

77.50 Introducción a los sistemas inteligentes

## Trabajo Práctico Final

### Integrantes:

Alumno	padron
Llauró, Manuel Luis	95736
Rial, Sebastián Andrés	90309
Blanco, Sebastian Ezequiel	98539

**Fecha de Entrega:** 24/06/2019

**GitHub:**

<https://github.com/BlancoSebastianEzequiel/7750-TPFinal>

# Índice

<b>1. Fase 1: Comprensión del negocio</b>	<b>1</b>
1.1. Determinar los objetivos del negocio	1
1.1.1. Escenario actual	1
1.1.2. Objetivos del negocio	1
1.1.3. Criterios de éxito del negocio	1
1.2. Evaluación de la situación	1
1.2.1. Inventario de recursos	1
1.2.2. Requisitos, supuestos y restricciones	1
1.2.2.1. Requisitos	1
1.2.2.2. Supuestos	1
1.2.2.3. Restricciones	2
1.2.3. Riesgos y contingencias	2
1.2.4. Terminología	2
1.2.4.1. Glosario de términos del negocio	2
1.2.4.2. Glosario de términos de la minería de datos	2
1.2.5. Costos y beneficios	3
1.3. Determinar objetivos de Minería de Datos	3
1.3.1. Objetivo de minería de datos	3
1.3.2. Criterios de éxito de minería de datos	4
1.4. Realizar el Plan del Proyecto	4
1.4.1. Plan de proyecto	4
1.4.2. Validación inicial de las herramientas	4
<b>2. Fase 2: Comprensión de los datos</b>	<b>5</b>
2.1. Recolectar los datos Iniciales	5
2.1.1. Reporte de la recolección de datos iniciales	5
2.2. Descubrir datos	5
2.2.1. Reporte de descripción de datos	5
2.3. Exploración de los datos	6
2.3.1. Reporte de exploración de datos	6
2.3.1.1. Analisis de categorias	6
2.3.1.2. Analisis de vistas en funcion de likes	7
2.3.1.3. Analisis de vistas en funcion de la diferencia de dias entre la fecha de tendencia y la de publicacion	10
2.3.1.4. Analisis de vistas en funcion de la diferencia de likes y dislikes	11
2.3.1.5. Analisis del progreso de views del video con mayor views del set de datos.	12
2.3.1.6. Analisis del progreso de views de los videos con mayor, mediano y menor vistas del set de datos.	13

2.3.1.7.	Análisis de la cantidad de views según el largo del título del video. . . . .	14
2.3.1.8.	Análisis la cantidad de videos borrados por categoría. . . . .	15
2.3.1.9.	Análisis de la cantidad de vistas de videos según tenga o no descripción. . . . .	16
2.4.	Verificación de calidad de datos . . . . .	17
2.4.1.	Reporte de calidad de datos . . . . .	17
<b>3.</b>	<b>Fase 3: Preparación de los datos . . . . .</b>	<b>18</b>
3.1.	Reporte de calidad de los datos . . . . .	18
3.2.	Seleccionar los datos . . . . .	19
3.2.1.	Inclusión/Exclusión de datos . . . . .	19
3.3.	Limpiar los datos . . . . .	19
3.3.1.	Reporte de limpieza de datos . . . . .	19
3.4.	Estructurar los datos . . . . .	19
3.4.1.	Derivación de atributos . . . . .	19
3.4.2.	Generación de registros . . . . .	20
3.5.	Integrar los datos . . . . .	20
3.5.1.	Unificación de los datos . . . . .	20
3.6.	Formato de los datos . . . . .	20
3.6.1.	Reporte de formato de los Datos . . . . .	20
<b>4.</b>	<b>Fase 4: Modelado . . . . .</b>	<b>21</b>
4.1.	Seleccionar una técnica de modelado . . . . .	21
4.1.1.	Técnica de modelado . . . . .	21
4.1.2.	Supuestos de modelado . . . . .	21
4.2.	Generar el diseño de las pruebas . . . . .	21
4.2.1.	Diseño de las pruebas . . . . .	21
4.3.	Construir el modelo . . . . .	21
4.3.1.	Configuración de parámetros . . . . .	21
4.3.1.1.	Árbol J48 . . . . .	21
4.3.1.2.	Multilayer Perceptron . . . . .	22
4.3.2.	Modelos . . . . .	23
4.3.2.1.	Resultados de Árbol de decisión J48: . . . . .	23
4.3.2.1.1.	Matriz de confusión . . . . .	23
4.3.2.1.2.	Resultados de la clasificación . . . . .	23
4.3.2.1.3.	Árbol . . . . .	24
4.3.2.1.4.	Reglas . . . . .	25
4.3.2.1.5.	Árbol con reglas . . . . .	32
4.3.3.	Descripción del modelo . . . . .	33
4.4.	Evaluar el modelo . . . . .	33
4.4.1.	Evaluación del modelo . . . . .	33
4.4.2.	Revisión de la configuración de parámetros . . . . .	33

<b>5. Fase 5: Evaluación</b>	<b>34</b>
5.1. Evaluar Resultado	34
5.1.1. Valoración de los resultados de minería de datos	34
5.1.2. Modelo aprobado	34
5.2. Proceso de revisión	34
5.2.1. Revisión del proceso	34
5.3. Determinar Próximos pasos	34
5.3.1. Listado de posibles acciones	34
<b>6. Conclusiones</b>	<b>35</b>

# **1. Fase 1: Comprensión del negocio**

## **1.1. Determinar los objetivos del negocio**

### **1.1.1. Escenario actual**

Al momento de escribir este informe, la empresa en cuestión tiene interés en desarrollar un canal de youtube, y requiere medir de alguna forma las probabilidades de éxito de llevar a cabo videos populares.

### **1.1.2. Objetivos del negocio**

El objetivo es en este caso poder cuáles son las condiciones que un canal de youtube y sus videos debe cumplir para que los mismos sean exitosos en la plataforma.

### **1.1.3. Criterios de éxito del negocio**

El proyecto se considerará exitoso si se llegan a detectar las variables clave que influyen en que los videos del canal de youtube sean exitosos o no, y en qué nivel influye cada una de ellas. Se considerará que los videos del canal de youtube si los mismos obtienen un porcentaje de X likes y más de 2 millones de vistas por video.

## **1.2. Evaluación de la situación**

### **1.2.1. Inventario de recursos**

Para desarrollar el presente proyecto se cuenta con una amplia gama de recursos que asegura un desarrollo de calidad y confianza del mismo. Se cuenta con un set de datos extraído de la pagina de kaggle sobre videos de youtube que fueron subidos a la plataforma. Además se cuenta con herramientas de software de análisis y visualización de datos líderes en el mercado, como Pandas. Por último, se cuenta con personal altamente calificado para la correcta interpretación de los mismos.

### **1.2.2. Requisitos, supuestos y restricciones**

#### **1.2.2.1 Requisitos**

Contar con datos suficientes y sobre todo representativos de la plataforma youtube

#### **1.2.2.2 Supuestos**

Los datos en estudio son lo suficientemente correctos como para poder sacar conclusiones confiables a partir de ellos.

### 1.2.2.3 Restricciones

Se cuenta solamente con datos de videos subidos a youtube Estados Unidos. No se asegura que los resultados sean válidos para otros países.

### 1.2.3. Riesgos y contingencias

Si bien se puede llevar a cabo un análisis lo más riguroso posible, siempre existe la posibilidad de que un video pueda no ser exitoso porque hay que tener en cuenta que el éxito de cada video puede depender de muchos factores de incertidumbre, y si bien se cumplen patrones, no hay nada que asegure al 100%. En caso de detectar que un video no está teniendo el éxito esperado, habrá que recurrir a las diversas métricas que pueda ofrecer Youtube a desarrolladores y analizar la situación.

### 1.2.4. Terminología

#### 1.2.4.1 Glosario de términos del negocio

**Youtube:** es una plataforma que te permite subir videos a partir de la creación de un canal.

**Canal de youtube:** Es el equivalente a crearse una cuenta en cierta aplicación, en la cual se almacenan los videos subidos.

**Categoría:** define el nombre de un grupo de videos con cualidades comunes.

**likes/dislikes:** Es un atributo de un video que muestra la cantidad de usuarios que le dieron like al video en cuestión.

**Vistas:** Es la cantidad de veces que el video fue visto.

**Video exitoso:** Aquel cuya cantidad de vistas es mayor a 200000 y promedio de likes versus dislikes es mayor al 80 %.

#### 1.2.4.2 Glosario de términos de la minería de datos

**Atributo:** dato sobre alguna característica de las observaciones.

**Atributo relevante:** atributo que juega un papel principal en la clasificación, por lo que la clase dependerá en alguna medida de qué valor tenga.

**Registro:** fila que representa una observación, está compuesto de atributos.

**Dataset:** conjunto de datos a ser utilizados para la ejecución de los algoritmos de Data Mining, está compuesto de registros.

**Filtrado de atributos:** especifica al dataset formado considerando sólo los atributos relevantes.

**Regla:** es una implicación, que representa una acción mediante una condición. Sigue la estructura “Si..., entonces...”.

**Soporte:** es la relación entre la cantidad total de registros del dataset que cumplen la regla y la cantidad de observaciones procesadas.

**Confianza:** es la relación entre la cantidad total de observaciones de la clase mayoritaria que cumplen la regla y la cantidad de observaciones que fueron afectadas por esa misma regla.

**Captura:** es la relación entre la cantidad de observaciones de la clase mayoritaria que cumplen la regla y la cantidad de observaciones procesadas pertenecientes a esa misma clase.

**Coeficiente de correlación de Pearson:** es la estadística de prueba que mide la relación estadística, o asociación, entre dos variables continuas. Es conocido como el mejor método para medir la asociación entre variables de interés porque se basa en el método de covarianza. Da información sobre la magnitud de la asociación, o correlación, así como la dirección de la relación

#### 1.2.5. Costos y beneficios

El beneficio del proyecto es detectar las características que hacen a un video de youtube sea exitoso. De esta manera se pueden tener en cuenta ciertos parametros para poder desarrollar un video que exitoso basandose en la historia. Al ser un trabajo final educativo, no hay costos.

### 1.3. Determinar objetivos de Minería de Datos

#### 1.3.1. Objetivo de minería de datos

El objetivo de minería de datos es el análisis de los datos obtenidos a partir de la información disponible, buscando obtener así información relevante que permita predecir las condiciones bajo las cuales una aplicación es exitosa.

### **1.3.2. Criterios de éxito de minería de datos**

Selección de al menos 4 reglas con soporte mayor o igual al 20

## **1.4. Realizar el Plan del Proyecto**

### **1.4.1. Plan de proyecto**

- Recolección de datos: 5 horas.
- Preparación de datos: 5 horas.
- Ejecución del algoritmo de Inducción: 4 horas.
- Análisis de resultados de algoritmo de Inducción: 4 horas.
- Combinación de resultados: 3 horas.
- Elaboración de reporte: 7 horas.

### **1.4.2. Validación inicial de las herramientas**

Se utilizarán las siguientes herramientas:

- Python
- Pandas
- Weka
- Jupiter Notebook
- Numpy



## 2. Fase 2: Comprensión de los datos

### 2.1. Recolectar los datos Iniciales

Los datos utilizados durante el transcurso del proyecto fueron obtenidos gratuitamente en el sitio de Kaggle y el mismo se encuentra en formato csv.

#### 2.1.1. Reporte de la recolección de datos iniciales

- El data set se encuentra en la carpeta “src/data/” del proyecto.
- Se uso pandas como metodo de recoleccion de datos.
- Para poder levantar el set de datos se usa como separado una coma.

### 2.2. Descubrir datos

#### 2.2.1. Reporte de descripción de datos

Se cuenta con dos datasets: por un lado se tienen datos estadísticos y generales de los videos descritos en 16 columnas; por otro lado hay un archivo json con el nombre de cada categoria segun el numero asignado. A continuación, los nombres de los features con su correspondiente descripción y tipo de dato:

Mombre de variable	tipo de dato	Descripcion
video_id	alfanumerica	identifica a cada video
trending_date	date	Es una fecha en un dia especifico luego de la publicacion del video
title	alfanumerica	Es el titulo del video
channel_title	alfanumerica	Es el nombre del canal del usuario que publico el video
category_id	numeric	it is the number that represent a category name
publish_time	timestamp	momento exacto de la publicacion del video
tags	alfanumerico	Etiquetas que se pueden agregar al video
views	numeric	Cantidad de vistas del video en el momento de la fecha de trending_date
likes	numeric	Cantidad de me gusta que el video tiene dados por usuarios de la plataforma youtube
dislikes	numeric	Cantidad de no me gusta que el video tiene dados por usuarios de la plataforma youtube
comment_count	numeric	Cantidad de comentarios del video
thumbnail_link	url	enlace a
comments_disabled	booleano	Dice si tiene o no los comentarios habilitados
ratings_disabled	booleano	Dice si tiene o no los likes/dislikes habilitados
video_error_or_removed	booleano	Dice si el video sufrio un error o fue borrado en el momento de la fecha de trending_date
description	alfanumeric	Texto escrito por el creador del video que describe el mismo

## 2.3. Exploración de los datos

### 2.3.1. Reporte de exploración de datos

#### 2.3.1.1 Analisis de categorias

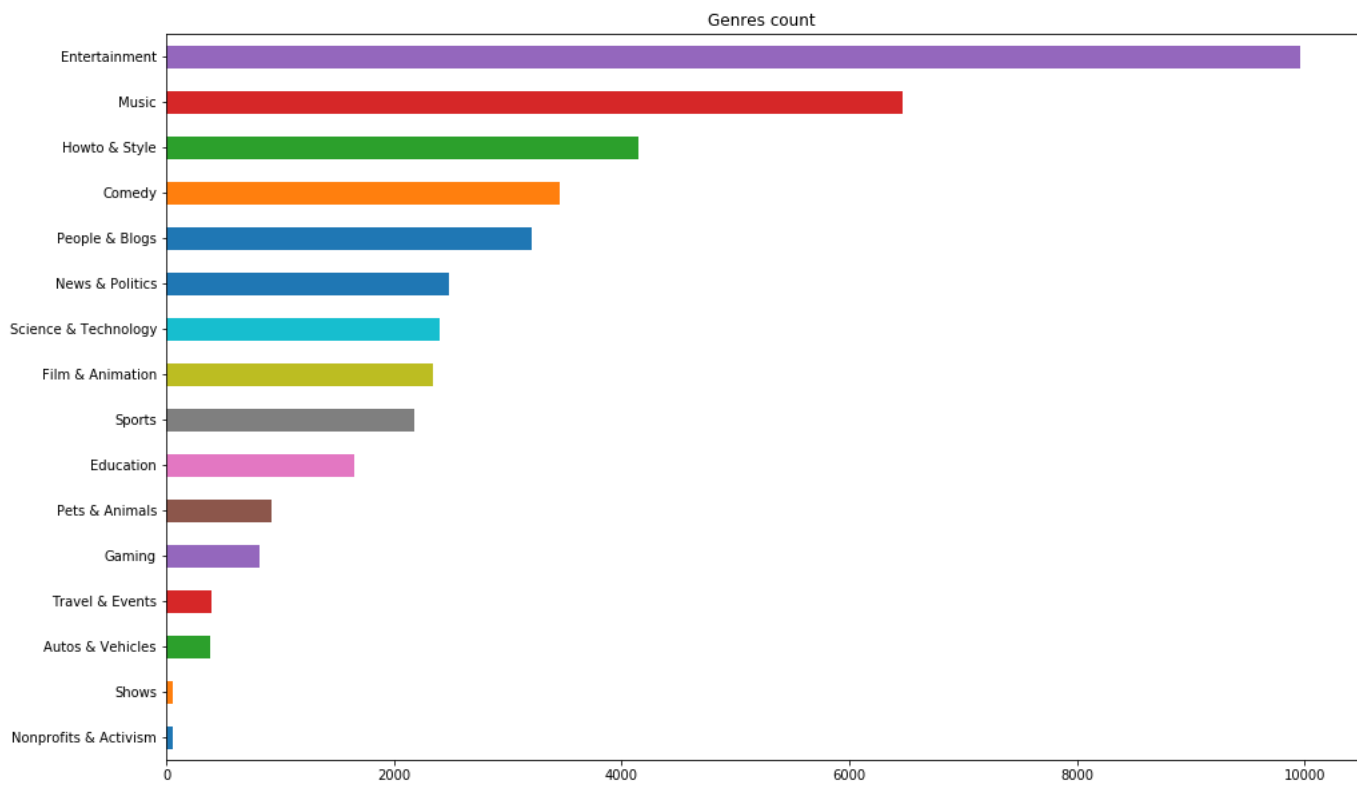


Figura 1: Cantidad de videos segun la categoria

#### 2.3.1.2 Analisis de vistas en funcion de likes

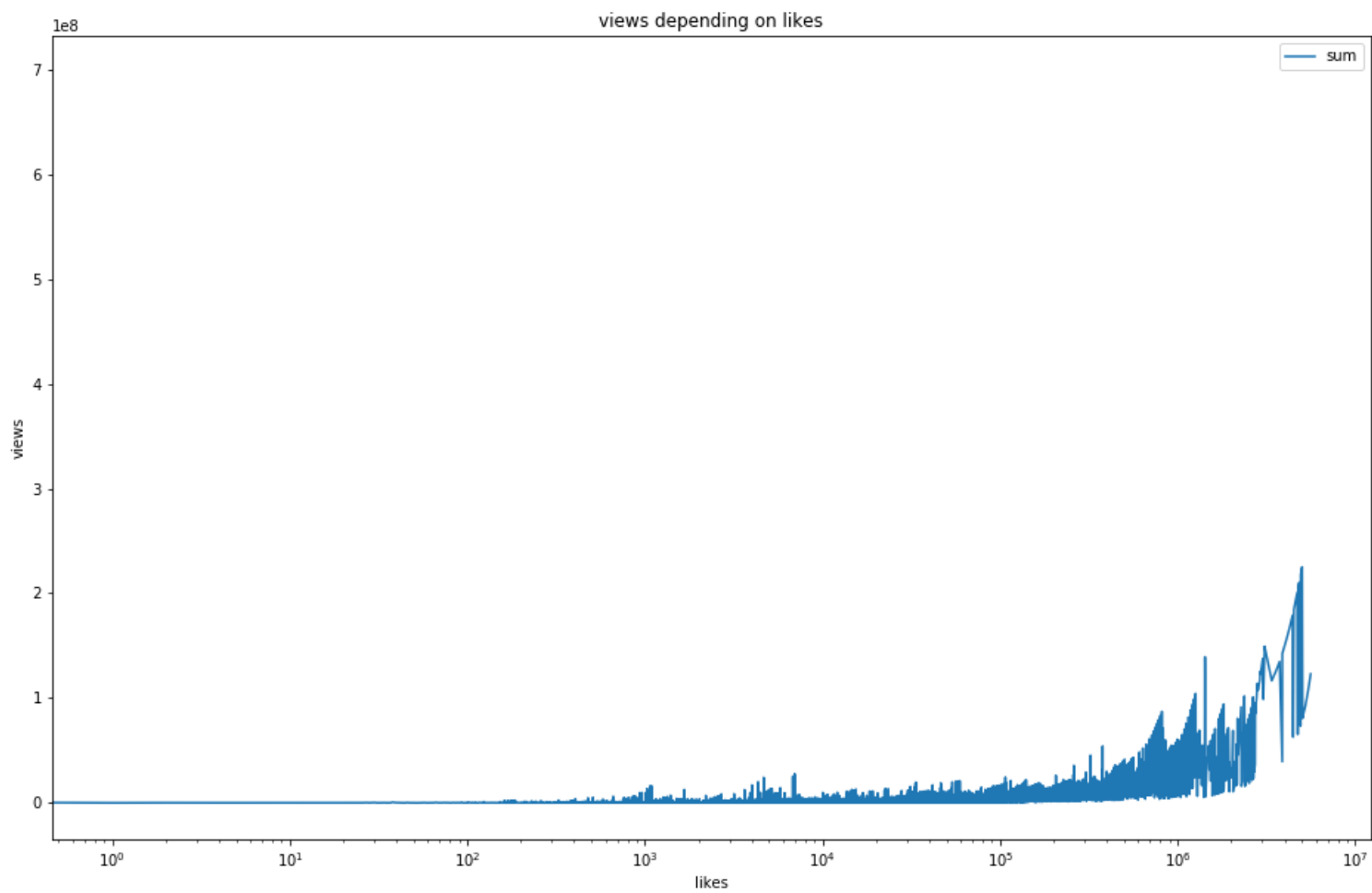


Figura 2: Cantidad de vistas segun la cantidad de likes

Podemos observar que hay un punto de quiebre en el cual a partir de cierta cantidad de likes los videos tienden a aumentar su cantidad de vistas

### 2.3.1.3 Analisis de vistas en funcion de la diferencia de dias entre la fecha de tendencia y la de publicacion

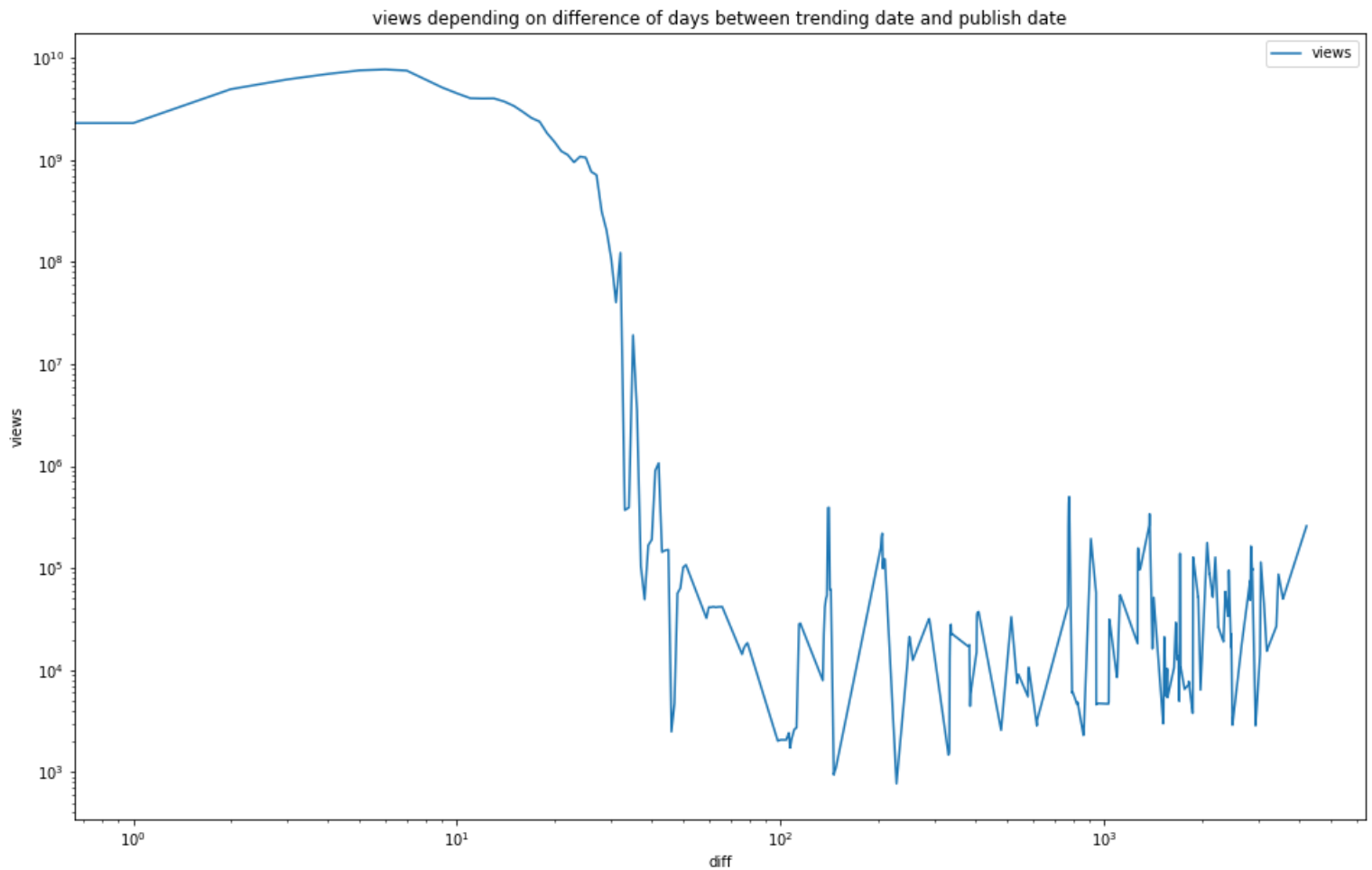


Figura 3: Cantidad de vistas a medida que pasan los dias de su publicacion

Podemos observar que el mayor incremento en la cantidad de vistas es a los pocos dias de la fecha de publicacion del video. Luego vemos que las vistan tienden a decaer

#### 2.3.1.4 Analisis de vistas en funcion de la diferencia de likes y dislikes

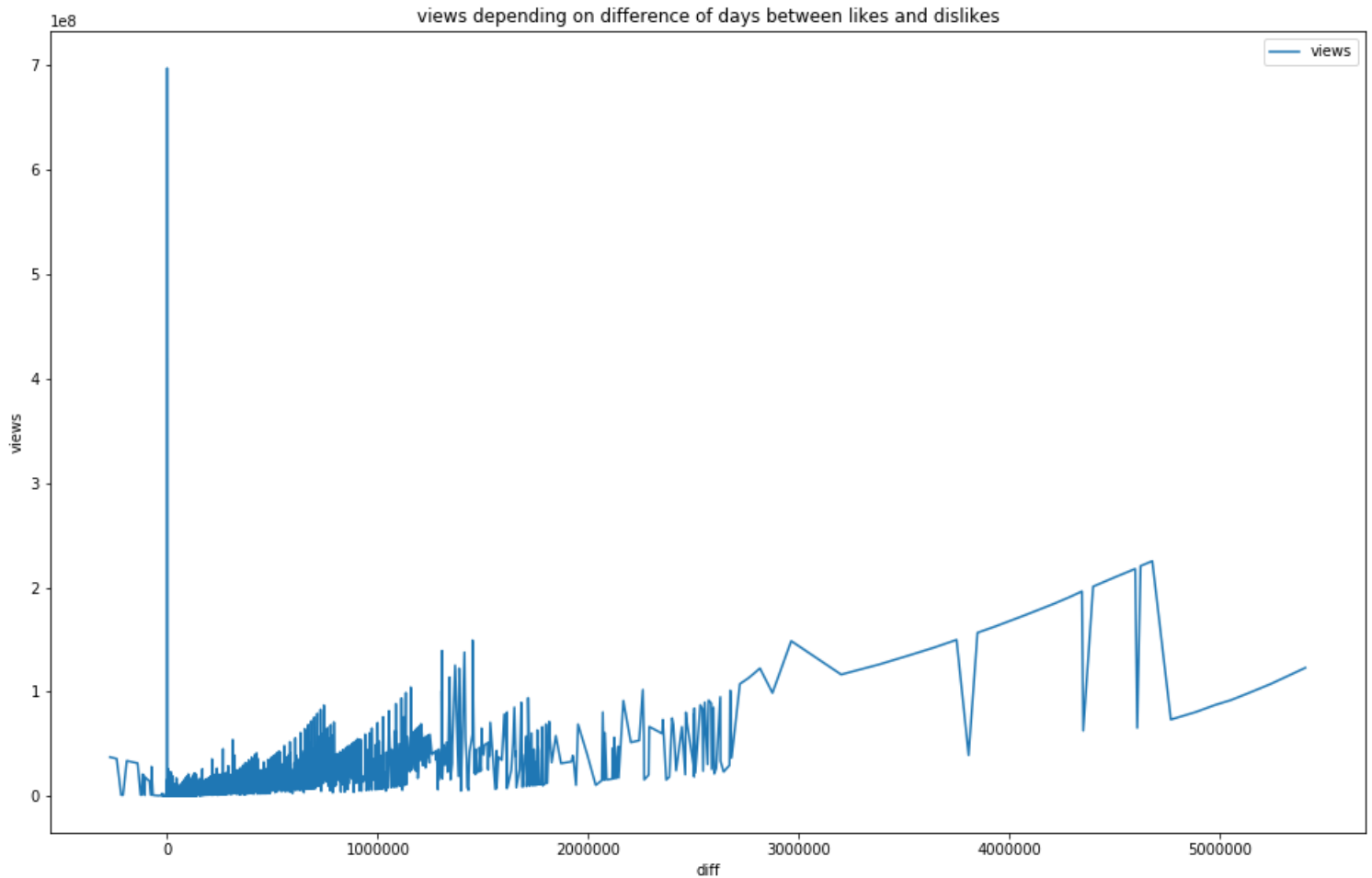


Figura 4: Cantidad de vistas segun la diferencia entre likes y dislikes

Podemos observar a medida que aumenta la distancia entre likes y dislikes, aumentan la cantidad de vistas. Esto es logico ya que en general un video tiene mas vistas cuanto mas gente descata que el mismo le gusto.

### 2.3.1.5 Analisis del progreso de views del video con mayor views del set de datos.

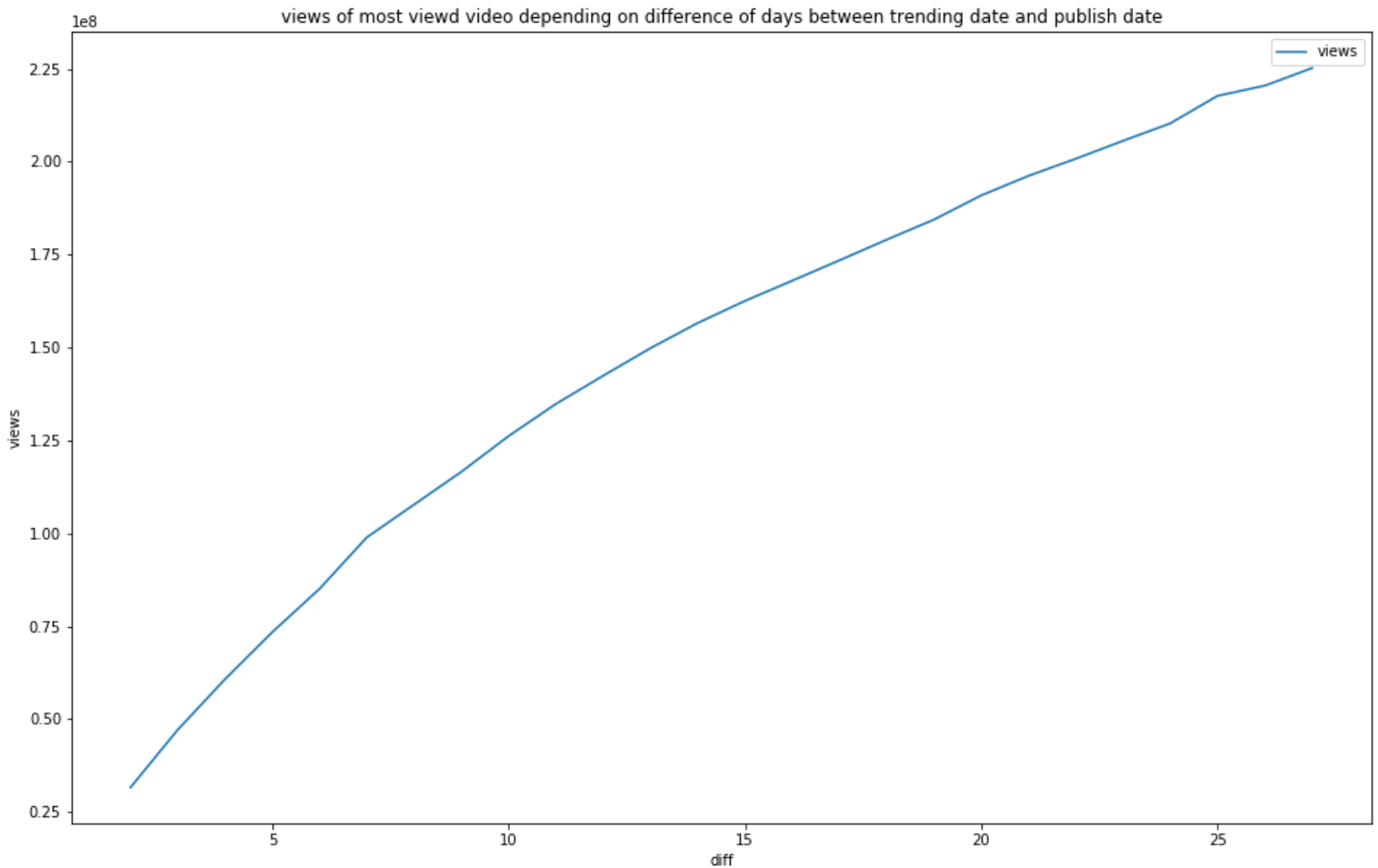


Figura 5: Progreso en dias de las vistas del video mas vistas

Podemos observar que a medida que pasan los dias respecto de la fecha de publicacion del video con mas vistas del set de datos, las vistas del mismo suben linealmente, lo cual tiene sentido ya que cuando un video es exitoso, sus vistan tienen a aumentar progresivamente.



### 2.3.1.6 Analisis del progreso de views de los videos con mayor, mediano y menor vistas del set de datos.

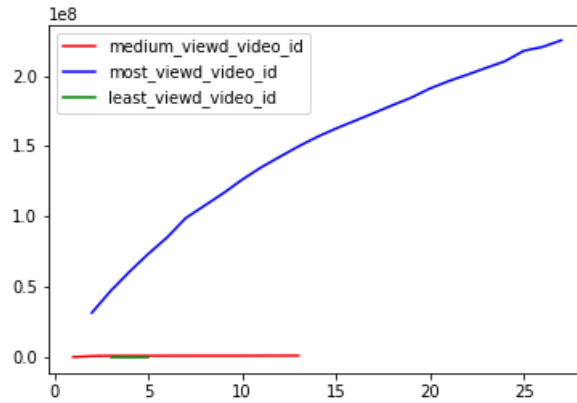


Figura 6: Progreso en dias de las vistas del video mas, mediano y menos vistas

Podemos observar que el video menos visto y el video cuya cantidad de vistas es la mediana del set de datos tienen un progreso de variación de vistas en el tiempo parecido. Y el video mas visto muestra una amplia diferencia. Esto tiene sentido ya que para tener un progreso como el que tiene el video mas visto se tiene que tener ciertas características parecidas. Esto nos dice que hay un grupo pequeño cercano al video mas visto que logra este progreso, y el resto se parece al menos visto

### 2.3.1.7 Analisis de la cantidad de views segun el largo del titulo del video.

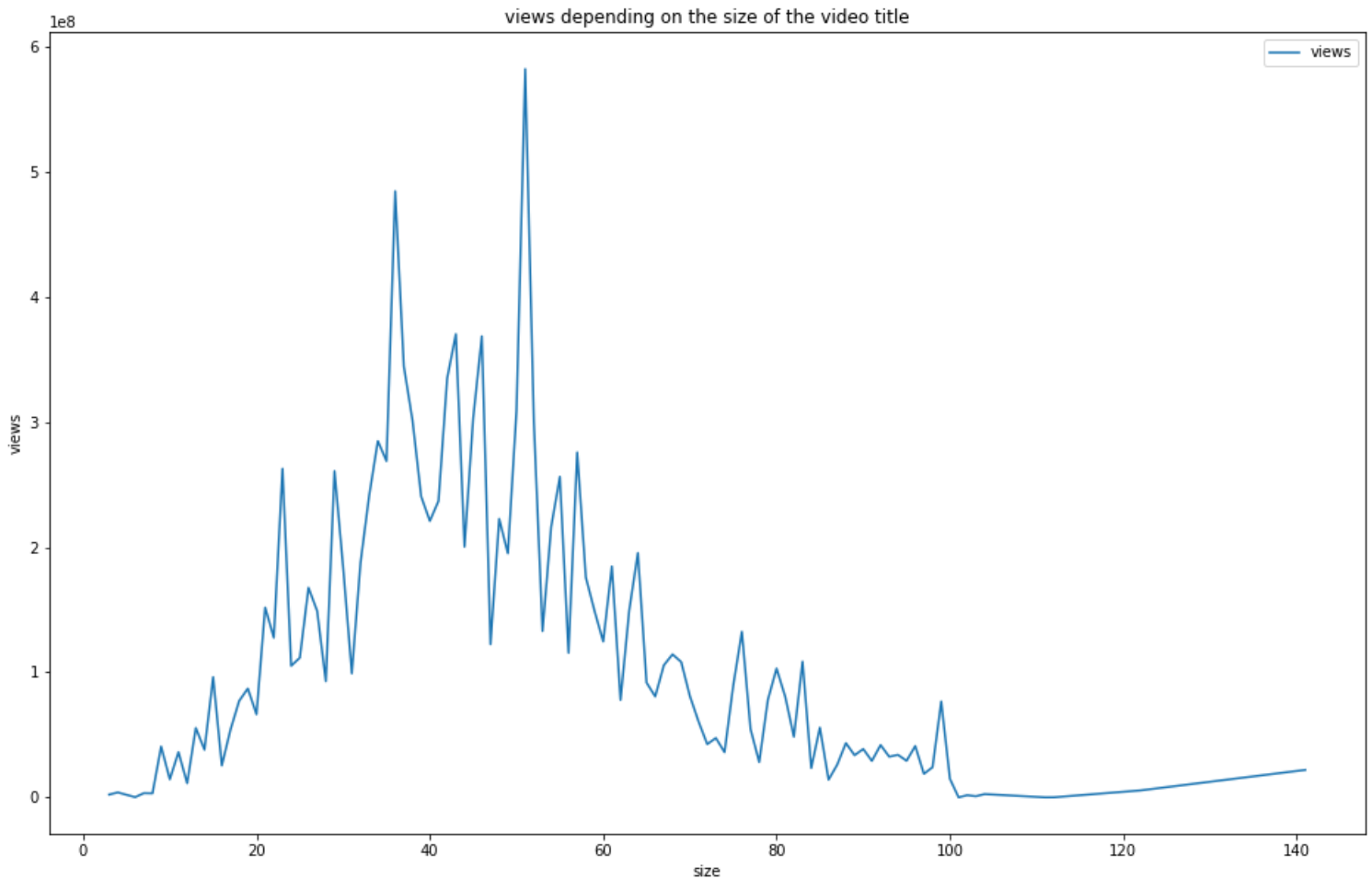


Figura 7: Cantidad de vistas segun el largo del video

Podemos observar que el largo del titulo tiene un rango de valores para su largo en el cual la cantidad de vistas del video logra un valor maximo. Esto tiene sentido ya que el ser humano tiende a leer cosas cortas que generen impacto y es por eso que este rango esta cercano al cero. Tambien si el titulo es muy corto es logico que las vistas no sean altas ya que quizas signifique que ese titulo corto no describe con impacto el video y por lo tanto no genera una atraccion para que la gente lo mire.

### 2.3.1.8 Analisis la cantidad de videos borrados por categoria.

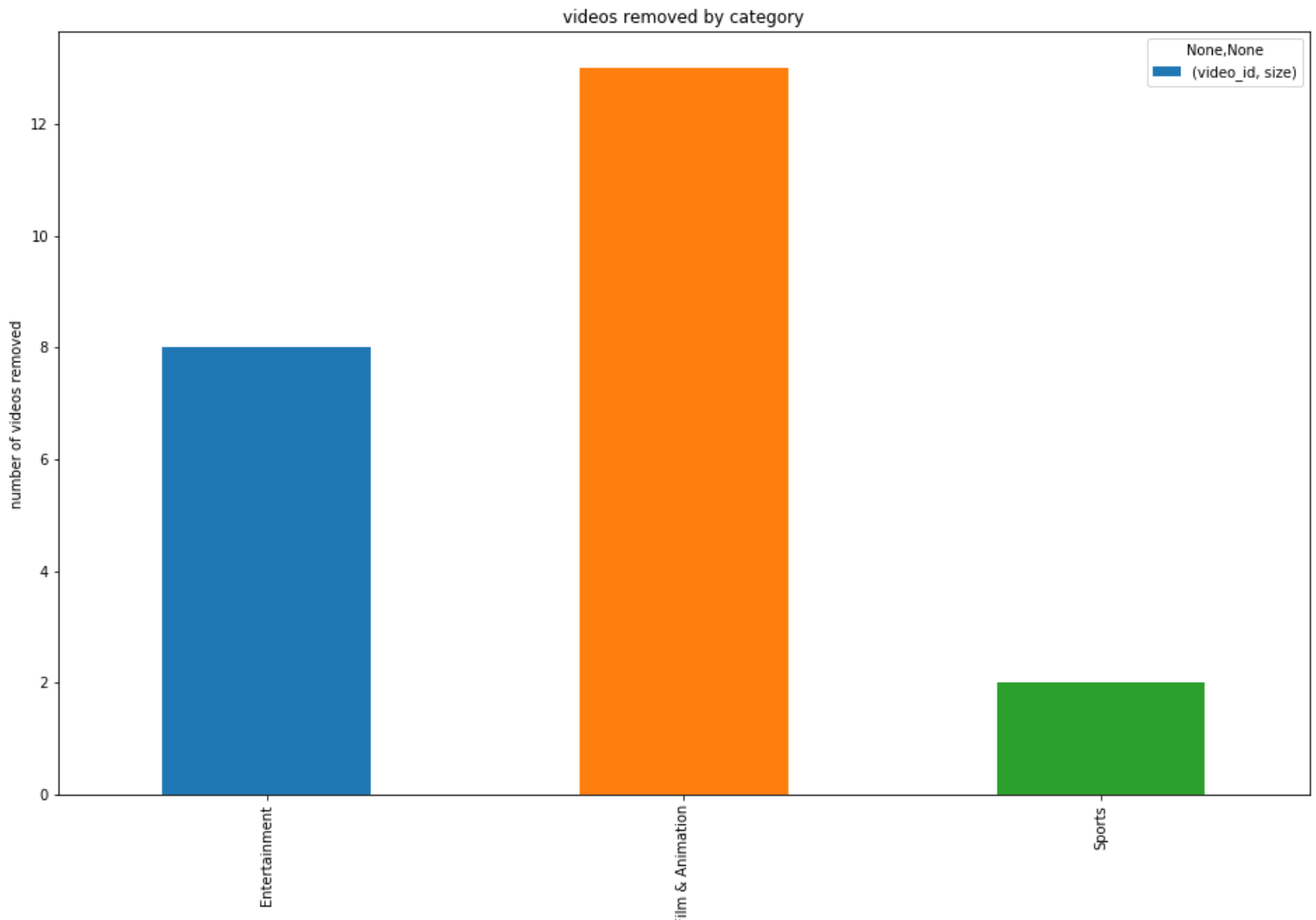


Figura 8: Cantidad de videos borrados por cateogria

Podemos observar que solo en las categorias entretenimeinto, deportes y animacion hay videos que fueron borrados. Esto es probable que se deba a problemas de copyright ya que youtube tiene ciertas resctricciones cuando se muestran imagenes que pertenecen a otras marcas, canciones o peliculas. Youtube ofrece un tiempo maximo para poner una cancion ajena hasta que te bloquean o desmonetizan el video por copyright. Lo mismo ocure con peliculas, o deportes.

### 2.3.1.9 Analisis de la cantidad de vistas de videos segun tenga o no descripcion.

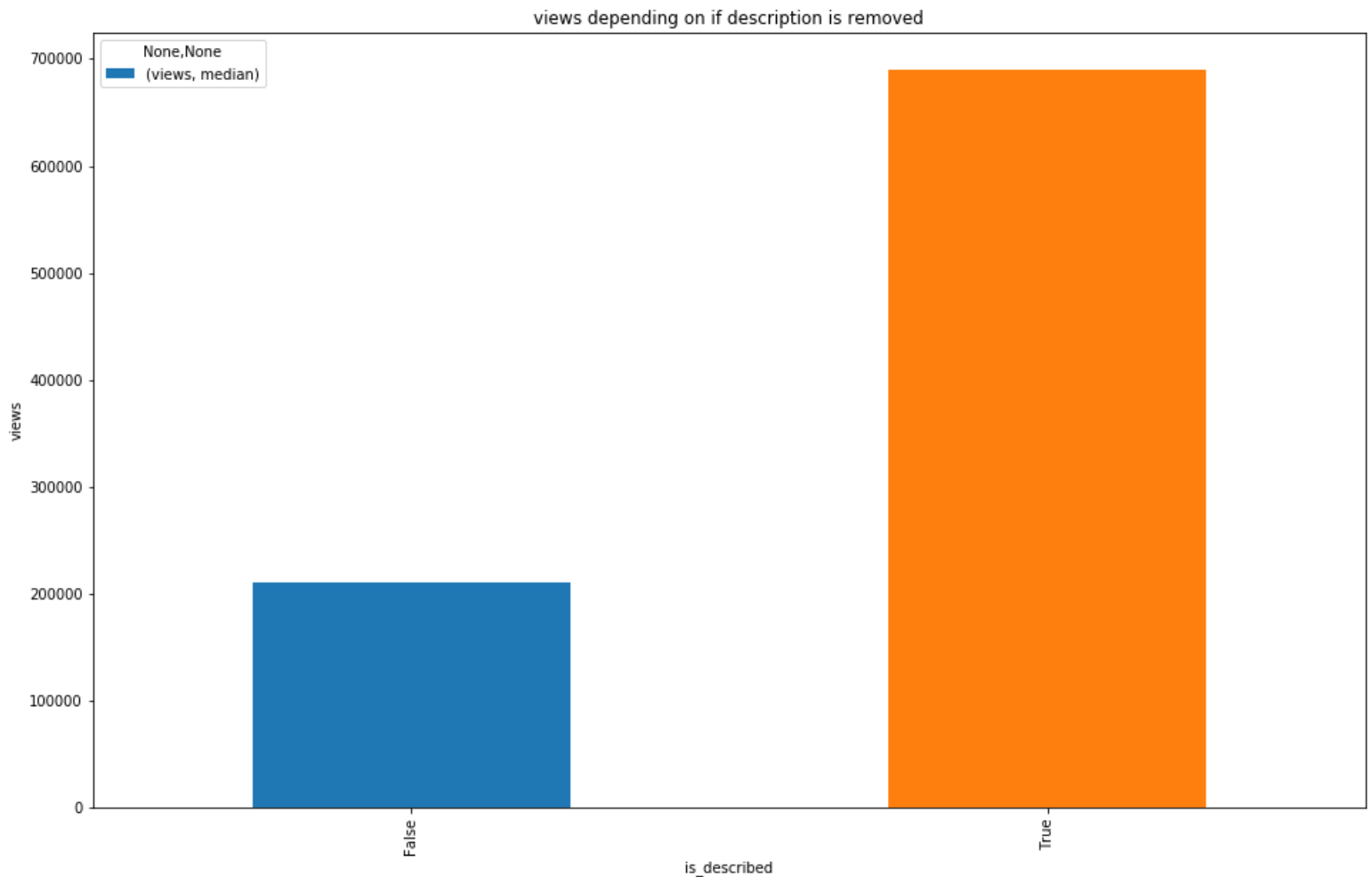


Figura 9: Cantidad de vistas segun su descripcion este o no habilitada

Podemos observar que hay una mayor cantidad de videos con vistas altas en la mediana lo cuales cuentan con una descripcion. Y los videos que no tiene descripcion en general tienden a tener pocas vistas. Esto es probable que se deba a que cuando uno ve una lista de videos en youtube, se tiene una vista previa del mismo, se lee el titulo y un poco la descripcion. Esto es importante a la hora de decidir si uno le interesa ver ese video o no ya que te puede brindar una explicacion que te interese.

## **2.4. Verificación de calidad de datos**

### **2.4.1. Reporte de calidad de datos**

Los datos utilizados durante el transcurso del proyecto fueron obtenidos gratuitamente en el sitio de Kaggle y el mismo se encuentra en formato csv. Con millones de aplicaciones en la actualidad, el conjunto de datos se ha convertido en la clave para obtener las mejores los mejores videos que se publicaron en youtube. Este conjunto de datos contiene más de 40000 detalles de videos publicados en youtube.

Fecha de recolección de datos (de API): Abril 2019

### 3. Fase 3: Preparación de los datos

#### 3.1. Reporte de calidad de los datos

A continuacion mostraremos un cuadro que muestra que atributos obtuvimos y de que tipo son:

Mombre de variable	tipo de dato	Descripcion
category_id	numeric	it is the number that represent a category nameo
publish_time	date	momento exacto de la publicacion del video
comments_disabled	booleano	Dice si tiene o no los comentarios habilitadoso
ratings_disabled	booleano	Dice si tiene o no los likes/dislikes habilitadoso
video_error_or_removed	booleano	Dice si el video sufrio un error o fue borrado en el momento de la fecha de trending_dateo
days_since_publication	numeric	Cantidad de dias desde que se publico el video
title_size	numeric	cantidad de caracteres que tiene el titulo del video
tags_quantity	numeric	cantidad de etiquetas que el video tiene
likes_ratio	numeric	porcentaje de likes respecto de dilikes
has_description	booleano	Dice si el video tiene descripcion o no
comments_per_view	numeric	Cantidad de cantidad de comentarios dividido cantidad de views

## 3.2. Seleccionar los datos

### 3.2.1. Inclusión/Exclusión de datos

Los datos que se terminaron utilizando fueron todos los de tipo alfanumerico, numérico y discreto.

## 3.3. Limpiar los datos

### 3.3.1. Reporte de limpieza de datos

Se limpiaron los siguientes campos:

- video\_id
- trending\_date
- title
- channel\_title
- tags
- likes
- dislikes
- thumbnail\_link
- description
- comment\_count

Algunos de estos campos se usaron para crear nuevos campos.

## 3.4. Estructurar los datos

### 3.4.1. Derivación de atributos

- **days\_since\_publication:** Cantidad de dias desde que se publico el video
- **title\_size:** Longitud en cantidad de caracteres del atributo title
- **publish\_time:** Se modifico el timestamp por un formato tal: YYYY-MM-DD
- **tags\_quantity:** cantidad de tags
- **likes\_ratio:** Ratio de likes y dislikes calculado como:  $\text{likes}/(\text{likes}+\text{dislikes})$

- **has\_description:** Es verdadero si el video tiene descripcion.
- **comments\_per\_view:** Cantidad de comentarios dividido la cantidad de vistas del video

#### 3.4.2. Generación de registros

Se genero el registro `is_sucessfull` que fue calculado en funcion de si el video tiene mas de 2 millones de viws y un ratio mayor o igual a 0.8

### 3.5. Integrar los datos

#### 3.5.1. Unificación de los datos

Se unieron las datos del dataset con el json de categorias para conocer la relacion entre el valor nuemrico y el nombre de la categoria.

### 3.6. Formato de los datos

#### 3.6.1. Reporte de formato de los Datos

`publish_time`: Se modifiko el timestamp por un formato tal: YYYY-MM-DD



## 4. Fase 4: Modelado

### 4.1. Seleccionar una técnica de modelado

Para este trabajo decidimos utilizar dos técnicas de modelado, J48 Decision Tree y Multilayer Perceptron. El motivo por el que elegimos estas técnicas es porque son algoritmos supervisados, lo cual es adecuado para el problema que intentamos resolver, y porque estos algoritmos, al funcionar sobre principios diferentes, reaccionan de distinta forma ante distintos tipos de datos de entrada, permitiéndonos de esta forma compararlos, y elegir el más apropiado. Se Aplicará cada una de estas técnicas, variando sus parámetros para obtener mejores resultados. La clase para estos algoritmos será el atributo `is_successfull`, el cual surgirá de aplicar una función de threshold a la cantidad de views de cada video. Para evitar que el algoritmo tenga información relacionada directamente con la clase, quitaremos el atributo `views` del set de datos para aplicar los algoritmos.

#### 4.1.1. Técnica de modelado

**J48 Decision Tree:** Se utilizará el algoritmo supervisado J48, el cual consiste en un árbol de decisión utilizado para la clasificación del set de datos.

**Multilayer Perceptron:** Se utilizará el algoritmo supervisado Multilayer Perceptron, el cual consiste en una Red Neuronal del tipo backtracking, utilizado para la clasificación del set de datos.

#### 4.1.2. Supuestos de modelado

Todos los datos usados para el modelado son numéricos.

### 4.2. Generar el diseño de las pruebas

#### 4.2.1. Diseño de las pruebas

Para ambas técnicas de modelado se eligió utilizar un 70 como set de entrenamiento y un 30 elegido al azar (se mezcla al azar todo el set de pruebas y se elige el 70 para darle suficientes datos a la red neuronal. En total quedan 12285 registros para prueba y 28664 registros para entrenamiento.

### 4.3. Construir el modelo

#### 4.3.1. Configuración de parámetros

##### 4.3.1.1 Arbol J48

Para el caso del árbol J48 debemos ajustar los parámetros de nivel de confianza y cantidad mínima de elementos por hoja.

### **Nivel de Confianza**

Un nivel de confianza demasiado alto provocará que el algoritmo no desestime ninguna regla lo cual llevará a un posible overfit.

### **Cantidad mínima de Elementos por hoja**

Lo mismo puede ocurrir en caso de usar un valor demasiado bajo de cantidad mínima de elementos por hoja.

Luego de Experimentar con distintos valores para cada caso, llegamos a los siguientes valores que nos dan los mejores resultados:

**Confidence Level:** 0.01

**Minium elements per Leaf:** 80

Luego de varios test, notamos que el valor de Minium elements per leaf que mejor resultado en términos de precisión era 60, pero el árbol quedaba demasiado grande como para comprenderlo fácilmente. Decidimos subirlo a 80 para obtener un árbol razonable, perdiendo una cantidad despreciable de precisión. Subirlo a mas de 80 ya podaba mucho el árbol y generaba pérdidas de precisión importantes. De la misma forma elegimos un Confidence level de 0.01. Disminuirlo mas generaba una pérdida de precisión, aumentarlo nos resultaba en un arbol mucho mas grande.

#### **4.3.1.2 Multilayer Perceptron**

Para el caso del Multilayer Perceptron debemos ajustar la cantidad hidden layers, el learning rate, el momentum, el training time y el decay.

### **Hidden Layers**

Si se colocan demasiadas hidden layers se incrementa mucho el tiempo de procesamiento, y ademas se corre el riesgo de overfitting. Si se colocan muy pocas puede que no sean suficientes para aproximar correctamente la función del problema.

### **Learning Rate**

El learning rate y el momentum afectan la velocidad a la que intenta converger el algoritmo. Un learning rate o momentum demasiado elevador provocaran que el algoritmo no logre encontrar un mínimo local. Un learning rate o momentum demasiado pequeños provocan que el algoritmo no logre encontrar un mínimo global.

### **Training Time**

Es la cantidad de pasadas de los datos a través de la red. Aumentarlo debería mejorar los resultados, sin embargo incrementa el tiempo de procesamiento y puede provocar overfitting.

### **Decay**

Si está activado El algoritmo Cambia dinámicamente el learning rate según los resultados anteriores. Sirve para evitar que el algoritmo se pase por alto los mínimos, pero aumenta el tiempo de procesamiento.

Luego de Experimentar con distintos valores para cada caso, llegamos a los siguientes valores que nos dan los mejores resultados:

**Hidden Layers:**

**Momentum:**

**Learning Rate:**

**Training Time:**

**Decay:**

### **4.3.2. Modelos**

#### **4.3.2.1 Resultados de Árbol de decisión J48:**

##### **4.3.2.1.1. Matriz de confusion**

is_successfull clasificado false	is_successfull clasificado true	
10106	174	Era originalmente false
1673	332	Era originalmente true

##### **4.3.2.1.2. Resultados de la clasificación**

Elementos clasificados correctamente	10438	84.9654 %
Instancias clasificadas en forma incorrecta	1847	15.0346 %



#### 4.3.2.1.4. Reglas

- Regla 1:

```
Si category_id <= 10
AND days_since_publication <= 7
Entonces
is_successfull = false
```

- Confianza: 0.787
- Soporte: 0.144
- Captura: 0.134

- Regla 2:

```
Si category_id >= 10
Entonces
is_successfull = false
```

- Confianza: 0.887
- Soporte: 0.775
- Captura: 0.816

- Regla 3:

```
Si category_id <= 10
AND days_since_publication > 7
AND days_since_publication > 32
Entonces
is_successfull = false
```

- Confianza: 1
- Soporte: 0.003
- Captura: 0.003

- Regla 4:

```
Si category_id <= 10
AND days_since_publication > 7
AND days_since_publication <= 32
AND days_since_publication <= 14
AND likes_ratio > 0.985
Entonces
Is_successfull = False
```

- Confianza: 0.713
- Soporte: 0.008
- Captura: 0.01

- Regla 5:

```
Si category_id <= 10
AND days_since_publication > 7
AND days_since_publication <= 32
AND days_since_publication > 14
AND tags_quantity > 32
Entonces
is_successfull = False
```

- Confianza: 0.67
- Soporte: 0.0021
- Captura: 0.0017

- Regla 6:

```
Si category_id <= 10
AND days_since_publication > 7
AND days_since_publication <= 32
AND days_since_publication > 14
AND tags_quantity <= 32
AND title_size <= 28
Entonces
is_successfull = False
```

- Confianza: 0.556
- Soporte: 0.003
- Captura: 0.002

- Regla 7:

```
Si category_id <= 10
AND days_since_publication > 7
AND days_since_publication <= 32
AND days_since_publication > 14
AND tags_quantity <= 32
AND title_size > 28
Entonces
is_Successfull = True
```

- Confianza: 0.726
- Soporte: 0.019
- Captura: 0.088

- Regla 8:

```
Si category_id <= 10
AND days_since_publication > 7
AND days_since_publication <= 32
AND days_since_publication <= 14
AND likes_ratio <= 0.985
AND tags_quantity > 39
Entonces
is_Successfull = False
```

- Confianza: 0.902
- Soporte: 0.0012
- Captura: 0.0013

- Regla 9:

```
Si category_id <= 10
AND days_since_publication > 7
AND days_since_publication <= 32
AND days_since_publication <= 14
AND likes_ratio <= 0.985
AND tags_quantity < 39
AND category_id <= 2
AND title_size <= 28
Entonces
is_Successfull = False
```

- Confianza: 0.883
- Soporte: 0.0023
- Captura: 0.0024

- Regla 10:

```
Si category_id <= 10
AND days_since_publication > 7
AND days_since_publication <= 32
AND days_since_publication <= 14
AND likes_ratio <= 0.985
AND tags_quantity < 39 & category_id <= 2
AND title_size > 28
```

```
AND title_size <= 40
Entonces
is_Successfull = True
```

- Confianza: 0.646
- Soporte: 0.0023
- Captura: 0.01

- Regla 11:

```
Si category_id <= 10
AND days_since_publication > 7
AND days_since_publication <= 32
AND days_since_publication <= 14
AND likes_ratio <= 0.985
AND tags_quantity < 39
AND category_id <= 2
AND title_size > 28
AND title_size >40
Entonces
is_Successfull = False
```

- Confianza: 0.652
- Soporte: 0.0081
- Captura: 0.0063

- Regla 12:

```
Si category_id <= 10
AND days_since_publication > 7
AND days_since_publication <= 32
AND days_since_publication <= 14
AND likes_ratio <= 0.985
AND tags_quantity <= 39
AND category_id >= 2
AND tags_quantity > 16
Entonces
Is_Successfull = True
```

- Confianza: 0.67
- Soporte: 0.01
- Captura: 0.045



- Regla 13:

```
Si category_id <= 10
AND days_since_publication > 7
AND days_since_publication <= 32
AND days_since_publication <= 14
AND likes_ratio <= 0.985
AND tags_quantity <= 39
AND category_id >= 2
AND tags_quantity <= 16
AND tags_quantity > 12
Entonces
is_Successfull = False
```

- Confianza: 0.79
- Soporte: 0.0022
- Captura: 0.0020

- Regla 14:

```
Si category_id <= 10
AND days_since_publication > 7
AND days_since_publication <= 32
AND days_since_publication <= 14
AND likes_ratio <= 0.985
AND tags_quantity <= 39
AND category_id >= 2
AND tags_quantity <= 16
AND tags_quantity <= 12
AND title_size > 47
Entonces
is_Successfull = False
```

- Confianza: 0.62
- Soporte: 0.0057
- Captura: 0.0042

- Regla 15:

```
Si category_id <= 10
AND days_since_publication > 7
AND days_since_publication <= 32
AND days_since_publication <= 14
AND likes_ratio <= 0.985
AND tags_quantity <= 39
AND category_id >= 2
```

```
AND tags_quantity <= 16
AND tags_quantity <= 12
AND title_size <= 47
AND likes_ratio > 0.97556
Entonces
is_Successfull = True
```

- Confianza: 0.776
- Soporte: 0.0026
- Captura: 0.0127

- Regla 16:

```
Si category_id <= 10
AND days_since_publication > 7
AND days_since_publication <= 32
AND days_since_publication <= 14
AND likes_ratio <= 0.985
AND tags_quantity <= 39
AND category_id >= 2
AND tags_quantity <= 16
AND tags_quantity <= 12
AND title_size <= 47
AND likes_ratio <= 0.97556
AND likes_ratio <= 0.965
Entonces
is_Successfull = True
```

- Confianza: 0.622
- Soporte: 0.0044
- Captura: 0.017

- Regla 17:

```
Si category_id <= 10
AND days_since_publication > 7
AND days_since_publication <= 32
AND days_since_publication <= 14
AND likes_ratio <= 0.985
AND tags_quantity <= 39
AND category_id >= 2
AND tags_quantity <= 16
AND tags_quantity <= 12
AND title_size <= 47
AND likes_ratio <= 0.97556
```

```
AND likes_ratio > 0.965  
Entonces  
is_Successfull = False
```

- Confianza: 0.66
- Soporte: 0.0027
- Captura: 0.0021

**4.3.2.1.5. Arbol con reglas** Aqui tenemos la salida de la aplicacion del modelo con las reglas de desicion:

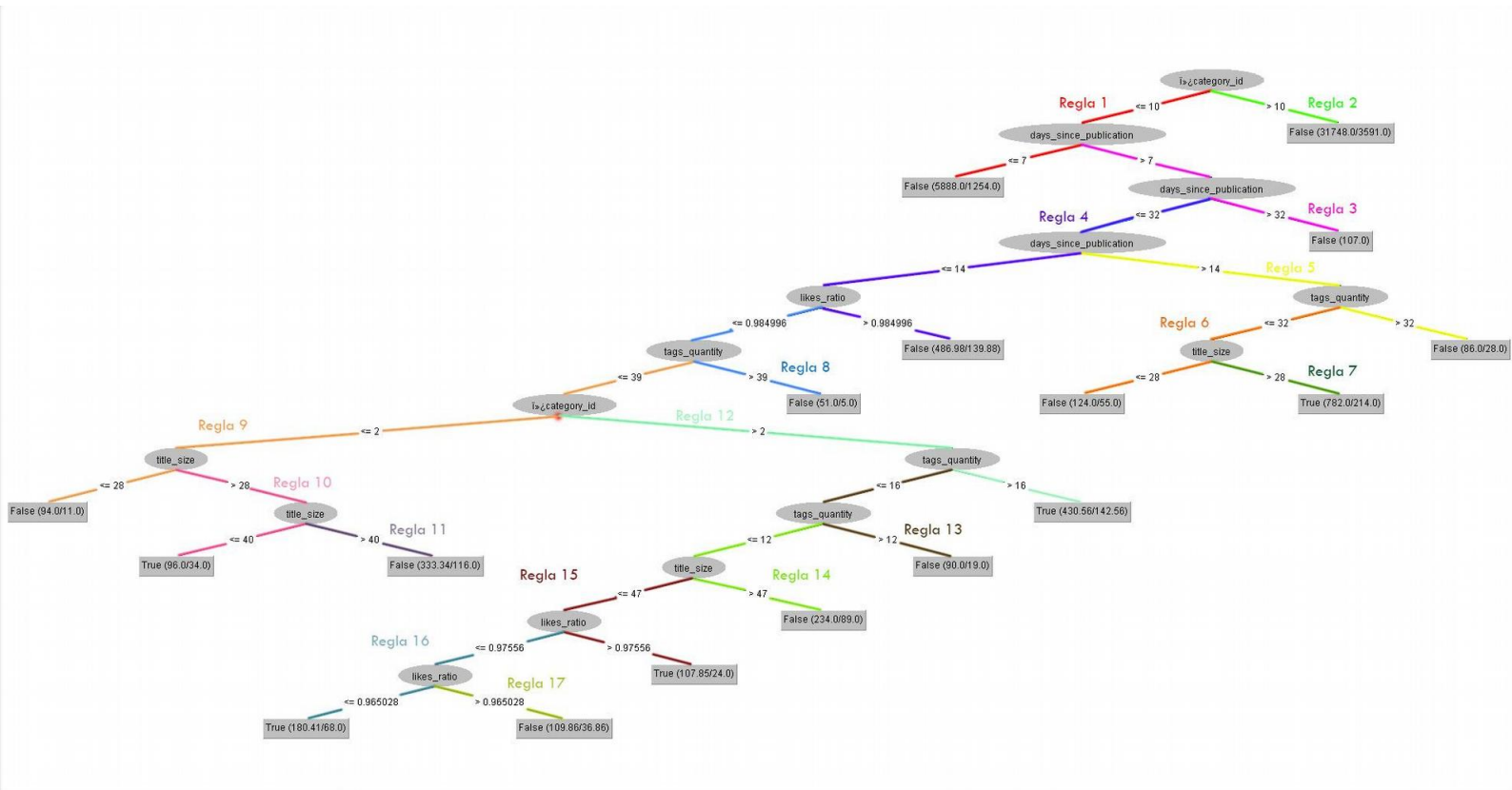


Figura 11: Arbol de desicion con reglas

4.3.3. Descripción del modelo

4.4. Evaluar el modelo

4.4.1. Evaluación del modelo

4.4.2. Revisión de la configuración de parámetros

## **5. Fase 5: Evaluación**

### **5.1. Evaluar Resultado**

5.1.1. Valoración de los resultados de minería de datos

5.1.2. Modelo aprobado

### **5.2. Proceso de revisión**

5.2.1. Revisión del proceso

### **5.3. Determinar Próximos pasos**

5.3.1. Listado de posibles acciones

## 6. Conclusiones

En base al set de datos, los resultados obtenidos, y las conclusiones a las que llegamos a partir de ellas, podemos decir que se cumple el objetivo de la metodología, para luego, a partir de esto tomar acciones para predecir si un video será exitoso dadas ciertas características.