

Modelo de Clasificación de Videos de YouTube: Caso DelcaVideography

Mariana Agudelo Zuluaga
Andrés Mauricio Cano Campiño
Esteban Castro Castaño
Juan David Gallego Montoya
Vanessa Osorio Urrea

Universidad EAFIT
Escuela de Ciencias Aplicadas e Ingeniería
Henry Laniado Rodas
Edwin Nelson Montoya Múnera
José Antonio Solano Atehortúa

Medellín, Colombia

Diciembre 9 de 2022

Contenido

1. Introducción	3
2. Marco teórico o de referencia	4
a. Algoritmo KNN	5
i) Métricas de distancia.....	5
ii) Definición de K.....	5
iii) Ventajas y desventajas del algoritmo KNN	6
b. Máquina de soporte vectorial (SVM).....	6
c. Regresión logística.....	6
d. Árboles de decisión.....	7
3. Desarrollo metodológico.....	9
a. Entendimiento del problema, pregunta de negocio o hipótesis.....	9
b. Análisis Exploratorio de Datos	10
i) Entendimiento de los datos	10
ii) Preparación de los datos.....	11
iii) Análisis descriptivo e insights importantes (correlación, causa-efecto, etc).....	16
iv) Ciclo de vida y arquitectura de los datos	23
c. Selección de modelos, Ingeniería de Características, Entrenamiento y Evaluación	30
i) Modelos.....	30
ii) Características e Ingeniería de Características	30
iii) Entrenamiento	32
iv) Evaluación.....	39
4. Modelo seleccionado	43
5. Análisis y Conclusiones.....	44
6. Bibliografía	45

1. Introducción

Según el historial de visualizaciones y demás datos recolectados para un canal de YouTube, es posible generar modelos de clasificación que permitan identificar cuáles son las variables más sensibles o significativas para apalancar el éxito de este. Particularmente se hace un análisis de DelcaVideography, un canal educativo donde se encuentran tutoriales gratuitos dedicados al diseño gráfico, audiovisual y fotográfico.

Para optimizar el aprendizaje de mi audiencia, aplico técnicas y procedimientos claros y fáciles de entender y, por supuesto, muy divertidos. Allí se pueden encontrar contenidos que incluyen desde diseño de logos, ilustraciones, arte abstracto, arte digital, imagen corporativa, tips o consejos de diseño, hasta reseñas de versiones de programas, diseño conceptual, diseño paramétrico, entre otros (DelcaVideography, 2022).

YouTube no ha dejado de ganar usuarios desde su lanzamiento y actualmente se trata de la plataforma de video abierto por excelencia, gracias al amplio y diverso contenido que ofrece, en su gran mayoría, de libre acceso. En concreto, el servicio de video bajo demanda financiado con publicidad (AVoD) contaba en 2021 con más de 2.500 millones de usuarios en todo el mundo, lo que supuso un incremento de 300 millones con respecto al año anterior (*Youtube: Usaurios a Nivel Mundial 2012-2021* / Statista, 2022).

YouTube realizó su análisis anual de las audiencias latinoamericanas en la plataforma, y esta vez registró un gran aumento. Por ejemplo, en Colombia alcanzó a más de 20 millones de personas en 8 meses, contando solo quienes son mayores de 18 años (*Así Creció YouTube En Latinoamérica En 2020*, 2020).

Tanto en la búsqueda principal como en la selección de sugerencias, YouTube encuentra videos para los usuarios en lugar de usuarios para los videos. El algoritmo no promociona ni expone los videos, sino que los muestra a cada usuario cuando este visita la plataforma. El objetivo del sistema de descubrimiento y búsqueda es unir a cada usuario con los videos que mejor respondan a sus intereses, generando así recomendaciones personalizadas (*Búsqueda y Descubrimiento de YouTube: Preguntas Frecuentes Sobre El Algoritmo y El Rendimiento - YouTube*, 2022).

Los videos de YouTube se clasifican según dos categorías: personalización del usuario y rendimiento del video.

La personalización de usuario se da según:

- Los videos que escogen ver
- Los videos que ignoran
- Los videos que rechazan
- La frecuencia con la que ven un canal o un tema

El rendimiento de los videos se da según el nivel de importancia que dan a estos los usuarios. Esto puede referirse, entre otras variables, a la duración media de la visualización, a cuánto del contenido se ha visto y a los “me gusta” que recibe el video. Lo anterior ayuda al algoritmo a acotar el mejor conjunto de videos para cada uno de los usuarios.

Por su lado, la cantidad de suscriptores, a pesar de ser una variable que puede influir en la cantidad de visualizaciones de un video, no debería ser el centro de atención del creador de contenido. Según YouTube (*Búsqueda y Descubrimiento de YouTube: Preguntas Frecuentes Sobre El Algoritmo y El Rendimiento - YouTube*, 2022), será más importante concentrarse en conocer a la audiencia y el tipo de contenido que les gusta, más que en el número de suscriptores y saber cómo funciona el algoritmo.

Otros factores que afectan la cantidad de visualizaciones son (*Búsqueda y Descubrimiento de YouTube: Preguntas Frecuentes Sobre El Algoritmo y El Rendimiento - YouTube*, 2022):

- *Interés en el tema:* Cuántas personas muestran interés en un tema concreto, hay unos temas que generan más interés que otros. Adicionalmente los intereses en los temas pueden cambiar con el paso del tiempo, es decir, pueden perder o ganar popularidad.
- *Competencia:* Puede que un video en particular tenga buenas métricas en cuanto al tiempo de visualización y el tiempo promedio de visualización, sin embargo, si un video del mismo tema tiene mejores métricas, este tiene mayor probabilidad de ser sugerido a los usuarios.
- *Estacionalidad:* El tráfico puede cambiar dependiendo de la época del año. También la etapa de la vida que están atravesando los espectadores puede influir en el éxito de los videos.

Buscando comprender las variables que afectan el posible éxito de un video se analizaron los datos provenientes de DelcaVideography, un canal de YouTube con 112 mil suscriptores, más de 9 millones de vistas y 8 años de antigüedad.

El resto del documento se encuentra organizado de la siguiente forma. En la sección 2 se aborda el marco teórico. En la sección 3 se describe el desarrollo metodológico, incluyendo el entendimiento del problema, el análisis exploratorio de los datos y la selección de modelos. También se plantea el ciclo de vida y la arquitectura de los datos. Finalmente, en la sección 4 y 5, se exponen las conclusiones del trabajo.

2. Marco teórico o de referencia

Ver videos se ha convertido en una actividad muy popular. En el pasado la mayoría de los servicios de transmisión eran unidireccionales, los perfiles de los usuarios no eran accesibles y por ende no era posible construir contenido (publicidad o programas) dirigido a un público específico. Con el incremento de las transmisiones por internet, la habilidad para conocer y consolidar información demográfica de los usuarios tales como su género o edad, basados en su historial de uso, hace viable la efectividad en la exposición a la publicidad y a los servicios de contenido ofrecidos (Nananukul, 2022).

Existen diversidad de casos de estudio en los cuales se ha identificado la posibilidad de generar videos cada vez más relacionados con los intereses de los usuarios. A modo de ejemplo se expone el artículo “Wild birds in YouTube videos: Presence of specific species contributes to increased views” (Kikuchi et al., 2022). En este caso, por ejemplo, se encontró que la presencia de especies específicas de aves incrementaba el número de visualizaciones de los videos, así mismo, se identifica que la duración de los videos, así como el número de días después de su carga en YouTube también incide en el éxito de él (Kikuchi et al., 2022).

Teniendo como objetivo identificar las variables que hacen que un canal sea más exitoso, es necesario tener en cuenta que, como se menciona en (Zappin et al., 2022a), la monetización se logra casi exclusivamente a

través de la publicidad reproducida en el contenido publicado por los creadores en sus canales. La publicidad de los videos aparece en tres momentos diferentes:

- Antes de iniciar la reproducción del video
- Durante el video
- En banners dentro del video.

Como también se menciona en (Zappin et al., 2022a) , la generación de publicidad en los videos es una variable de gran relevancia para los creadores de contenido, y es a la vez uno de los apalancadores del crecimiento de YouTube. Considerando como la plataforma determina si un video aplica para ser monetizado, esta tiene políticas establecidas para cada uno de los creadores de contenido y canales. Específicamente debe cumplir los siguientes criterios: i) tener más de 1000 suscriptores, ii) tener más de 4000 horas de tiempo de visualización durante el último año y iii) seguir las condiciones de la comunidad de la plataforma (Zappin et al., 2022b). No obstante, sigue siendo potestad de YouTube monetizar un video y, a pesar de que pareciera haber indicios de las razones de la exposición a los usuarios de un video, se menciona también en (Zappin et al., 2022a) que no se ha publicado oficialmente una guía sobre lo que puede determinar claramente el éxito de un video, dando así cabida al tipo de análisis realizado en este trabajo.

En cuanto a las técnicas usadas para el modelado de los datos, se tiene:

a. Algoritmo KNN

Tal como se define en (Hastie et al., 2009), el algoritmo de k vecinos más cercanos, también conocido como KNN o k-NN, es uno de los Métodos de Clasificación de aprendizaje supervisado no paramétrico, que utiliza la proximidad para hacer clasificaciones o predicciones sobre la agrupación de un punto de datos individual. Si bien se puede usar para problemas de regresión o clasificación, generalmente se usa como un algoritmo de clasificación, partiendo de la suposición de que se pueden encontrar puntos similares cerca uno del otro.

El objetivo del algoritmo KNN es identificar los vecinos más cercanos de un punto de consulta dado, de modo que se pueda asignar una etiqueta de clase a ese punto. Para hacer esto, KNN tiene algunos requisitos:

i) Métricas de distancia

Para determinar qué puntos de datos están más cerca de un punto de consulta determinado, es necesario calcular la distancia entre el punto de consulta y los otros puntos de datos. Estas métricas de distancia ayudan a formar límites de decisión, que dividen los puntos de consulta en diferentes regiones.

ii) Definición de K

El valor k en el algoritmo KNN define cuántos vecinos se verificarán para determinar la clasificación de un punto de consulta específico. Los valores más bajos de k pueden tener una varianza alta, pero un sesgo bajo, y los valores más grandes de k pueden generar un sesgo alto y una varianza más baja. En general, se recomienda tener un número impar para k para evitar

empates en la clasificación y las tácticas de validación cruzada pueden ayudarlo a elegir la k óptima para su conjunto de datos.

iii) **Ventajas y desventajas del algoritmo KNN**

Al igual que cualquier algoritmo de machine learning, KNN tiene sus puntos fuertes y débiles. Dependiendo del proyecto y la aplicación, puede o no ser la elección correcta.

a. Ventajas

- Fácil de implementar
- Se adapta fácilmente
- Pocos hiperparámetros

b. Desventajas

- No escala bien
- Propenso al sobreajuste

b. **Máquina de soporte vectorial (SVM)**

Este clasificador es uno de los más empleados en tareas de clasificación en aprendizaje automático, ya que ofrece muy buenos resultados de forma general y para la mayoría de datasets.

Como en la mayoría de los métodos de clasificación supervisada, los datos de entrada (los puntos) son vistos como un vector p -dimensional.

La SVM busca un hiperplano que separe de forma óptima a los puntos de una clase de la de otra, que eventualmente han podido ser previamente proyectados a un espacio de dimensionalidad superior.

En ese concepto de "separación óptima" es donde reside la característica fundamental de las SVM: este tipo de algoritmos buscan el hiperplano que tenga la máxima distancia con los puntos que estén más cerca de él mismo. De esta forma, los puntos del vector que son etiquetados con una categoría estarán a un lado del hiperplano y los casos que se encuentren en la otra categoría estarán del otro lado. (*Funcionamiento de SVM - Documentación de IBM*, n.d.)

En la literatura de las SVM, se llama atributo a la variable predictora y característica a un atributo transformado que es usado para definir el hiperplano. La elección de la representación más adecuada del universo estudiado se realiza mediante un proceso denominado selección de características.

Al vector formado por los puntos más cercanos al hiperplano se le llama vector de soporte (*Funcionamiento de SVM - Documentación de IBM*, n.d.)

c. **Regresión logística**

El objetivo primordial que resuelve esta técnica es el de modelar cómo influye en la probabilidad de aparición de un suceso, habitualmente dicotómico, la presencia o no de diversos factores y el valor o nivel

de los mismos. También puede ser usada para estimar la probabilidad de aparición de cada una de las posibilidades de un suceso con más de dos categorías (Molinero, 2001).

La regresión logística puede ser representada de la siguiente manera:

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n$$

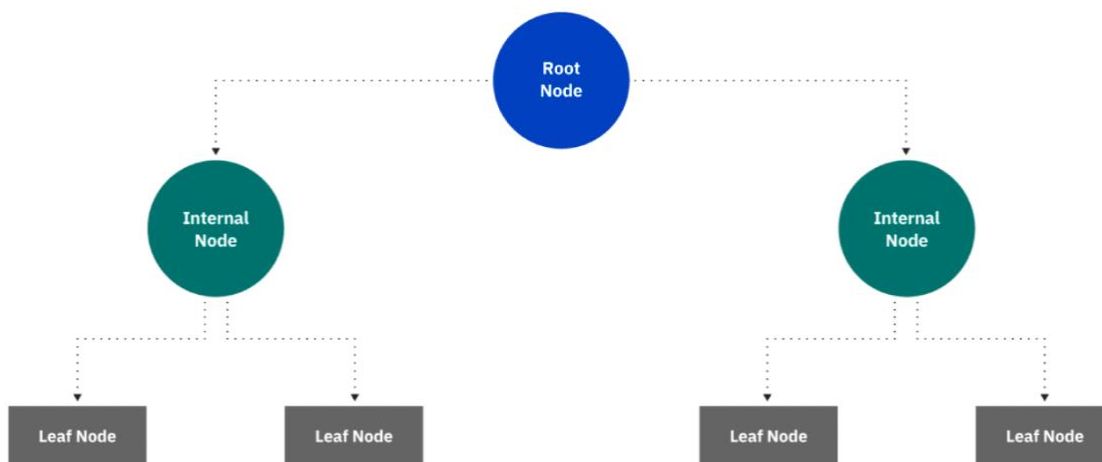
Donde X_1, X_2, \dots, X_n representan las variables de entrada al modelo y $\beta_0, \beta_1, \beta_2, \dots, \beta_n$ representan los coeficientes de regresión de las variables de entrada.

Como se menciona en (Nananukul, 2022), regresión logística tiene varias características que son adecuadas para implementar el modelo de inferencia.

- Para la regresión lineal regular, las probabilidades podrían llegar a ser mayores que uno y menores que cero. Esos valores son inadmisibles para representar probabilidad.
- La función logística proporciona una varianza más realista de la salida, especialmente cuando hay dos resultados. En contraste con la regresión regular donde la varianza de la salida es constante a través de los valores de las entradas, la varianza de la salida de la función logística se acerca a cero mientras p se acerca a uno o cero.

d. Árboles de decisión

Un árbol de decisión es un algoritmo de aprendizaje supervisado no paramétrico, que se utiliza tanto para tareas de clasificación como de regresión. Tiene una estructura de árbol jerárquica, que consta de un nodo raíz, ramas, nodos internos y nodos hoja (IBM, n.d.).



Fuente: IBM, n.d.

El árbol siempre comienza en un nodo raíz sin ramas entrantes. Las ramas salientes del nodo raíz, alimentan los nodos de decisión, es decir los nodos internos. De acuerdo con la información disponible, estos nodos

forman subconjuntos homogéneos que se conocen como nodos hoja o nodos terminales, los cuales “representan todos los resultados posibles dentro del conjunto de datos” (IBM, n.d.).

El aprendizaje del árbol funciona identificando los puntos de división óptimos dentro de él.

Este proceso de división se repite de forma recursiva de arriba hacia abajo hasta que todos o la mayoría de los registros se hayan clasificado bajo etiquetas de clase específicas. Que todos los puntos de datos se clasifiquen o no como conjuntos homogéneos depende en gran medida de la complejidad del árbol de decisión (IBM, n.d.).

En los árboles más pequeños, es más fácil tener nodos de hojas puros, es decir puntos de datos de una sola clase. En contraste, cuando el árbol se hace más grande es más difícil mantener esa pureza, lo que generalmente hace que haya muy pocos datos de subárboles determinados y cuando esto sucede es conocido como fragmentación de datos, lo cual puede crear un overfitting. Por esta razón, para los árboles de decisión, es preferible crear árboles pequeños, “lo cual es consistente con el principio de parsimonia en la Navaja de Occam. Es decir, “las entidades no deben multiplicarse más allá de la necesidad”” (IBM, n.d.).

Para evitar la complejidad y sobreajuste, se usa la técnica de poda, en donde se eliminan las ramas que se dividen en características de poca importancia y son convertidas en una hoja terminal.

Adicionalmente, también se conoce que cuando los datos de entrenamiento son pocos o incoherentes, se produce un overfitting (Lior Rokach and Oded Maimon, 2008).

Otra forma para que los árboles mantengan su precisión, es mediante el uso de Bosques aleatorios.

Existen varios tipos de árboles de decisión, entre los más populares se encuentran (IBM, n.d.):

- **ID3:** Este desarrollo se atribuye a A Ross Quinlan. ID3 es la abreviatura de "Iterative Dichotomiser 3". Este algoritmo aprovecha la entropía y la ganancia de información como métricas para evaluar las divisiones de candidatos.
- **C4.5:** Es considerado una iteración posterior de ID3, también desarrollada por Quinlan. “Puede utilizar la ganancia de información o las proporciones de ganancia para evaluar los puntos de división dentro de los árboles de decisión”.
- **CART:** CART es una abreviatura de "árboles de clasificación y regresión" (“classification and regression trees”), introducido por Leo Breiman. El algoritmo normalmente utiliza “la impureza de Gini para identificar el atributo ideal para la división. La impureza de Gini mide la frecuencia con la que se clasifica incorrectamente un atributo elegido al azar. Cuando se evalúa usando la impureza de Gini, un valor más bajo es más ideal”.

Generalmente, los árboles de decisión son superados por otros algoritmos, sin embargo, son útiles para minería de datos y descubrimiento o conocimiento de la información.

3. Desarrollo metodológico

a. Entendimiento del problema, pregunta de negocio o hipótesis

Para desarrollar este proyecto, nos enfocaremos en el caso específico de DelcaVideography, un canal educativo creado en Medellín, Colombia en el año 2014.

A diciembre de 2022 cuenta con 112,000 suscriptores, se encuentra en la posición 723 del ranking de canales de YouTube en Colombia y en el puesto 1,432 del ranking de canales educativos en YouTube (*Social Blade*, n.d.).

Por cada contenido que se publica, lo ven aproximadamente entre el 0.1% y el 0.3% de sus suscriptores.

Por las razones anteriores el dueño del canal DelcaVideography plantea la necesidad de responder a la pregunta ¿cómo saber si un video que publicará será exitoso o no?

Para esto, con base en el criterio experto, iniciamos con definir qué se considera un video exitoso, así:

Un video es exitoso si su porcentaje de tiempo de visualización es superior a la mediana del porcentaje de tiempo de visualización de los videos analizados (publicaciones entre el 16 de febrero de 2021 y 30 de agosto de 2022).

Específicamente, los datos se normalizaron a 8 días quedando definido el éxito de la siguiente forma (más adelante en el documento se nombrará como la marca_exito3):

$$\frac{\text{Tiempo Visualizaciones 8 días}}{\text{Total Visualizaciones 8 días} * \text{Duración del video}} \geq \text{Mediana}$$

Esta pregunta, decidimos resolverla con un modelo de clasificación, el cual detallaremos más adelante.

Si bien no se conocen todas las variables que influyen en el algoritmo de recomendación de YouTube, se sabe que las primeras horas de publicación de un video son claves para su éxito futuro (teniendo en cuenta las estadísticas del canal y la bibliografía consultada). Por esto, el hecho de apoyar en la validación del éxito de los videos publicados con base en el porcentaje de visualización ayudará al canal a tener más impresiones (recomendaciones) por parte de YouTube y por lo tanto mayores ingresos para el canal.

b. Análisis Exploratorio de Datos

i) Entendimiento de los datos

El canal cuenta con información a *posteriori*, *priori* y *pública* de cada uno de los videos publicados.

Sobre la información a *posteriori*, cuenta con estadísticas agrupadas, referentes a información sociodemográfica del público que ve los videos, entre las que se encuentran:

- a. Área geográfica: países, regiones
- b. Ciudades
- c. Rangos de edad
- d. Género
- e. Estado de suscripción: inscrito o no al canal
- f. Tipo de dispositivo usado

En cuanto a la información a *priori*, definida por el creador de contenidos, se tienen los siguientes tipos de variables:

- g. *Nombre del video*
- h. *Cantidad de recomendaciones incluidas en cada uno de los videos.* Las recomendaciones se refieren a enlaces incluidos en el transcurso del video para recomendar otros contenidos asociados al tema del video (generalmente son otros videos del mismo canal).
- i. *Tiempo de recomendación:* corresponde a los minutos y segundos del momento exacto donde aparece la recomendación. Se crearon 3 variables de este tipo, que corresponden al máximo de recomendaciones incluidas en los videos analizados.
- j. *Publicidad al inicio del video:* marcación de si el creador del contenido seleccionó esta opción para incluir publicidad.
- k. *Publicidad al fin del video:* Marcación de si el creador del contenido seleccionó esta opción para incluir publicidad.
- l. *Cantidad de publicidad durante el transcurso del video:* cuando los videos tienen una duración mayor a 8 minutos, YouTube permite que el creador del contenido seleccione la cantidad de publicidades que quiere incluir en el transcurso del video y su momento exacto.
- m. *Tiempo publicidad:* corresponde a los minutos y segundos del momento exacto donde aparece la publicidad en el transcurso del video. Se crearon 8 variables de este tipo, que corresponden al máximo de publicidades incluidas en los videos analizados.

En cuanto a la información *pública*, mediante un web scraping extrajimos las siguientes variables:

- n. Duración del video
- o. Cantidad de likes

- p. Cantidad de dislikes
- q. Cantidad de visualizaciones
- r. Comentarios de los videos

ii) Preparación de los datos

Para poder procesar y contar con la información necesaria para la estimación de los modelos, se crearon las siguientes variables calculadas:

- s. *Tema del video*: con el dueño del canal se clasificaron cada uno de los videos analizados.
- t. *Porcentaje de visualizaciones del video*:

$$\frac{\text{Tiempo de visualización}}{\text{Duración Video Minutos} * \text{Cant. Visualizaciones}}$$

- u. *Densidad Recomendaciones*:

$$\left(\frac{\text{Duración Video en Minutos}}{\text{Cantidad de Recomendaciones}} \right)^{-1}$$

- v. *Densidad Publicitaria*:

$$\left(\frac{\text{Duración Video en Minutos}}{\text{Cantidad de Publicidad}} \right)^{-1}$$

- w. *Minutos de cada recomendación (3 variables)*
- x. *Minutos de cada publicidad (8 variables)*
- y. *Porcentaje de minutos de cada recomendación (3 variables) sobre el tiempo total del video*
- z. *Porcentaje de Minutos de cada publicidad (8 variables) sobre el tiempo total del video*
- aa. *Día de la semana*
- bb. *Polarity*: calificación obtenida como resultado del procesamiento de los comentarios de los videos, mediante un análisis de sentimientos que describiremos más adelante.
- cc. *Marca éxito*: Calculamos 4 tipos de marcas para decidir cuál se ajustaba mejor a los datos:
 - i. *Marca éxito*: porcentaje de visualización mayor o igual a la mediana del porcentaje de visualización de los datos analizados.
 - ii. *Marca éxito 2*: de acuerdo con el criterio experto del dueño del canal, es el porcentaje de clicks a las impresiones mayor o igual al 3%.
 - iii. *Marca éxito 3*: porcentaje de visualización mayor o igual a la mediana del porcentaje de visualización de los datos analizados, en los primeros 8 días de haberse publicado el video.
 - iv. *Marca éxito 4*: porcentaje de clicks a las impresiones mayor o igual al 3%, en los primeros 8 días de haberse publicado el video.

Finalmente, la marca seleccionada fue marca_exito3.

Adicionalmente, teniendo en cuenta que la información para la construcción del modelo, estaba en tiempo real en YouTube y que cuando la descargamos cada video tenía diferentes antigüedades de haberse publicado, calculamos las siguientes variables para normalizar la información a 2 y 8 días después de haberse publicado el video:

dd. Ingresos estimados

ee. Impresiones: Las impresiones corresponden a la cantidad de veces que YouTube ha promocionado el video tanto a suscriptores como no suscriptores del canal.

ff. Porcentaje de visualización

gg. Porcentaje de clicks de las impresiones

hh. Tiempo de visualización en horas

ii. Cantidad de likes

jj. Cantidad de dislikes

De acuerdo con las variables descritas anteriormente, al final de la preparación de datos el total de las variables disponibles fueron las siguientes 94:

#	Nombre Variable	Tipo de dato
1	nombre_video	object
2	fecha_publicacion	int64
3	visualizaciones	int64
4	tiempo_de_visualizacion_(horas)	float64
5	suscriptores	int64
6	tus_ingresos_estimados_(usd)	float64
7	impresiones	int64
8	porcentaje_de_clicks_de_las_impresiones (%)	float64
9	nombre_video_original	object
10	info_videos	object
11	duracion_video	object
12	id	object
13	datecreated	object
14	likes	int64
15	dislikes	int64
16	rating	float64
17	viewcount	int64
18	tema	object
19	cant_recomendaciones	int64
20	tiempo_recomendación1	object
21	tiempo_recomendación2	object
22	tiempo_recomendación3	object
23	minutos_recom1	float64
24	minutos_recom2	float64

25	minutos_recom3	float64
26	publicidad_inicio	int64
27	publicidad_fin	int64
28	cant_publicidad_durante	int64
29	tiempo_publicidad_1	object
30	tiempo_publicidad_2	object
31	tiempo_publicidad_3	object
32	tiempo_publicidad_4	object
33	tiempo_publicidad_5	object
34	tiempo_publicidad_6	object
35	tiempo_publicidad_7	object
36	tiempo_publicidad_8	object
37	polarity	float64
38	minutos_publi1	float64
39	minutos_publi2	float64
40	minutos_publi3	float64
41	minutos_publi4	float64
42	minutos_publi5	float64
43	minutos_publi6	float64
44	minutos_publi7	float64
45	minutos_publi8	float64
46	fecha_publicacion_ff	datetime64[ns]
47	dia_semana_str	object
48	dia_semana	int64
49	duracion_video_minutos	float64
50	duracion_video_segundos	int64
51	duracion_video_horas	float64
52	porc_visualizacion	float64
53	marca_exito	int64
54	marca_exito2	int64
55	rangos_duracion_video	object
56	rangos_cant_publicidad	object
57	rangos_cant_recomendaciones	float64
58	consecutivo_tema	int64
59	cantida_publicidad	int64
60	cantida_recomendacion	int64
61	duracion_minutos	float64
62	densidad_publicitaria	float64
63	densidad_recomendacion	float64
64	fecha_consulta	object
65	dias_publi_a_cons	int64
66	vistas_2_dias	int64
67	porcentaje_de_clicks_de_las_impresiones_(%)_de_2_dias	int64
68	tiempo_de_visualizacion_(horas)_de_2_dias	int64



69	tus_ingresos_estimados_(usd)_de_2_dias	int64
70	impresiones_de_2_dias	int64
71	likes_de_2_dias	int64
72	dislikes_de_2_dias	int64
73	vistas_8_dias	int64
74	porcentaje_de_clics_de_las_impresiones_(%)_de_8_dias	int64
75	tiempo_de_visualizacion_(horas)_de_8_dias	int64
76	tus_ingresos_estimados_(usd)_de_8_dias	int64
77	impresiones_de_8_dias	int64
78	likes_de_8_dias	int64
79	dislikes_de_8_dias	int64
80	porc_visualizacion_8d	float64
81	porc_visualizacion_2d	float64
82	marca_exito3	int64
83	marca_exito4	int64
84	porc_min_recom1	float64
85	porc_min_recom2	float64
86	porc_min_recom3	float64
87	porc_min_publi1	float64
88	porc_min_publi2	float64
89	porc_min_publi3	float64
90	porc_min_publi4	float64
91	porc_min_publi5	float64
92	porc_min_publi6	float64
93	porc_min_publi7	float64
94	porc_min_publi8	float64

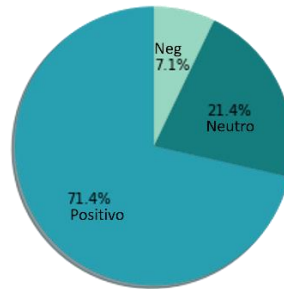
Análisis de sentimientos

Con este proceso se pretende identificar y categorizar la reacción de los espectadores al contenido de un video. Los comentarios obtenidos a través del método mencionado anteriormente son el insumo para la implementación de este análisis. Se siguieron los siguientes pasos:

1. Se detectaron y eliminaron los comentarios hechos por el creador del canal, dado que no aportan información relevante. Son comentarios del tipo:
 - a. ¡Suscríbete y sigue aprendiendo! ► <https://bit.ly/3z6p40B2x1> CURSO DISEÑO GRÁFICO Y AUDIOVISUAL CUPÓN SÚPER DESCUENTO AQUÍ ► <https://bit.ly/3x4IiIL> Curso Completo y Gratuito Corel Draw en YouTube ► <https://bit.ly/3PVi1xs>
 - b. ► Invideo Registro Premium con descuento <https://invideo.io/?ref=delcavideography> ► Invideo cupón de descuento DELCA20

- Comentario Neutro 0: comentarios con calificación igual a 0
- Comentario Negativo -1: comentarios con calificación menor a 0

El valor obtenido permite resumir la reacción de los usuarios al video. Por ejemplo, para  Sigue esta ESTRATEGIA para CREAR tus LOGOS (Bien EXPLICADO ) en CoreIDRAW, se tiene:



Esto nos muestra que de 14 comentarios, 71.4% son positivos, 21.4% son neutros y 7.1% negativos.

- Finalmente se obtiene un promedio de la métrica *polarity* para cada uno de los videos. Esta variable no se encontró como significativa para el modelo, razón por la cual no hace parte de la solución.

iii) Análisis descriptivo e insights importantes (correlación, causa-efecto, etc)

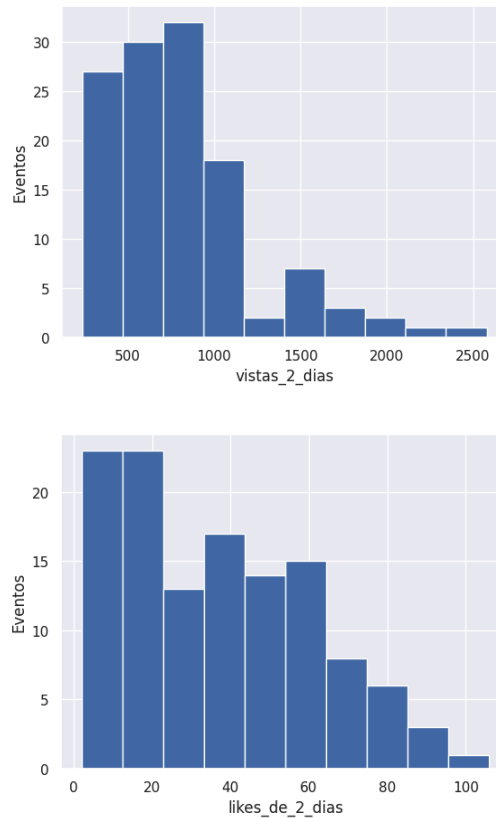
A continuación, se presenta un análisis descriptivo de los datos utilizados para la evaluación del problema.

Inicialmente se consolida la información de los datos obtenido a priori, si bien una vista inicial no permite obtener conclusiones, sí permite empezar a abordar el problema identificando diferencias de escala entre las variables y demás.

	cant_recomendaciones	densidad_publicitaria	dia_semana	duracion_video_minutos	consecutivo_tema	densidad_recomendacion	porcentaje_de_clicks_de_las_impressiones_de_2_dias	vistas_2_dias	porcentaje_visualizacion_2d	likes_de_2_dias	dislikes_de_2_dias	marca_exito_2	marca_exito_3	marca_exito_4
Count	123	123	123	123	123	123	123	123	123	123	123	123	123	123
mean	2732	0,157	1829	13068	9878	0,284	0,179	803,65	0,276	36846	0,358	0,504	0,179	0,504
std	0,702	0,113	1226	8258	5098	0,215	0,406	430378	0,084	24663	0,574	0,502	0,385	0,502
min	0	0	0	1017	0	0	0	235	0,012	2	0	0	0	0
25%	3	0	1	6808	5	0,148	0	484,5	0,221	16,5	0	0	0	0
50%	3	0,194	1	11,9	10	0,234	0	753	0,279	34	0	1	0	1
75%	3	0,23	3	17433	16	0,359	0	971,5	0,317	55,5	1	1	0	1
max	3	0,41	6	47433	17	1,2	2	2578	0,523	106	3	1	1	1

Se identifica como las variables duración del video, consecutivo del tema y vistas de 2 días se presentan en una escala muy diferente a las demás variables analizadas a priori.

Partimos del análisis preliminar de las variables que inicialmente se consideraron de interés de cara a analizar el problema.

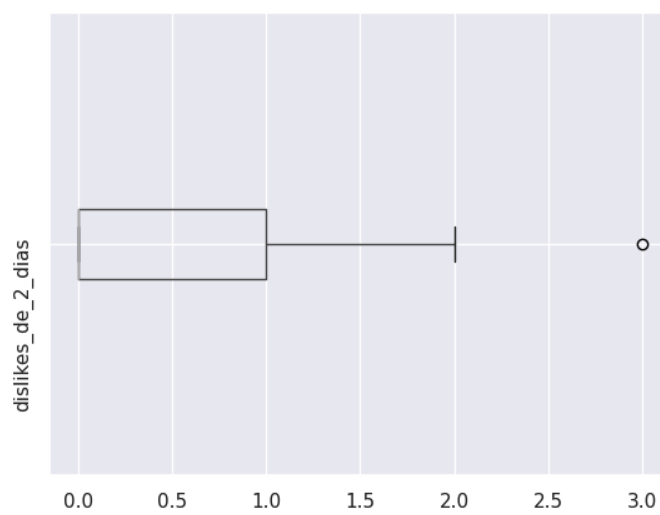
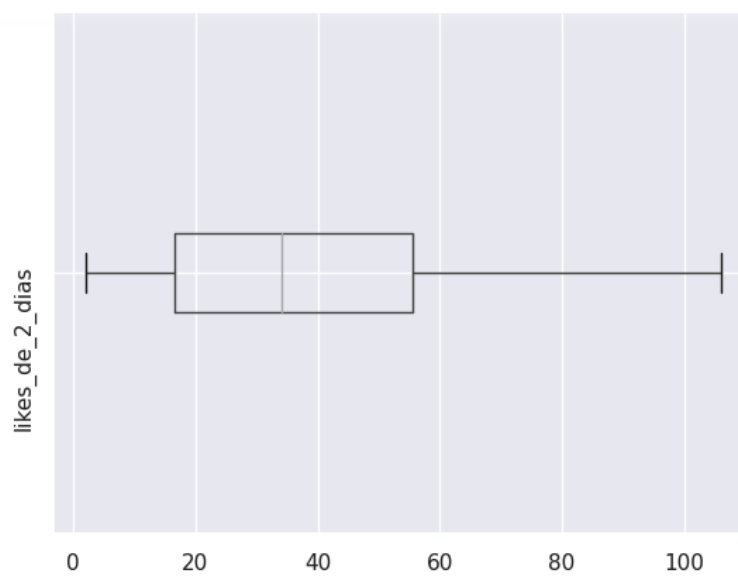
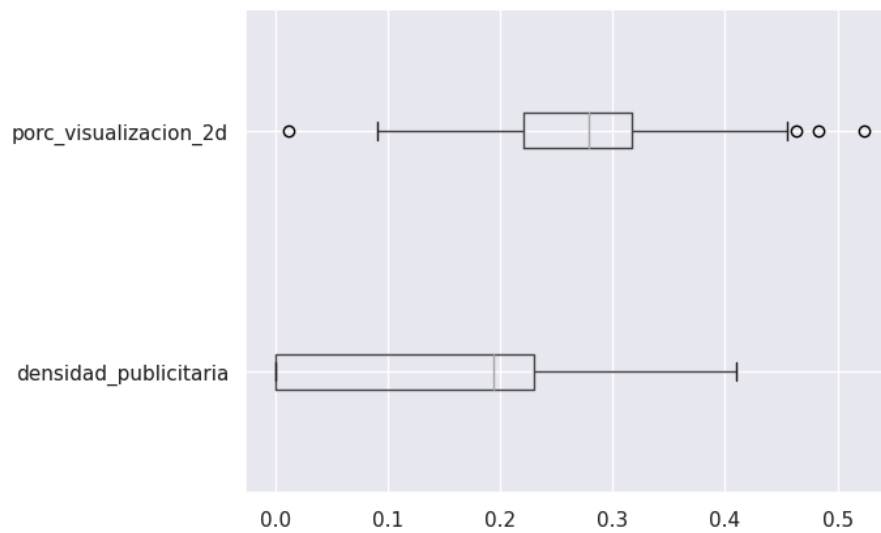


Se observa cómo los histogramas de las variables “Vistas de los primeros dos días” y “Likes de los primeros dos días” tienen un comportamiento similar, lo cual es un indicio de que más adelante nos podemos encontrar una correlación fuerte entre ambas.

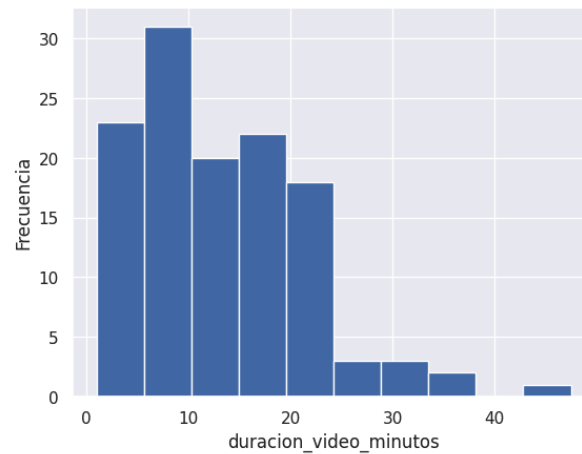
Continuando con el análisis, se plantea según el siguiente boxplot que la variable $\text{porc_visualizacion_2d}^1$ tiene presencia de 4 puntos que pueden ser considerados preliminarmente como outliers, no obstante, también podemos decir que las observaciones de la muestra se encuentran concentradas en el intervalo $[0, 0.45]$.

Por otra parte, la variable $\text{densidad_publicitaria}$ y los likes de dos días no presentan outliers según el boxplot. La cantidad de dislikes a los dos días se concentra en el intervalo $[0, 2]$ y presenta un caso particular donde los dislikes fueron 3.

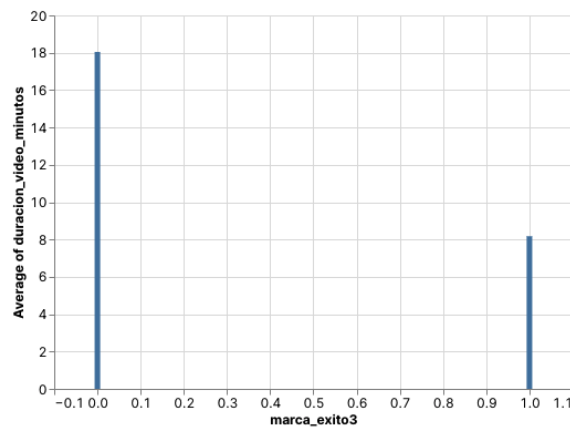
¹ Tiempo de visualización de 2 días / (Duración del vídeo x Cantidad de visualizaciones de 2 días)



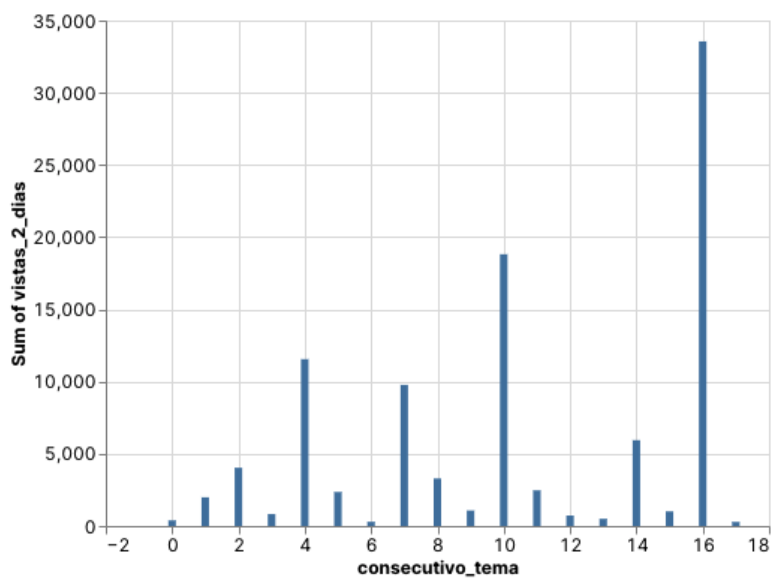
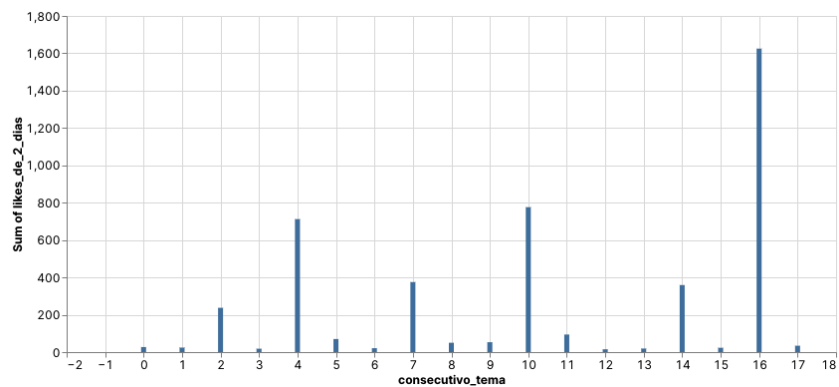
Otra de las variables analizadas exploratoriamente fue la duración del video en minutos ya que según las indagaciones previas sobre el algoritmo de Youtube, puede ser significativa de cara a tener buenas métricas entre los usuarios. Se observa como los videos del canal tienen mayoritariamente una duración de hasta 10 minutos.



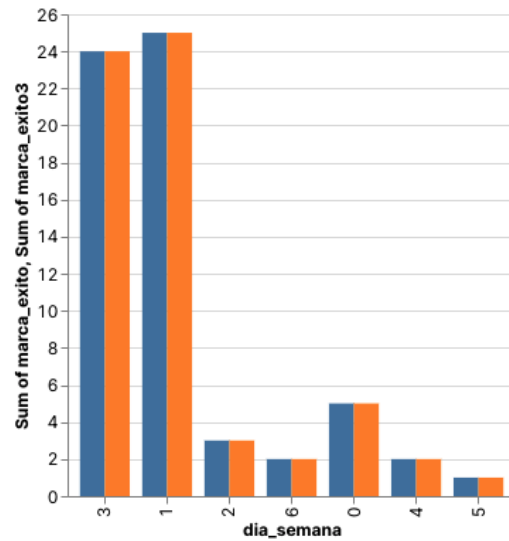
La sospecha de la importancia de la duración en el éxito de los videos se incrementa cuando se analiza de forma preliminar la comparación entre la marca de éxito y la duración de los videos, se identifica claramente como los videos exitosos son los que menor duración promedio tienen.



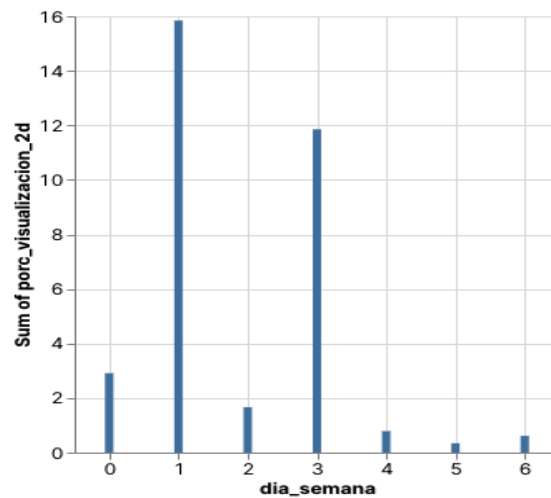
Dentro de la literatura consultada respecto al algoritmo de Youtube, otra de las variables que podría ser considerada como significativa de cara a lograr que un video sea exitoso es la temática del video. Al consultar el comportamiento de los videos según su temática, se observa como aquellos relacionados con Corel Draw (Temas 10 y 16) tienen más visualizaciones y más likes, lo cual podría ayudar a suponer que los videos con dicha temática tienen mayor probabilidad de ser exitosos.



Otra de las explicaciones brindadas por Youtube respecto a su algoritmo (YouTube Creators, 2022) menciona que una variable a considerar para el éxito de un video es la época del año, para este caso de estudio analizamos el éxito según el día de la publicación y se identifica como con la muestra obtenida se marca claramente que los videos más exitosos son aquellos publicados los días martes (1) y jueves (3).



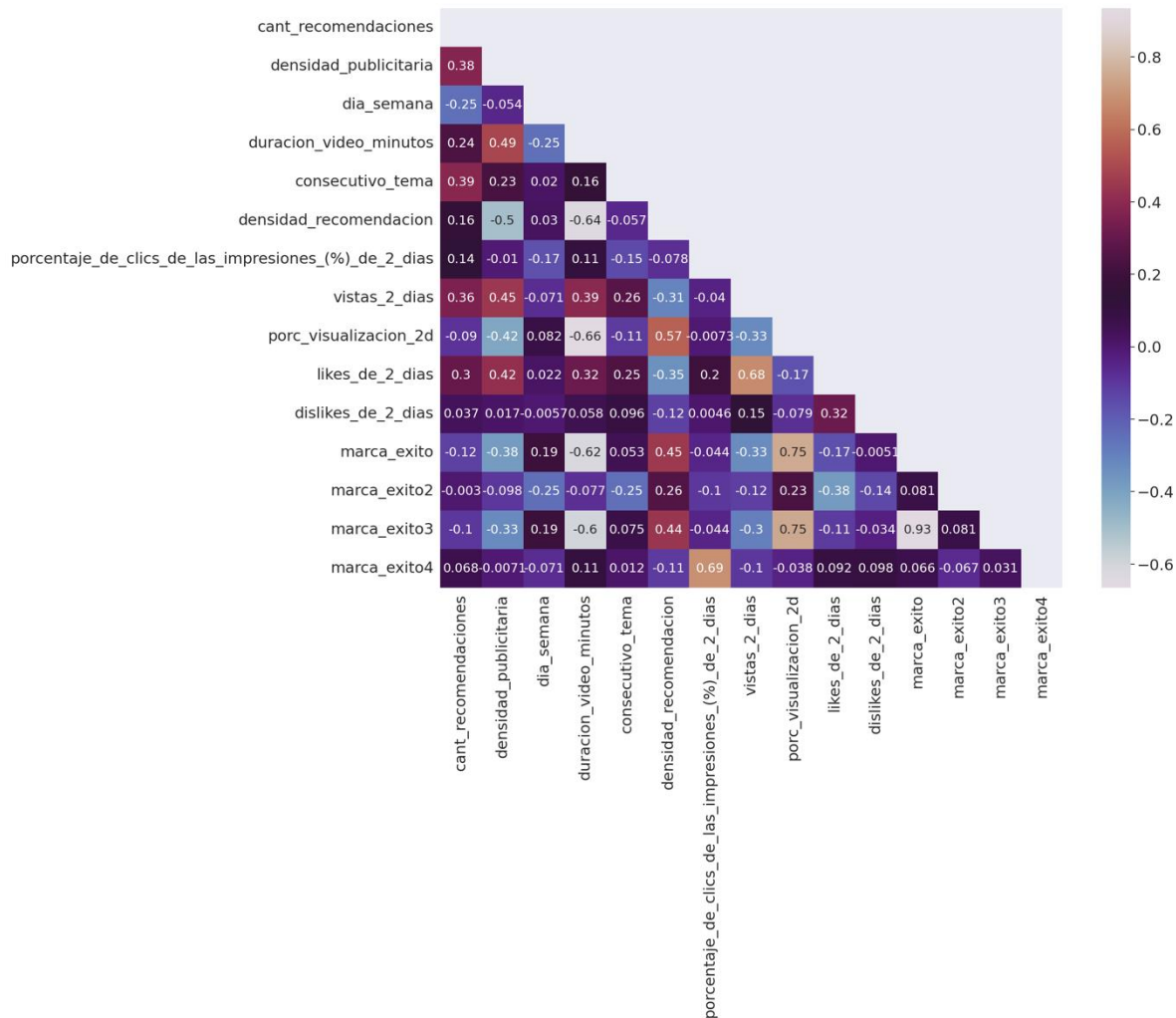
Por lo anterior, no es de sorprender que estos días sean publicados los videos que tienen mayor porcentaje de visualización.



Pasamos ya a construir la matriz de correlaciones de las variables que a priori se tienen para el modelo y se identifica lo siguiente:

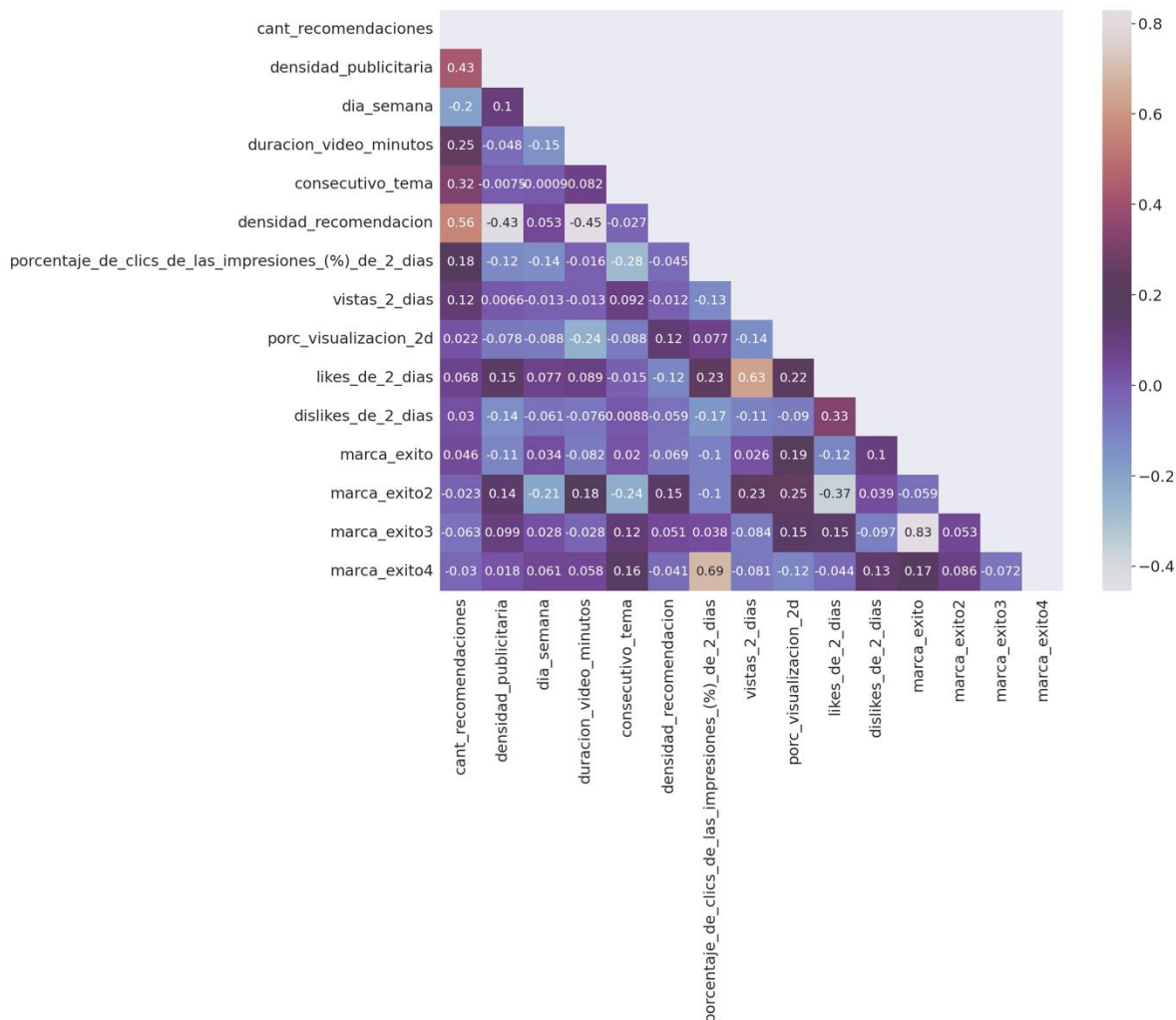
- La variable duracion_video_minutos muestra una correlación negativa con las variables densidad de recomendación, porcentaje de visualización de 2 días, marca de éxito y marca de éxito 3. Esto nos lleva a sospechar que la duración de los videos tiene en efecto una relación inversa con el éxito de los videos.
- Las variables vistas de dos días y porcentaje de visualización de dos días parecen tener una correlación importante con el éxito del video, es decir, se puede sospechar que entre más vistas tenga en los primeros dos días cada video, mayor será su éxito más adelante.
- También es llamativo ver que los likes de los videos no son garantía del éxito del video.

- Dado que no hay otras correlaciones significativas, se interpreta hasta este punto que no hay otras variables además de las mencionadas que puedan ser expresadas en función de las demás. Adicionalmente, hay indicios de que se puede agregar valor al construir un modelo para resolver el problema mencionado ya que no hay una o más variables que estén marcadamente indicando posibilidades de éxito.



Para complementar el análisis de las correlaciones de Pearson, calculamos también la correlación parcial que permite encontrar la relación entre dos variables haciendo que las demás estén fijas.

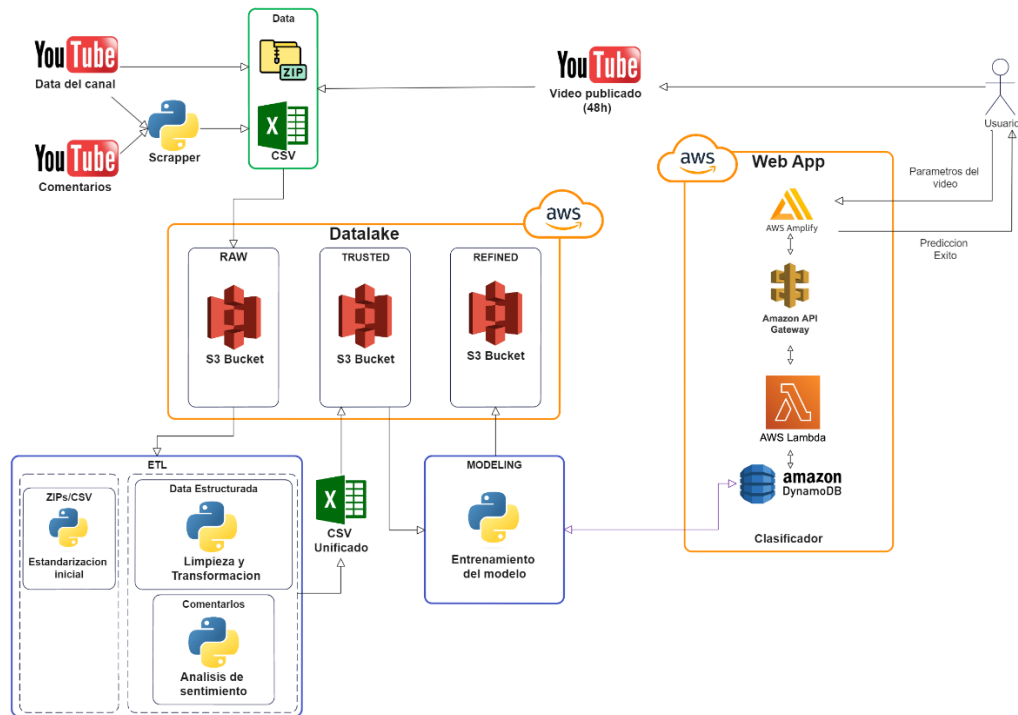
En este caso tampoco encontramos variables que puedan ser explicadas en términos de otras, incluyendo a las diferentes marcas de éxito.



Cómo se ha mencionado anteriormente, también se cuenta con un conjunto de datos a posteriori los cuales no han sido utilizados para el desarrollo del problema y la construcción de los modelos. No agrega valor construir un modelo para predecir si un video será exitoso meses después de la publicación del mismo. La información de dicho análisis exploratorio de datos quedará incluida en el repositorio de GitHub.

iv) Ciclo de vida y arquitectura de los datos

Para el desarrollo del presente proyecto y analizando tanto requerimientos de desarrollo como de usuario hemos implementado una arquitectura batch, con alimentación inicial manual (futuramente automática) cada dos días con todos los datos de los videos hasta el momento, esto debido a que un video puede cambiar entre exitoso y no exitoso en este lapsus de tiempo, tomando esto en cuenta aplicamos la siguiente arquitectura:

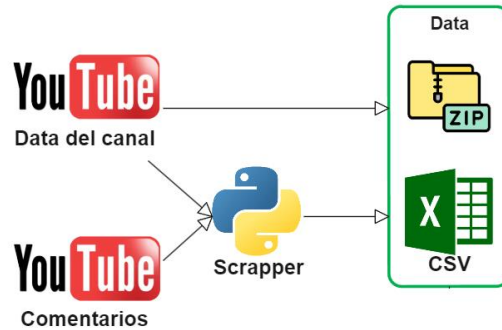


Como podemos ver la estructura general se componen de áreas macro:

- Extracción de datos
- Almacenamiento de datos
- Tratamiento y transformación de datos
- Modelado
- Aplicación web para la implementación del modelo
- Uso continuo

Ahondando en cada una de estas áreas podremos entender a mayor detalle su funcionamiento y rol dentro del funcionamiento de modelo final.

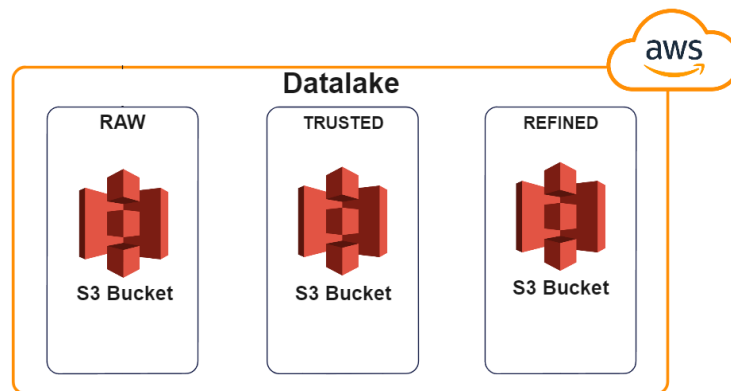
1) Extracción de datos



Como fuentes principales de datos tenemos la información general del canal y sus videos, donde teniendo acceso directo al canal desde la plataforma de *YouTube* podemos exportar datos de los suscriptores (edades, área geográfica, cantidad de suscriptores, etc.) y datos de los videos (nombre, duración, vistas, publicidades, etc.). Estos datos son exportados por la plataforma en archivos *Zip* y deben ser descomprimidos para su uso en un paso siguiente. Adicional, realizamos un scraping con Python para generar *CSVs* que contenga información general de los videos obtenible sin acceso directo al canal (likes, dislikes, fecha publicación, etc.).

Como fuente secundaria de datos tenemos un scraping de los comentarios del video, de esta forma obtenemos de manera rápida y actualizada los datos para un futuro análisis de sentimientos, este al igual que el scraping de la data principal está en formato *CSV*.

2) Almacenamiento de datos



Después de la extracción de los datos, estos son directamente almacenados en nuestro *Bucket AWS S3* en la zona *Raw* la cual será usada en el futuro para la limpieza y tratamiento inicial de los datos.

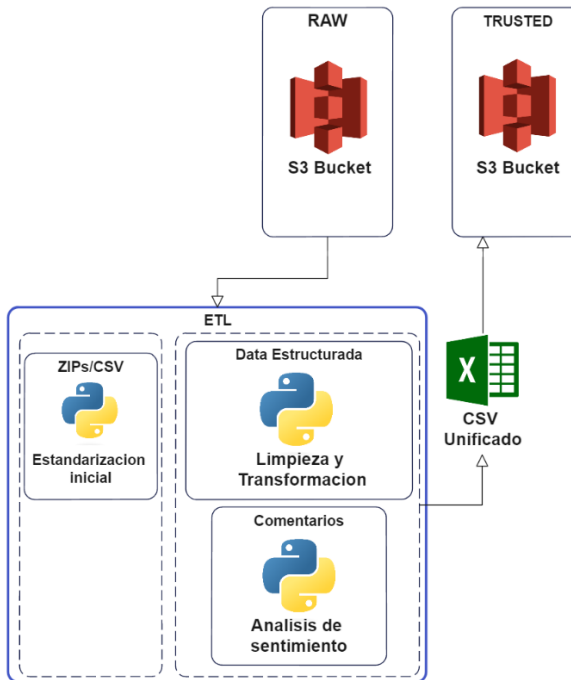
Así mismo en este *Bucket AWS S3* encontramos la zona *Trusted* de nuestra arquitectura, en la cual se almacenarán las tablas ya transformadas y limpiadas para el consumo del modelo y la zona

Refined donde se almacenará el historial de modelos usados y el modelo actual para su rápido uso por parte del usuario y el sistema.

El bucket puede encontrarse en:

<https://s3.console.aws.amazon.com/s3/buckets/proyectointegrador-grupo2-20222?region=us-east-1&tab=objects>.

3) Tratamiento y transformación de datos



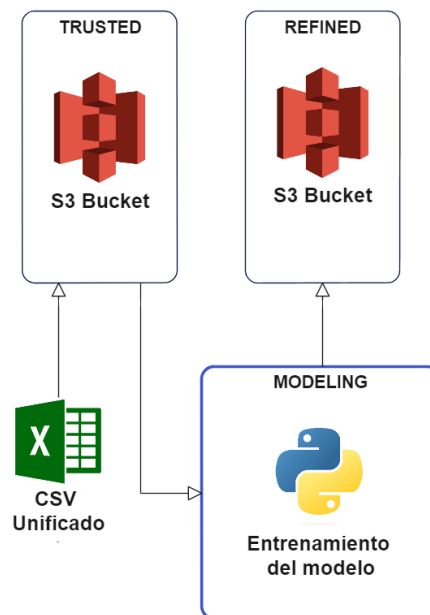
Una vez se encuentra la información en la zona Raw, procedemos a descomprimir los ZIP en formato XLSX para mayor facilidad de uso. Tan pronto tenemos los datos de forma que pueden ser unificados en un Pandas Dataframe procedemos con las siguientes limpiezas y transformaciones:

- (1) Se unifican todas las tablas, tanto las descargadas de la plataforma como las de scraping para tener un Dataframe unificado.
- (2) Reemplazar caracteres con tildes por su equivalente sin acento, este cambio ya que la mayoría de las librerías procesan texto en inglés y no en español, y ya que no se está realizando ningún análisis de texto a los títulos u otros campos de texto para el modelo final podemos utilizar esta limpieza para el fácil uso de las librerías.
- (3) Se eliminan las columnas vacías.
- (4) Se normalizan las fechas para que estén en el mismo formato Datetime.

- (5) Se convierten todos los datos de texto que aún quedan en la tabla a minúsculas y se cambian lo que usan guiones bajos (“_”) intermedios por espacios, para su mejor entendimiento y uso.
- (6) Se renombran las columnas (variables) para su mejor análisis y entendimiento.
- (7) Se agrupan las columnas para dejar solo 1 observación o dato por video en la tabla.
- (8) Se realiza un análisis exploratorio inicial para determinar el tipo de distribución de las variables y filtrar aquellas que pueden generar sesgos en el modelo o que tienen 1 dato que se repite en toda la variable.

Una vez tenemos este Dataframe unificado, limpio y transformado se envía como un CSV a nuestra zona Trusted para el consumo y creación del modelo.

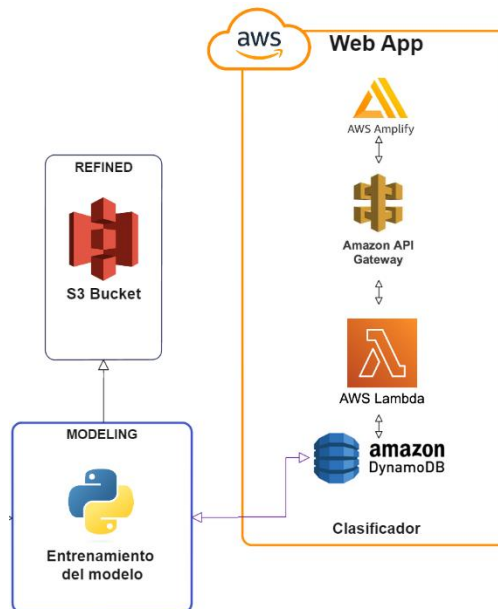
4) Modelado



Ya con nuestra base de datos inicial y unificada, podemos pasar al modelado, en este paso tomamos los datos unificados, limpios y transformados de la zona Trusted y los utilizamos para para la exploración de diferentes modelos y variables que puedan explicar de forma adecuada la marca de éxito seleccionada por el usuario.

Una vez se entrena el modelo, se optimizan los parámetros que este recibe para encontrar la combinación que genera la mejor métrica de precisión, este modelo optimizado con sus parámetros y betas es guardado en la zona *Refined* para el uso rápido de la aplicación web.

5) Aplicación web para la implementación del modelo:



Con relación al despliegue del modelo, este lo planeamos realizar en una segunda iteración, en AWS de la siguiente forma (*Crear Aplicación Web Básica En AWS*, n.d.):

- (1) Usando la Consola de AWS Amplify, crearemos una aplicación Web con un formulario HTML estático, que al final será el encargado de recolectar las variables que son discriminantes para el modelo y por lo tanto permitirá realizar la predicción de si el video será exitoso o no.
- (2) Posteriormente usaremos el servicio de *AWS Lambda* para crear una función sin servidor, la cual se ejecutará bajo demanda. *AWS Lambda* es un servicio en la nube que elimina la necesidad de aprovisionamiento y mantenimiento de infraestructuras complejas y permite la conexión con el Back End.
- (3) En el siguiente paso, usaremos *Amazon API Gateway* para crear una API RESTful que se encargará de invocar la función Lambda desde el navegador web del usuario final, implementando métodos HTTP como Get y Post.
- (4) A continuación, crearemos una tabla de datos usando *Amazon DynamoDB*, la cual almacenará la información recolectada en el formulario estático. Igualmente, usaremos el servicio *AWS Identity and Access Management (IAM)* para activar los permisos necesarios y que la función Lambda escriba en la tabla de DynamoDB mediante una política IAM y SDK AWS con la librería boto3 de Python.
- (5) Finalmente, conectamos el sitio web estático invocando la API REST para que pueda verse la información que se ingresa en el formulario web.

Adicionalmente, habiendo almacenado la información en la tabla *DynamoDB*, se conectará con el modelo entrenado para alimentar y calcular su resultado, y finalmente retornar una respuesta al usuario.

El aspecto del formulario web será de la siguiente forma:

Delcavideography

¿El video será Exitoso?

1. Duración del video en minutos

Registrar minutos:

2. Cantidad de Recomendaciones

Digite la cantidad en el video:

3. Cantidad de Publicidad

Digite la cantidad en el video:

4. Cantidad de Vistas 2 días

Digite la cantidad:

5. Horas de visualizaciones 2 días

Digite la cantidad separada por punto:

6. Cantidad de Likes 2 días

Digite la cantidad:

7. Cantidad de Dislikes 2 días

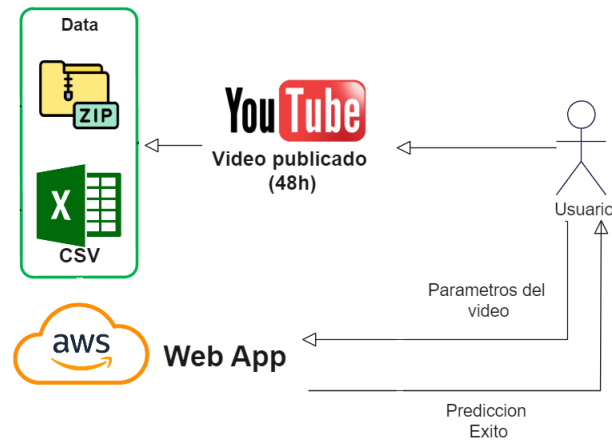
Digite la cantidad:

☐ Acepta los términos y condiciones. Debes estar de acuerdo antes de continuar.

Limpiar

Continuar

6) Uso continuo



Una vez está en línea el aplicativo web el usuario puede enviar los parámetros del video para obtener su predicción. Así mismo, el usuario podrá alimentar el video a la base de datos después de 48 horas de su publicación, permitiendo la refinación del modelo con el tiempo.

c. Selección de modelos, Ingeniería de Características, Entrenamiento y Evaluación

i) Modelos

Partiendo de los datos con la etiqueta de videos exitosos y no exitosos explicada anteriormente y construida con el criterio experto del dueño del canal, concluimos que estamos frente a un problema de aprendizaje supervisado. Tenemos unos datos de entrada que son las variables descritas en la sección anterior y la salida que es la marca de 1 y 0 de éxito de los videos.

En este caso el objetivo de los modelos es la clasificación, permitiendo la implementación de los modelos mencionados en el marco teórico. Estos son: el algoritmo KNN, máquina de soporte vectorial (SVM), regresión logística y árbol de decisión.

ii) Características e Ingeniería de Características

Después de consolidar todos los datos en un solo dataframe con una observación por video, revisamos el porcentaje de poblamiento con el objetivo de detectar nulos. Con esto encontramos que los campos *minutos de la publicidad* y *minutos de la recomendación* tienen información faltante, ya que dependen del número de recomendaciones y publicaciones de cada video. En este caso para la selección e ingeniería de características es fundamental tener un dataframe sin datos nulos, por eso definimos como método para imputar esos datos, llenar los campos faltantes con la duración de cada video, es decir, el minuto final del video.

Para poder implementar métodos estadísticos, observar patrones y correlaciones, es importante aplicar una codificación (*label encoder*) a los campos categóricos, en este caso al día de la semana y el consecutivo del tema.

Por su lado aplicamos el método ***Precisión de variable*** para la remoción de dimensiones. esta técnica estadística nos permite identificar aquellas variables que están siendo explicadas en términos de otras y poder eliminarlas con base en un umbral definido que es un 70%. Todas las variables que se encuentren por encima de este umbral fueron eliminadas.

$$R = 1 - \frac{1}{D - DI}$$

Donde,

D = diagonal de la matriz de variables explicativas.

DI = diagonal de la inversa de la matriz de variables explicativas.

Selección de variables después de aplicar la técnica descrita anteriormente:

```
cant_recomendaciones      0.561862
densidad_publicitaria     0.517274
dia_semana                0.180076
duracion_video_minutos    0.645294
consecutivo_tema          0.236910
densidad_recomendacion    0.675248
porcentaje_de_clicks_de_las_impressiones_(%)_de_2_dias  0.225992
vistas_2_dias             0.575312
porc_visualizacion_2d     0.514356
likes_de_2_dias           0.626334
dislikes_de_2_dias        0.148772
```

Variables seleccionadas y su descripción:

#	Nombre Variable	Tipo de dato	Descripción
1	cant_recomendaciones	int64	Cantidad de recomendaciones incluidas en cada uno de los videos. Las recomendaciones se refieren a enlaces incluidos en el transcurso del video para recomendar otros contenidos asociados al tema del video (generalmente son otros videos del mismo canal).
2	densidad_publicitaria	float64	$\left(\frac{\text{Duración Video en Minutos}}{\text{Cantidad de Publicidad}}\right)^{-1}$
3	duracion_video_minutos	float64	Duración total del video calculada en minutos.

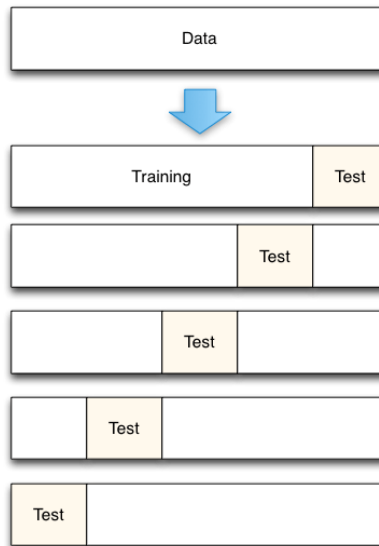
4	densidad_recomendacion	float64	$\left(\frac{\text{Duración Video en Minutos}}{\text{Cantidad de Recomendaciones}}\right)^{-1}$
5	vistas_2_dias	int64	Cantidad de Visualizaciones en los 2 primeros días de haberse publicado el video. Porcentaje del tiempo de visualización en 2 días:
6	porc_visualizacion_2d	float64	$\frac{\text{Tiempo de visualización de 2 días}}{\text{Duración Video Minutos} * \text{Cant. Visualizaciones de 2 días}}$
7	likes_de_2_dias	int64	Cantidad de likes en los 2 primeros días de haberse publicado el video.
8	dislikes_de_2_dias	int64	Cantidad de dislikes en los 2 primeros días de haberse publicado el video.
9	dia_semana	int64	0: lunes, 1: martes, 2: miércoles, 3: jueves, 4: viernes, ...
10	consecutivo_tema	int64	0: Archivo CorelDraw, 1: Características CorelDraw, 2: Consejos CorelDraw, 3: Curso CorelDraw, ...

iii) Entrenamiento

Posterior a la selección de características, quedamos con un dataset de 123 observaciones con lo cual usamos el 30% para testeo (37 observaciones) y 70% para entrenamiento (86 observaciones).

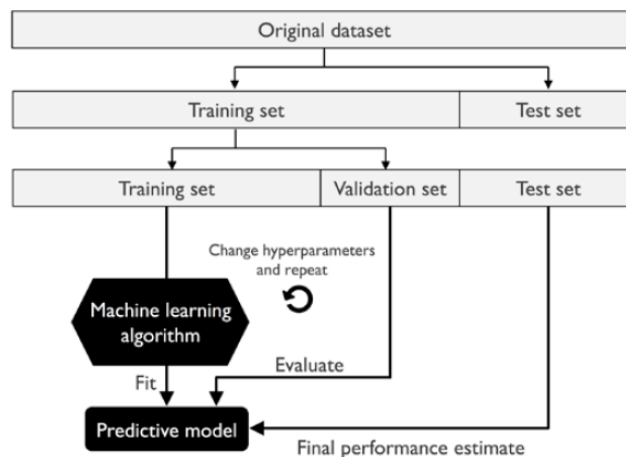
También se utilizó la validación cruzada con los porcentajes de testeo y entrenamiento definidos previamente, con el objetivo de entrenar y testear con diferentes submuestras del conjunto de datos. En este caso generamos 5 folds mediante la función StratifiedKFold del submódulo de model_selection de la librería de Sklearn. Esta función garantiza que cada muestra de datos contenga las clases balanceadas, esto es fundamental para resolver un problema de clasificación.

Esta validación cruzada nos genera una lista de scores con lo cual podemos ver el mínimo, máximo y el score mediano que ese obtiene con el conjunto de datos.



Fuente: mathworks, 2022

Vale la pena aclarar que cada modelo se entrenó con sus parámetros por default, y luego se reentrenaron los modelos, pero ajustando los hiperparámetros de cada algoritmo para encontrar la mejor combinación. Este proceso nos permite de forma iterativa mejorar considerablemente el modelo, esto fue realizado con la función `RandomSearchCV` del submódulo de `model_selection` de Sklearn, esta función es más eficiente computacionalmente con respecto a otras como `GridSearchCV`, porque para nuestro caso seleccionamos 10 combinaciones aleatorias y seleccionamos la mejor.



Fuente: imgur, 2021

I. Regresión Logística

Con el objetivo de identificar aquellas variables significativas, estimamos los parámetros de la regresión logística para todas las variables seleccionadas y definidas previamente. Se usa un nivel de significancia del 5% que permite detectar aquella variable con el valor p más alto para eliminarla y volver a estimar el modelo. Esto quiere decir que se no se rechaza la hipótesis nula de que el parámetro sea igual a cero, concluyendo entonces que esta variable no ayuda a explicar si un video será o no exitoso.

Iteración 1 con todas las variables

Con este modelo inicial obtenemos un Pseudo R-squ de 76.35%, procedemos a eliminar la variable día de la semana que tiene un valor p de 0.985.

```
Optimization terminated successfully.
Current function value: 0.163320
Iterations 11

Logit Regression Results
=====
Dep. Variable:      marca_exito3  No. Observations:      86
Model:              Logit         Df Residuals:              75
Method:              MLE          Df Model:                  10
Date:                Wed, 07 Dec 2022  Pseudo R-squ.:          0.7635
Time:                22:35:39          Log-Likelihood:         -14.046
converged:           True            LL-Null:                -59.401
Covariance Type:     nonrobust        LLR p-value:            3.870e-15
=====

```

	coef	std err	z	P> z	[0.025	0.975]
cant_recomendaciones	0.6683	1.056	0.633	0.527	-1.402	2.739
densidad_publicitaria	-8.5517	7.962	-1.074	0.283	-24.158	7.054
día_semana	0.0140	0.731	0.019	0.985	-1.420	1.447
duracion_video_minutos	-0.8640	0.317	-2.728	0.006	-1.485	-0.243
consecutivo_tema	-0.0270	0.122	-0.222	0.824	-0.266	0.212
densidad_recomendacion	-22.9847	8.308	-2.766	0.006	-39.269	-6.701
porcentaje_de_clicks_impressiones_de_2_dias	0.5508	1.514	0.364	0.716	-2.417	3.518
vistas_2_dias	-0.0070	0.004	-1.623	0.105	-0.015	0.001
porc_visualizacion_2d	69.8460	24.169	2.890	0.004	22.475	117.217
likes_de_2_dias	0.0843	0.047	1.812	0.070	-0.007	0.176
dislikes_de_2_dias	-1.2264	1.142	-1.074	0.283	-3.464	1.011

Iteración 2 eliminando día de la semana:

Con esta segunda iteración, sin tener en cuenta el día de la semana obtenemos un Pseudo R-squ de 76.35% igual al modelo con todas estas variables. En este caso podemos concluir que el día de semana no es un variable que explique si un video será o no exitoso. Procedemos a eliminar el consecutivo tema que tiene el mayor valor-p que es de 0.82.

```

Optimization terminated successfully.
Current function value: 0.163322
Iterations 11

Logit Regression Results
=====
Dep. Variable:      marca_exito3  No. Observations:      86
Model:              Logit         Df Residuals:             76
Method:             MLE           Df Model:                 9
Date:               Wed, 07 Dec 2022  Pseudo R-squ.:           0.7635
Time:               22:31:50         Log-Likelihood:         -14.046
converged:          True            LL-Null:                 -59.401
Covariance Type:    nonrobust       LLR p-value:            1.172e-15
=====

```

	coef	std err	z	P> z	[0.025	0.975]
cant_recomendaciones	0.6677	1.055	0.633	0.527	-1.401	2.736
densidad_publicitaria	-8.5057	7.576	-1.123	0.262	-23.354	6.343
duracion_video_minutos	-0.8653	0.310	-2.794	0.005	-1.472	-0.258
consecutivo_tema	-0.0263	0.116	-0.227	<u>0.820</u>	-0.253	0.201
densidad_recomendacion	-22.9817	8.307	-2.767	0.006	-39.263	-6.700
porcentaje_de_clicks_impressiones_de_2_dias	0.5507	1.516	0.363	0.716	-2.421	3.522
vistas_2_dias	-0.0070	0.004	-1.633	0.102	-0.015	0.001
porc_visualizacion_2d	69.9319	23.762	2.943	0.003	23.359	116.505
likes_de_2_dias	0.0845	0.046	1.855	0.064	-0.005	0.174
dislikes_de_2_dias	-1.2320	1.103	-1.117	0.264	-3.394	0.930

Iteración 3 eliminando consecutivo tema de la iteración 2:

Procedemos a eliminar la variable porcentaje_de_clicks_impressiones_de_2_dias que tiene el valor-p más alto de todos con un 0.68, en este caso el Pseudo R-squ se reduce levemente.

```

Optimization terminated successfully.
Current function value: 0.163627
Iterations 10

Logit Regression Results
=====
Dep. Variable:      marca_exito3  No. Observations:      86
Model:              Logit         Df Residuals:             77
Method:             MLE           Df Model:                 8
Date:               Thu, 08 Dec 2022  Pseudo R-squ.:           0.7631
Time:               17:02:57         Log-Likelihood:         -14.072
converged:          True            LL-Null:                 -59.401
Covariance Type:    nonrobust       LLR p-value:            3.418e-16
=====

```

	coef	std err	z	P> z	[0.025	0.975]
cant_recomendaciones	0.5530	0.928	0.596	0.551	-1.266	2.372
densidad_publicitaria	-8.5088	7.575	-1.123	0.261	-23.356	6.338
duracion_video_minutos	-0.8349	0.274	-3.049	0.002	-1.372	-0.298
densidad_recomendacion	-22.3131	7.694	-2.900	0.004	-37.394	-7.232
porcentaje_de_clicks_impressiones_de_2_dias	0.6084	1.519	0.400	<u>0.689</u>	-2.369	3.586
vistas_2_dias	-0.0068	0.004	-1.655	0.098	-0.015	0.001
porc_visualizacion_2d	68.0193	21.981	3.094	0.002	24.937	111.102
likes_de_2_dias	0.0804	0.041	1.942	0.052	-0.001	0.161
dislikes_de_2_dias	-1.1600	1.057	-1.097	0.273	-3.233	0.913

Iteración 4: Modelo final

Después de eliminar las variables de día de la semana, consecutivo tema y porcentaje de clicks de impresiones de 2 días por tener valores p muy altos, podemos observar que el Pseudo R-squ no cambia drásticamente, podemos concluir que estas variables no aportan información para explicar si un video será o no exitoso.

Por otro lado, con nivel de significancia del 5% definido previamente observamos que las variables significativas son duración del video en minutos, densidad de recomendaciones, porcentaje de visualización de 2 días, likes de 2 días y revisando las vistas a 2 días podríamos considerarla estadísticamente significativa con un nivel del 10%.

Estas variables van alineadas con el criterio experto y tienen sentido a la hora de explicar si un video es exitoso o no, por ejemplo, a medida que la duración del video en minutos y densidad de recomendaciones aumentan reduce la probabilidad de éxito del video. Si aumentan los likes a 2 días y el porcentaje de visualización a 2 días aumenta la probabilidad de éxito del video.

Vale la pena resaltar que algunas variables que no son estadísticamente significativas pero que consideramos relevantes para explicar el fenómeno, existe la posibilidad que esas variables por si solas no aporten mucha información, pero combinadas con otras si ayuden a explicar el fenómeno en estudio.

```

Optimization terminated successfully.
      Current function value: 0.164571
      Iterations 10

Logit Regression Results
=====
Dep. Variable:      marca_exito3      No. Observations:      86
Model:              Logit             Df Residuals:          78
Method:              MLE              Df Model:              7
Date:               Thu, 08 Dec 2022   Pseudo R-squ.:         0.7617
Time:               17:04:55           Log-Likelihood:        -14.153
Converged:           True              LL-Null:               -59.401
Covariance Type:     nonrobust         LLR p-value:           9.785e-17
=====
               coef    std err          z      P>|z|      [0.025    0.975]
-----
cant_recomendaciones    0.6603    0.896     0.737     0.461    -1.096     2.417
densidad_publicitaria  -8.8187    7.506    -1.175     0.240   -23.531     5.893
duracion_video_minutos  -0.8145    0.267    -3.050     0.002    -1.338    -0.291
densidad_recomendacion -22.4671    7.741    -2.902     0.004   -37.640    -7.295
vistas_2_dias           -0.0074    0.004    -1.926     0.054    -0.015     0.000
porc_visualizacion_2d   68.4072   22.042     3.103     0.002    25.205   111.609
likes_de_2_dias          0.0812    0.041     1.994     0.046     0.001     0.161
dislikes_de_2_dias      -1.0793    1.101    -0.980     0.327    -3.237     1.078
=====

```

Después de seleccionar estas variables procedemos a ajustar los modelos con los conjuntos de datos de entrenamiento con la librería sklearn y validar con el conjunto de testeo mediante métricas como la matriz de confusión, f1-score, recall y accuracy.

En el caso de la regresión logística sin optimizar hiperparámetros obtenemos unos scores de:

- Accuracy train: 0.86
- Accuracy test: 0.78

Con la validación cruzada el accuracy estaría entre 0.70 y 0.88 con un k-fold de 5, con una mediana de 0.77.

Optimizando hiperparámetros

- Penalidad: ["l1", "l2", "elasticnet", "none"]
- Constante: entre 0 y 3 100 datos
- Solver (forma de optimización): ["liblinear", "sag", "saga"]

El mejor estimador es: Best estimador: {'solver': 'liblinear', 'penalty': 'l1', 'C': 2.63}

En la documentación para conjuntos de datos pequeños el metodo 'liblinear' funciona bien.

- Testing accuracy train: 0.90
- Testing accuracy test: 0.81

Con esta regresión logística con regulación L1 (LASSO) y constante de 2.63 mejoramos considerablemente las métricas del modelo.

II. Árbol de Decisión

En el caso del árbol de decisión sin optimizar hiperparametros obtenemos unos scores de:

- Accuracy train: 1
- Accuracy test: 0.94

Con la validación cruzada el accuracy estaría entre 0.76 y 1 con un k-fold de 5, con una mediana de 0.88.

Optimizando hiperparámetros

- max_depth: 1 a 25
- Criterio: ["gini", "entropy", "log_loss"]
- min_samples_split: 1 a 10
- min_samples_leaf: 1 a 10

El mejor estimador es: Best estimador: {'min_samples_split': 6, 'min_samples_leaf': 6, 'max_depth': 11, 'criterion': 'gini'}

- Testing accuracy train: 0.97
- Testing accuracy test: 0.94

Con este modelo random forest se reduce levemente el problema de overfitting que se presentaba en el modelo sin optimizar que generaba un accuracy 1 de los datos de entrenamiento.

III. KNN (K-Nearest-Neighbor)

En el caso del KNN sin optimizar hiperparámetros obtenemos unos scores de:

- Accuracy train: 0.77
- Accuracy test: 0.45

Con la validación cruzada el accuracy estaría entre 0.47 y 0.64 con un k-fold de 5, con una mediana de 0.47.

Optimizando hiperparámetros

- n_neighbors: 1 a 30
- pesos: uniform o distance
- Métricas: distancia de manhattan, mahalanobis, euclídea.

Donde,

1. Uniforme: pesos uniformes. Todos los puntos en cada barrio se ponderan por igual (Sklearn, 2022).
2. Distancia: puntos de peso por el inverso de su distancia. En este caso, los vecinos más cercanos de un punto de consulta tendrán una mayor influencia que los vecinos que están más lejos (Sklearn, 2022).

El mejor estimador es: Best estimator: {'weights': 'uniform', 'n_neighbors': 6, 'metric': 'manhattan'}

- Testing accuracy train = 0.75
- Testing accuracy test= 0.54

IV. SVM (Support Vector Machine)

En el caso del SVM sin optimizar hiperparámetros obtenemos unos scores de:

- Accuracy train: 0.63
- Accuracy test: 0.48

Optimizando hiperparámetros

- Kernel: linear, rbf, sigmoid, poly
- Constante: 0 a 2 1000 veces

El mejor estimador es: Best estimator: Best estimator: {'kernel': 'linear', 'C': 1.08}

- Testing accuracy train = 0.81
- Testing accuracy test= 0.62

iv) Evaluación

Los modelos son evaluados a través de los siguientes métodos:

1. *Matriz de confusión*: permite identificar, de forma gráfica, el desempeño que está teniendo el modelo de clasificación supervisada. Las columnas representan los resultados arrojados por el modelo para cada una de las clases, mientras que en las filas figuran los valores reales para cada categoría.

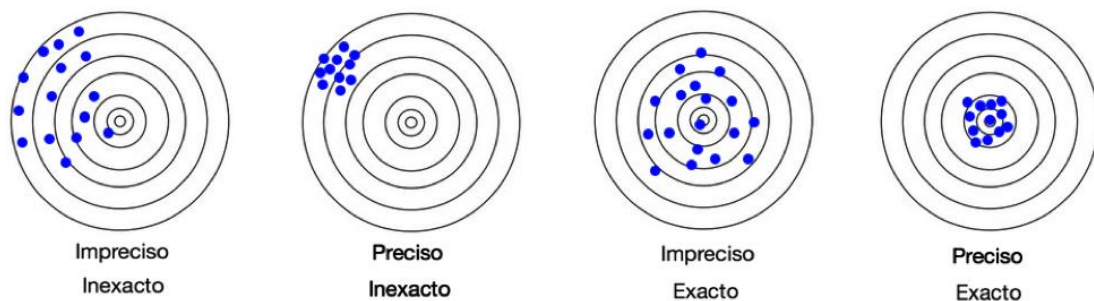
Para interpretar los resultados es útil contar con métricas como la exactitud y la precisión que se describen a continuación.

2. *Accuracy (exactitud)*: es el número de predicciones correctas realizadas sobre el número total de observaciones, en otras palabras, representa la proporción de las predicciones que fueron realizadas de forma correcta. Este indicador podría tomar cualquier valor entre 0 y 1, siendo 1 la máxima calificación, al indicar que todas las predicciones son adecuadas. Es importante resaltar que la exactitud solo será una medida válida del rendimiento del modelo cuando el conjunto de datos es equilibrado.

Se calcula el accuracy tanto para los datos de entrenamiento como los de testeo.

3. *Precision*: por su lado, la precisión se relaciona con la dispersión de los valores predichos por el modelo. En este caso se busca una menor dispersión, que indicaría una mayor precisión. Este valor se calcula como el número de positivos reales dividido por el número de los resultados positivos, incluyendo aquellos que no fueron identificados de forma adecuado (scikit-learn, 2022b).

Es importante resaltar que esta métrica y la anterior son independientes (Moreno Díaz, n.d.). Gráficamente se puede ver.



Fuente: Moreno Díaz, n.d.

4. *Recall (exhaustividad)*: esta métrica explica la proporción de positivos encontrados, comparado con el total de positivos existentes en los datos (scikit-learn, 2022b). No es relevante si los positivos fueron predichos de forma correcta o no. Al igual que la precisión, un valor igual a 1 representará la máxima calificación.

En términos generales, la exactitud indica cuántas veces el modelo arrojó un resultado correcto en general. La precisión es qué tan bueno es el modelo para predecir una categoría específica. La exhaustividad dice cuántas veces el modelo pudo detectar una categoría específica (Moreno Díaz, n.d.).

5. *F1- score*: finalmente, el tradicional F1-score es una media armónica de la precisión y la exhaustividad (scikit-learn, 2022a). La fórmula general se ve de la siguiente forma:

$$F_1 = 2 \cdot \frac{\text{Precisión} \cdot \text{Exhaustividad}}{\text{Precisión} + \text{Exhaustividad}}$$

Por otro lado, un F-score general, usa un factor de tal forma que la exhaustividad se considere β veces tan importante como la precisión (scikit-learn, 2022a).

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{Precisión} \cdot \text{Exhaustividad}}{(\beta^2 \cdot \text{Precisión}) + \text{Exhaustividad}}$$

El valor más alto posible es 1, indicando precisión y exhaustividad perfectas.

Con esta información es posible evaluar los modelos previamente mencionados:

1) *Regresión Logística*

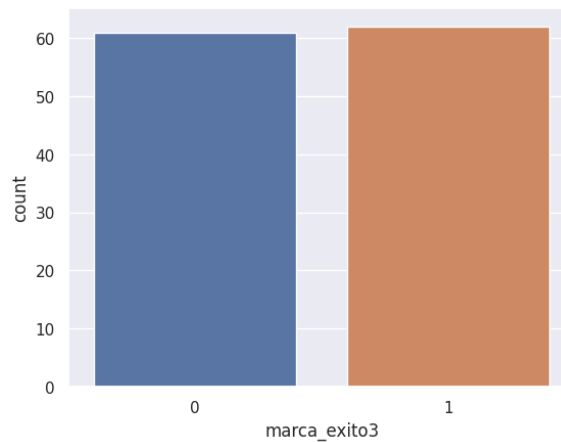
Una vez ajustados los hiperparámetros del modelo como se menciona en la sección anterior, se obtienen las diferentes métricas para la evaluación de este.

En cuanto a exactitud, se tiene:

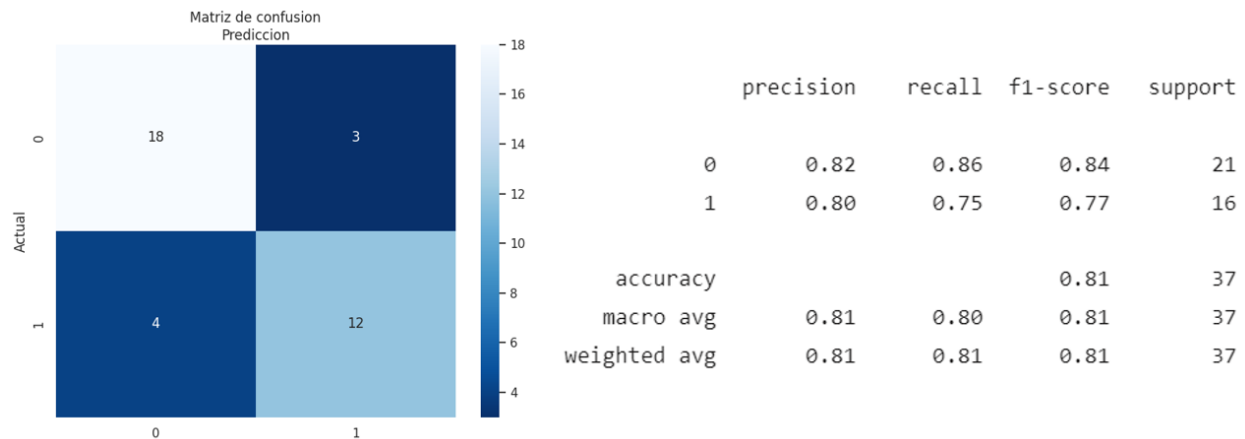
- Training accuracy: 0.90
- Testing accuracy: 0.81

Esto nos indica que, para el momento del entrenamiento, el 90% de las observaciones son adecuadamente clasificadas, mientras que en el testeo el número se reduce al 81%. Esta disminución en la métrica del entrenamiento al testeo es esperable, y la magnitud del cambio no es lo suficientemente grande para indicar un sobreajuste del modelo.

Cabe resaltar que esta métrica puede ser usada en la presencia de una muestra balanceada, como es el caso de los datos aquí utilizados.



Ahora, al hacer la revisión de la matriz de confusión y las métricas precisión, recall y F1-score es posible extraer información adicional. El modelo es más preciso y tiene un recall mayor para la predicción de los videos no exitosos. Esto lleva a que el F1-score sea superior al compararlo con la categoría de videos exitosos.



El F1-score general se situó en 81%, indicando una buena habilidad del modelo para capturar tantos los casos de videos exitosos como no exitosos.

2) *Árbol de decisión*

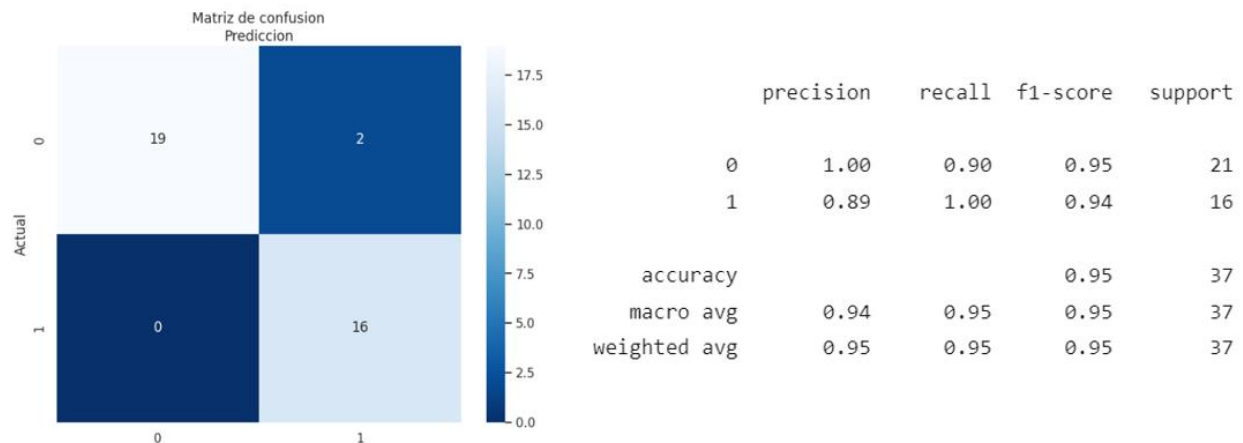
Continuando con la evaluación del árbol de decisión, se obtienen los siguientes resultados:

- Training accuracy: 0.97
- Testing accuracy: 0.94

Esto nos indica que para el momento del entrenamiento, el 97% de las observaciones son adecuadamente clasificadas, mientras que en el testeo el número se reduce al 94%.

La matriz de confusión y las métricas asociadas nos indican que el modelo es ligeramente más preciso para la predicción de los videos no exitosos. Por su lado, el recall es mayor para la categoría de videos exitosos.

En este caso el F1-score general se situó en 95%, indicando una buena habilidad del modelo para capturar tantos los casos de videos exitosos como no exitosos. A pesar de esto, estos resultados podrían apuntar a un overfitting del modelo.



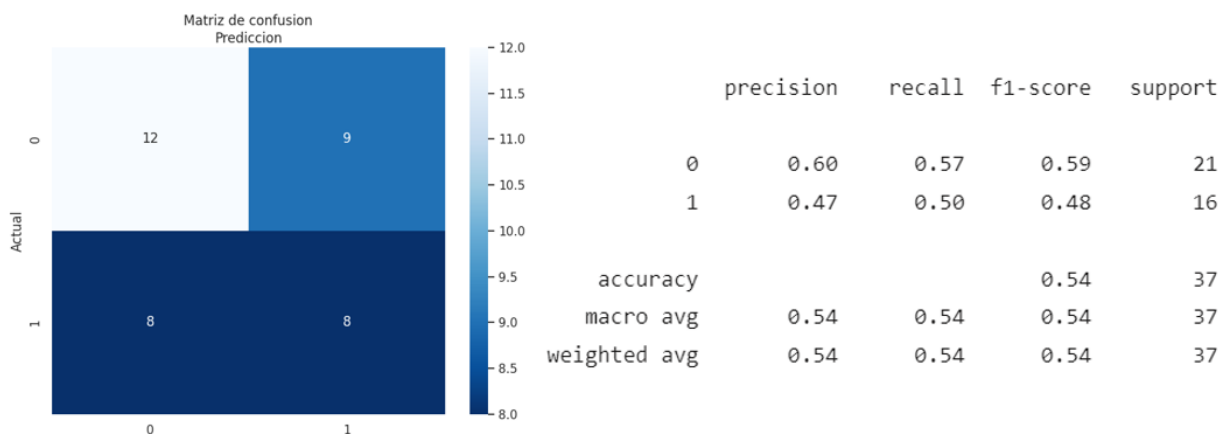
3) KNN

Después del ajuste de hiperparámetros, los resultados arrojados por el KNN nos muestran las siguientes métricas de accuracy:

- Training accuracy: 0.75
- Testing accuracy: 0.54

Al momento del entrenamiento, el 75% de los videos fueron clasificados correctamente, mientras que en el testeo el valor se reduce drásticamente al 54%.

Por su lado, los resultados a continuación muestran que el modelo es más preciso y tiene un recall mayor para la predicción de los videos no exitosos. Además, al revisar el valor de las métricas para los casos



exitosos, es posible observar que disminuyen de manera importante, llevando el F1-score de la clase a ubicarse en 0.49.

Para este caso el F1-score general se situó en 54%, mostrando la falta de habilidad para capturar tanto los casos de videos exitosos como no exitosos. Esto se puede ver en la matriz de confusión, donde es posible encontrar valores similares para cada uno de los cuadrantes.

Este modelo no se consideraría un buen clasificador, pues es cercano a no tener información alguna. Es decir, es comparable con tirar una moneda.

4) *Support Vector Machine (SVM)*

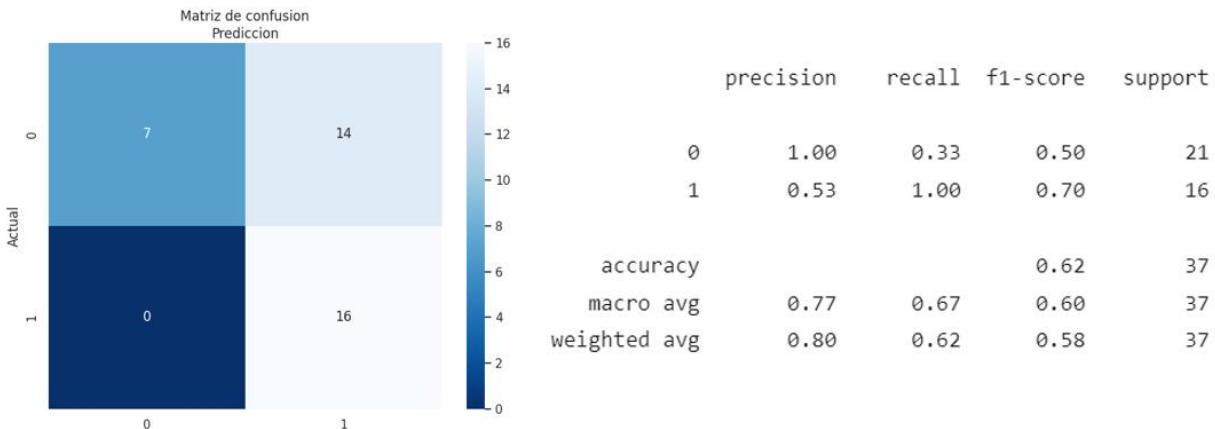
Finalmente, SVM arroja métricas de exactitud correspondientes a:

- Training accuracy: 0.81
- Testing accuracy: 0.62

Similar a la situación presentada en el KNN, existe una diferencia significativa entre la exactitud de entrenamiento y del testeo. Al momento del entrenamiento, el 81% de los videos fueron clasificados correctamente, mientras que en el testeo el valor se reduce drásticamente al 62%.

Los resultados de precisión y recall, por su lado, muestran una distorsión en la habilidad del modelo para predecir los videos no exitosos. A pesar de que cuenta con un valor alto en la precisión, el valor de recall es apenas 0.33, llevando el F1-score de esta clase a 0.5.

Para este caso el F1-score general se situó en 62%. A pesar de esto, los resultados individuales muestran que el modelo es mejor para predecir videos exitosos que no exitosos.



4. Modelo seleccionado

Una vez realizado el análisis de todos los modelos se selecciona la regresión logística como el más adecuado para el proyecto. Esto surge de la facilidad para el entendimiento del modelo, la demanda computacional, y los resultados obtenidos en la evaluación.

5. Análisis y Conclusiones

- A pesar de que el algoritmo de YouTube es desconocido, y que según la documentación consultada son muchas las variables que influyen, mediante las variables propias de un video publicado en un canal, es posible estimar si el contenido será exitoso o no.
- Para tratar de entender un problema de clasificación y las variables que lo explican es importante combinar el análisis estadístico mediante pruebas de hipótesis, intervalos de confianza, entre otros, con modelos de machine learning con el objetivo de no perder interpretabilidad del fenómeno en estudio.
- Desde la parte del modelo, es fundamental todo el proceso de ingeniería de datos y ETL para consolidar los datos de una forma sencilla para el proceso de modelación, luego es importante realizar un proceso detallado de selección de características, acompañado de técnicas estadísticas y criterio experto para lograr seleccionar aquellas variables significativas para el modelo.
- Los modelos por su naturaleza tienen unos parámetros por defecto, los cuales puede ser ajustados para mejorar el rendimiento y la clasificación de los modelos, como se evidenció este proceso mejora significativamente los resultados, también es relevante aplicar varios tipos de modelos y empezar a comparar para seleccionar el mejor en función de métricas definidas previamente.
- Versión 2.0: En una etapa posterior, esperaríamos mejorar los siguientes aspectos:
 - *Web scrapping*: Aparte de la extracción automática de información, también programar momentos específicos para alimentar los modelos.
 - *Análisis no estructurado de script y tópicos*
 - *Datos sintéticos*
 - *Implementación de Front End más robusto*
 - *Backtesting de modelos*

6. Bibliografía

- Así creció YouTube en Latinoamérica en 2020*. (2020, November 14).
<https://folou.co/internet/youtube-audiencias-latinoamerica/>
- Búsqueda y descubrimiento de YouTube: Preguntas frecuentes sobre el algoritmo y el rendimiento - YouTube*. (2022, August 9).
<https://www.youtube.com/watch?v=fApg7tzLjY>
- Crear aplicación web básica en AWS*. (n.d.). Retrieved December 7, 2022, from
<https://aws.amazon.com/es/getting-started/hands-on/build-web-app-s3-lambda-api-gateway-dynamodb/>
- DelcaVideography - YouTube*. (n.d.). Retrieved December 3, 2022, from
<https://www.youtube.com/@DelcaX>
- Funcionamiento de SVM - Documentación de IBM*. (n.d.). Retrieved December 8, 2022, from
<https://www.ibm.com/docs/es/spss-modeler/saas?topic=models-how-svm-works>
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*.
<https://doi.org/10.1007/978-0-387-84858-7>
- IBM. (n.d.). *¿Qué es un árbol de decisión?* Retrieved December 8, 2022, from
<https://www.ibm.com/es-es/topics/decision-trees>
- imgur. (2021, April 6). *tuning hyperparameters*. <https://imgur.com/SPOUx7W>
- Kikuchi, Y., Nishimura, I., & Sasaki, T. (2022). Wild birds in YouTube videos: Presence of specific species contributes to increased views. *Ecological Informatics*, 71, 101767.
<https://doi.org/10.1016/J.ECOINF.2022.101767>
- Lior Rokach and Oded Maimon. (2008). *Data mining with decision trees: theory and applications*. World Scientific.
- mathworks. (2022). *mathworks*. <https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/233459a6-523d-4cf7-91f3-ff539a1b58ce/f6c9980c-ed0d-4564-8289-e95c9274b48e/images/screenshot.png>
- Molinero, L. M. (2001). *LA REGRESION LOGISTICA*.
<https://web.archive.org/web/20130629005527/http://www.seh-lelha.org/rlogis1.htm>
- Moreno Díaz, Ó. (n.d.). *Exactitud y precisión*. Retrieved December 7, 2022, from
https://formacion.intef.es/pluginfile.php/246707/mod_resource/content/1/exactitud_y_precisin.html
- Nananukul, N. (2022). An Inference Model for Online Media Users. *Journal of Data Science*, 11(1), 143–155. [https://doi.org/10.6339/JDS.2013.11\(1\).1129](https://doi.org/10.6339/JDS.2013.11(1).1129)
- scikit-learn. (2022a). *sklearn.metrics.f1_score — scikit-learn 1.2.0 documentation*.
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html
- scikit-learn. (2022b). *sklearn.metrics.precision_recall_fscore_support — scikit-learn 1.2.0 documentation*. https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html

learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html#sklearn.metrics.precision_recall_fscore_support

Sklearn. (2022). *Sklearn KNeighborsClassifier*.

Social Blade. (n.d.). Retrieved December 7, 2022, from <https://socialblade.com/youtube/channel/UCDDQf-3UNw3B66FzV53KykQ>

YouTube Creators. (2022, August 9). *Búsqueda y descubrimiento de YouTube: Preguntas frecuentes sobre el algoritmo y el rendimiento - YouTube*. <https://www.youtube.com/watch?v=fApg7tzlLjY>

Youtube: usuarios a nivel mundial 2012-2021 | Statista. (2022, April 27). <https://es.statista.com/previsiones/1289041/usuarios-de-youtube-en-todo-el-mundo>

Zappin, A., Malik, H., Shakshuki, E. M., & Dampier, D. A. (2022a). YouTube Monetization and Censorship by Proxy: A Machine Learning Prospective. *Procedia Computer Science*, 198, 23–32. <https://doi.org/10.1016/J.PROCS.2021.12.207>

Zappin, A., Malik, H., Shakshuki, E. M., & Dampier, D. A. (2022b). YouTube Monetization and Censorship by Proxy: A Machine Learning Prospective. *Procedia Computer Science*, 198, 23–32. <https://doi.org/10.1016/J.PROCS.2021.12.207>