Escuela Superior
de Ingeniería y Tecnología
Universidad de La Laguna

Trabajo de Fin de Grado

Grado en Ingeniería informática

Análisis de la reacción del consumidor en Youtube

Analysis of consumer reaction on Youtube

Diego Álvarez Padrón

La Laguna, 6 de Junio de 2019

Dña. **Rosa María Aguilar Chinea**, con N.I.F. 43.778.956-C Catedrática de Universidad adscrita al Departamento de Ingeniería Informática y de Sistemas de la Universidad de LaLaguna, como tutora

D. **Jesús Miguel Torres Jorge**, con N.I.F. 43.826.207-Y profesor Contratado Doctor de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como cotutor

CERTIFICA(N)

Que la presente memoria titulada:

"Análisis de la reacción del consumidor en Youtube"

ha sido realizada bajo su dirección por D. Diego Álvarez Padrón, con N.I.F. 79.060.357-L.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 6 de junio de 2019.

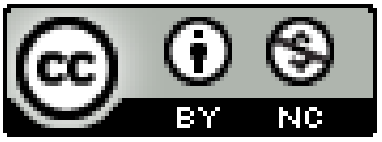
Agradecimientos

Quiero agradecer principalmente a mi familia, amigos y demás personas que han estado ahí en todo momento.

Agradecer enormemente a mis tutores por el gran apoyo brindado siempre que se lo he requerido tanto en tutorías, correos como en cualquier circunstancia a cualquier hora.

Por último agradecer a las personas que se han cruzado en mi vida durante los estudios en estos últimos años.

Licencia



Resumen

La presente memoria describe el proyecto de Fin de Grado titulado “Análisis de la reacción del consumidor en Youtube”, este permite el análisis de sentimientos en comentarios de Youtube de un vídeo concreto. El análisis se basa en el turismo en Tenerife, en el video de Youtube que tuviera más comentarios en inglés dentro del ámbito turístico y en Tenerife conjuntamente.

Para la realización de este proyecto se obtienen los comentarios utilizando la API de Youtube habiendo creado previamente un proyecto en esta. Además de los comentarios obtenemos todos los datos del video de Youtube y del canal; los “likes”, número de visualizaciones, título del vídeo y del canal, fechas de publicación así como los propios comentarios.

Con estos datos se realiza el análisis de sentimientos de los comentarios del video clasificándolos en positivos, negativos o neutros asignando un valor: 1, -1, 0 respectivamente. A partir de estos datos sacamos las conclusiones en relación a los sentimientos de las personas sobre el contenido.

Palabras clave

Youtube, Turismo, Comentarios, Consumidores, Análisis de sentimientos, Tenerife.

Abstract

Here is the end of the degree project titled “Analysis of consumer reaction on Youtube”, where a project has been developed that allows the analysis of feelings in Youtube comments of a specific video. The analysis focused on tourism in Tenerife, specifically the YouTube video that had more comments in English focusing on tourism and Tenerife together.

For the realization of this project the comments are obtained using the Youtube API having previously created a project in it. In addition to comments we obtain all the data from the YouTube video and the channel: the "likes", number of visualizations, video and channel title, publication dates as well as the comments themselves.

With all this data the feelings analysis of the comments of the video is made, classifying them in positive, negative or neutral assigning a value: 1, -1, 0 respectively. From this we draw conclusions in relation to the feelings of people about this content.

Keywords

Youtube, Tourism, Comments, Consumers, Feelings analysis, Tenerife.

Índice General

Índice de figuras

Índice de tablas

1 Introducción

A día de hoy la importancia de las redes sociales en el día a día es incuestionable, a través de ellas no solo podemos comunicarnos sino que también podemos realizar diferentes alternativas sobre la creación de contenido en muchos aspectos de nuestra vida.

Centrándonos en el impacto de las redes sociales en la industria Canaria, podemos observar que dicho impacto ha crecido de forma exponencial a lo largo de los últimos años. Hoy en día se han convertido en una herramienta tremendamente útil e incluso imprescindible para el correcto desarrollo y crecimiento de cualquier empresa en el sector Canario.

Un estudio[17] realizado por Nick Earle (vicepresidente Senior de Cisco Services) a diversas empresas a nivel mundial enuncia que “no tener en cuenta la influencia de las redes sociales expone a las empresas al deterioro de su imagen”. El problema radicaliza en que el aumento de usuarios conectados está impulsando un cambio de actitud ante el mercado en la empresa, generando “negocios impulsados por personas”, donde las herramientas de redes sociales y tecnologías de cooperación son el combustible en la productividad de próxima generación", afirmó el vicepresidente Senior en “Cisco Services”.

Estas herramientas acercan la tecnología a las empresas, proporcionando nuevas experiencias de colaboración, nuevos canales de información para las personas y estableciendo herramientas potenciales de venta, que mejoran la percepción de marca y la confidencialidad de los usuarios.

En Canarias observamos grandes ejemplos que avalan este estudio. La Consejería de Economía, Industria, Comercio y Conocimiento del Gobierno de Canarias, a través de la Agencia Canaria de Investigación, Innovación y Sociedad de la Innovación (ACIISI), ha aumentado su presencia en redes sociales de una manera progresiva con el objeto de acercarse cada vez más al ciudadano del siglo XXI. Las cuentas en Facebook de Ciencia Canaria y de la ACIISI lograron juntas 12.163 nuevos seguidores en este tiempo, con un alcance medio mensual de 382.857 personas y una media de 3.965.000 impresiones de las noticias. Con esta iniciativa, Economía pretende facilitar la implicación de la ciudadanía a través de las redes sociales, donde se propaga la información de forma efectiva, rápida y amplia, complementando los canales de comunicación tradicionales en la Administración Pública de la Comunidad Autónoma de Canarias.

Las redes sociales tienen mucho impacto en todos los sectores canarios, pero nos centraremos en el que tiene mayor impacto económico en nuestra comunidad, el turismo. El turismo genera el 34% de la riqueza en Canarias por lo que es de vital importancia dar la mejor imagen y saber plasmar todo lo que Canarias tiene que ofrecer de la mejor manera posible en las redes sociales.

Youtube ha cambiado la forma en que el contenido en vídeo se almacena, transmite y se ve en el mundo. La televisión de ayer, es el YouTube de hoy. YouTube ha puesto la capacidad de publicar contenido en video para el mundo al alcance de cualquier persona con conexión a Internet. Prueba de ello son grandes campañas turísticas como la denominada “Notwinter Games” impulsada por el Gobierno de Canarias. Dicha campaña logró en menos de dos semanas más de 100 millones de impactos en Europa, concretamente del 12 al 25 de febrero, coincidiendo con los Juegos Olímpicos de Invierno.

'Notwinter Games' obtuvo cerca de 95 millones de impresiones de sus anuncios y publicaciones en Internet y contabilizó más de 6,7 millones de visualizaciones de sus vídeos en YouTube y redes sociales, el vídeo oficial de la campaña fue remitido por email a más de 700.000 turistas que conforman la base de datos de Turismo de Canarias. Esto promovió un gran revuelo y tráfico en Youtube. La campaña mezcló contenidos propios con otros desarrollados por “influencers”, entre ellos el patinador Javier Fernández, medalla de bronce en los Juegos Olímpicos de Invierno.

A pesar de estas grandes campañas, el uso de Youtube en relación con el turismo en Canarias tiene aún un gran crecimiento por delante en los próximos años. Existen pequeñas iniciativas creadas por la cámara de comercio de Santa Cruz de Tenerife llamada “Fit Canarias” (https://www.Youtube.com/user/fitcanarias) para promover la innovación turística, cuenta tan solo con 37 suScriptores. Youtube tiene una capacidad de explotación muy superior a la actual, que veremos plasmada en el futuro, no solo por parte de empresas sino de particulares que colgarán en Youtube sus propias experiencias personales muchísimo más de lo que se hace actualmente.

1.1 Antecedentes

El análisis o detección de sentimientos, es el estudio por el cual se determina la opinión de las personas en Internet sobre algún tema en específico, prediciendo la polaridad de los usuarios (a favor, en contra, neutro, etc), abarcando temas que van desde productos, películas, servicios a intereses socio-culturales como elecciones, guerras, fútbol, etc.

En nuestro caso particular nos vamos a centrar en un canal de Youtube sobre el turismo en Canarias, más concretamente en los comentarios a partir de cada video. Teniendo esto como base, vamos a utilizar un algoritmo pre-entrenado llamado VADER (Valence Aware Dictionary and Sentiment Reasoner) al que le pasaremos una sentencia concreta y nos dirá como resultado si el comentario es positivo, negativo o neutro.

El análisis de todos estos datos se realizará mediante dicho algoritmo en el lenguaje de programación Python, expresamente en inglés y el resultado, será mostrado en diversas tablas de una manera muy visual y muy clara.

1.2 Big data

“Big Data” es un término que describe el gran volumen de datos, tanto estructurados como no estructurados, que inundan los negocios cada día. Pero no es la cantidad de datos lo que es importante, lo que importa con el “Big Data” es lo que las organizaciones hacen con los datos. “Big Data” se puede analizar para obtener ideas que conduzcan a mejores decisiones y movimientos de negocios estratégicos.

Cuando hablamos de “Big Data” nos referimos a conjuntos de datos o combinaciones de conjuntos de datos cuyo tamaño (volumen), complejidad (variabilidad) y velocidad de crecimiento (velocidad) dificultan su captura, gestión, procesamiento o análisis mediante tecnologías y herramientas convencionales, tales como bases de datos relacionales y estadísticas convencionales o paquetes de visualización, dentro del tiempo necesario para que sean útiles.

Teniendo en cuenta lo anterior, vemos que la herramienta en la que se basa este proyecto (Youtube) es junto otras redes sociales como Facebook, Twitter y el resto de redes sociales una de las fuentes de “Big Data” más masivas que existen en la actualidad. Generan miles de millones de datos diarios en forma de comentarios, opiniones, visualizaciones, encuestas y demás maneras de expresar las opiniones

por parte de los usuarios. Gracias a esto las empresas pueden hacer varios estudios de mercado combinando la ingeniería social para sacar su propio beneficio a la hora de vender y distribuir un producto.

1.3 Análisis de sentimientos

El análisis de sentimientos o minería de opinión es un tipo de procesamiento del lenguaje natural, para realizar el seguimiento del estado de ánimo del público sobre un producto en particular; concretamente el procesamiento de lenguaje natural, análisis de texto y lingüística computacional para identificar y extraer información subjetiva de los recursos. En nuestro caso esos recursos son las palabras que forman cada comentario de nuestro vídeo en Youtube.

Una tarea muy importante para las empresas de hoy en dia es, a partir del Big Data, realizar un potente análisis de sentimientos para su propio beneficio.Los usuarios/clientes expresan sus opiniones constantemente en las redes sociales, ya sea en temas de actualidad, gustos o productos que consuman, a partir de estas opiniones es muy interesante para las empresas automatizar un análisis de sentimientos ya sea en Facebook, Twitter o Youtube para poder poder cuantificar y clasificar las opiniones de los usuarios, esto les permitirá el objetivo principal de cualquier empresa, maximizar beneficios, minimizar costes y por supuesto conseguir una valoración lo más positiva posible para sus productos.

A pesar de estos pensamientos bastante extendidos y estandarizados queda aún mucho trabajo que realizar en este campo de análisis a la hora de automatizar el proceso mediante inteligencia artificial(IA). La situación se complica exponencialmente cuando se utiliza la ironía o el sarcasmo junto con otros caracteres raros del alfabeto que pueden expresar opinión positiva, negativa, neutra o incluso combinarse con ironía y sarcasmo.

1.4 Planificación del trabajo

Este apartado detalla la planificación base que se ha seguido para el desarrollo del proyecto.

Nuestra tarea principal es el análisis de sentimientos de comentarios de Youtube de un video determinado.Como tema central se ha escogido el turismo en Tenerife, para ello se ha analizado el estado actual del tema, se ha hecho un estudio de la información y se elegido un vídeo concreto para obtener todos sus datos.

El desarrollo del proyecto se ha dividido en cuatro etapas principales como se puede observar a continuación.

- Tarea 1. Obtención de datos en Youtube sobre Turismo en Tenerife.
- Tarea 2. Teniendo en cuenta el video con más comentarios extraer y preparar los datos (Tokenización,conversión a minúsculas, eliminación de “Stopwords”, eliminación de puntuación, eliminación de palabras de menos de dos caracteres)
- Tarea 3.- Detección de sentimientos con el algoritmo pre-entrenado “VADER”
- Tarea 4.- Generación de informe:
  ¿Cómo evoluciona el sentimiento a lo largo del tiempo? ¿Hay algún pico?
  ¿Cuáles son los períodos en los que los usuarios comentan más?
  ¿Cuál es el día de la semana en que los usuarios hacen los comentarios más positivos?
  ¿Cuál es el día de la semana en que se comenta más activamente?

2 Objetivos del proyecto

El objetivo principal de este trabajo es el análisis de sentimientos de los comentarios de un video alojado en Youtube a través de una línea de tiempo.

En primer lugar tendremos que crearnos un proyecto y una cuenta en la API de Youtube, de esta manera podemos empezar a realizar peticiones para traer los datos que posteriormente trataremos.

Una vez tengamos dichos datos los “limpiaremos” para que se pueda proceder a realizar su análisis, para ello removemos signos de puntuación, palabras con menos de 2 caracteres, cualquier tipo de símbolo y cualquier emoticono.

Una vez realizada la limpieza usaremos varios datos pertenecientes a un vídeo de Youtube , estos son: número de “likes”, “dislikes”, visualizaciones, título y por supuesto los comentarios que es la base principal de nuestro análisis. A partir de los comentarios determinaremos un análisis de sentimientos realizando una clasificación de la siguiente manera:

- Positivo (1)
- Negativo(-1)
- Neutro(0)

A partir de estos valores representaremos los resultados en varias gráficas para poder ver de manera visual y clara el desarrollo de los sentimientos de las personas a través de una línea de tiempo.

2.1 Canal y video elegido

El canal[1] elegido para dicho proyecto se llama “Tuberides”, canal que se dedica a viajar por el mundo para descubrir los mejores parques acuáticos con el fin de publicar videos sobre sus atracciones, virtudes y demás cosas de interés para el público en general.

Se trata de un canal con un tráfico y una antigüedad importante, se incorporó el 21/06/2009 y cuenta actualmente con más de 779 millones de reproducciones, más de medio millón de suScriptores y decenas de videos.

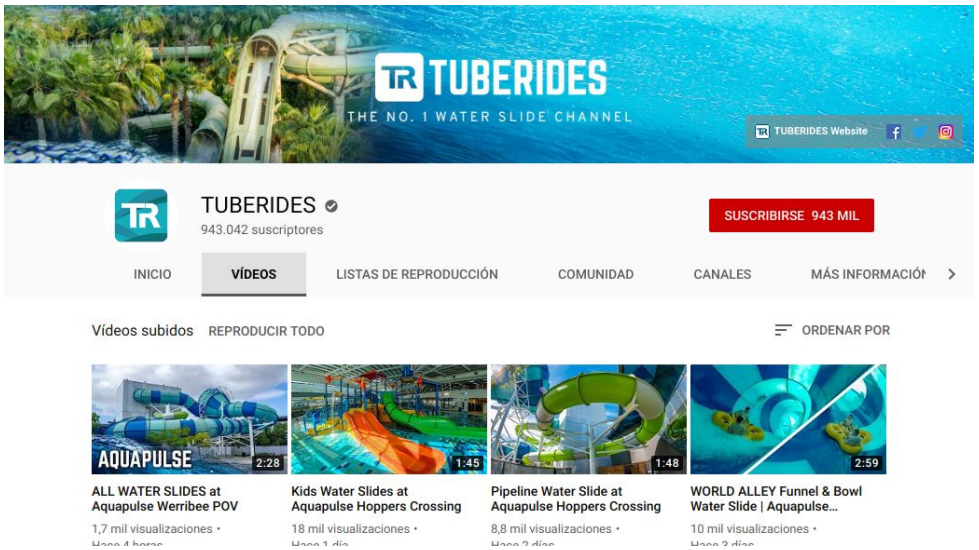


Ilustración 1: Canal de Youtube escogido

Dentro de todos los videos que posee este canal de Youtube se ha escogido un vídeo[2] qué hace una recopilación sobre las atracciones de un parque acuático de Canarias dado el tráfico aceptable para el proyecto, trata sobre el turismo en Tenerife y posee un número de comentarios en inglés apropiado.

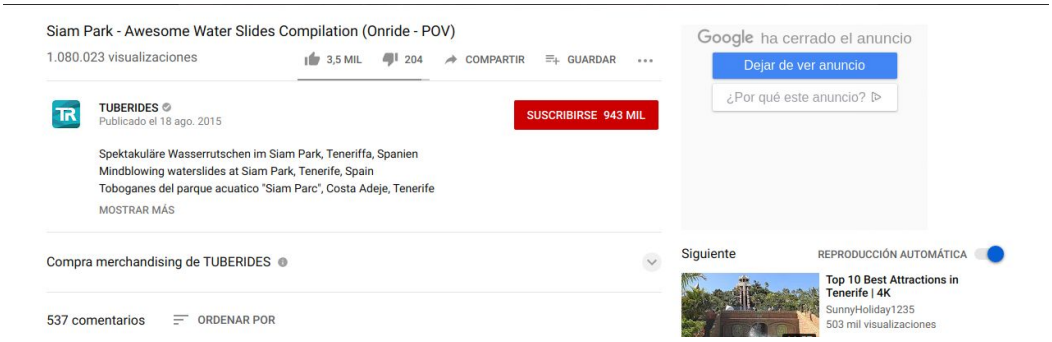


Ilustración 2: Video escogido

3 Tecnologías y aspectos técnicos

En este apartado comentaré los entornos y tecnologías utilizados en el trabajo y realización de todo el proyecto.

3.1 Entorno de implementación

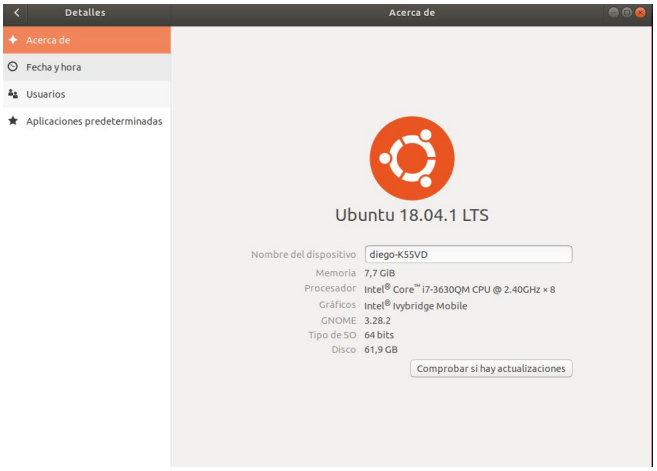


Ilustración 3: Sistema operativo.

De principio a fin el desarrollo del proyecto se ha realizado en el sistema operativo Linux, concretamente la versión 18.04 . La decisión de haber elegido este entorno es puramente subjetiva y personal ya que me encuentro más cómodo trabajando en Linux, tiene un conjunto de herramientas y facilidades para desarrollar código que no tiene otro sistema operativo como Windows en ninguna de sus versiones.

Ilustración 4: Editor de texto Atom

Este sistema operativo tiene innumerables editores de texto para desarrollar código, en mi caso he decidido elegir Atom para el completo desarrollo del Script.

Atom[3] es un editor de código de fuente de código abierto para MacOS, Linux, y Windows con soporte para múltiples plug-in escritos en “Node.js” y control de versiones “Git” integrado, desarrollado por “GitHub”. Atom es una aplicación de escritorio construida utilizando tecnologías web

Tiene varias características destacables como “IDE”:

- **Funcionalidades extra (Packages).** Con el "package manager", instalado por defecto, podemos instalar y desinstalar fácilmente casi cualquier función imaginable. A día de hoy más de 6500 paquetes de modificaciones se encuentran disponibles.

- **Integración con Git.** Nuestro proyecto de Atom se sincronizará automàticamente con el repositorio de “Git” y veremos en todo momento si se encuentra en la misma versión que nuestro repositorio o en qué documentos hay divergencias.

- Funciona con una gran cantidad de lenguajes de programación.

- Atom dispone de un documento totalmente editable donde podemos ajustar el estilo de trabajo a nuestras más detalladas preferencias.

3.2 Lenguaje de programación y bibliotecas

Para el desarrollo del proyecto se ha utilizado de principio a fin el lenguaje de programación Python[4]. Esto es debido principalmente a las ventajas y posibilidades que da este lenguaje a la hora de realizar un estudio y análisis gracias a la cantidad de paquetes o bibliotecas que posee para hacer el trabajo más eficiente.

En los últimos años el lenguaje se ha hecho muy popular, gracias a varias razones como:

- La cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, su ayuda para realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero.
- La sencillez y velocidad con la que se crean los programas.
- Además es gratuito y usable tanto en Linux como Windows y Mac Os.

La principal razón para realizar este proyecto en Python es la cantidad de librerías útiles para este proyecto(limpieza de datos o análisis de sentimientos).

Para instalar las librerías que vamos a comentar a continuación hacemos uso de **pip install** el instalador de paquetes principal de Python, para usarlo simplemente **sudo pip install <paquete>**

- Librería **requests[5]**. hace que la integración con servicios web sea transparente, peticiones HTTP en este caso para llamar a la API
- Librería **Json[6]**. Codifica o decodifica archivos en formato JSON
- Librería **pandas[7]**. Análisis de datos y usos de estructuras de datos, el bloque de comentarios de Youtube en nuestro caso.
- Librería **nltk[8]**. Limpia todas las palabras en distintas fases antes de ejecutar el análisis de sentimientos. Usa “from nltk.tokenize import sent_tokenize, word_tokenize” para tokenizar las palabras, “from nltk.corpus import stopwords” para eliminar los signos de puntuación y “from nltk.sentiment.vader import SentimentIntensityAnalyzer” para usar el algoritmo pre-entrenado Vader que ejecuta el análisis de sentimientos
- Librería **matplotlib[9]**. Englobando “import matplotlib.pyplot as plt” lo utilizamos en este proyecto para expresar mediante una gráfica el promedio mensual del sentimiento de los comentarios de Youtube.

- Librería **string[10]**. Necesaria para la limpieza de los datos antes de ser analizados, concretamente la depuración de los signos de puntuación.

3.3 API de Youtube

La API[11] o Interfaz de Programación de Aplicaciones de Youtube hace referencia al conjunto de procesos, métodos o funciones de las distintas bibliotecas que posee Youtube para poder extraer la información, almacenarla en nuestro proyecto y posteriormente analizarla.

Para hacer uso de esta API necesitamos acceder a la consola de desarrolladores de Google y crear un proyecto con nuestro usuario y contraseña. Una vez hecho esto tenemos a nuestra disposición la posibilidad de activar varias APIs, en nuestro caso vamos a hacer uso de la API Youtube v3 que tiene como operaciones principales las siguientes:

Operaciones	Definición
list	Recupera (GET) una lista de cero o más recursos.
insert	Crea (POST) un nuevo recurso
update	Modifica (PUT) un recurso existente para reflejar los datos de tu solicitud.
delete	Elimina un (DELETE) recurso específico.

Tabla 1: Operaciones principales de la API de Youtube

Una vez tengamos creado el proyecto en la consola de desarrolladores de Google necesitaremos, por un lado, activar y añadir la API a nuestro desarrollo y por otro lado, para que sea seguro, crear un token de acceso único que usaremos para hacer las llamadas desde nuestro Script.

Utilizaremos la API para hacer las peticiones en las que descargamos estadísticas del video y todos los comentarios en cadenas de texto para procesarlos y analizarlos.

3.4 Algoritmo pre-entrenado Vader

Para la ejecución del análisis de sentimientos utilizaremos la librería de **nltk vaderSentiment** ,específicamente **from nltk.sentiment.vader import SentimentIntensityAnalyzer.** Tendremos que instalarlo en nuestro sistema, para lo cual ejecutamos el instalador de paquetes de Python[12] **$ pip install vaderSentiment.**

Esto se basa en el algoritmo VADER (Valence Aware Dictionary and sEntiment Reasoner)[13] las cuales son un conjunto de reglas léxicas basadas en el análisis de sentimientos aplicado en redes sociales. Decimos que es un algoritmo pre-entrenado porque tiene una serie de reglas que vienen definidas por defecto a la hora de mandarle a realizar un análisis de sentimientos. Esto hace que sea una herramienta muy eficaz a la hora de analizar los resultados porque ha sido testeado con una gran cantidad de palabras que denotan los distintos tipos de sentimientos.

El resultado principal sobre el que podemos sacar conclusiones son los sentimientos positivos, negativos o neutros que el algoritmo arroja a partir de analizar las distintas palabras de, en nuestro caso, los comentarios de Youtube de los usuarios.
Internamente el algoritmo hace un análisis comparando esos parámetros o reglas léxicas que he comentado anteriormente con cada una de las palabras de los comentarios de Youtube una vez estén preparados para analizar. A continuación se comentarán algunas de esas reglas léxicas.

- *Incremento empírico de la calificación de intensidad de sentimiento promedio para la palabra de refuerzo*

  B_INCR = 0.293
  B_DECR = -0.293

- *empíricamente derivado de la intensidad de sentimiento media aumentó de calificación para el uso de MAYÚSCULAS para enfatizar una palabra*

  C_INCR = 0.733

  N_SCALAR = -0.74

Además tenemos varios conjuntos de datos almacenados en variables que sirven para el análisis, datos de palabras positivas y negativas con las que emitir el veredicto final. Este algoritmo también tiene en cuenta cuestiones como la ironía o el sarcasmo, ya que hace una estimación comparando el resto de palabras de la frase

y determinando si cada palabra individualmente aporta su significado literal o tiene otra connotación.

Puntualizar que esta clase de pre-entrenado solo está disponible en inglés así que para este proyecto se ha escogido un video sobre el turismo en Tenerife por lo que teniendo en cuenta el factor idioma tanto para el inglés Británico como para el americano. En el resultado de este proyecto se puede observar observar que cada comentario muestra la fecha de publicación y un valor de sentimiento que oscila entre -1 (negativo), 0 (neutro) y 1 (positivo).

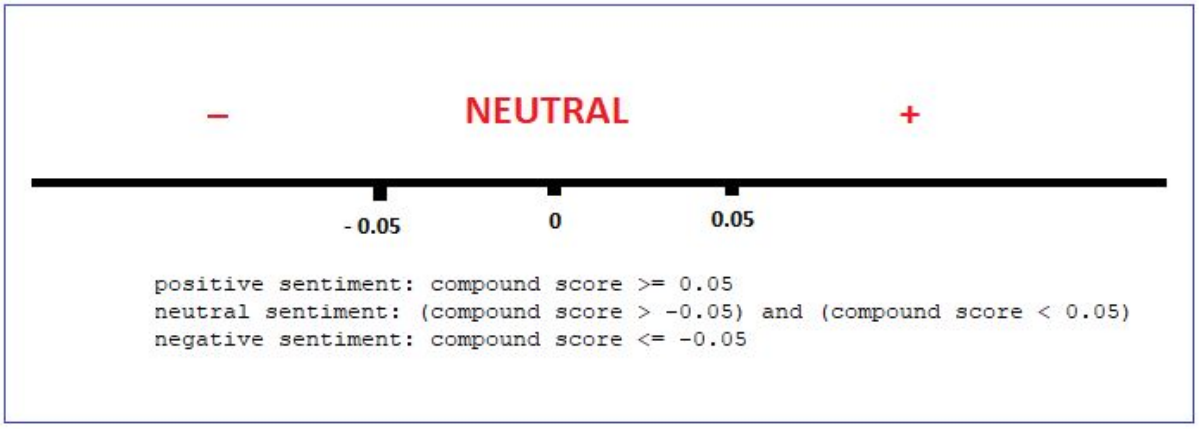


Ilustración 5: Vader escala de sentimientos

4. Desarrollo del proyecto

4.1 Petición y almacenamiento de datos.

En este primer apartado del desarrollo, nos centraremos en hacer una petición formal autenticada a la API de Youtube para extraer los datos exactos que necesitamos.

Para ello necesitamos declarar en nuestro Script una serie de constantes:

- **videoId** del vídeo del canal de Youtube
- **channelId** del canal de Youtube
- **key** de autenticación
- **url** de la API con parámetros internos.
- **type**: el tipo de contenido que estamos buscando. Puede ser video, lista de reproducción, canal.

- **part**: El tipo de datos que obtenemos en la respuesta. Sólo podemos elegir un fragmento, que devuelve toda la información básica, como el título, la descripción y las miniaturas.No podemos obtener ninguna estadística.
- **order**: campo en el que los resultados son ordenados, hemos elegido **viewCoun**t, pero puede ser también **likeCount** o **dislikeCount**. Por defecto, los resultados están ordenados por pertinencia.
- **maxResults**: el número de resultados en una respuesta. Es un número entero entre 0 y 50. Aunque se puede aumentar para realizar menos peticiones, en este caso está delimitado en 100.

De esta manera podemos realizar un **requests** para almacenar los datos.

Inicialmente se realizó un estudio del canal de Youtube que nos reportó la lista de los videos asociados con el canal que hemos elegido y otro método que nos devolverá todas las estadísticas asociadas a un vídeo concreto.

- **get_channel_videos()**
- **get_statistics()**

```
def get_channel_videos(channelId):
        url = 'https://www.googleapis.com/youtube/v3/search'
        pms = {'type': 'video', 'key': 'AIzaSyAohEyyDRPOQja4XJWbZfNFJqjr2p4R1F0', 'channelId': channelId ,
        'part':'snippet', 'order':'viewCount','maxResults':50}
        res = requests.get(url, params = pms)
        data = res.json()
        lst = []
        for video in data['items']:
            video_stats = get_statistics(video['id']['videoId'])
            results_json = {
            'channelTitle' : video['snippet']['channelTitle'],
            'title' : video['snippet']['title'],
            'publishedAt' : video['snippet']['publishedAt'],
            'videoId' : video['id']['videoId'],
            'viewCount' :
             video_stats['items'][0]['statistics']['viewCount'],
            'commentCount' : video_stats['items'][0]['statistics']
            ['commentCount'],
            'likeCount' :
             video_stats['items'][0]['statistics']['likeCount'],
            'dislikeCount' : video_stats['items'][0]['statistics']
            ['dislikeCount'],
            }
            lst.append(results_json)

            df = pd.read_json(json.dumps(lst))
            return(df)
#print(get_channel_videos(channelId)) #
```

Ilustración 6: Método get_channel_videos

```
def get_statistics(videoId):#video id
    url = 'https://www.googleapis.com/youtube/v3/videos'
    pms = {'key': 'AIzaSyAohEyyDRPOQja4XJWbZfNFJqjr2p4R1F0', 'id': videoId,
    'part':'contentDetails,statistics'}
    res = requests.get(url, params = pms)
    data = res.json()
    return(data)
```

Ilustración 7: Método get_statistics

Como resultado ambas funciones nos dan un objeto Json que contiene los datos correspondientes en cada caso. Podemos observar como se le pasan por parámetro el **channelId** y el **videoId** respectivamente y mediante una petición requests previamente autenticada almacenamos los datos. Para realizar esto tenemos que obtener como se ha comentado anteriormente la key de acceso.

4.1.1 Key de acceso

Para obtener una key de la API de Youtube y poder realizar todo lo descrito anteriormente tenemos que seguir distintos pasos.

1. Logearnos en: https://console.developers.google.com/

2. Habilitar la API pulsando **enable**

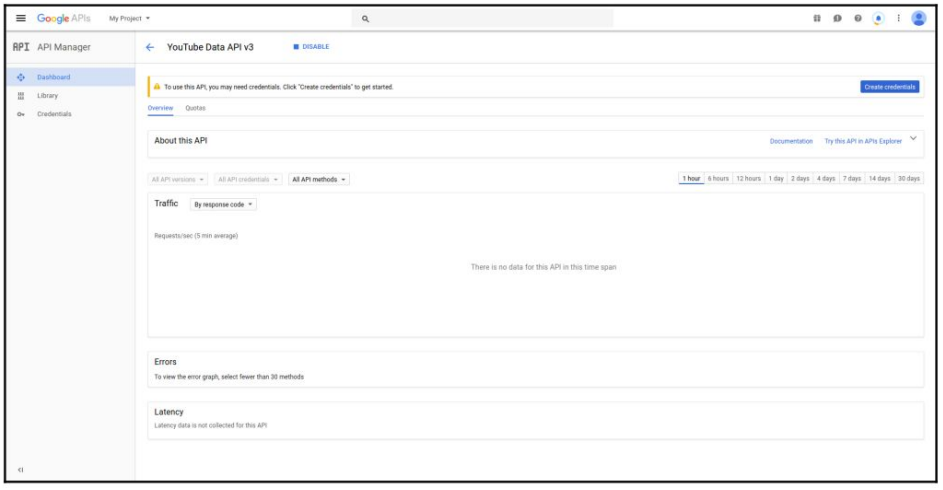


Ilustración 8: Habilitación API

3. Crear un proyecto y ponerle nombre

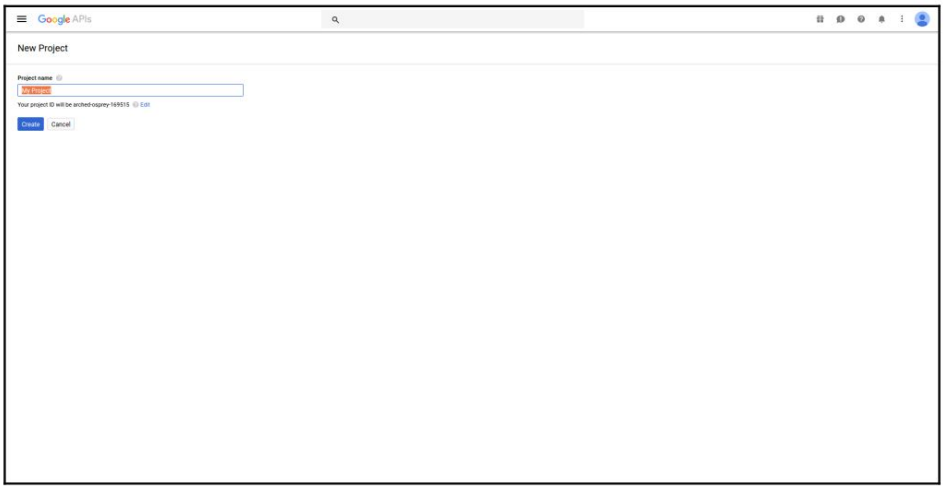


Ilustración 9: Creación de proyecto

4. Agregar credenciales y crear una nueva clave en el área de acceso de API pública

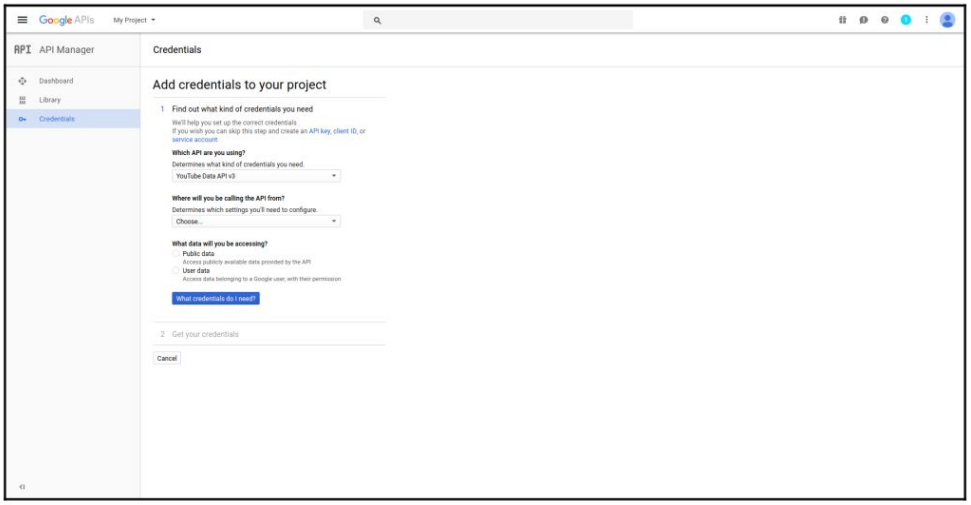


Ilustración 10: Agregar credenciales.

5. Finalmente copiar las credenciales de acceso a nuestro Script como vemos a continuación

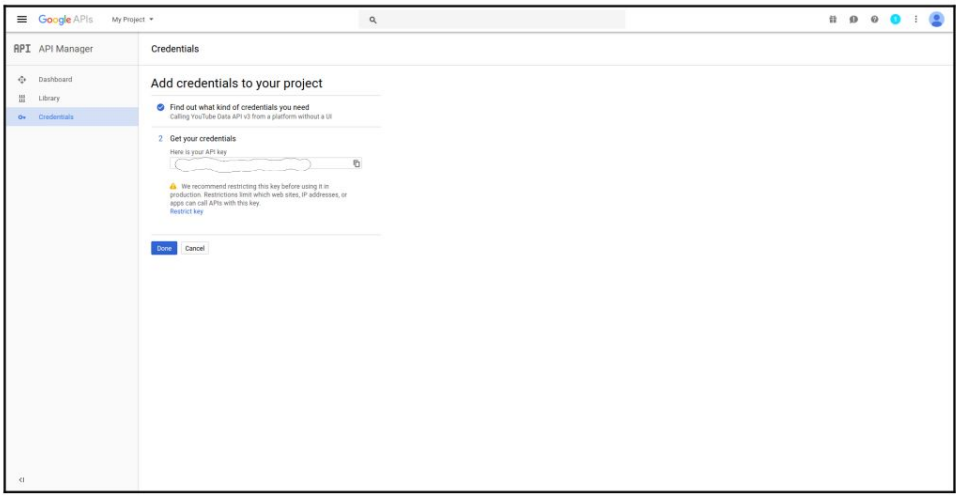


Ilustración 11: Copiar credenciales

Una vez realizados estos pasos podemos realizar las peticiones necesarias a la API para traer los datos con nuestro Script. Dichas peticiones quedarán registradas en distintos gráficos dentro de nuestro proyecto en la consola de desarrolladores de Google.

4.2 Script para obtención de datos del video

La librería **requests** es usada para hacer llamadas a la API de Youtube como podemos ver a continuación:

```
url = 'https://www.googleapis.com/youtube/v3/videos'
pms = {'key': 'AIzaSyAohEyyDRPOQja4XJWbZfNFJqjr2p4R1F0', 'id': videoId,
'part':'contentDetails,statistics'}
res = requests.get(url, params = pms)
#print_response(res)
data = res.json()
#print(data)
```

Ilustración 12: Librería requests

Realizamos la llamada a la API mediante la URL con los datos de autenticación comentados anteriormente, como resultado nos da un objeto de tipo **res** que transformamos en un objeto Json con la siguiente estructura:

'statistics': { 'commentCount': '8962',
'dislikeCount': '3713',
'favoriteCount': '0',
'like Count': '47479',
'view Count': '4140155'
}

Especificar que si realizamos un print de este objeto Json realmente nos devuelve un conjunto de datos mayor junto con el esquema anterior, muchos más parámetros. Ejecutandolo como ejemplo en nuestro proyecto nos reporta lo siguiente:


```
{  'kind': 'youtube#videoListResponse',
'etag': '"XpPGQXPnxQJhLgs6enD_n8JR4Qk/6NG41hcWjdcjlI7-HgkLV8LFZp4"',
 'pageInfo': {'totalResults': 1, 'resultsPerPage': 1},
 'items': [{'kind': 'youtube#video', 'etag': '"XpPGQXPnxQJhLgs6enD_n8JR4Qk/8hOyo_Q0ybd4I-sfVQz5Rj78pVM"',
  'id': 'vvR9fTZl6fU',
  'contentDetails': {'duration': 'PT8M28S', 'dimension': '2d', 'definition': 'hd', 'caption': 'false', 'licensedContent': True, 'projection': 'rectangular'},
  'statistics': {'viewCount': '1078421', 'likeCount': '3499', 'dislikeCount': '204', 'favoriteCount': '0', 'commentCount': '536'}}]}
```


Ilustración 13: Print del objeto Json

El resultado nos devuelve en forma de resumen todos los parámetros relacionados con los aspectos técnicos del video, páginas totales, etiquetas y el tipo de dato que

tenemos. Duración, definicion del video, licencia, proyección y distintos datos que podemos observar en la Ilustración.

Sin embargo estos datos pedidos a la API[14] sobre el video son meramente informativos, nos dan una visión general del vídeo pero no son los datos que vamos a analizar en profundidad. Para ello necesitamos realizar otra llamada a la API de Youtube, esta vez para traer los comentarios que tiene el vídeo.

```
#VAMOS A TRAER AHORA LOS COMENTARIOS
url = 'https://www.googleapis.com/youtube/v3/commentThreads'
full_data = [] #return list
page = ''      #initialization of paging
while True:
    pms = {'part': 'snippet', 'videoId': videoId, 'maxResults': 100, 'key': 'AIzaSyAohEyyDRPOQja4XJWbZfNFJqjr2p4R1F0'
           'pageToken': page
           }
    res = requests.get(url, params = pms)
    #print_response(res1) RESPUESTA DEL GET CON COMENTARIOS
    print("Estado de la conexion: %s" % res) # check the status of the connection
    data = res.json()
    full_data.extend(data['items'])
    print("Descargado: %s, Total: %s" % (len(data['items']),
    len(full_data)))
    try:
        page = data['nextPageToken']
        #print(page)
    except:
        break
```

Ilustración 14: Llamada a la API commentThreads

Como vemos en la Ilustración realizamos otra llamada a la API, concretamente **commentThreads**, esto nos retornará una lista de comentarios que coincidan con los parámetros pasados por parámetro en la petición requests, es decir, los parámetros que hemos comentado anteriormente como nuestro **videoId**, **channelId** y la **key** de nuestro proyecto.

La variable **full_data** contiene en este momento todos los datos asociados al video en cuestión, al contener todos los datos incluido los comentarios y toda su información se observa una extensión demasiado grande para plasmarla en una Ilustración. A continuación se mostrará un ejemplo genérico del contenido de dicha variable.

{ 'etag': '"m2yskBQFythfE4irbTIeOgYYfBU/XgRZI3UhbKRFQZGd-n-2OCOKR8A"',
'id': 'z13jynay3seygpvzy04cef5r2tm5ihh4d0k',
'kind': 'Youtube#commentThread',
'snippet': {'canReply': False,
'isPublic': True,
'topLevelComment': {'etag':
'"m2yskBQFythfE4irbTIeOgYYfBU/jW3EfLcy4MnIFrtFloQEjokBPXU"',
'id': 'z13jynay3seygpvzy04cef5r2tm5ihh4d0k',
'kind': 'Youtube#comment',
'snippet': {'authorChannelId': {'value': 'UCuJkT4Bsd1qiIQoI8Rbf_hg'},
'authorChannelUrl':
'http://www.Youtube.com/channel/UCuJkT4Bsd1qiIQoI8Rbf_hg',
'authorDisplayName': 'Vince / FCB',
'authorProfileImageUrl': 'https://yt3.ggpht.com/-9DZP7TJ0J-
4/AAAAAAAAAAI/AAAAAAAAAAA/dvUasDtmZFw/s28-c-k-no-mo-rj-
c0xffffff/photo.jpg',
'canRate': False,
'likeCount': 0,
'publishedAt': '2017-05-06T17:52:51.000Z',
'textDisplay': 'i don't like this',
'textOriginal': "i don't like this",
'updatedAt': '2017-05-06T17:52:51.000Z',
'videoId': 'YQUpg795iBo',
'viewerRating': 'none'}},
'totalReplyCount': 0,
'videoId': 'YQUpg795iBo'}}

Nos centraremos en el fragmento de código donde podemos encontrar **textOriginal** y **publishAt** valores.

Los **Dataframe[15]** son formas muy eficientes de almacenar y procesar datos; por ello agregaremos información relevante a dichas estructuras, las cadenas de palabras que forman los comentarios así como su fecha de publicación.

```
df = pd.DataFrame()
df['comments'] = [k['snippet']['topLevelComment']['snippet']['textDisplay'] for k in full_data]
df['date'] = [k['snippet']['topLevelComment']['snippet']['publishedAt'] for k in full_data]
```

Ilustración 15: Dataframe con comentarios y su fecha

En nuestro caso se utilizarán los comentarios como una función a lo largo del tiempo. Para hacer esto tenemos que configurar el **index** del **Dataframe** como un objeto de fecha y hora tal y como se muestra a continuación:

```
df = df.set_index(['date'])
df.index = pd.to_datetime(df.index)#ponemos como indice la fecha del comentario y lo convertimos a un objeto de datetime
```

Ilustración 16: Date index Dataframe

Por tanto nuestro data frame tendrá ahora dos columnas,una formada por la fecha y la hora en la que se creó el comentario y otra con el propio comentario, de esta manera queda más clara la representación de dicha información. Se mostraría de la siguiente manera:

	comments
date	
2019-05-22 18:16:30	ik im going tomz
2019-05-21 18:53:39	you can find this waterpark on roblox lol
2019-04-30 17:23:23	I've been. It's lit
2019-04-19 17:05:43	VIVA ESPAÑA!!! ESESESES
2019-04-18 16:27:51	I already have been there . It is amazing!❤
2019-04-15 18:30:53	Fuck you! Stupid lifeguard in once
2019-04-14 17:24:50	Ich gehe am Mittwoch da hin 😍
2019-04-09 08:41:24	Im goin there today!😍
2019-04-06 02:06:08	song name?
2019-03-19 01:02:50	I've been tenerife 3 time n I'm going ...

Ilustración 17: Muestra de Dataframe

Podemos observar que ya tenemos los datos almacenados pero no listos para analizar. Hay que proceder a la limpieza de los datos quitando emoticonos que se muestran en la ilustración 17 así como demás signos de puntuación y caracteres raros.

4.3 Procesamiento y preparación de datos

Los comentarios están en el **Dataframe** tal y como fueron extraídos de Youtube mediante la API. Por lo que, dichos comentarios contienen muchos caracteres, signos de puntuación, emojis y otros elementos que crean ruido en el análisis de los sentimientos. Antes de calcular el análisis de sentimiento en cada comentario debemos procesar el texto para eliminar todos estos elementos. El flujo del procesamiento de los datos consta de las siguientes partes:

- Tokenización
- Conversión a minúsculas
- Eliminación de palabras claves
- Eliminación de la puntuación
- Eliminación de palabras de menos de dos caracteres.

La forma de proceder será crear un método que ejecute todos los pasos del procesamiento de los datos uno detrás de otro para finalmente aplicar dicho método a nuestro Dataframe.

Para las tareas de procesamiento de lenguaje utilizaremos la biblioteca NLTK, concretamente con estos subpaquetes:

*from **nltk.tokenize import sent_tokenize, word_tokenize***
*from **nltk.corpus import stopwords***

En la siguiente ilustración podemos observar el método **clean** que hará todo el procesamiento de los datos:

```
def clean(text):
    tokens = word_tokenize(text.strip())
    clean = [ i.lower() for i in tokens]
    clean = [ i for i in clean if i not in stopwords.words('english')]
    clean = [ i.strip(''.join(punctuation)) for i in clean if i not in list(string.punctuation)]
    clean = [ i for i in clean if len(i) > 1]
    return " ".join(clean)
    df['clean_comments'] = df['comments'].apply(clean)
    return (" ".join(clean))

df['clean_comments'] = df['comments'].apply(clean)
```

Ilustración 18: Método clean

Invocamos los métodos **word_tokenize** que **mediante la función strip()** separa cada comentario en palabras individuales, el **método lower()** donde convertimos todas las posibles mayúsculas a minúsculas, eliminamos mediante **stopwords** los símbolos de puntuación siempre que sean en inglés, otros casos serán ignorados. Por último eliminamos las palabras que tengan menos de 2 caracteres mediante el método **len()** ya que no nos aportaran nada para el análisis de sentimientos.

Como resultado de toda esta preparación, se adicionan todas las fases del análisis para aplicarlas a la cadena de caracteres pasada por parámetros en el método clean().

A continuación definimos una nueva columna en el Dataframe llamada **clean_comments** que será el resultado de la columna de comentarios sin limpiar **comments aplicando el método clean() mediante apply(clean)**

date	clean_comments
2019-05-22 18:16:30	ik im going tomz
2019-05-21 18:53:39	find waterpark roblox lol
2019-04-30 17:23:23	39 39 lit
2019-04-19 17:05:43	viva españa ESESESES
2019-04-18 16:27:51	already amazing
2019-04-15 18:30:53	fuck stupid lifeguard
2019-04-14 17:24:50	ich gehe mittwoch da hin
2019-04-09 08:41:24	im goin today
2019-04-06 02:06:08	song name
2019-03-19 01:02:50	39 tenerife time 39 going like 62 days second ...
2019-03-18 16:09:01	ques like siam water park
2019-03-05 19:36:58	shinga kinaree best slides

Ilustración 19: Dataframe al aplicarle clean()

Como vemos en la Ilustración tenemos nuestra nueva columna del Dataframe llamada clean_comments lista para poder realizarle un análisis de sentimientos a excepción de un detalle. Siguen existiendo caracteres raros como emoticonos, tenemos que limpiarlo para no enturbiar el análisis, en el caso de que no hiciéramos ninguna criba, estos caracteres el algoritmo ignoraria dichos comentarios y pasaría a los siguientes pero puede reducir el porcentaje de aciertos notablemente.

Por tanto se aplicará un segundo método a nuestro Dataframe, elim_emoji(), al cual le pasamos por parámetro los comentarios infectados con emojis y este método mediante la función compile() los transforma a objetos en código ASCII; a continuación filtramos los caracteres que coincidan con emoticonos para dejar nuestro comentario limpios. De nuevo mediante apply() se lo aplicamos a nuestro Dataframe y quedaría listo para analizar sin ningun caracter extraño.

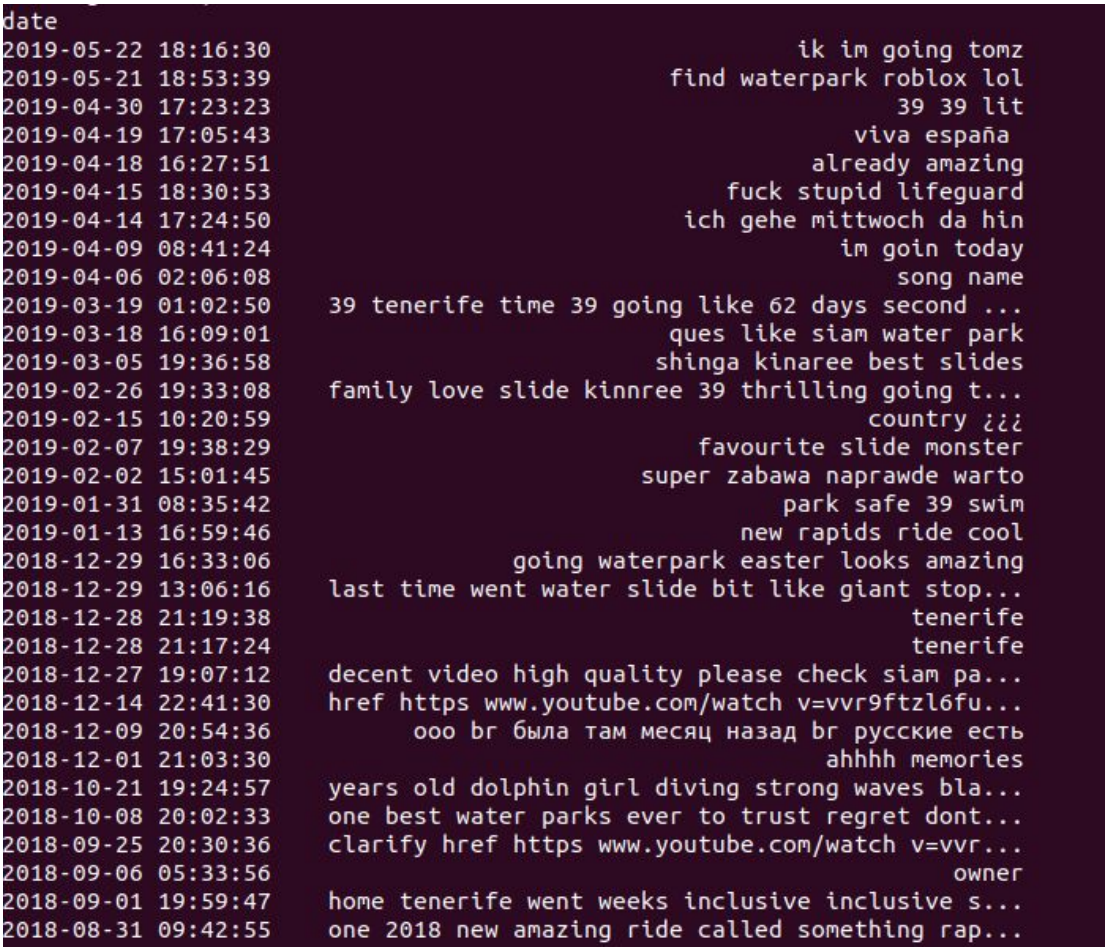
A continuación se muestra el resultado obtenido en las siguientes dos Ilustraciones. Puntualizar que utilizamos un método para imprimir el Dataframe seleccionando las columnas presentes así como el índice, siendo la fecha en este caso, como podemos ver en las ilustraciones 20 y 21 la lista de comentarios está limpia de cualquier tipo de emoticono:

```
def elim_emoji(string):
    emoji_pattern = re.compile("["
                           u"\U0001F600-\U0001F64F"  # emoticonos
                           u"\U0001F300-\U0001F5FF"  # simbolos y pictogramas
                           u"\U0001F680-\U0001F6FF"  # transport & map symbols
                           u"\U0001F1E0-\U0001F1FF"  # flags (IOS)
                           u"\U00002702-\U000027B0"
                           u"\U000024C2-\U0001F251"
                           "]+", flags=re.UNICODE)
    return emoji_pattern.sub(r'', string)

df['clean_comments'] = df['clean_comments'].apply(elim_emoji)

with pd.option_context('display.max_rows', None, 'display.max_columns', None):
    print(df['clean_comments'])
```

Ilustración 20: Método elim_emoji Ilustración



date	
2019-05-22 18:16:30	ik im going tomz
2019-05-21 18:53:39	find waterpark roblox lol
2019-04-30 17:23:23	39 39 lit
2019-04-19 17:05:43	viva españa
2019-04-18 16:27:51	already amazing
2019-04-15 18:30:53	fuck stupid lifeguard
2019-04-14 17:24:50	ich gehe mittwoch da hin
2019-04-09 08:41:24	im goin today
2019-04-06 02:06:08	song name
2019-03-19 01:02:50	39 tenerife time 39 going like 62 days second ...
2019-03-18 16:09:01	ques like siam water park
2019-03-05 19:36:58	shinga kinaree best slides
2019-02-26 19:33:08	family love slide kinnree 39 thrilling going t...
2019-02-15 10:20:59	country ¿¿¿
2019-02-07 19:38:29	favourite slide monster
2019-02-02 15:01:45	super zabawa naprawde warto
2019-01-31 08:35:42	park safe 39 swim
2019-01-13 16:59:46	new rapids ride cool
2018-12-29 16:33:06	going waterpark easter looks amazing
2018-12-29 13:06:16	last time went water slide bit like giant stop...
2018-12-28 21:19:38	tenerife
2018-12-28 21:17:24	tenerife
2018-12-27 19:07:12	decent video high quality please check siam pa...
2018-12-14 22:41:30	href https www.youtube.com/watch v=vvr9ftzl6fu...
2018-12-09 20:54:36	ooo br была там месяц назад br русские есть
2018-12-01 21:03:30	ahhhh memories
2018-10-21 19:24:57	years old dolphin girl diving strong waves bla...
2018-10-08 20:02:33	one best water parks ever to trust regret dont...
2018-09-25 20:30:36	clarify href https www.youtube.com/watch v=vvr...
2018-09-06 05:33:56	owner
2018-09-01 19:59:47	home tenerife went weeks inclusive inclusive s...
2018-08-31 09:42:55	one 2018 new amazing ride called something rap...

Ilustración 21: Dataframe tras ejecutar elim_emoji

4.4 Cálculo del análisis de sentimientos

El análisis de sentimientos será aplicado sobre la columna de comentarios limpia mediante la librería **nltk** usando el clasificador de Vader:

```
#SENTIMIENTOS
sentiment = SentimentIntensityAnalyzer()
df['sentiment'] = df['clean_comments'].apply(lambda txt: sentiment.polarity_scores(txt) ['compound'])
```

Ilustración 22: Análisis de sentimientos

La nueva columna contiene un sentimiento compuesto para cada comentario, donde los valores más cercanos a +1 describen la actitud positiva y los valores más cercanos a -1 actitud negativa, respectivamente.

Se realizó mediante la creaciń de un objeto del tipo **SentimentIntensityAnalyzer()** que será el analizador, utilizado junto con el texto que le pasemos, en este caso los comentarios. Mediante la función polarity_scores() de la librería nltk obtenemos la polaridad a la que se inclina cada comentario, positiva, negativa o neutra en valores uno, menos uno o cero respectivamente. Usando el score de **compound** se suman los score de cada palabra en el léxico. Se ajusta de acuerdo con las reglas y luego se normaliza entre -1 (negativo extremo) y +1 (positivo extremo).

4.4.1 Sentimientos a lo largo del tiempo

4.4.1.1 Sentimientos a lo largo de los meses.

En primer lugar realizaremos el análisis de sentimientos de los datos obtenidos centrándonos en la variación de estos a lo largo de los meses. Específicamente en los periodos comprendidos entre el inicio del año 2015 hasta abril del vigente año. Ya que sería inviable contabilizar los datos de cada mes, haremos saltos de 6 meses como veremos posteriormente en la gráfica.

El código para realizar dicha tarea se basa en la **librería matplotlib** y en la función **resample()** pasándole el parámetro **M** que designa meses.

```
#Average score per month
#La M es de mes, se puede poner B business day, D calendar day, W weekly...
df['sentiment'].resample('M').mean()
df['sentiment'].plot(title = "Sentimiento a lo largo de los meses", lw=2, ylim = (-1,1))

plt.axhline(0, color='k', lw = 2)
plt.xlabel('Fecha')
plt.ylabel('Sentiment score')

plt.show()
```

Ilustración 23: Sentimiento a lo largo de los meses

4.4.1.2 Sentimiento promedio por día de la semana.

Para calcular el sentimiento promedio por día de la semana utilizaremos el método **groupby() de la librería matplotlib** lo que nos permitirá delimitar por **weekday**. En este análisis se contempló que el inicio se empiece a contar desde una fecha lo más cercana posible al inicio de los comentarios en el video, en este caso **"2015-07-10"**

```
#SENTIMENT BY WEEKDAY
dx = df[df.index > '2015-07-18']
ax = dx.groupby(dx.index.weekday)['sentiment'].mean().plot(kind = 'bar',title = 'Sentimiento promedio por dia de semana')
ax.set_xticklabels(['Lunes', 'Martes', 'Miercoles', 'Jueves', 'Viernes','Sabado', 'Domingo'])
plt.show()
```

Ilustración 24: Sentimiento promedio por día

4.4.1.3 Comentarios a lo largo de los años

Resulta interesante saber en qué periodos a lo largo de los años los usuarios han comentado más activamente, es por ello que filtramos de nuevo por meses usando el **método resample(M) e invocando la función count()** para contar el número de comentarios y delimitarlos.

```
"""
COMMENTS IN TIME
"""

df['sentiment'].resample('M').count().plot()

plt.axhline(0, color='k', lw = 2)
plt.xlabel('Fecha')
plt.ylabel('Numero de comentarios')

plt.show()
```

Ilustración 25: Comentarios a lo largo de los años

4.4.1.4 Después de la fecha de publicación

La mayoría de videos reciben más tráfico en cuanto a comentarios e interacciones de los usuarios en momentos muy cercanos a la fecha de publicación del video. En este caso se modificó el inicio de la fecha en la que se cuenta el número de comentarios situandola 1 año después de la publicación del video, concretamente el 18 de Octubre de 2016, observaremos más adelante los resultados.

```
dx = df[df.index > '2016-10-18']
dx['sentiment'].resample('M').count().plot()

plt.axhline(0, color='k', lw = 2)
plt.xlabel('Fecha')
plt.ylabel('Numero de comentarios meses despues')

plt.show()
```

Ilustración 26: Comentarios después de la fecha marcada

4.4.1.5 Promedio de comentarios por dia de la semana

Para abordar este apartado ilustraremos la media de comentarios agrupados por dia de la semana. Comenzaremos a contar 1 mes después de la fecha de publicación del video tal y como hemos hecho anteriormente.

Para ello agrupamos los comentarios comenzando con el índice de la fecha comentada un mes después de la fecha de publicación del video y usando la columna del Dataframe donde se alojan los comentarios; le pasamos los datos al método count() para hacer una cuenta, mediante el parámetro **kind** designamos el tipo de gráfico que queremos y lo mostramos.

```
dx = df[df.index > '2015-09-18']
ax = dx.groupby(dx.index.weekday)['comments'].count().plot(kind = 'bar',
title = 'Numero de comentarios por dia')
ax.set_xticklabels(['Lunes', 'Martes', 'Miercoles', 'Jueves', 'Viernes','Sabado', 'Domingo'])
plt.show()
```

Ilustración 27: Promedio de comentarios por dia de la semana.

5 Conclusiones y resultados

A continuación presentaremos las conclusiones tras haber realizado el proyecto y los resultados de manera gráfica de las distintas cuestiones.

5.1 Evolución del sentimiento a lo largo del tiempo

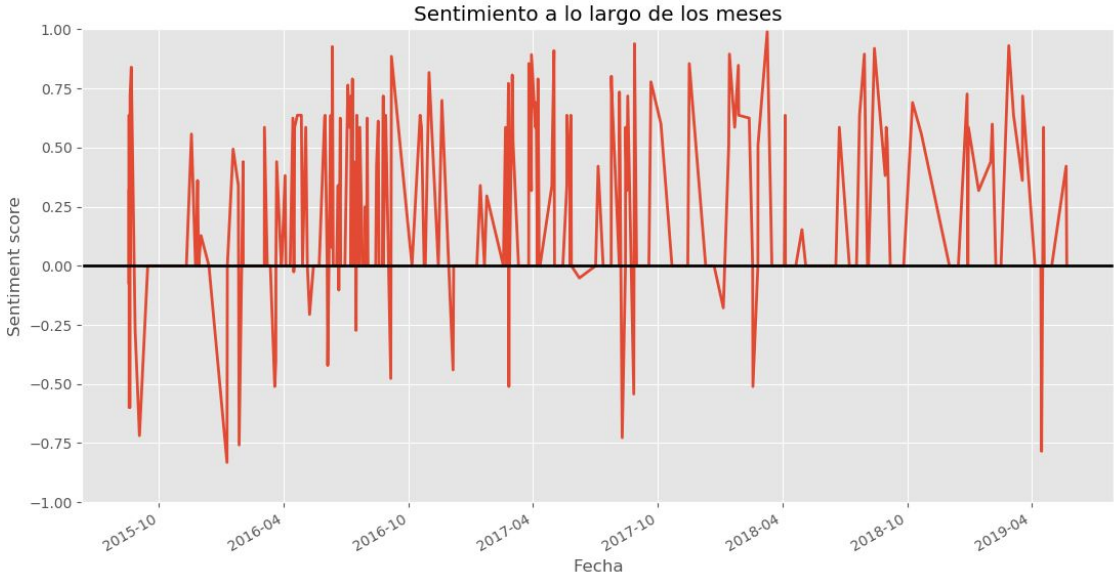


Ilustración 28: Gráfica de sentimientos a lo largo del tiempo

El gráfico muestra la evolución del sentimiento para el video analizado. Hemos delimitado el eje x con periodos de seis meses desde agosto de 2015 hasta mayo de 2019. De manera global el sentimiento para este video es bastante positivo. Sin embargo, hay momentos en los que la curva pasa claramente al lado negativo. Tratándose de un tema de turismo en un lugar concreto, podríamos achacar dichos picos a distintos momentos del año en relación a la meteorología, las ofertas de viaje o los periodos con más afluencia de extranjeros.

Se observa el pico más negativo aproximadamente entre los meses de diciembre y enero de 2015 y 2016, momentos del año en los que el tiempo es más frío y puede causar el descontento de los clientes para con el parque acuático.

Por otro lado el pico más positivo no se produce en el mes más caluroso del año pero haciendo una media sí que todos los picos más positivos se producen en los meses menos fríos de cada año. Esta y otras cuestiones pueden ser la causa de esta distribución.

5.2 Periodos con más comentarios

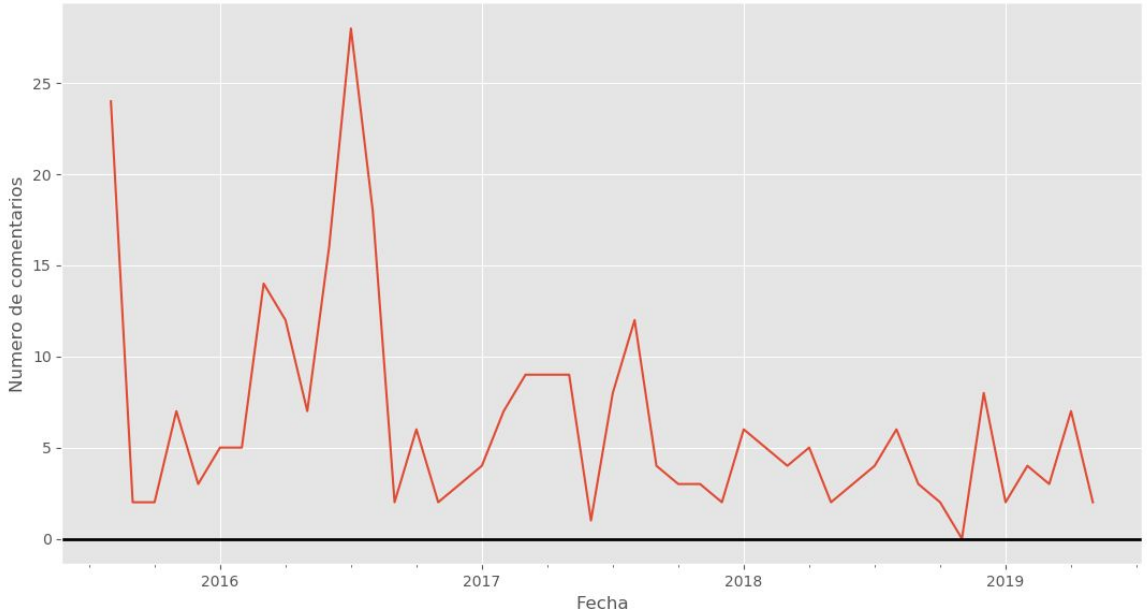


Ilustración 29: Gráfica de comentarios a lo largo del tiempo

Podemos observar que el primer gran pico de comentarios es justo en la fecha de publicación del video y en fechas cercanas, sin embargo el mayor pico de comentarios no ocurre hasta un mes después, acontecimiento que no es normal. Puede ser debido a distintos factores relacionados con el parque acuático como nuevas atracciones, promociones especiales u ofertas que han provocado un pico de actividad por parte de los usuarios.

Como conclusión podemos deducir que el periodo más adecuado para el público objetivo de este tipo de videos no es justo la fecha de publicación de este ni días posteriores sino varios meses o años después. Recordamos que estamos hablando del turismo en Tenerife, específicamente de un parque acuático así que tiene sentido que las opiniones, vivencias o experiencias de la gente en este lugar se vayan formando a lo largo del tiempo.

Si la temática del video fuera sobre un tema de actualidad que cambia radicalmente de forma diaria o semanal, es trivial pensar que el pico de comentarios sería la

fecha de publicación ya que con el paso del tiempo el tema del video en concreto estaría desactualizado o ya no interesaría.

En la siguiente gráfica se muestran los comentarios cambiando la fecha de inicio a varios meses más adelante para evitar el pico inicial y ver realmente la evolución del número de comentarios a lo largo del tiempo, de esta manera veremos de una manera más concisa el comportamiento que tendrá el video en el futuro.

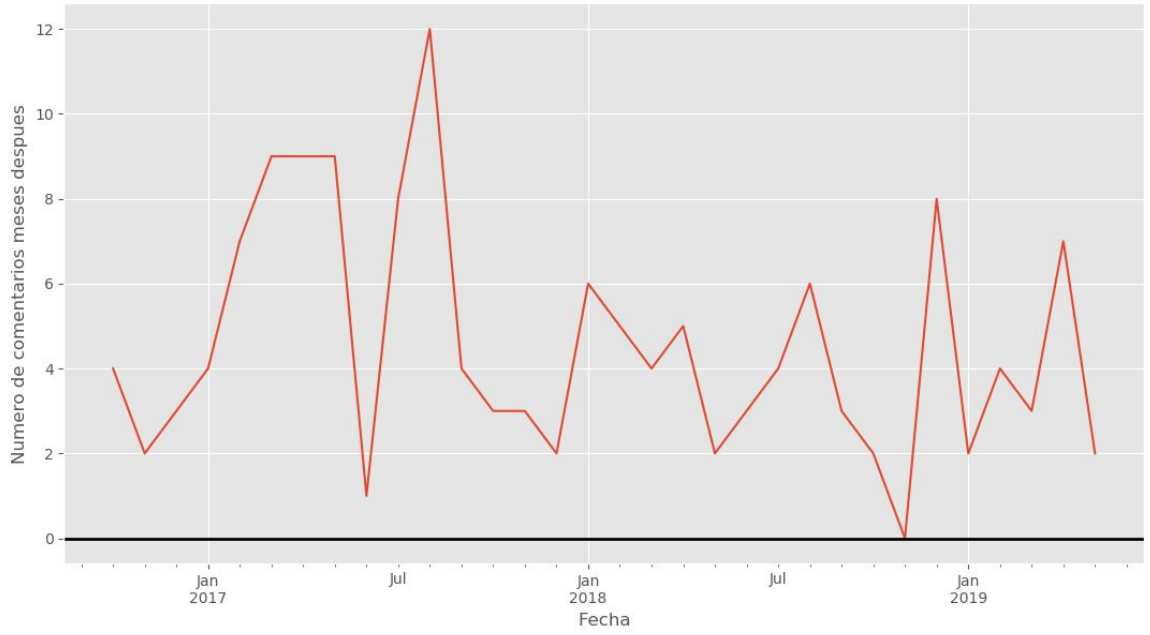


Ilustración 30: Gráfica de comentarios después de fecha prefijada

Se observa que no es hasta dos años después de la fecha de publicación del video cuando la media de número de comentarios se estabiliza, esto probablemente se deba al hecho de que se trate el turismo como tema principal y otros como los videojuegos, “vlogs” o cosméticos.

5.3 Dia de comentarios más positivos

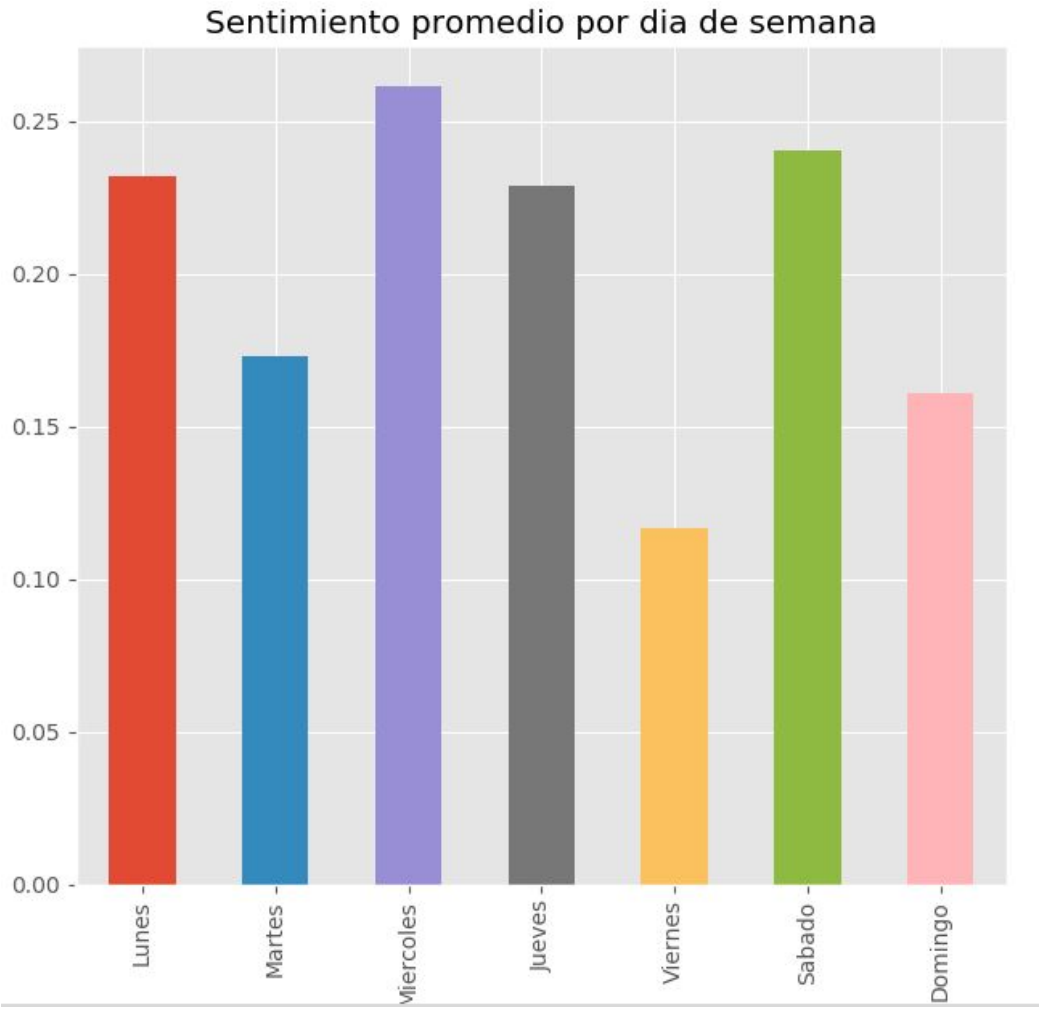


Ilustración 31: Gráfica de días con puntuación de sentimientos por comentarios

El análisis de sentimientos promedio agrupado por día de la semana muestra una tendencia interesante. Mientras que el máximo el número de comentarios es el sábado, el sentimiento más positivo es el miércoles, seguido muy de cerca por el propio sábado.

Como conclusión no podemos asegurar que entre más comentarios haya más positivos sean ni viceversa, el miércoles es el tercer día en el que menos comentarios de media se publican pero es el día en el que más positivos son. Es por ello que no se puede deducir una correlación entre el número de comentarios y su sentimiento, por supuesto no es una regla que pueda generalizarse.

Se podría realizar un estudio de tendencias de combinación de contenido de manera cuantitativa(recuento de comentarios) y cualitativas(sentimientos) para obtener una conclusión más real basándose en estadísticas.

5.4 Dia que se comenta más activamente

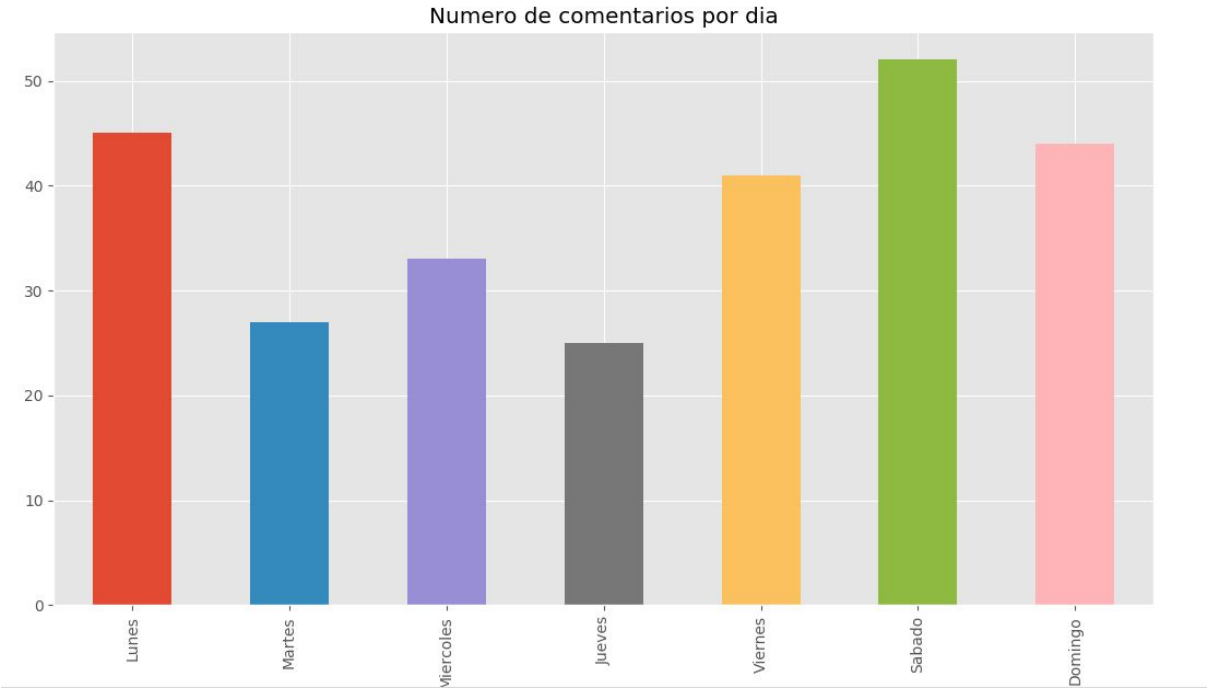


Ilustración 32: Gráfica de dia que se comenta más activamente

El resultado de la visualización nos muestra que el video que elegimos tiene una tasa estable de compromiso con respecto a los días de la semana,aunque cuando observamos más actividad es en el fin de semana, concretamente el sábado, cuando más se comenta, muy cerca del lunes.

Este fenómeno podría explicarse por el hecho de que la audiencia de este video sobre turismo tiene más tiempo para revisar, dar opiniones o vivencias el fin de semana. Podría ser un ejercicio interesante hacer el mismo estudio de tendencias realizado en el subapartado anterior.

Habría que realizar un análisis similar con el resto de videos del canal para ver si es una tendencia que se repite o es un caso aislado. Tales conocimientos sobre los comentarios a lo largo del tiempo resultan excelentes aportaciones para los equipos de marketing de las empresas que necesitan priorizar la publicación de contenidos e interacción con su público.

6 Conclusiones

- Se han realizado peticiones para traer distintos datos mediante la API de Youtube para preparar y analizar una gran cantidad de datos sobre el tráfico de opiniones que han realizado distintos usuarios de Youtube sobre un video.
- Se ha realizado un análisis de sentimientos, una vez limpiados y preparados los datos anteriormente mencionados, sobre el que se han sacado distintas conclusiones.

En base al análisis realizado podemos decir a “grosso” modo que debido a la temática del video, este no sigue la tendencia general sobre temas de actualidad y actualizacion diaria. En el caso de los videos sobre videojuegos, tecnología o contenido creado para entretener, podemos deducir varias máximas que suelen cumplirse con bastante exactitud en la totalidad de los vídeos que se van publicando.

- El mayor pico de visualizaciones se produce el día de publicación del video o los días inmediatamente posteriores.
- El mayor número de comentarios, likes y dislikes se produce exactamente en el mismo periodo de tiempo comentado anteriormente.
- El vídeo tendrá mayor impacto entre más actualidad tenga el tema del que se trata.

En este caso el tema tratado, no entra dentro de ninguna temática de actualidad ni algo que sea extremadamente novedoso. El parque acuático no resulta novedoso debido a su apertura varios años atrás.

Es por ello que podemos observar una tendencia clara de comentarios ni un pico excesivamente grande de comentarios en la fecha de publicación del video.
Sin embargo el video mantiene un flujo constante de comentarios a lo largo del tiempo. En el momento de realización de esta memoria se puede observar que hace pocas horas se han contabilizado varios comentarios así como hace varios días y meses. Estos datos son extravagantes y se salen un poco de las normas generales puesto que el video fue publicado en agosto de 2015, hace casi cuatro años.

Los usuarios o turistas en este caso van aportando su opinión o sus expectativas sobre el parque acuático una vez lo han visitado o lo vayan a visitar a lo largo del tiempo por lo que podemos sacar como una conclusión firme que el video seguirá teniendo tráfico mientras el parque acuático continúe en funcionamiento y con una afluencia masiva de turistas como ha sido el caso desde que abrió sus puertas.

Adentrándonos en la cuestión de cuál es el dia que se comenta más activamente no podemos dar tampoco una respuesta cien por cien. Para poder responder a esta cuestión hay que tener en cuenta varios puntos que son distintos dependiendo del usuario o turista. Podríamos pensar que los días de las semana en los que más se comenta son el fin de semana y que la época del año que más tráfico hay es en periodos vacacionales además de en periodos de verano, donde el tiempo en Canarias es más cálido.

Teniendo en cuenta estas cuestiones, en una buena parte la tendencia de comentarios sigue esa línea pero no podemos decir que la mayoría sea igual porque en este caso los millones de turistas que visitan el parque y luego aportan su comentario no tienen el mismo periodo de vacaciones ni tampoco comentan el mismo dia que han consumido sus días en el parque, es decir, se puede dar el caso de un usuario que comenta su experiencia varios días, semanas o incluso meses después de haber realizado su viaje a Canarias.

Por otro lado siguiendo la reflexión anterior y centrándonos en averiguar cuál es el periodo donde se comenta más positivamente vemos una tendencia similar a lo descrito anteriormente pero nada exacta. Los fines de semana es donde los comentarios son más numerosos, pero el lunes tiene una cifra muy similar, si miramos los sentimientos no cuadra con estos datos. El día que más positivamente se comenta es el miércoles, unos de los días en los que menos comentarios se producen asi que podriamos deducir que a menos número de comentarios más positivos son pero con una diferencia no lo suficientemente grande como para considerarlo una regla.

Tras analizar todas estas cuestiones podemos concluir diciendo que la temática de este vídeo no es la más oportuna para realizar el análisis y posteriormente estimar el comportamiento de los usuarios para las empresas a la hora de sacar el máximo beneficio en base a las opiniones de los clientes. Probablemente es debido a que los datos se obtienen de manera muy dispar, dependiendo del gusto de cada usuario de manera individual y de la percepción que haya tenido en el viaje; un usuario puede ser que comente exactamente al acabar su experiencia y sea coherente en cuanto a el tiempo meteorológico y no basando su opinión en ese tipo de cuestiones mientras que otro aporte su comentario varios meses después cuando quiera recomendarselo a otra persona. Es por cuestiones de este estilo por lo que los datos siguen un patrón lógico pero a muy bajo nivel, no podemos estimar de una manera cien por cien empírica la totalidad del comportamiento como si se puede hacer en vídeos con otras temáticas y que generan una afluencia mucho más masiva de tráfico en menos tiempo como es el caso de videos sobre videojuegos o de creadores de contenido influyentes.

Lo ideal sería realizar un análisis y comparar un canal con millones de visitas diarias para ver las diferencias que se pueden observar con un canal relacionado con el turismo en Tenerife.

6.1 Líneas futuras

Una vez realizado el proyecto se pueden aplicar distintas mejoras lógicas al proyecto una vez se haya realizado una revisión a fondo del proyecto.

POdría resultar útil y productivo para las empresas realizar un análisis y una comparación de distintos videos de la misma temática, con la finalidad de obtener conclusiones más claras y verídicas a la hora de ejecutar, por ejemplo, distintas estrategias de marketing. Para ello se podría desarrollar la herramienta necesaria para introducir manualmente distintos id de distintos vídeos y poder así mostrar los resultados.

Junto a esto se podría adaptar el idioma para que se pueda realizar el análisis no solo en inglés, sino en español y otros idiomas. Esto permitiría llevar el estudio a otro nivel de amplitud bastante superior.

Por último se podría hacer un sistema de ventanas para ir pidiendo, almacenando y eligiendo los distintos datos así como una sistema web o aplicativo para mostrar los resultados y las conclusiones de los distintos análisis.

7 Summary and conclusions

7.1 Conclusions

- We have made requests to bring different data through the Youtube API to prepare and analyze a large amount of data on the traffic of opinions that have been made by different YouTube users about a video.

- An analysis of feelings has been carried out, once the previously mentioned data has been cleaned and prepared, on which different conclusions have been drawn.

Based on the analysis we can say that the theme of the video, it does not follow the general trend on current issues and daily updates.In the case of videos about video

games, technology or content created to entertain, we can deduce several maxims that are usually fulfilled quite accurately in all the videos that are published.

- The largest peak of views occurs on the day of publication of the video or the days immediately following.

- The greatest number of comments, likes and dislikes occurs exactly in the same period of time previously commented.

- The video will have a greater impact if the topic is actual.

In this case the topic, does not fall within any current theme or something that is extremely novel. The water park is not new due to its opening several years ago.

That is why we can observe a clear trend of comments or an excessively large peak of comments on the date of publication of the video. However, the video maintains a constant flow of comments over time. At the time of making this report it can be seen that a few hours ago several comments were counted as well as several days and months ago. These data are extravagant and they go a little out of the general rules since the video was published in August 2015, almost four years ago.

The users or tourists in this case are contributing their opinion or their expectations about the water park once they have visited it or will visit it over time so we can draw as a firm conclusion that the video will continue to have traffic while the Water park continues to operate and with a massive influx of tourists as has been the case since it opened its doors.

Going into the question of which day is most actively discussed, we can not give a hundred percent response either. In order to answer this question it is necessary to take into account several points that are different depending on the user or tourist. We could think that the days of the week in which most comments are the weekend and that the season of the year with the most traffic is in holiday periods as well as in summer periods, where the weather in the Canary Islands is warmer.

Taking into account these issues, in a large part the trend of comments follows that line but we can not say that the majority is the same because in this case the millions of tourists who visit the park and then contribute their comment do not have the same holiday period nor do they comment on the same day that they have spent their days in the park, that is, it can be the case of a user who comments his

experience several days, weeks or even months after having made his trip to the Canary Islands.

On the other hand, following the previous reflection and focusing on finding out which is the period where it is most positively commented, we see a trend similar to that described above but not exact. On weekends is where the comments are more numerous, but Monday has a very similar figure, if we look at the feelings does not match with this data. The day that is most positively commented is on Wednesday, one of the days in which fewer comments are produced so we could deduce that the fewer number of more positive comments are but with a difference not large enough to consider it a rule.

After analyzing all these issues we can conclude by saying that the theme of this video is not the most appropriate to perform the analysis and then estimate the behavior of users for companies when making the maximum benefit based on the opinions of customers . Probably it is because the data is obtained in a very different way, depending on the taste of each user individually and the perception that he / she has had during the trip; a user may comment exactly at the end of his or her experience and be consistent about the weather and not basing his or her opinion on that type of question while another person contributes his comment several months later when he or she wants to recommend it to someone else. It is for reasons of this style that the data follow a logical pattern but at a very low level, we can not estimate in an entirely empirical way the whole behavior as if it can be done in videos with other themes and that generate an influx much more massive traffic in less time as is the case of videos about video games or influential creators of content.

The ideal would be to perform an analysis and compare a channel with millions of daily visits to see the differences that can be observed with a channel related to tourism in Tenerife.

7.2 Future lines

Once the project is completed, different logical improvements can be applied to the project once a thorough revision of the project has been carried out.

It could be useful and productive for companies to perform an analysis and comparison of different videos of the same subject, with the aim of obtaining more clear and true conclusions when executing, for example, different marketing strategies. For this, the necessary tool could be developed to manually enter different ids of different videos and thus be able to show the results.

Along with this, the language could be adapted so that the analysis can be done not only in English, but also in Spanish and other languages. This would allow to take the study to a much higher level of amplitude.

Finally, a window system could be made to ask for, store and choose the different data as well as a web or application system to show the results and conclusions of the different analyzes.

8. Presupuestos

Tareas del proyecto			
Tareas	Tiempo (horas)	Costo(euros)	Total(euros)
Obtención de datos de canales de Youtube	28	25	700
Extracción y preparación de datos	50	25	1250
Detección y análisis de sentimientos	40	25	1000
Informe de conclusiones	30	25	750
Cuenta final			3700

Tabla 2: presupuestos

9 Bibliografía y referencias

[1] Canal elegido [Online]. Aviable:
https://www.Youtube.com/channel/UC1vKZUIDvfvnQcpxUfp5tSw
[2] Video elegido [Online]. Aviable:
https://www.Youtube.com/watch?v=vvR9fTZl6fU
[3] Editor de código atom [Online]. Aviable:
https://atom.io/
[4] Python [Online]. Aviable:
https://www.python.org/
https://stackoverflow.com/questions/393843/python-and-regular-expression-with-unicode
https://www.tutorialspoint.com/python/string_lower.html
[5] Librería requests [Online]. Aviable:
https://2.python-requests.org//es/latest/
[6] Librería Json [Online]. Aviable:
https://www.analyticslane.com/2018/07/16/archivos-Json-con-python/
[7] Librería pandas [Online]. Aviable:
https://pandas.pydata.org/
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.plot.html
[8] Librería nltk [Online]. Aviable:
https://www.nltk.org/
[9] Librería matplotlib [Online]. Aviable:
https://matplotlib.org/
[10] Librería string [Online]. Aviable:
https://docs.python.org/2/library/string.html
[11] API Youtube [Online]. Aviable:
https://console.developers.google.com/
https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces
https://developers.google.com/Youtube/v3/
[12] Pip [Online]. Aviable:
https://pypi.org/project/pip/
[13] Algoritmo Vader [Online]. Aviable:
https://pypi.python.org/pypi/vaderSentiment
https://medium.com/@aneesha/quick-social-media-sentiment-analysis-withvader-da44951e4116
https://github.com/cjhutto/vaderSentiment
[14]API Youtube commentThreads [Online]. Aviable:
https://www.googleapis.com/Youtube/v3/commentThreads
[15] Dataframe [Online]. Aviable:
https://github.com/mobileink/data.frame/wiki/What-is-a-Data-Frame%3F

[16] Siddhartha Chatterjee; Michael Krystyanczuk, Python Social Media Analytics, Birmingham: Packt Publishing.

[17] europapress.com (10-02-10). [Online]. Aviable: https://www.europapress.es/portaltic/sector/noticia-no-tener-cuenta-influencia-redes-sociales-expone-empresas-deterioro-imagen-20100210144127.html