, mayor cantidad de comentarios y un

canal popular, tienden a registrar más likes. Esto significa que la interacción de los usuarios actúa como un indicador clave para predecir el éxito de un video, ya que los videos que generan reacciones suelen ser impulsados por los algoritmos de las plataformas, alcanzando a más usuarios.

I. CONCLUSIONES

- El proyecto permitió comprender cómo los datos generados por YouTube en Canadá pueden ser analizados para identificar patrones de popularidad, interacción y comportamiento del público en videos en tendencia.
- Se evidenció que categorías como “Entertainment”, “Music” y “Comedy” concentran gran parte del contenido viral, y que variables como vistas, comentarios y canal de publicación influyen directamente en la cantidad de likes que recibe un video.
- Se aplicó un modelo de árbol de regresión para predecir el número de "me gusta", utilizando como variables predictoras: vistas escaladas, comentarios escalados y codificación del canal. El modelo mostró buen desempeño con una precisión (R^2) cercana a 1 y errores aceptables dados los valores extremos de likes que pueden superar las 100,000 interacciones.
- A través del análisis, se confirmó una relación positiva entre vistas, comentarios y likes, indicando que cuanto mayor es la interacción del público, mayor es la probabilidad de éxito del video. Esta información puede ser valiosa para marcas, creadores de contenido y agencias que deseen optimizar su estrategia en plataformas digitales.
- Como equipo, se reforzó la importancia de iterar entre fases del proceso CRISP-DM. Las decisiones de limpiar, transformar y escalar los datos fueron claves para mejorar el rendimiento del modelo. Se concluye que un enfoque colaborativo y metodológico en Data Science puede generar conocimiento útil para la toma de decisiones basada en datos reales.

II. ANEXOS

Link del github:
https://github.com/KeniChris/FDS2025-1-258

III. BIBLIOGRAFÍA