**UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS**

Administración de la Información

Trabajo Final

Sección: CC51


INTEGRANTES:

Chávez Arias, Bill Brandon - U20171C042

Peralta Ireijo, Sebastian Fernando - U201816030

Costa Morales, Juan Josemaria - U201822001

PROFESORA:

Reyes Silva, Patricia Daniela


Julio, 2021-1

# 1. Objetivos del proyecto

Como objetivo tenemos desarrollar un proyecto de analitica con la finalidad de descubrir información relevante sobre los videos que son tendencia en YouTube en Francia. Ya que el cliente es una importante empresa de marketing digital, el cual desea obtener respuestas a varios requerimientos de información que se detallan a continuación.

**Por Categoría de Videos:**

1. ¿Qué categorías de videos son las de mayor tendencia?
2. ¿Qué categorías de videos son los que más gustan? ¿Y las que menos gustan?
3. ¿Qué categorías de videos tienen la mejor proporción (ratio) de “Me gusta” / “No me gusta”?
4. ¿Qué categorías de videos tienen la mejor proporción (ratio) de “Vistas” / “Comentarios”?

**Por el tiempo transcurrido:**

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

**Por Canales de YouTube:**

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

**Por la geografía del país:**

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

**Adicionalmente, al cliente le gustaría conocer si:**

¿Es factible predecir el número de “Vistas” o “Me gusta” o “No me gusta”?
¿Los videos en tendencia son los que mayor cantidad de comentarios positivos reciben?

# 2. Caso de análisis

Los datos con los que se trabajó en este proyecto consisten en información sobre videos virales en la conocida plataforma YouTube. Incluye múltiples meses de data sobre los videos virales que surgen a diario en la plataforma en varios países. Por ejemplo: el título del vídeo, cantidad de vistas, cantidad de likes, etc. Para este trabajo se trabajará exclusivamente con la data perteneciente a Francia. Según Kaggle, la data a utilizar tiene como autor a Mitchell, J. y fue publicada el 13 de noviembre del 2017. La última vez que fue actualizada fue el 2 de junio del 2019.

El estudio de esta información, permite observar las características que cumplen los videos que se vuelven virales en un país. De tal manera que, se puede aprender sobre qué clase de videos son los que son virales con más frecuencia. Esta información le es de interés a todo aquel que utilice Youtube como creador. Por ejemplo, a un director de videos de música le puede interesar qué temática de videos suelen ser viral con más frecuencia para adaptar sus futuros proyectos a la tendencia. En resumen, este estudio beneficiará a aquellos que quieran averiguar sobre las cualidades de los videos virales que surgen a diario en la plataforma.

## 3. Conjunto de datos

| # | Nombre | Tipo | Descripción | Valores |
|---|---|---|---|---|
| 1 | video_id | Object | Identificador único para los videos, cuenta con caracteres alfanuméricos y guiones | 783GObTk-I4 8L3bSNzTVhQ PKBfGeMSTb4 Cu_I5qtYqaw |
| 2 | trending_date | Object | Muestra la fecha cuando se alcanzó trending | 2017-14-11 (yyyy-mm-dd) |
| 3 | title | Object | Muestra el título del video | “vu_du10022018. mov” “VENOM - Official Teaser Trailer (HD)” |
| 4 | channel_title | Object | Muestra el nombre del Canal que publicó el video | “Sony Pictures Entertainment” “Shahid TV” “Mai-Lie” |
| 5 | category_id | Int64 | Identificador de la categoría del video | 1 - 44 |
| 6 | publish_time | Object | Fecha de publicación | 2017-14-11 (yyyy-mm-dd) |
| 7 | tags | Object | Etiquetas de los videos | “senegal”\|”video” \|... |
| 8 | views | Int64 | Muestra la cantidad de vistas | 223 - 101m |
| 9 | likes | Int64 | Muestra la cantidad de likes | 0 - 4.75m |
| 10 | dislikes | Int64 | Muestra la cantidad de dislikes | 0 - 1.35m |
| 11 | comment_count | Int64 | Muestra la cantidad de comentarios | 0 - 1.04m |

| | | | | |
|---|---|---|---|---|
| 12 | thumbnail_link | Object | Muestra el enlace para miniatura | “https://…” |
| 13 | comments_disabled | bool | Muestra los comentarios deshabilitados | True - False |
| 14 | ratings_disabled | bool | Muestra las clasificaciones deshabilitadas | True - False |
| 15 | video_error_or_removed | bool | Muestra si el video a sido eliminado | True - False |
| 16 | description | Factor | Muestra la descripción del video | “Une nouvelle dose de dessins...”, ... |
| 17 | state | Factor | Muestra nombre del estado del país | Alsace, Corse, Moyotte, ... |
| 18 | lat | float64 | Muestra la latitud geográfica de la ubicacion del Estado | -21 - 50 |
| 19 | lon | float64 | Muestra la longuitud geografica de la ubicacion del Estado | -62 - 56 |
| 20 | geometry | Factor | Muestra el registro de coordenadas del Estado dentro del Planeta | POINT(-0.56526 527 47.48651651) |

# 4. Análisis exploratorio de los datos

- ## Cargar los datos

  La carga del conjunto de datos a utilizar en el trabajo se utilizó la siguiente función: `Frvids = pd.read_csv('FRvideos_cc50_202101.csv')`. El contenido de dicha función tendrá que adaptarse a la ubicación del archivo .csv en el equipo.

- ## Inspeccionar los datos

  Tras cargar los datos es posible observar las características de la data con la que se va a trabajar. Inicialmente, se puede visualizar algunas de las primeras filas para explorar su contenido con la función `Frvids.head()`:

| | video_id | trending_date | title | channel_title | category_id | publish_time | tags | views | likes | dislikes | cc |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Ro6eob0LrCY | 17.14.11 | Malika LePen : Femme de Gauche - Trailer | Le Raptor Dissident | 24 | 2017-11-13T17:32:55.000Z | Raptor"\|"Dissident"\|"Expliquez"\|"moi"\|"cette"\|... | 212702 | 29282 | 1108 | |
| 1 | Yo84eqYwP98 | 17.14.11 | LA PIRE PARTIE ft Le Rire Jaune, Pierre Croce,... | Le Labo | 24 | 2017-11-12T15:00:02.000Z | [none] | 432721 | 14053 | 576 | |
| 2 | ceqntSXE-10 | 17.14.11 | DESSINS ANIMÉS FRANÇAIS VS RUSSES 2 - Daniil... | Daniil le Russe | 23 | 2017-11-13T17:00:38.000Z | cartoon"\|"pokémon"\|"école"\|"ours"\|"мультфильм | 482153 | 76203 | 477 | |
| 3 | WuTFI5qftCE | 17.14.11 | PAPY GRENIER - METAL GEAR SOLID | Joueur Du Grenier | 20 | 2017-11-12T17:00:02.000Z | Papy grenier"\|"Metal Gear Solid"\|"PS1"\|"Tirage... | 925222 | 85016 | 550 | |
| 4 | ee6OFs8TdEg | 17.14.11 | QUI SAUTERA LE PLUS HAUT ? (VÉLO SKATE ROLLER ... | Aurelien Fontenoy | 17 | 2017-11-13T16:30:03.000Z | vélo"\|"vtt"\|"bmx"\|"freestyle"\|"bike"\|"mtb"\|"di... | 141695 | 8091 | 72 | |

Otra opción es la función info(), la cual nos mostrará detalles sobre cada columna del conjunto de datos como su nombre, cantidad de Nulos y el tipo de dato.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40724 entries, 0 to 40723
Data columns (total 21 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   video_id                40724 non-null  object
 1   trending_date           40724 non-null  object
 2   title                   40724 non-null  object
 3   channel_title           40724 non-null  object
 4   category_id             40724 non-null  int64
 5   publish_time            40724 non-null  object
 6   tags                    40724 non-null  object
 7   views                   40724 non-null  int64
 8   likes                   40724 non-null  int64
 9   dislikes                40724 non-null  int64
 10  comment_count           40724 non-null  int64
 11  thumbnail_link          40724 non-null  object
 12  comments_disabled       40724 non-null  bool
 13  ratings_disabled        40724 non-null  bool
 14  video_error_or_removed  40724 non-null  bool
 15  description             37812 non-null  object
 16  state                   40724 non-null  object
 17  lat                     40724 non-null  float64
 18  lon                     40724 non-null  float64
 19  geometry                40724 non-null  object
 20  category_title          40610 non-null  object
dtypes: bool(3), float64(2), int64(5), object(11)
memory usage: 5.7+ MB
```

- Pre-Procesamiento de datos

- Título de categoría (Json) y conversión de formato de *trending_date* y *publish_time*

  Antes de realizar la limpieza del conjunto de datos es necesario incluir la data de la categoría recuperada del archivo .json (FR_category_id.json). La extracción de la data del archivo .json se hizo de la siguiente manera:

```
with open("FR_category_id.json") as file:
    jdata = json.load(file)
```

En la variable jdata se almacena todo el contenido del .json permitiendo manipularla. Paso seguido, se procedió a agregar una columna *category_title* al Data Frame, evaluando que el título de categoría se añada dependiendo del category_id de la fila correspondiente. (Proceso detallado en el .pynb de Pre-Procesamiento https://github.com/SebsPER/EB-2021-1-CC51/blob/main/code/Carga-Inspeccion-PreProcesamiento.ipynb )

Luego, también se requiere que el formato de las columnas *trending_date* y *publish_time* sea estandarizado a yyyy-mm-dd, ya que en el inicio tienen formatos diferentes para representar una fecha. Encima de eso, se pide que las columnas tengan un tipo de dato datetime, a diferencia del tipo object que tienen al cargar el archivo .csv.

El resultado de la adición y manipulación de columnas del Data Frame es el siguiente:

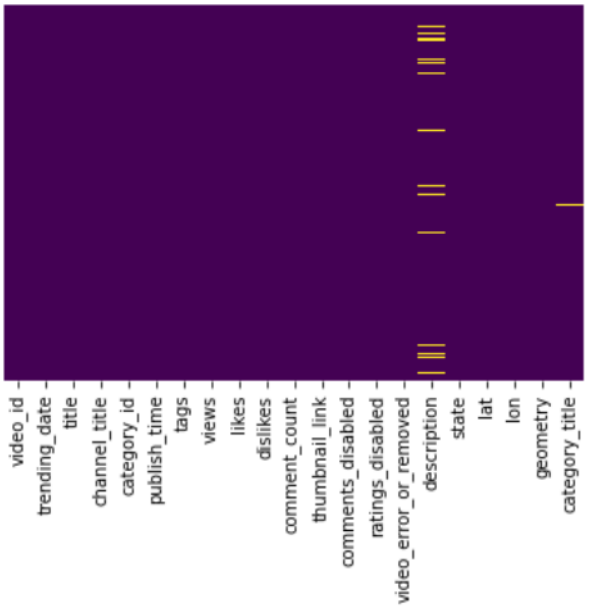
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40724 entries, 0 to 40723
Data columns (total 21 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   video_id                40724 non-null  object
 1   trending_date           40724 non-null  datetime64[ns]
 2   title                   40724 non-null  object
 3   channel_title           40724 non-null  object
 4   category_id             40724 non-null  int64
 5   publish_time            40724 non-null  datetime64[ns]
 6   tags                    40724 non-null  object
 7   views                   40724 non-null  int64
 8   likes                   40724 non-null  int64
 9   dislikes                40724 non-null  int64
 10  comment_count           40724 non-null  int64
 11  thumbnail_link          40724 non-null  object
 12  comments_disabled       40724 non-null  bool
 13  ratings_disabled        40724 non-null  bool
 14  video_error_or_removed  40724 non-null  bool
 15  description             37812 non-null  object
 16  state                   40724 non-null  object
 17  lat                     40724 non-null  float64
 18  lon                     40724 non-null  float64
 19  geometry                40724 non-null  object
 20  category_title          40610 non-null  object
dtypes: bool(3), datetime64[ns](2), float64(2), int64(5), object(9)
memory usage: 5.7+ MB
```

- Detección y manipulación de valores Na (faltantes/vacíos)

Este paso se hace para detectar aquellos valores que por alguna razón o motivo fueron ingresados de manera incorrecta y nos retorna valores NA o vacíos que no ayudan para el análisis que se quiere realizar.
Para identificar esos valores se utilizaron las siguientes funciones:

```
print(Frvids.isna().sum())
sns.heatmap(Frvids.isnull(),yticklabels=False,cbar=False,cmap='viridis
')
```

```
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              2912
state                       0
lat                         0
lon                         0
geometry                    0
category_title            114
dtype: int64
```

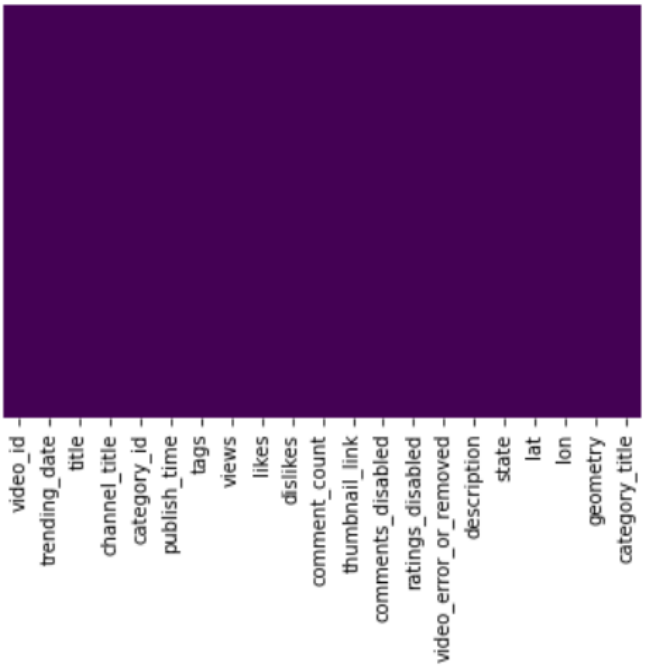



Al ejecutar esas funciones en nuestro Data Frame conseguimos un conteo de la cantidad de filas por columna que tienen un valor vacío y una representación gráfica de la misma data.
Se puede observar que las columnas de *description* y *category_title* tienen filas con estos valores de los cuales nos queremos deshacer. Teniendo en cuenta que esta cantidad de filas representa una cantidad mínima del Data Frame se decidió eliminar esas filas para obtener un Data Frame limpio de la siguiente manera:

```
FrvidsClean = Frvids[Frvids['description'].notna()]
FrvidsClean=FrvidsClean[FrvidsClean['category_title'].notna()]
```

Posterior a la eliminación de las filas que tenían valores vacíos se puede volver a hacer la búsqueda de valores vacíos y se mostrará que han sido completamente removidos del conjunto de datos.

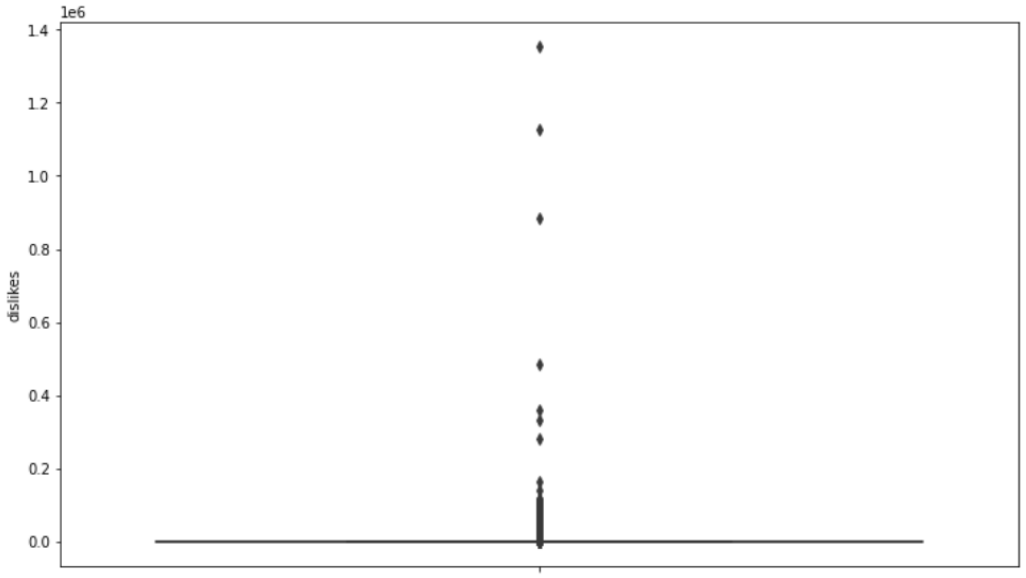
```
video_id                  0
trending_date             0
title                     0
channel_title             0
category_id               0
publish_time              0
tags                      0
views                     0
likes                     0
dislikes                  0
comment_count             0
thumbnail_link            0
comments_disabled         0
ratings_disabled          0
video_error_or_removed    0
description               0
state                     0
lat                       0
lon                       0
geometry                  0
category_title            0
dtype: int64
```



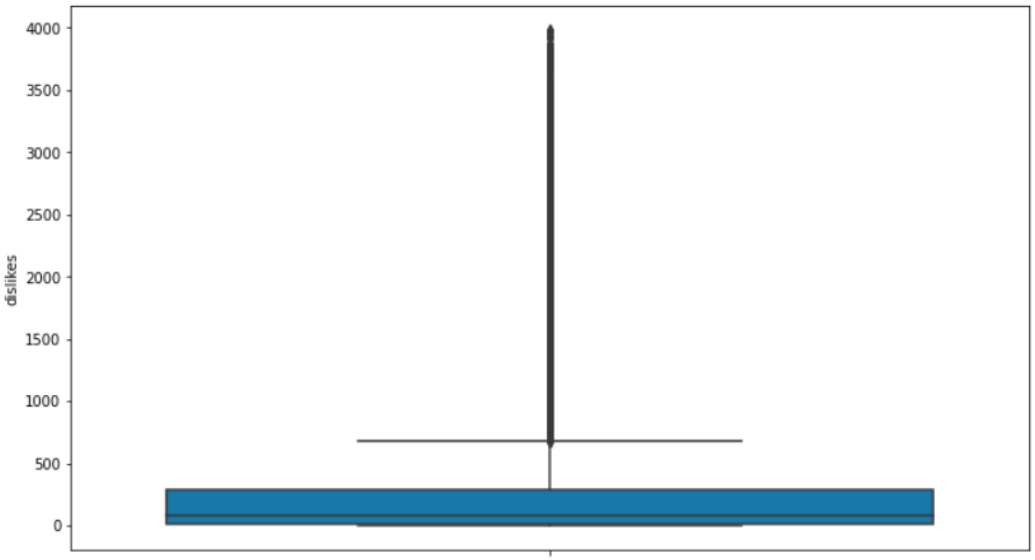

- Detección y manipulación de Outliers (valores atípicos)

  Para detectar valores atípicos en el conjunto de datos hay que evaluar las columnas numéricas. Para el conjunto de datos con el que se está trabajando en esta ocasión, las columnas que se necesita evaluar son: *views, likes, dislikes y comment_count*. Una manera rápida de visualizar la distribución de los datos de una columna es con un boxplot:

```
plt.figure(figsize=(12, 7))
sns.boxplot(y=FrvidsClean['dislikes'], x=FrvidsClean['category_id'],
data=FrvidsClean,palette='winter')
```



A través del boxplot podemos apreciar que hay una gran cantidad de valores que se desvían de la mayoría para la columna *dislikes*. Lo mismo sucede para las columnas *views*, *likes* y *comment_count*. Entonces lo que queda decidir es si estos valores son coherentes para evaluar, si es que no han sido producto de un error. Sin embargo, teniendo en cuenta que nuestro conjunto de datos trabaja con información sobre videos, el que varios videos se vuelvan virales y consigan números mucho más altos que el promedio no es inusual ni surrealista. Por otro lado, en esas mismas columnas hay una gran cantidad de valores bajos que ciertamente no son muy coherentes para videos tendencia. Que muchos vídeos tendencia tuvieran 0 likes o menos de 500 views afecta notoriamente los resultados del posterior análisis a realizar. Para deshacernos de estos videos con valores incoherentes utilizamos la técnica de Z-Score Method, la cual consiste en identificar qué valores tienen una desviación mayor a tres. Dichos casos procedieron a ser eliminados. (El método del rango intercuartil no era posible ya que el cálculo del “bigote” inferior era negativo nuestros valores no pueden ser negativos).




Al finalizar el proceso de limpieza, el dataset con el que se trabajará cuenta con un total de 35712 filas.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 35712 entries, 0 to 40722
Data columns (total 21 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   video_id                35712 non-null  object
 1   trending_date           35712 non-null  datetime64[ns]
 2   title                   35712 non-null  object
 3   channel_title           35712 non-null  object
 4   category_id             35712 non-null  int64
 5   publish_time            35712 non-null  datetime64[ns]
 6   tags                    35712 non-null  object
 7   views                   35712 non-null  int64
 8   likes                   35712 non-null  int64
 9   dislikes                35712 non-null  int64
 10  comment_count           35712 non-null  int64
 11  thumbnail_link          35712 non-null  object
 12  comments_disabled       35712 non-null  bool
 13  ratings_disabled        35712 non-null  bool
 14  video_error_or_removed  35712 non-null  bool
 15  description             35712 non-null  object
 16  state                   35712 non-null  object
 17  lat                     35712 non-null  float64
 18  lon                     35712 non-null  float64
 19  geometry                35712 non-null  object
 20  category_title          35712 non-null  object
dtypes: bool(3), datetime64[ns](2), float64(2), int64(5), object(9)
memory usage: 5.3+ MB
```

- Visualizar Datos

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

   Podemos observar en el gráfico a continuación la cantidad de videos que posee cada categoría, también se adjunta el código utilizado para la elaboración del gráfico.

```
df = pd.DataFrame(Frvids, columns=["category_title"])

cat_tendencia=Frvids.groupby("category_title").count()
plt.bar(cat_tendencia.index,cat_tendencia['category_id'])
plt.title("Cantidad de videos por Categoría")
plt.xlabel("Categoría de videos")
plt.ylabel("Cantidad de videos")
plt.xticks(rotation='vertical')
```

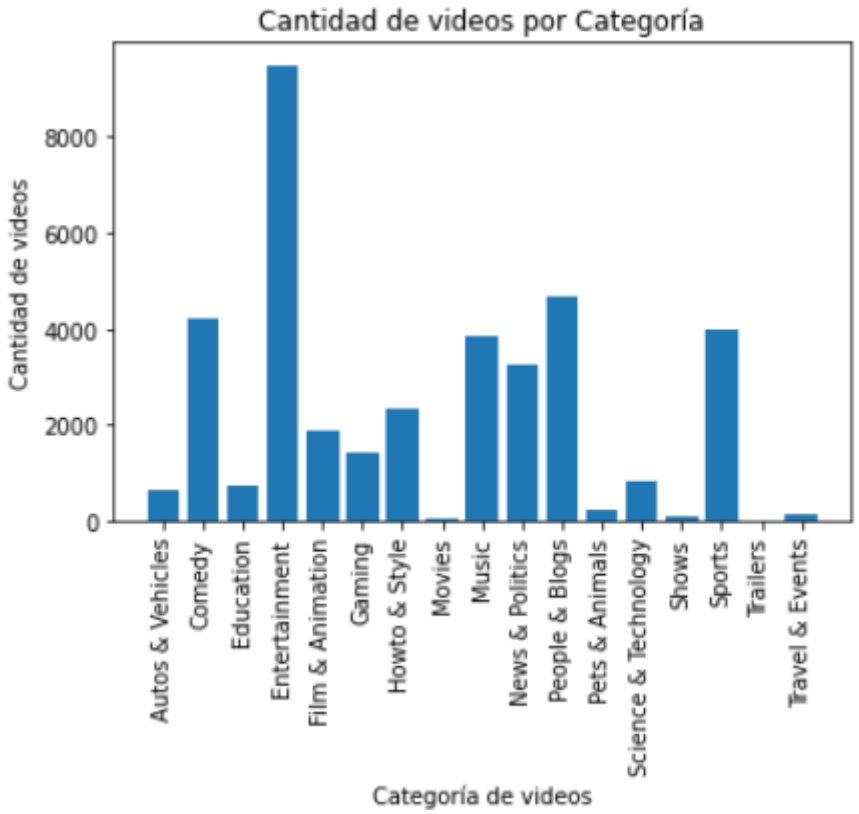



Figura 1

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

En el cuadro a continuación se visualiza la cantidad de likes por Categoría, fue necesario el agrupamiento de category_title como se muestra en el código a continuación para la visualización del gráfico de barra.

```
cat_gustan=Frvids.groupby(['category_title']).sum()

plt.bar(cat_gustan.index,cat_gustan['likes'])
plt.title("Cantidad de likes por Categoría")
plt.xlabel("Categoría de videos")
plt.ylabel("Cantidad de likes")
plt.xticks(rotation='vertical')
```

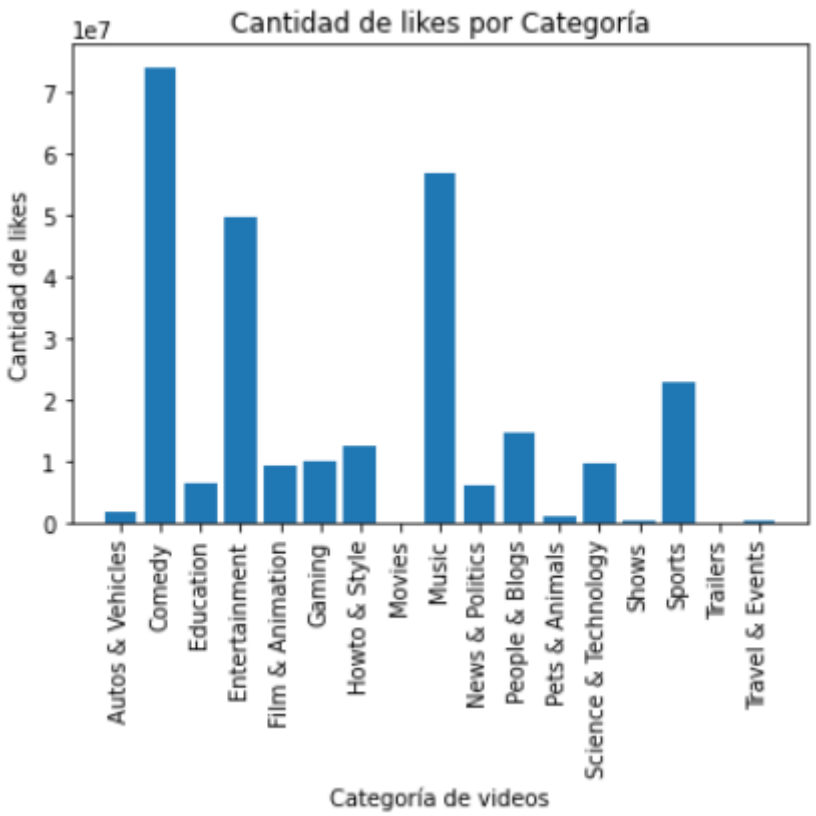



Figura 2.1

También se realizó una gráfica donde se visualiza la cantidad de dislikes por categoría. Para el ingreso de datos al `plt.bar` se utilizó la misma estructura del DataFrame para la gráfica anterior, el código se muestra a continuación.

```
plt.bar(cat_gustan.index,cat_gustan['dislikes'])
plt.title("Cantidad de dislikes por Categoría")
plt.xlabel("Categoría de videos")
plt.ylabel("Cantidad de dislikes")
plt.xticks(rotation='vertical')
```

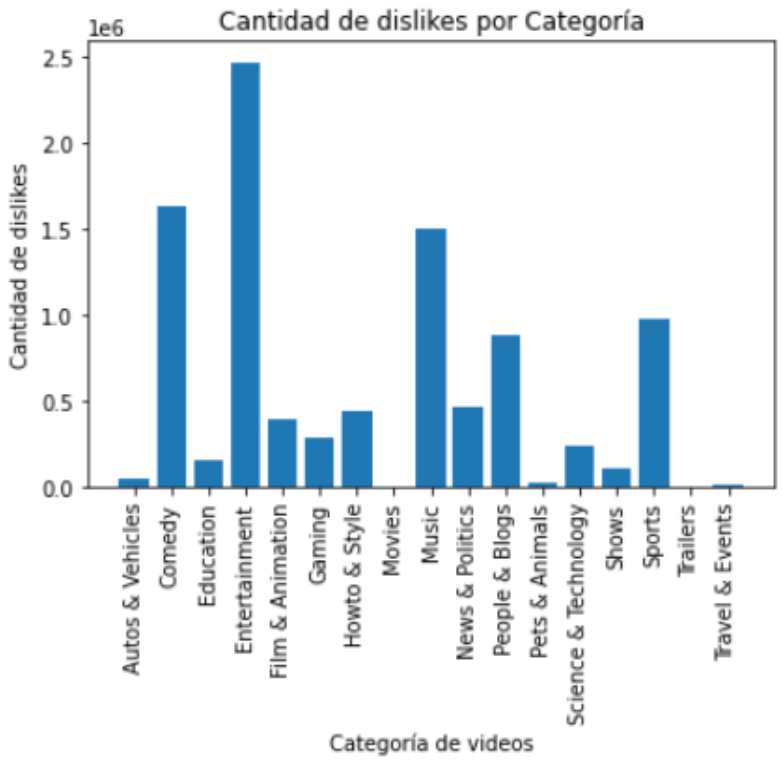



Figura 2.2

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

   Para el cálculo de la mejor proporción de `Me gusta` / `No me gusta`, se hizo un agrupamiento por `Category_title` donde el resultado del agrupamiento se suma y guarda en la variable `cat_pro`. Con la data resultante se puede realizar el gráfico a continuación donde se visualiza la proporción de likes y dislikes por categoría.

```
cat_pro_likes=Frvids.groupby("category_title").sum().sort_values('views',ascending=False)

plt.bar(cat_pro_likes.index,(cat_pro_likes['likes']/cat_pro_likes['dislikes']))
plt.title("Proporción de likes y dislikes segun Categoría")
plt.xlabel("Categoria de videos")
plt.ylabel("Proporción de likes/dislikes")
plt.xticks(rotation='vertical')
```


Figura 3

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

   Para hallar la proporción de vistas por comentarios que existen en los videos según categorías, fue necesario el agrupamiento nuevamente de `category_title`, donde el resultado del agrupamiento debe sumarse dando como resultado el nuevo Data Frame que se utilizara para para la elaboración de la gráfica siguiente.

```
cat_pro_vist_coment=Frvids.groupby("category_title").sum().s
ort_values('views',ascending=False)

plt.bar(cat_pro_vist_coment.index,(cat_pro_vist_coment['view
s']/cat_pro_vist_coment['comment_count']))
plt.title("Proporción de vistas y comentarios segun
Categoría")
plt.xlabel("Categoria de videos")
plt.ylabel("Proporción de vistas/comentarios ")
plt.xticks(rotation='vertical')
```

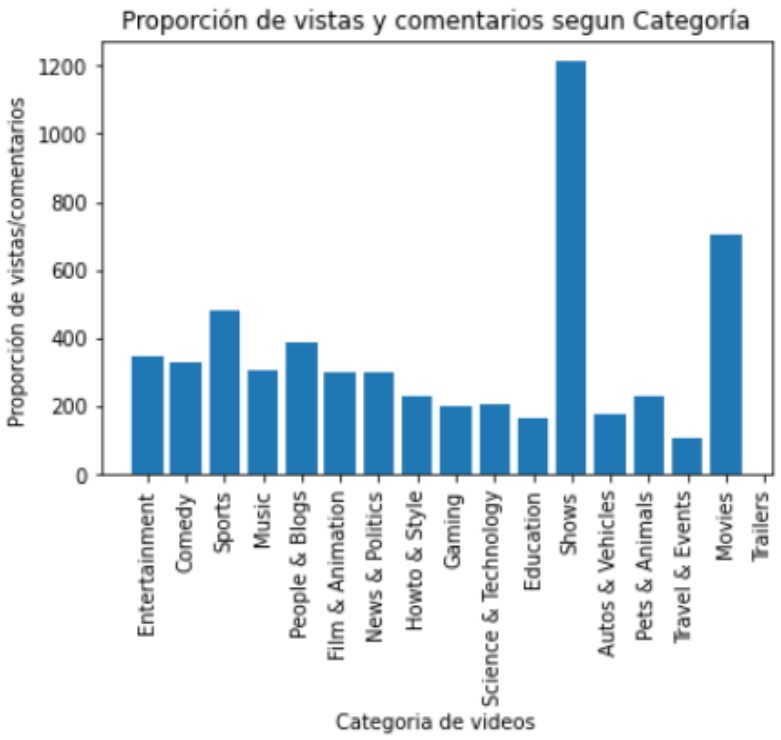



Figura 4

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

   Las gráficas a continuación, tienen como objetivo evidenciar el cambio del volumen del surgimiento de videos en tendencia con el pasar del tiempo. Para hacer esto se contó la cantidad de videos que se volvieron tendencia en cada día y mes.

```
plt.figure(figsize=(20, 7))
Frvids.groupby(Frvids['trending_date'].rename('Fechas')).size().plot()
plt.ylabel("Cantidad de videos")
plt.title("Distribucion de videos tendencia con el tiempo")
```

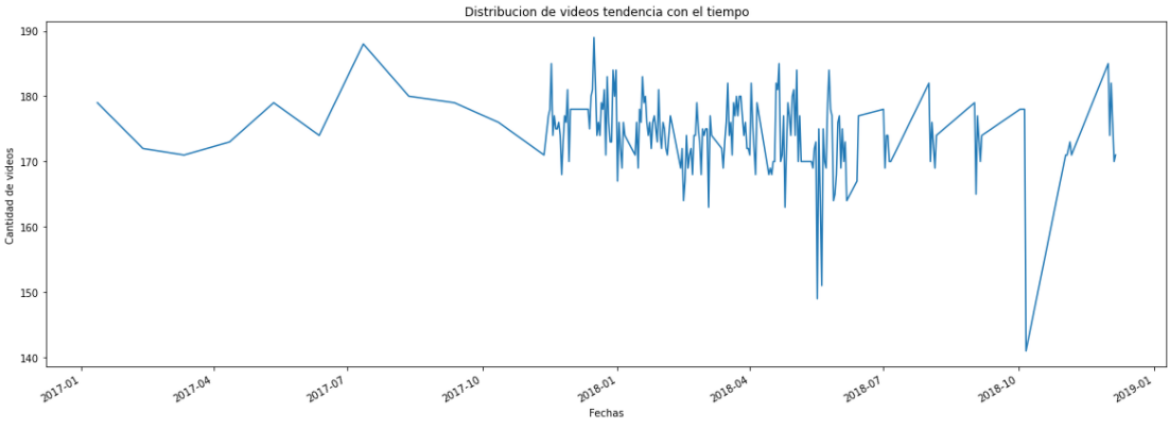



Figura 5.1

```
plt.figure(figsize=(10, 7))
Frvids.groupby(Frvids['trending_date'].rename('Meses').dt.month).size(
).plot(marker='o', linestyle='--')
```

```
plt.ylabel("Cantidad de videos")
plt.title("Distribución de videos tendencia por mes")
```

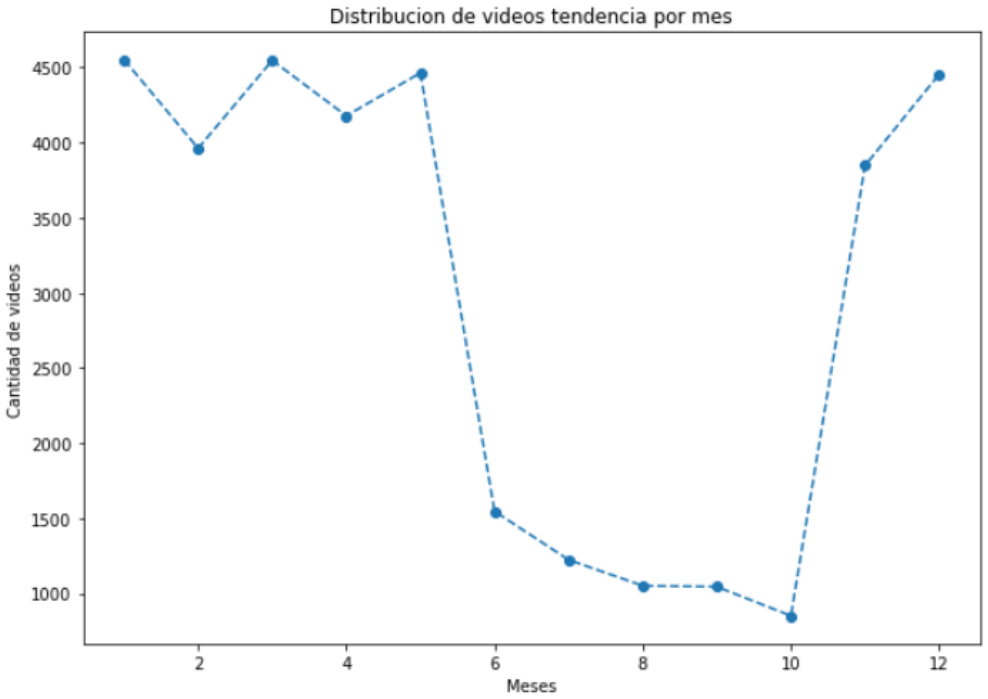



Figura 5.2

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

   Para esta pregunta lo que se hizo fue encontrar que canales eran los que tenían más videos en el dataset. Esto es interpretado como que el canal tiene una mayor capacidad de producir videos que son tendencia. Por lo tanto, dichos canales son tendencia más frecuentemente. A continuación se visualizan los 10 canales que son tendencia más frecuentemente:

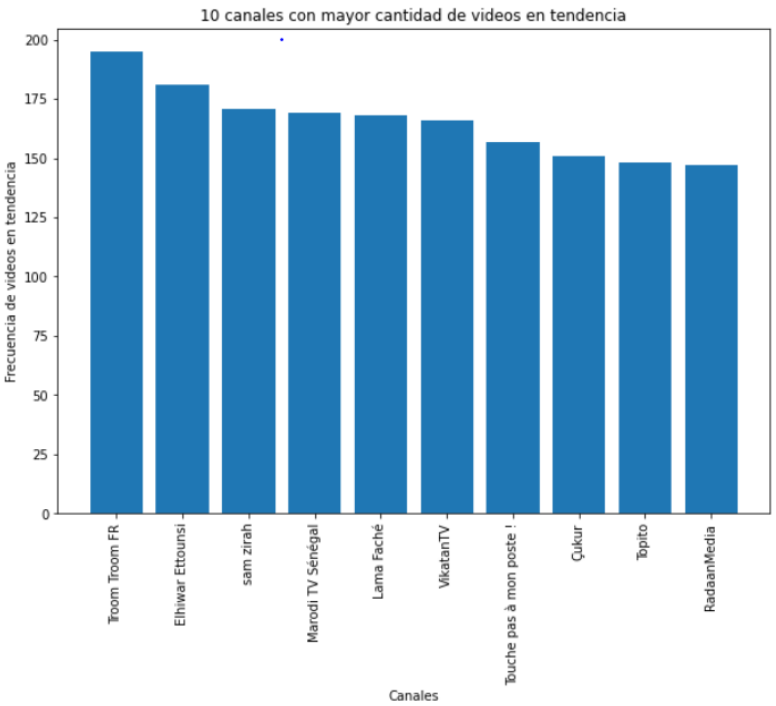



Figura 6.1

Por el otro lado, al querer obtener información sobre los canales de menor frecuencia de tendencia la metodología aplicada previamente no nos ayudaba, ya que solo nos retornaba canales que tenían un solo video. Entonces, lo que se hizo fue agrupar los 200 canales con menor frecuencia de tendencia y agruparlos según Categoría.

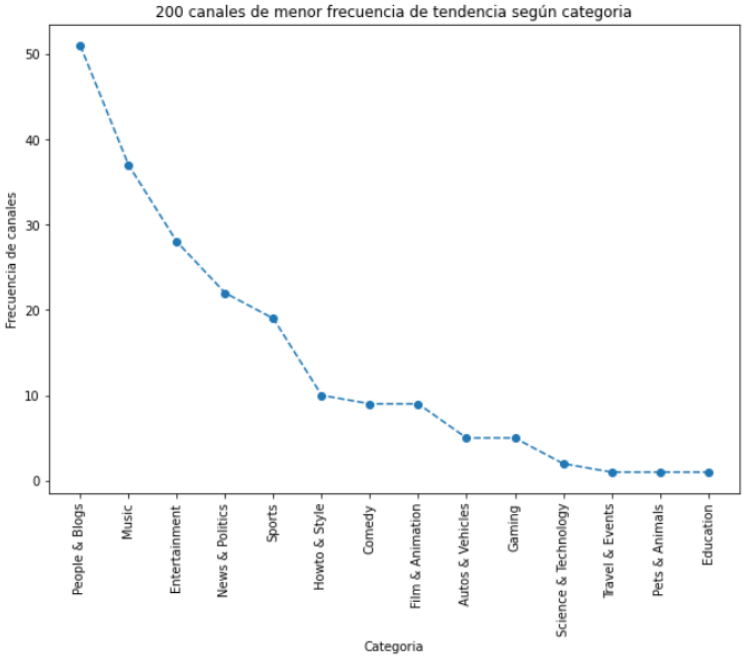



Figura 6.2

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

En esta pregunta utilizamos principalmente, la librería panda y plotly para realizar el gráfico geográfico. Para esto, se utilizaron las columnas del dataset, latitud y longitud, para calcular la posición de los estados. Una vez realizado esto agrupamos el dataset según el estado al que pertenece. Finalmente, utilizamos la gráfica Scatter Mapbox de Plotly para evaluar los valores de cada estado para ‘vistas’, ‘me gusta’ y ‘no me gusta’ donde podremos acercar y alejar el mapa para visualizar mejor los datos obtenidos.

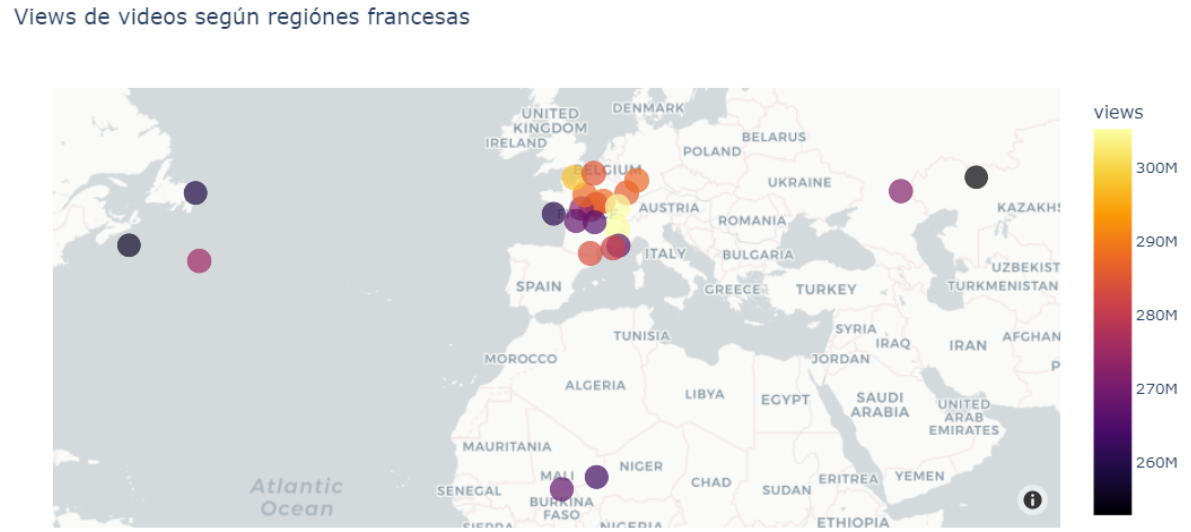



Figura 7.1

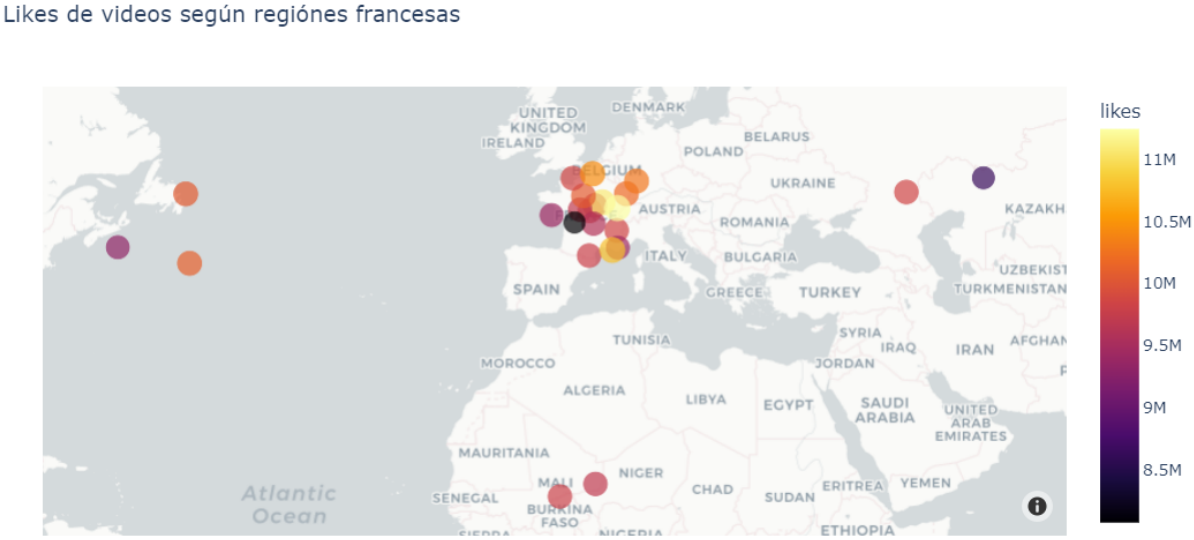



Figura 7.2

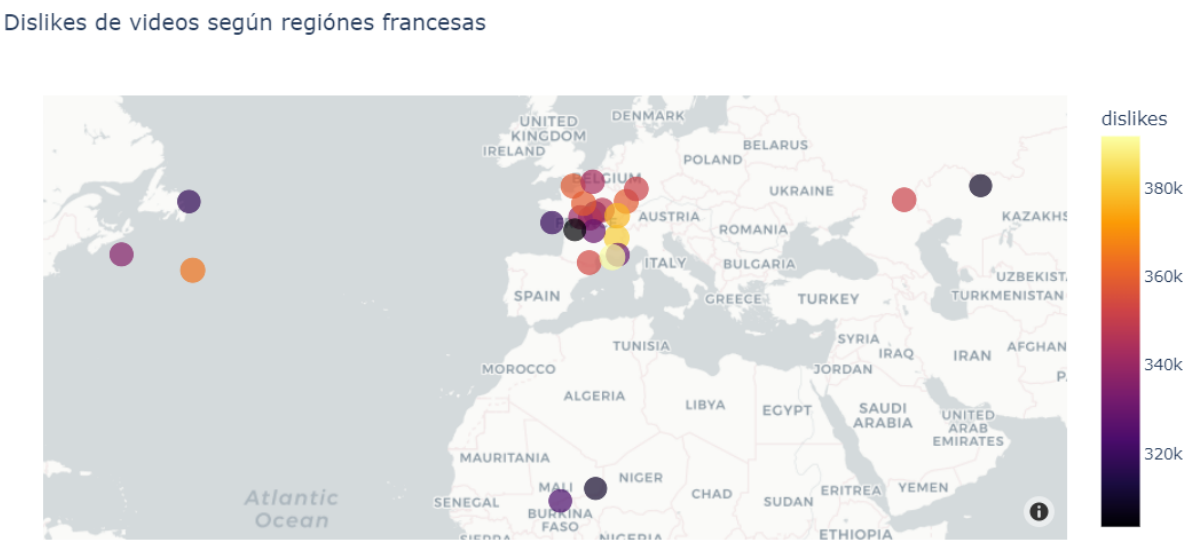



Figura 7.3

Adicionales:

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

  Para la predicción se realizó un modelo, el cual tiene como variable de salida a `likes` y datos entrada `comment_count`, `dislike` y `views`, los resultados, el error y el residuo, son indicadores de que el modelo predice con gran eficacia, este tema se ve a detalle en el punto 5. De modo que es factible predecir el número de “Vistas”, “Me gusta” y “No me gusta”.

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

  La respuesta es no, debido a la elaboración del código, el cual nos da la información en porcentaje de comentarios positivos y negativos de los videos, podemos observar que la mayor parte de los comentarios hacia los videos en tendencia son los negativos. El código y el gráfico se muestran a continuación.

  En la porción de código siguiente se puede observar que se halló la media de likes y dislikes, luego se utilizó en una función para cada video, designando el valor 1 en una nueva columna si los likes eran mayor que el promedio, y 0 si era menor.

```
media_like=Frvids['likes'].mean()
media_dislike=Frvids['dislikes'].mean()

def analize_sentiment(comment):
    if comment > media_like :
        return 1
```

```
    else:
        return 0

Frvids['polaridad'] =
np.array([analize_sentiment(comment) for comment in
Frvids['likes']])
Frvids.head()
```

| | views | likes | dislikes | comment_count | polaridad |
|---|---|---|---|---|---|
| 0 | 212702 | 29282 | 1108 | 3817 | 1 |
| 1 | 432721 | 14053 | 576 | 1161 | 0 |
| 2 | 482153 | 76203 | 477 | 9580 | 1 |
| 3 | 925222 | 85016 | 550 | 4303 | 1 |
| 4 | 141695 | 8091 | 72 | 481 | 0 |

Tabla. 1

Luego hallamos la cantidad de comentarios positivos y negativos, guardamos estos valores para luego expresarlos en porcentaje como se muestra a continuación.

```
pos_com= [ i for index, i in enumerate(Frvids['likes'])
if Frvids['polaridad'][index] > 0]
neg_com = [ i for index, i in enumerate(Frvids['likes'])
if Frvids['polaridad'][index] == 0]

print("Percentaje de positivos :
{}%".format(len(pos_com)*100/len(Frvids['likes'])))
print("Percentaje de negativos :
{}%".format(len(neg_com)*100/len(Frvids['likes'])))
```

```
Percentaje de positivos : 23.25548835125448%
Percentaje de negativos : 76.74451164874552%
```

Tabla. 2

Para tener una visualización clara de esto se procedió a generar un gráfico.

```
cantidad = Frvids['polaridad'].value_counts()
plt.xticks(rotation='horizontal')
plt.bar(['negativo','positivo'], cantidad.values)
```

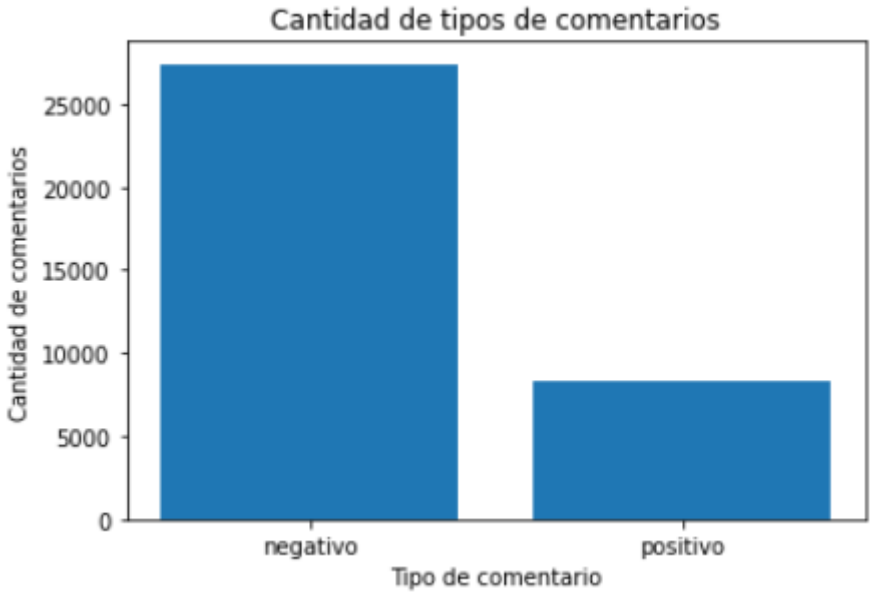



Fig. 8

## 5. Modelizar y evaluar los datos

- Identificar qué variables en el conjunto de datos son susceptibles a ser modeladas.

Para realizar el modelo es necesario que los datos ingresados en este caso sean numéricos, por lo tanto, tendríamos variables como `dislikes`, `comment_count`, `views` y `likes`, que son variables numéricas, las cuales pueden pertenecer al modelo. De estas variables elegimos como variable de salida a `likes`, y como variables de entrada `dislikes`, `comment_count` y `views`.

- • Describir el "conocimiento" que se intenta extraer a partir de la aplicación de un modelo de datos.

A partir de la aplicación de un modelo, es posible estimar, según sea el algoritmo a utilizar, un dato, en este caso nosotros disponemos como variable para estimar a `likes`. De esta manera podemos estimar los `likes` de algún video, a partir de otras variables como `dislikes`, `comment_count` y `views`.

```
X = Frvids[['dislikes','views','comment_count']]
y = Frvids['likes']
```

De otra forma podemos limpiar el Data Frame, dejando solo variables que utilizaremos.

```
varialbes = Frvids
varialbes.drop(['video_id','trending_date','title','channel_title'
,'category_id','publish_time','tags'
,'thumbnail_link','comments_disabled','ratings_disabled','video_er
ror_or_removed','description'
```

```
,'state','lat','lon','geometry','category_title'],axis='columns',
inplace=True)
```

| | views | likes | dislikes | comment_count | polaridad |
|---|---|---|---|---|---|
| **0** | 212702 | 29282 | 1108 | 3817 | 1 |
| **1** | 432721 | 14053 | 576 | 1161 | 1 |
| **2** | 925222 | 85016 | 550 | 4303 | 1 |
| **3** | 141695 | 8091 | 72 | 481 | 1 |
| **4** | 141253 | 14354 | 202 | 417 | 1 |
| **...** | ... | ... | ... | ... | ... |
| **35707** | 560827 | 8688 | 147 | 696 | 1 |
| **35708** | 58758 | 307 | 86 | 485 | 0 |
| **35709** | 264639 | 2011 | 999 | 1397 | 0 |
| **35710** | 47510 | 4601 | 61 | 615 | 0 |
| **35711** | 78117 | 244 | 74 | 46 | 0 |

35712 rows × 5 columns

Tabla. 3

- • Utilizar algún algoritmo innovador (según la pregunta a responder) para crear un modelo de datos (e. algoritmo de regresión lineal).

El algoritmo que se utilizó para la creación del modelo de Regresión Lineal, a continuación se muestran las importaciones necesarias y declaración de datos en código para el entrenamiento y la validación.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=50)

from sklearn.linear_model import LinearRegression

lm = LinearRegression()
lm.fit(X_train,y_train)
```

- • Obtener métricas y realizar una evaluación del resultado obtenido en el modelo.

Gracias a la librería `sklearn.linear_model` model mediante la importación de `LinearRegression` podemos evaluar los coeficientes de los datos de entrega, el código y la tabla se ven a continuación.

```
coef_df = pd.DataFrame(lm.coef_,X.columns,columns=['Coeficiente'])
coef_df
```

| | Coeficiente |
|---|---|
| views | 0.010061 |
| dislikes | 4.779857 |
| comment_count | 7.319839 |

Tabla. 4

Esto significa que por cada vista, encontramos 0.023642 likes, significa también que por cada comentario encontramos 4.481385 likes. Con esta información pasamos a la validación, donde se ingresaron los datos de entrada para la validación, esto se muestra en el código a continuación, y también la gráfica resultante de la predicción.

```
predictions = lm.predict(X_test)

plt.scatter(y_test,predictions)
plt.xlabel('Y Prueba')
plt.ylabel('Y Predichos')
```

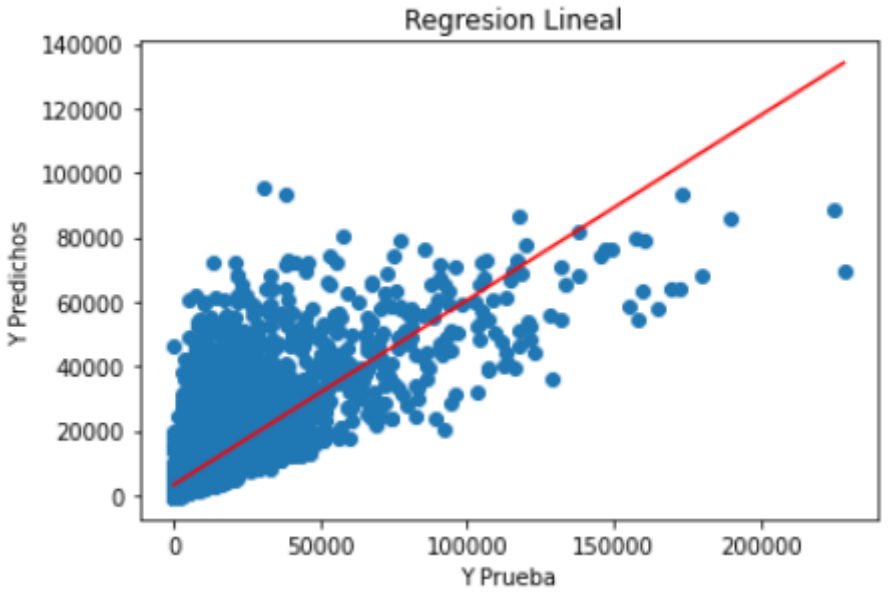



Fig. 9.1

Se puede observar un gran número de predicciones “correctas”, pero esta validación con exactitud la podemos ver a continuación, donde mostramos un gráfico de barras el cual tiene como eje y a la cantidad de videos y en el eje x la cantidad de residuo (resta de los datos de salida de la validación - la predicción).

```
sns.displot((y_test-predictions),bins=50);
```

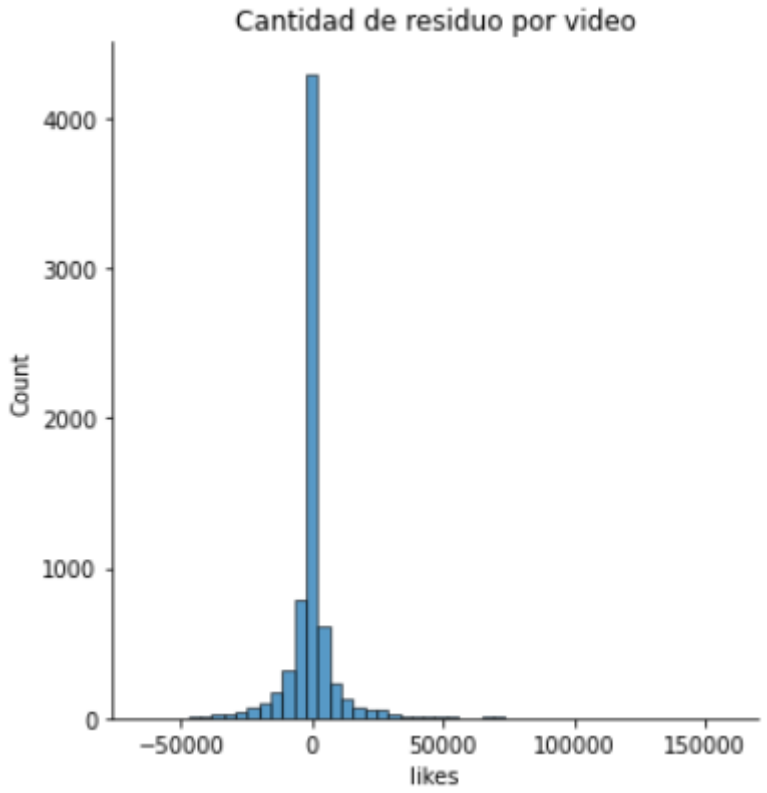



Fig.9.2

Como se puede observar en el gráfico anterior, la mayoría tiene como residuo 0, esto significa que los datos para la validaciones (datos de salida), tuvieron en su mayoría una correcta predicción.
Por último tenemos las métricas utilizadas para hallar el error, utilizamos 3, MAE, MSE, RMSE, estas pueden ser utilizadas a través de la librería `sklearn` importando `LinearRegression`, el código se muestra a continuación y el resultado de estas métricas.

```
print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test,
predictions)))
```

```
MAE: 5024.686346716741
MSE: 121283400.82974611
RMSE: 11012.87432189009
```

Tabla. 5

# 6. Conclusiones del proyecto

- **Por Categoría de Videos**

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

Los videos ingresados en el Data Frame se consideran todos tendencia, partiendo de este punto, la cantidad de videos que contiene una categoría, definiría qué categoría tiene mayor tendencia. Como se puede observar en la figura 1, las categorías que contienen más vídeos (mayor tendencia) son, Entertainment que tiene aproximadamente 90,000 videos, luego sigue People & Blogs por casi la mitad con aproximadamente 44,000 videos,y siguen Comedy con Sports con cantidad de videos muy similares aproximadamente 40,000 videos.

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

Como se puede observar en la figura 2.1, la categoría de videos que más gustan es la de Comedy con aproximadamente 7x10^7 likes, seguida por Music con aproximadamente 6x10^7 likes, y en tercer lugar se encuentra Entertainment con aproximadamente 5x10^5 likes. Por otro lado en la figura 2.2 tenemos las categorías con más dislikes o las que menos gustan, esta lista está encabezada por Entertainment con aproximadamente 2.5x10^5 dislikes, seguida por Comedy con aproximadamente 1.6x10^6 dislikes, y en tercer lugar se encuentra Music con aproximadamente 1.5x10^6 dislikes. En conclusión, la categoría con más *likes* es Comedy, y la categoría con más *dislikes* es Entertainment.

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

La categoría de videos que tiene la mejor proporción (ratio) de *like/dislike*, según la figura 3, son Pets & Animals quien encabeza esta lista con aproximadamente 50 likes/dislikes, en segundo lugar está Travel & Events con aproximadamente 45 likes/dislikes, quién es seguida por Comedy con aproximadamente 44 likes/dislikes. En conclusión, la figura 3 da a entender que la categoría que tiene más likes por un “No me gusta” es Pets & Animals, sin embargo en general, existe una buena proporción entre “Me gusta” / “No me gusta”.

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

Según la figura 8 que refiere a la cantidad de tipos de comentarios, podemos ver que la mayoría de los comentarios son negativos, en la tabla 2, podemos ver los porcentajes de comentarios negativos y positivos, el total de comentarios negativos es del 78.74%, dando a entender que en general los comentarios de los videos en tendencia son negativos. Partiendo de este punto, las categorías que tienen la mejor proporción (ratio) “Vistas” / “Comentarios” según la figura 4, en primer lugar tenemos a la categoría Show con aproximadamente 1200 vistas/comentarios, la cual está seguida por Movies con aproximadamente 700 vistas/comentarios, y en tercer lugar tenemos Sports con aproximadamente 500 vistas/ comentarios. En conclusión, la categoría que tiene mejor proporción (ratio) “Vistas” / “Comentarios” es Shows.

- **Por el tiempo transcurrido**

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

Para los videos pertenecientes al dataset en cuestión, se volvieron tendencia entre el inicio del 2017 y el final del 2018. En la figura 5.1, se puede observar la frecuencia videos según la fecha en la que se volvieron tendencia. Dicha figura nos muestra que el mayor volumen de videos está en la primera mitad del 2018. En cambio, el 2017 y la segunda mitad del 2018 el surgimiento de videos tendencia es más esporádico. Sin embargo, a través del tiempo siempre que surgen videos tendencia lo hacen en cantidades entre 165-190 videos.
Por otro lado en la figura 5.2, se tiene un enfoque un poco diferente que agrupa la data de la figura 5.1 por mes. Se puede concluir en base a la figura que desde Diciembre a Mayo se tiene un volumen constante de alrededor de 4000 videos por mes. No obstante, el volumen decrece radicalmente entre los meses de Junio hasta Octubre. Durante este periodo de tiempo el volumen de videos oscila alrededor de los 1600 a 1000 videos.

- **Por Canales de YouTube**

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

Para resolver esta duda se contabilizó cuantos vídeos tendencia tenía cada canal y se extrajeron los diez mejores. Esto se refleja en la figura 6.1, donde se puede observar que los diez canales que más han sido tendencia tienen todos más de 150 videos. El canal *Troom Troom FR* es el canal que tiene la mayor cantidad de videos tendencia con 195 videos, es decir, es el canal que ha sido tendencia más frecuentemente en el alcance de tiempo de nuestro dataset. Los canales *Elhiwar Ettounsi* y *sam zirah* son el segundo y tercero con 181 y 171 videos, respectivamente.
Por el lado de los canales con la menor frecuencia, el enfoque precio no nos dio mucha información, ya que habían cientos de canales con solo un video tendencia. Entonces, se agruparon los 200 canales con la menor cantidad de videos tendencia por categoría. De tal manera se puede analizar qué categoría tiene la mayor cantidad de canales que no suelen ser tendencia a menudo. El resultado es que la categoría de *People & Blogs, Music* y *Entertainment* son las categorías que más canales contienen.

- **Por la geografía del país**

7. ¿En qué Estados se presenta el mayor número de “Vistas”, “Me gusta” y “No me gusta”?
En el dataset filtrado se presenta el mayor número de “Vistas” en el estado de Rhone Alpes con un número total de 305,266,731, asimismo Rhone Alpes también cuenta con el mayor número de “Me gusta” con 11,247,990. Adicionalmente, el estado con más “No me gusta” es Provence Alpes Cote D'azur el cual cuenta con 391,752. Esto está vinculado con las vistas debido a que los videos que más vistos tienen a tener mayor interacción de los usuarios.

Finalmente, a pesar de ser los ejemplos más extremos todos los estados cuentan con un número similar de “Vistas” y “No me gusta”, sin embargo, las interacciones con “Me gusta” presentan una diferencia notoria, esto sucede debido a la predisposición de la gente a expresar una opinión negativa en lugar de una positiva.

- **Adicionalmente, al cliente le gustaría conocer si:**

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

  Es factible, en la figura 9.1 podemos observar en la gráfica de la regresión lineal del modelo, que hay algunos puntos fuera de línea, sin embargo estos no tienen gran impacto en el resultado final, ya que la mayoría está cercana a la línea. Para confirmar esto, en la figura 9.2 podemos ver que al residuo igual a 0 se encuentran aproximadamente 4,000 videos, casi la mayoría de los valores, dando a entender que la predicción en su mayoría fueron aciertos, esto podemos observar también con “Vistas” y “No me gusta” en el código del repositorio, donde se debe cambiar los valores de entrada y de salida, teniendo en fin resultados similares.

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

  No, como se puede ver en la figura 8, hay un mayor margen de comentarios negativos en los videos, aproximadamente cerca de los 25,000, por otro lado están los comentarios positivos con cerca de un valor total de 7,500. Existe una gran diferencia, esto se puede ver más claro en los porcentajes de la tabla 2, donde la cantidad de comentarios positivos es el 23.26% del total de comentarios, esto hace que el 76,74% de los comentarios sean negativos.

# 7. Archivar y publicar

Enlace a repositorio de github: https://github.com/SebsPER/EB-2021-1-CC51