Universidad Peruana de Ciencias Aplicadas

Ciclo 2021-2

CC50 - Administración de la Información
Sección: CC51

Trabajo Final
Tema: Creación de conocimiento a partir de los datos en Python
Profesora: Ing. Patricia Reyes Silva

César Mosqueira
Sebastian Arana

Noviembre, 2021

1 Abstract

En el presente reporte, explicaremos el trabajo analítico realizado. Para comenzar pondremos en contexto explicando el objetivo de la investigación junto con las pautas del trabajo. A continuación, veremos una descripción de los datasets brindados con los cuales realizaremos el trabajo de analítico. Explicaremos la fuente de los mismos y daremos una descripción detallada de cómo están estructurados y el proceso de limpieza que consideramos que requiere. Posteriormente, explicaremos los métodos a usar, tanto para la limpieza como para el análisis de los datos y, finalizaremos con el análisis obtenido, visualizaremos los datos y llegaremos a conclusiones del proyecto para responder las preguntas iniciales explicadas en las pautas.

2 Introducción

El objetivo del informe es explicar el proyecto de analítica que hemos llevado a cabo, en el cuál tomamos el papel de realizar un proyecto de analítica para una consultora internacional con la finalidad de conocer las tendencias de videos de YouTube en siete importantes países. Debimos tomar en cuenta algunos requerimientos que el cliente presenta.

2.1 Requerimientos

A continuación se detalla una lista de preguntas que al finalizar el proyecto deberá dar respuesta:

2.1.1 Por categoría de Videos

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

2.1.2 Por el tiempo transcurrido

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

2.1.3 Por Canales de Youtube

- ¿Qué Canales de YouTube son tendencia más frecuentemente? ¿Y cuáles con menos frecuencia?

2.1.4 Por la geografía del país

- ¿En qué Estados se presenta el mayor número de “Vistas”, “Me gusta” y “No me gusta”? Adicionalmente, al cliente le gustaría conocer si:

2.1.5 Preguntas

- ¿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?

3 Dataset

El conjunto de datos motivo de análisis se denomina Tendencias de las estadísticas de videos de YouTube (Trending YouTube Video Statistics). Este conjunto de datos es un registro diario de los videos de YouTube de mayor tendencia, cuyo contenido incluye varios meses sobre datos de tendencias diarias de videos en los siguientes países:

- EE. UU. (US)
- Gran Bretaña (GB)
- Alemania (DE)
- Canadá (CA)
- Francia (FR)
- Rusia (RU)
- México (MX)
- Corea del Sur (KR)
- Japón (JP)
- India (IN)

Los datos de cada país se encuentran en archivos individuales en formato CSV y la descripción de sus categorías en un archivo de tipo JSON. Este conjunto de datos, en su versión original se obtiene desde el sitio web Kaggle, sin embargo, para este proyecto, ha sido modificado incorporándole cuatro nuevas columnas:

- state: nombre del Estado perteneciente al país (incorporado de forma aleatoria).
- lat: latitud geográfica de ubicación del Estado.
- lon: longitud geográfica de ubicación del Estado.
- geometry: (opcional) registra las coordenadas de las geometrías donde su ubica el Estado dentro del planeta. Es de utilidad si se decide utilizar la librería GeoPandas para la elaboración de mapas.

En nuestro caso analizaremos el dataset de México: '`data/MX_category_id.json`' y '`data/MXvideos_cc50_202101.csv`'. El .csv presenta los siguientes datos:

Nombre	Definición
video_id	Codigo del video
trending_date	Fecha en la que el video fue recomendado
title	Titulo del video
channel_title	Nombre del canal
category_id	ID de la categoria
publish_time	Fecha de publicacion
tags	Tags del video
views	Cantidad de vistas
likes	Cantidad de likes
dislikes	Cantidad de dislikes
comment_count	Cantidad de comentarios
thumbnail_link	Link del thumnail
comments_disabled	Bool de comentarios desactivados
ratings_disabled	Bool de rating desactivado
video_error_or_removed	Bool de error de video
description	Descripcion del video
state	Nombre del estado donde se publicó
likes.mean	Media de likes
dislikes.mean	Media de dislikes
comment_count.mean	Media de cantidad de comentarios
category_description	Descripcion de la categoria

Table 1: Nombre y descripción de columnas en el dataset

3.1 Análisis Propio

3.1.1 ¿Para quién sería importante el análisis de estos datos?

Tomando en cuenta las preguntas que se van a responder en este proyecto, el análisis le sería de utilidad mayormente a personas que de alguna forma dependen de la plataforma de YouTube. Ya sean creadores de contenido, publicistas, anunciantes, etc. Pues al ser una ciencia volátil el hecho de si un video es tendencia o no, el análisis de datasets como estos ayudan y acercan a entender el funcionamiento de las mentes de quienes consumen esta plataforma. Esta intención ha sido acertada por YouTube mismo (google). La ciencia de datos que aplican es impecable. Generan datasets de tamaños gigantes y con algoritmos de big data[2] logran dar una experiencia de calidad. Este es el objetivo de el análsis de datos.

4 Métodos

4.1 Pre-Procesamiento

4.1.1 Descripción de categoría

Se nos brindó un archivo .json en el que contenía información respecto a las categorías, para lo cual se utilizaba la columna de `category_id` descrita en la tabla1. Entonces para tener datos más útiles a la mano, creamos una nueva columna al dataset llamada `category_description` en donde se almacena la descripción.

4.1.2 Limpieza de datos vacios

Decidimos optar por 3 criterios de eliminación de filas, que fueron el ID del video, la cantidad de vistas y el ID de la categoría. Esto quiere decir que, en caso una o más, de estas 3 columnas esté vacía, la fila del dataset será eliminada, ya que no necesitamos información de un video donde no podemos saber la cantidad de vistas ni tampoco no tenga información de sobre qué trata el video.

```
data <- data[!is.na(data$video_id),]
data <- data[!is.na(data$category_id),]
data <- data[!is.na(data$views),]$
```

Donde `data` es el dataframe original.
Posteriormente, para limpiar el resto de columnas verificamos cuáles son las que contienen datos vacíos para saber qué hacer con ellas exactamente. Utilizando:

```
empty_data <- is.na(data)
summary(empty_data)
```

Tuvimos un resultado como la tabla que se ven la figura 1. Por lo tanto decidimos reemplazar los datos vacíos con

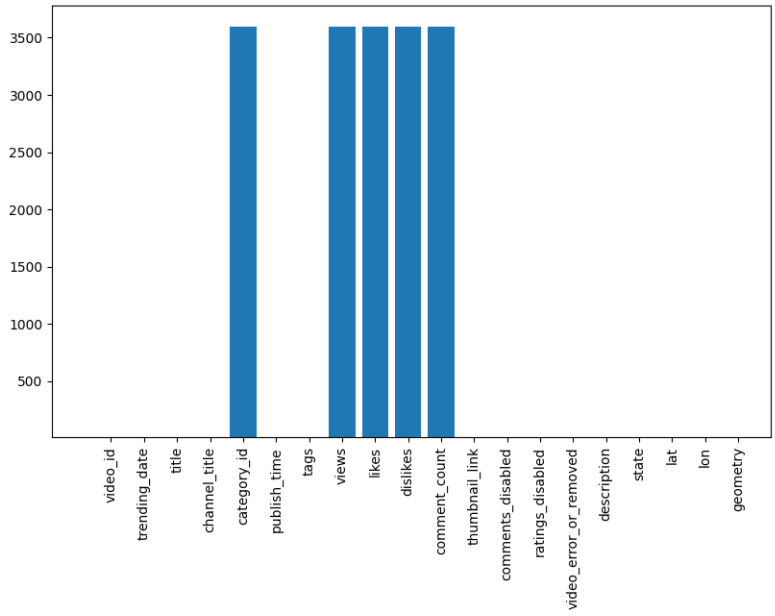


Figure 1: Cantidad de filas vacías por columna.

el promedio de la columna ya que, a diferencia de los datos anteriores, estos no afectarían drásticamente el resultado del análisis al adulterarlos.

```
lm <- mean(data$likes, na.rm = TRUE
dlm <- mean(data$dislikes, na.rm = TRUE)
```

```
ccm <- mean(data$comment_count, na.rm = TRUE)
data$likes.mean <- ifelse(is.na(data$likes), lm), data$likes)
data$dislikes.mean <- ifelse(is.na(data$dislikes), dlm, data$dislikes)
data$comment_count.mean <- ifelse(is.na(data$comment_count), ccm, data$comment_count)
```

5 Análisis

Trataremos de responder las preguntas planteadas anteriormente de la mejor manera.

5.1 Por Categoria

5.2 Por Tiempo Transcurrido

5.3 Por Canales de YouTube

El análisis que se realizó con los canales de youtube es para encontrar un índice que defina que tan común es que dichos canales llegen a estar en tendencia. Para eso aproximamos una formula en donde q es es la cantidad de videos que tiene el canal y v la cantidad de visualizaciones totales, el indice de tendencia sería $T = v/q$. Variables que serán cargadas aquí:

```
fs = dict(df['channel_title'].value_counts())
feature = {}
for channel in channels:
    feature[channel] = {
        'q': fs[channel],
        'v': np.nansum(df[df['channel_title'] == channel]['views'])
    }
```

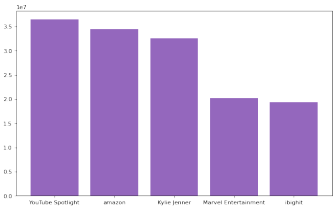
Para posteriormente conseguir tanto el que tenga mayor índice como el que tenga menor. Se buscó 5 resultados para poder compararlos.

```
SEARCHES = 5
keys = []
for _ in range(SEARCHES):
    key = min(feature, key=lambda x: feature[x]['v']/feature[x]['q'])
    keys.append({key : feature[key]})
    del feature[key]
```

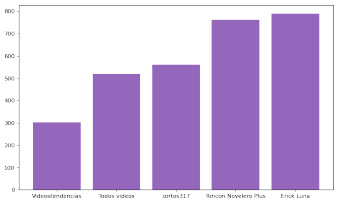
Si quisieramos cambiar si queremos el mayor o menor índice, modificaríamos `min` por `max` y `SEARCHES` es la cantidad de resultados que buscamos.
El resultado que obtenemos es el siguiente:

5.4 Por Geografía del País

Para hacer un análisis geográfico utilizamos tratamos de no solo dar una respuesta cuantitativa, por lo cual con ayuda de los geojsons de méxico[1]. Guardamos el archivo .json en `data/mexico-geo.json`.
Tuvimos que reemplazar algunos nombres de los estados del dataset, con los nombres de los estados del archivo .json para realizar la visualización. El detalle se encuentra en el repositorio.



(a) Canales con mas tendencia



(b) Canales con menos tendencia

```
df = pd.read_csv('../data/cleaned_MXvideos_cc50_202101.csv')
geo_mexico = json.load(open('../data/mexico-geo.json','r'))

df.loc[df['state'] == 'San Luis Potosi', 'state'] = 'San Luis Potosí'
df.loc[df['state'] == 'Michoacan', 'state'] = 'Michoacán'
df.loc[df['state'] == 'Yucatan', 'state'] = 'Yucatán'
df.loc[df['state'] == 'Queretaro', 'state'] = 'Querétaro'
df.loc[df['state'] == 'Nuevo Leon', 'state'] = 'Nuevo León'
df.loc[df['state'] == 'Distrito Federal', 'state'] = 'Ciudad de México'
df.loc[df['state'] == 'Mexico', 'state'] = 'México'
```

Posteriormente creamos las características del plot. Recordemos que lo que queremos hallar son los likes, dislikes y views por estado. Por lo tanto por cada estado guardaremos los 3 resultados, para luego mostrarlos por separado.

```
states = list(df['state'].unique())
features = {}
for state in states:
    features[state] = { 'likes': np.nansum(df[df['state'] == state]['likes']),
                        'dislikes': np.nansum(df[df['state'] == state]['dislikes']),
                        'views': np.nansum(df[df['state'] == state]['views']) }
```

La ventaja de realizarlo de esa forma es que ahora podemos reemplazar una palabra y tendremos el filtro para cada uno. Y como por ahora nos preocupan más los resultados que el uso de recursos, podemos permitírnoslo.

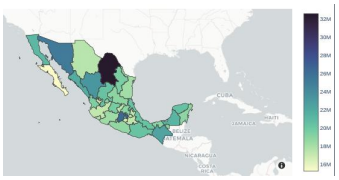
```
from plotly import graph_objects as go

#Create figure object
fig = go.Figure(
    go.Choroplethmapbox(
        geojson = geo_mexico, #Assign geojson file
        featureidkey = "properties.name", #Assign feature key
        locations = list(features), #Assign location data
        z = [features[x]['views'] for x in features], # 'views' is the filter
        zauto = True,
        colorscale = 'deep',
        showscale = True,
    )
)

#Update layout
```

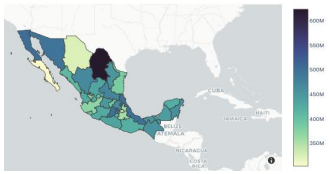
```
    fig.update_layout(
        mapbox_style = "carto-positron", #Decide a style for the map
        mapbox_zoom = 3, #Zoom in scale
        mapbox_center = {"lat": 23.43, "lon": -99.13}, #Center location of the map
        title = "Views por estado"
    )}
```

Y el resultado que obtenemos:



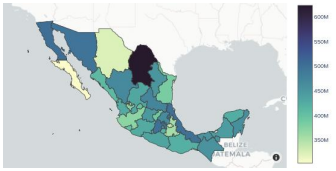
(a)

Figure 3: Likes por estado



(a)

Figure 4: Dislikes por estado



(a)

Figure 5: Vistas por estado

6 Conclusiones del proyecto

Definitabmente este trabajode análisis de datos nos va servir en el futuro.La capacidad de extraer conocimiento de datos crudos es indispensable hoy en día.

6.1 Repuestas

AL realizar el analisis geográfico nos dimos cuenta de un detalle. El estado de chihuaha (el más grande de todos), es el que tenía más dislikes y views. Sin embargo, no era el que tenía más likes. Esto nos lleva a creer que en ese estado, la plataforma de YouTube es usada para medios de entretenimiento.
Asi mismo, cuando filtra

References

[1] Daver Angelnmara. *Geojson*. Nov. 2019. URL: https://github.com/angelnmara/geojson.

[2] Paul Covington, Jay Adams, and Emre Sargin. "Deep Neural Networks for YouTube Recommendations". In: *Proceedings of the 10th ACM Conference on Recommender Systems*. ACM, Sept. 2016. DOI: 10.1145/2959100.2959190. URL: https://doi.org/10.1145/2959100.2959190.