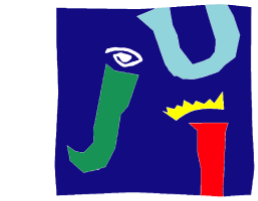


UNIVERSITAT JAUME·I

TRABAJO DE FIN DE MÁSTER

MÁSTER UNIVERSITARIO
EN SISTEMAS INTELIGENTES
CURSO 2017-2018

Detección automática de tweets no relevantes en streams guiados por consulta

---

Autor:

Víctor García Pérez

Tutor:

Rafael Berlanga Llavori

Lectura del Trabajo
Castellón, octubre 2018

Resumen

Desde principio de los años 90 cuando surgieron las redes sociales, el número de usuarios y la cantidad de información compartida y publicada en ellas ha experimentado un crecimiento exponencial.

En este trabajo nos centraremos en la red social Twitter, que contaba a principios de 2018 con 330 millones de usuarios. El objetivo de este trabajo es conseguir predecir cuáles de todos los tweets recogidos a través de una consulta de dominio son relevantes o irrelevantes para una fase de análisis posterior. Para ello, en primer lugar, se ha realizado un barrido bibliográfico para consultar el estado del arte en temas similares.

En segundo lugar, se ha elaborado un método semi-manual para realizar el etiquetado del dataset donde se han identificado los tweets en función de la clase a la que pertenecen, relevantes o irrelevantes. Después se ha realizado un análisis estadístico de los datos para buscar un método de clasificación adecuado según las métricas de evaluación seleccionadas. Todos los experimentos han sido realizados con la ayuda de las librerías de minería de datos y tratamiento de texto disponibles para Python.

Palabras clave

Análisis de Redes Sociales, Clasificación automática, Minería de textos.

Abstract

Early in the 90s when social networks emerged, the number of users and the amount of information shared and published in them has undergone an exponential growth. In this work we will focus on the social network Twitter, which had at the beginning of 2018 with 330 million users.

The goal of this work is to predict which of all the tweets obtained through a domain query are relevant or irrelevant for a subsequent analysis phase. For this, first, a bibliographic search has been made to find out the state of the art on similar topics.

Secondly, a semi-manual method has been developed to perform the tagging of the dataset where the tweets have been identified according to the type they belong to, namely: relevant or irrelevant. Then a statistical analysis of the data has been carried out to find an adequate automatic classification method according to the selected evaluation metrics. All the experiments have been carried out with the help of data mining and text processing libraries available for Python.

Agradecimientos

Quería dedicar este pequeño apartado para agradecer a todas aquellas personas que han estado cerca durante la realización de este trabajo y que, de diferentes formas, han contribuido a que la realización del máster y de este trabajo final llegue a su fin.

En primer lugar, me gustaría agradecer a mi tutor, Rafael Berlanga Llavori, la oportunidad de realizar este trabajo, y el tiempo que me ha dedicado para desarrollar el mismo y mi formación en este campo.

A mi hija Arlet, que, pese a su corta edad, me ha dado fuerzas para ofrecer lo mejor de mí para que este trabajo saliera adelante.

A María José, por su apoyo y amor incondicional en todo momento.

A mis padres Floren y Fina que siempre están ahí presentes para darme el cariño y la fuerza que necesito en todo momento.

Y no puedo olvidarme de mi hermana Sheila que, pese a la distancia, siempre cerca está apoyándome en todos mis proyectos.

A mis compañeros de máster, en especial Miguel, Carlos y Héctor, junto a los cuales formamos el equipo Robin Hood en la competición Datathom UniversityHack 2018, viviendo una agradable experiencia de trabajo en equipo.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

En los últimos tiempos se ha incrementado el análisis de los mensajes transmitidos a través de la red social Twitter, la finalidad es obtener información de los usuarios referente a eventos políticos o para conocer opiniones sobre alguna temática. El surgimiento del paradigma de Inteligencia de Negocio Social (Social Business Intelligence) [1] ofrece nuevas perspectivas de análisis y acción en las redes sociales basadas en métricas e indicadores obtenidos de los mensajes y usuarios de las redes sociales. Dentro de este nuevo paradigma, la reducción del ruido e identificación de usuarios y mensajes relevantes son cruciales para realizar un análisis de calidad.

El objetivo principal de este Trabajo Final de Máster consistirá en, dado un stream de tweets obtenidos mediante una consulta de dominio (por ejemplo, marcas de un sector), detectar qué tweets son irrelevantes para esa labor de análisis posterior.

En este trabajo se considera que los tweets son relevantes para una futura fase de análisis si las palabras clave de la consulta se utilizan con el sentido correcto, y el contenido del tweet está en el contexto de la tarea de análisis según la temática de los tweets obtenidos. Ejemplos de tweets irrelevantes son los mensajes spam y memes, los cuales utilizan las palabras clave de la consulta con una intencionalidad completamente diferente a la esperada por los usuarios del dominio. Estos mensajes podrían ser muy numerosos y sesgar cualquier análisis que se quiera realizar del dominio.

A modo de ejemplo, mostramos los tweets más re-direccionados de un stream generado para el dominio de coches:

> **DanMentos** - "I think I have ADHD, doc" why? "I keep forgetting whereI parked my Ford" that's not- "yeah I keep losing my Focus" get out of my office (🔁 4766)
> **NotYikYaks** - "Left my Adderall in my Ford Fiesta. Now it's a Ford Focus." (🔁 4200)
> **CarAutoDaily** - Future Cars: 2016 Mitsubishi Pajero Sport SUV... - https://t.co/HoP7ianCax (🔁4167)
> **CuntsWatching** - Crying because he's got GCSE's to sit in 4 days, poor woman musthave thought she was giving birth to a Ford Mondeo http://t.co/hctk8PWdVl (🔁4164)
> **CarAutoDaily** - 2016 Toyota RAV4 Formal Pictures and Data... - https://t.co/Lg6j9RoATX (🔁4069)

Puede apreciarse que de los cinco primeros, tres son efectivamente memes y dos son campañas de promoción de coches publicados por un medio relevante del sector. Así pues, debemos eliminar de cualquier análisis mensajes como el primero, cuya finalidad y semántica son totalmente ajenas al dominio de análisis.

Como hipótesis de partida, en este trabajo consideramos que tanto los contenidos textuales como las métricas sociales asociadas a los mensajes pueden ser de gran utilidad para clasificarlos como relevantes o irrelevantes. Un requerimiento importante a la hora de seleccionar los métodos de clasificación es que sean lo más insensible posible al idioma de los tweets. En este TFM se trabajarán con tweets tanto en español como en inglés.

1.1 Objetivos

Los objetivos de este Trabajo Final de Máster se han centrado en detectar o predecir cuáles son los tweets irrelevantes que no deben tenerse en cuenta en la fase de análisis de información social. Estos casos englobarán tweets considerados spam, memes que utilizan las palabras clave de la consulta, y expresiones hechas que utilizan las palabras clave de las consultas en un sentido inesperado.

Para conseguir este objetivo principal se ha realizado, en primer lugar, una revisión bibliográfica para conocer los antecedentes científico-técnicos relacionados con la problemática tratada en el TFM, y así poder elegir las técnicas más apropiadas para realizar predicciones y evaluar su calidad.

Otro sub-objetivo ha sido el análisis del stream de tweets obtenidos mediante una consulta de dominio a través de la Filter Realtime API de Twitter. Con el análisis realizado se ha desarrollado un método para realizar el etiquetado de tweets como relevantes o irrelevantes.

Una vez realizado el análisis del stream de tweets se han seleccionado las características del conjunto de datos que sirven para identificar qué tweets son irrelevantes. Con esto se ha podido definir la mejor manera de construir el dataset para la fase de detección de tweets considerados irrelevantes.

La selección de características que debían formar el conjunto de datos de entrenamiento ha servido como fase inicial para diseñar los experimentos a realizar, y para entrenar clasificadores que ayuden a la selección del modelo de clasificación.

El principal objetivo consiste en encontrar un clasificador para resolver el problema planteado. Para la selección del modelo de clasificación, se han preparado unos experimentos que han sido ejecutados con el set de algoritmos seleccionados. Estos experimentos han sido evaluados con una serie de métricas que han permitido valorar los clasificadores de manera objetiva.

Para terminar el trabajo se han obtenido unas conclusiones e ideas de mejora y trabajo futuro.

1.2 Contextualización del trabajo

1.2.1 Redes sociales

A principios de los años 90 surgieron las primeras redes sociales en internet. Las finalidades de ellas fueron diversas. En 1995 nació Classmates como un lugar donde encontrar antiguos compañeros de clase o trabajo. Un año antes, en 1994 se fundó GeoCities, una red que lo que pretendía era que los usuarios creasen sus páginas web y las alojarán por temáticas en un barrio, por ejemplo: Entretenimiento en Hollywood, ordenadores en Silicon Valley, etc.

Unos años después, en el 2003 apareció MySpace, en 2004 se lanza Facebook, concebida originalmente como una plataforma para conectar universitarios, y en 2006 se inauguró la red de mirco-blogging denominada Twitter. El ecosistema de redes sociales no ha parado de crecer en todo el mundo. En la figura 1 se pueden ver el número de usuarios de redes sociales en 2018 según el informe de Hootsuit. [2]

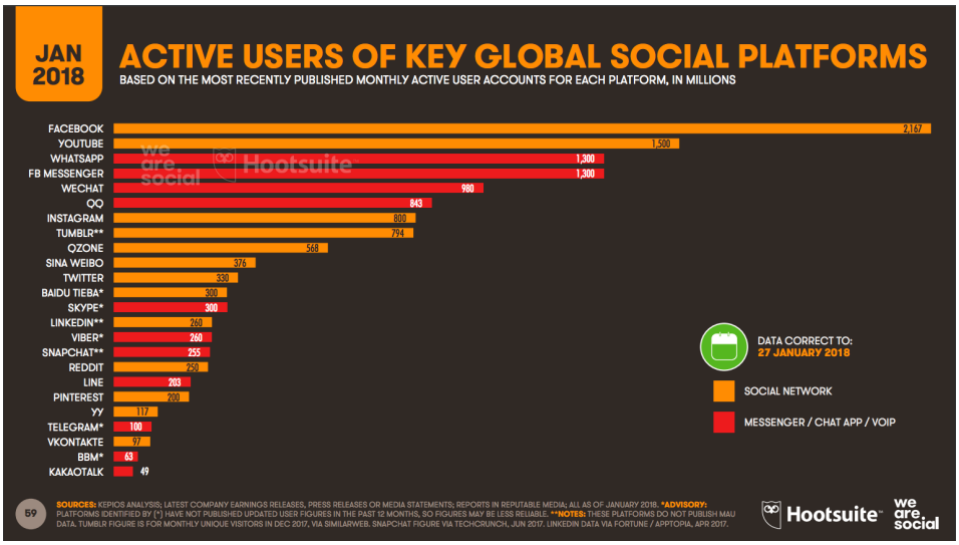


Figura 1: Usuarios de redes sociales en el mundo

Con este escenario de multitud de redes sociales y millones de usuarios interaccionando entre ellos y opinando sobre temas de actualidad, se puede intuir que actualmente constituyen una fuente de información importantísima para recabar información y opiniones de los usuarios.

Cada acción que realiza un usuario en las redes sociales es considerada como una forma de opinar sobre algo, ya sea desde la pulsación de "Me gusta" a una foto o comentario, un re-tweet o un comentario o mensaje sobre ese hilo o tópico.

Las redes sociales tienen cada una su característica o finalidad para la que se han diseñado, y las formas de relacionarse son diferentes, por ejemplo, en Facebook los usuarios deben hacerse amigos, aceptando una solicitud de amistad, mientras que Twitter cada usuario puede seguir o ser seguido sin implicar una reciprocidad.

1.3 Twitter

El funcionamiento de Twitter básicamente consiste en que un usuario puede publicar mensajes cortos para el público en general. En sus inicios Twitter permitía a los usuarios escribir mensajes de hasta 140 caracteres y en la actualidad de 280 caracteres. Un usuario puede seguir a tantos usuarios como desee y puede ser seguido por los usuarios que consideren interesante a ese usuario. Por ello, los usuarios que más contenido o influencia tienen, adquieren mayor número de seguidores, convirtiendo a Twitter en uno de los medios de comunicación más importantes de la actualidad.

En la figura 1 se puede ver que Twitter cuenta con 330 millones de usuarios en todo el mundo, por lo que puede ser considerado una muestra importante de la población a la hora de realizar cualquier análisis.

Una de las primeras tareas que se deben realizar a la hora de realizar cualquier análisis de información de Twitter, es filtrar los tweets que no son relevantes en la tarea de análisis que se va a realizar. Un primer paso para este filtrado es utilizar la API de Twitter indicando palabras clave que deben contener los tweets que van a ser analizados. A estas palabras clave las denominamos consultas de dominio.

El objetivo de este trabajo es realizar un filtrado adicional de los tweets devueltos por la API que consideramos claramente no relevantes tareas de análisis. Para ello a continuación [3] se explican una serie de conceptos que serán de utilidad para la comprensión de capítulos posteriores.

Los trending topics (hastags #)
Trending topic (TT): Es una de las palabras o frases más repetidas en un momento concreto en Twitter. Los diez más relevantes se muestran en la página de inicio, pudiendo el usuario escoger el ámbito geográfico que prefiera, mundial o localizado, o personalizadas, en función además de a quién sigue el propio usuario.

Metadatos de Twitter: Métricas
Las métricas en Twitter nos dan una visión de la relevancia de una cuenta. La métrica más común es fijarnos en el número de seguidores de una cuenta. En cierta forma, los seguidores son importantes, pero no son ni lo único ni lo más importante para medir el impacto de una cuenta en Twitter.

Otra métrica es la **ratio seguidores/seguidos**. Cuanto más alta sea esta ratio, mayor será la relevancia real de la cuenta. Otra forma de medir la relevancia es la **ratio de seguidores por tweet**. Un usuario que lleve un año en Twitter y tenga mil tweets,

tendrá una cuenta más relevante que otro que haya conseguido el mismo número de seguidores en medio año o con la mitad de tweets.

Los **retweets/nº de tweets** es otra de las métricas para medir la relevancia de una cuenta. Cuantos más retweets tenga un usuario, más interesante o relevante es lo que dice.

Listas

Las listas de Twitter son grupos de cuentas de Twitter con las que se crea un directorio de contactos sobre una temática específica. El objetivo de las listas es poder agrupar unos contactos particulares en Twitter, para luego, poder leer sus tweets de manera concentrada. En general, tienen las siguientes características:

- Se pueden crear de usuarios ya seguidos, o de usuarios no seguidos.
- Pueden ser públicas, o privadas. Las listas públicas se reflejan en el perfil del usuario que las crea. Twitter envía una notificación a los usuarios que son agregados a la lista.
- Las listas tienen un nombre o título, y una descripción.
- Otros usuarios pueden "seguir" tu lista de Twitter.
- Hay límites para las listas de Twitter: 1,000 listas por usuario que las crea, y hasta 5,000 usuarios por listas.

Ventajas de las listas de Twitter

- Ayudan a no perderse los tweets de usuarios específicos, dentro de todo el ruido de Twitter.
- **Mejoran la experiencia de lectura de tweets,** cuando se usan columnas, liberando de tener que leer necesariamente el timeline completo.
- Ayudan a tener los contactos organizados y bien identificados.

Momentos

Se trata de una función de Twitter pensada para consultar y organizar tweets por temas.

De toda la información que se genera en Twitter, la que se ha utilizado en este trabajo son los textos de los tweets y las métricas de usuario. Se deja para el trabajo futuro incorporar métricas referentes a los otros elementos asociados a un tweet.

1.3.1 API Twitter

En la actualidad la red social Twitter es una de las mayores fuentes de información en tiempo real de lo que está sucediendo en el mundo, donde millones de usuarios publican información de que está pasando.

Para poder explotar esta información, Twitter ha desarrollado una serie de APIs que ayudan a desarrolladores de todo el mundo a poder explotar la información disponible en la red social, de una manera ágil y sin el uso de la interfaz de usuario que ven todos los usuarios.

En la fecha de edición de este trabajo, Twitter cuenta con seis APIs diferentes, cada una para un uso concreto. Estas son: Search Tweets, Filter realtime Tweets, Account Activity API, Direct Message API, Twitter for websites y Ads API. [4]

En este trabajo nos centraremos en la Filter realtime Tweets, más conocida como API Streaming. Twitter presenta dos versiones posibles de está API, donde la diferencia principal radica en el precio (Una gratuita y otra de pago) y la cantidad de información que entrega al desarrollador.

Las principales diferencias entre ambas versiones están descritas en la tabla 1. La extracción de tweets para este trabajo ha sido realizada mediante la versión Standard de la Filter realtime Tweets. Aunque en este trabajo se ha utilizado la API para obtener tweets, finalmente se ha utilizado un dataset proporcionado para agilizar la generación de un conjunto grande de datos.

Categoría	Número de filtros	Reglas de uso
Standard	400 palabras clave, 5,000 user-ids y 25 localizaciones	Solo se permite un solo filtro en cada conexión. Para cambiar el filtro es necesario desconectarse
Enterprise	Hasta 250.000 filtros por stream, hasta 2.048 caracteres	Pueden utilizarse miles de reglas en una misma conexión. No es necesario desconectarse para añadir o modificar filtros

Tabla 1: Versiones API Streaming Twitter

Para poder conectarse al stream de tweets hay que seguir los siguientes pasos:

1. Crearse una cuenta en Twitter.

2. Después crear una solicitud de desarrollador en Twitter [5].

3. Implementar un mecanismo de autentificación denominado Aouth (Open Authorization) que es un estándar abierto que permite flujos simples de autorización para sitios web. El proceso de autentificación se muestra en la figura 2.

4. Realizar la conexión al stream con el filtro de palabras clave que se desean filtrar.

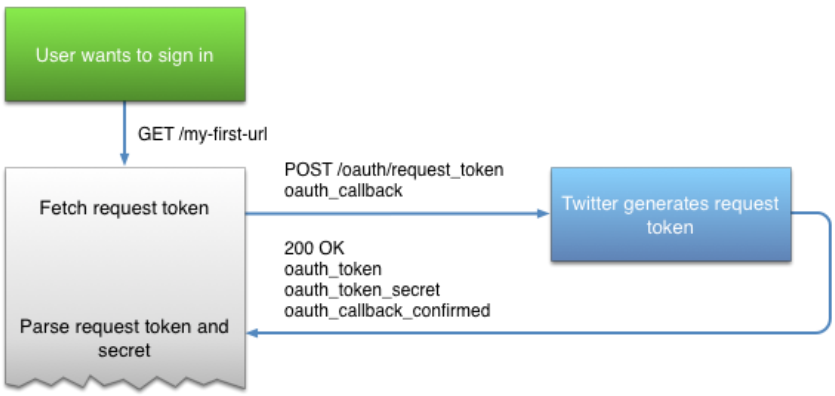


Figura 2: Sistema de autentificación OAuth de Twitter[6]

Capítulo 2

Estado del Arte

Como se explicaba en el apartado anterior, Twitter es una red social muy amplia en usuarios, la cual se ha convertido en un medio de comunicación de eventos en tiempo real. Gracias al concepto de "Hastag", término que empieza con el carácter "#", se permite a los usuarios realizar un seguimiento sobre lo que otros están escribiendo en relación a una temática.

La estructura de Twitter [7] permite que los sistemas de información puedan realizar seguimiento de noticias para descubrir qué está sucediendo en el mundo. Por ello Twitter se puede considerar un sistema de opinión o encuestas a nivel mundial. Para poder realizar este sondeo de información, lo primero que se necesita realizar es un filtrado de todos aquellos tweets considerados inicialmente como no pertenecientes al dominio de análisis, principalmente:

i. Spams que utilizando los términos del dominio tienen como objetivo llevar al usuario a páginas no deseadas mediante grandes descuentos o regalos. Suelen utilizar marcas muy conocidas como gancho para que el usuario acceda a los enlaces mostrados.
ii. Memes que consisten en comentarios jocosos o irónicos donde se usan las palabras del dominio con un sentido totalmente distinto al esperado.
iii. Malware que al igual que el spam puede servir de gancho, pero esta vez para hacer una suplantación de identidad del usuario (Phishing), o recolectar información del usuario (Histórico de llamadas, SMS, localización del usuario, acceder a la información del Smartphone del que se ha conectado a la red, hacer llamadas con el teléfono del usuario, etc.) [8,9]

	Irrelevancia/Ruido						
	Spam			Memes		Malware	
Características	Cuenta de usuario	Información del tweet	Grafo de relación	Viral o con mucha popularidad	Aparición en una ventana temporal	Phishing	Virus

Tabla 2: Clasificación de factores de Irrelevancia

En la tabla 2 se hace un resumen de las características de los tipos de irrelevancia o ruido a la hora de hacer un análisis de tweets sobre una temática concreta.

2.1 Detección de Spam

Los denominados spammers invaden los trending topics de Twitter mediante la emisión de tweets denominados spam [10]. La detección de spam es importante, dado que Twitter es usado por muchas empresas o políticos para medir la satisfacción de sus productos o el número de usuarios con potencial voto hacia dichos políticos [11].

Las características para la detección de spam de Twitter se clasifican de la siguiente manera:

1. **Características basadas en la cuenta:** Los Spammers pueden detectarse basándose en su cuenta de usuario, que contiene algunas características que no son controladas por el usuario, como las que se pueden ver en la tabla 3, y otras que sí son controladas por el usuario y no servirán para utilizarse en la detección, tales como su Nick, foto, fecha de nacimiento, localización, fecha de creación de la cuenta, etc.

Característica	Descripción	¿Controlada por el usuario?
Número de tweets	Número total de tweets tiene la cuenta	NO
Número de “followings”	Número total de cuentas que el usuario sigue	NO
Número de “followers”	Número total de seguidores que el usuario tiene	NO
Número de “likes”	Número total de “likes” que tiene la cuenta	NO
Número de retweets	Número total de retweets que tiene la cuenta	NO
Número de listas	Número total de listas que tiene la cuenta	NO
Número de momentos	Número total de momentos que tiene la cuenta	NO

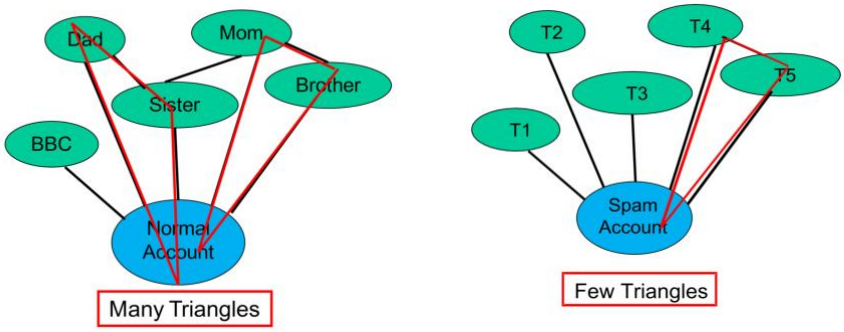
Tabla 3: Características basadas en la cuenta de usuario

2. **Características basadas en tweets:** Los metadatos del tweet también proporcionan una serie de características, de las cuales las que no son controladas por el usuario también proporcionan diferencias para poder clasificar un tweet como spam o no. En la tabla 4 se muestran las características extraídas de los metadatos clasificadas por la posibilidad de manipularse por el usuario o no.

Característica	Descripción	¿Controlada por el usuario?
Emisor	Emisor del tweet	SI
Menciones	Las menciones usadas en el tweet	SI
Hastags	Los hastags usados en el tweet	SI
Link	Los links a otras páginas usados	SI
Número de "likes"	Número total de likes que ha conseguido el tweet	NO
Número de retweets	Número total de retweets que ha conseguido el tweet	NO
Número de respuestas	Número total de respuestas que ha conseguido el tweet	NO
Fecha de emisión	Fecha en la que el tweet ha sido publicado	SI
Localización	Localización detectada de donde se ha publicado del tweet	SI

Tabla 4: Características basadas en el tweet

3. **Relación entre el emisor y receptor del tweet:** Los usuarios generadores de spam [12] tienen un perfil de seguir a muchas cuentas de Twitter para conseguir seguidores que entre ellos suelen tener muy poca relación, mientras que las relaciones del resto de cuentas suelen tener relaciones por intereses, por afinidad de amistad o familia, etc. [13]. De este modo perfiles de cuentas donde existan muchas relaciones entre usuarios se podrán caracterizar como cuentas no generadoras de spam, mientras que las cuentas spammers tendrán un perfil mayoritario en relaciones disgregadas. Esto se puede ver en la figura 3.



Cuenta normal Cuenta spammer

Figura 3: Grafo de relaciones en los perfiles de cuentas [13]

2.2 Detección de Memes

Meme es un término creado por Richard Dawkins en 1976 en [14] y aparece definido como la unidad cultural más simple (una idea, comportamiento, moda o uso) que puede propagarse de persona a persona dentro de una cultura sin importar ubicación geográfica, con lazos de comunicación entre ellos y ubicados en un mismo espacio temporal. Sin embargo, un meme de internet se usa para describir una idea, concepto o pensamiento manifestado en un medio virtual, en cualquier formato, sean los más típicos en formato texto, imagen, video, audio…

Los memes en internet [15] principalmente se propagan mediante links a sitios web, o en cualquiera de los formatos citados a través de las redes sociales. [16]

Los memes tienen un factor temporal muy importante en la evolución de su popularidad. La popularidad no depende únicamente del propio meme, si no que depende también de otros memes que sean virales en ese momento. En [17] Coscia et al. analizan que los memes pueden competir o colaborar entre ellos. Si compiten entre ellos significa que hay una influencia negativa de un meme sobre el otro. En caso de colaboración implica que la influencia entre ellos es positiva y hacen el uno al otro que aumente su difusión.

Coscia et al. se centran en las cuatro principales características que denominan que tienen los memes:

- Número de colaboradores: Número de memes en colaboración
- Número de competidores: Número de memes en competición
- Hecho de que un meme esté correlacionado con otros
- La entidad de su pico de popularidad sobre su popularidad promedia

Un contenido será viral cuando se distribuye de forma masiva en las redes sociales y blogs y suelen ser con carácter humorístico y claman a las emociones. Por lo general, salvo para ciertas marcas, los contenidos que se hacen virales no nacen pensados para serlo. Sin embargo, las campañas de marketing de las marcas nacen para hacerse virales, ya que implicaría ser un éxito de marketing, aunque no es fácil que lo consigan.

La viralidad de un mensaje es la probabilidad de que se envíe ese mensaje. En Twitter, viralidad es la probabilidad de retweet de un mensaje. En Twitter por lo general, la probabilidad de retweet no depende de relaciones sociales existentes entre usuarios, esto es un caso parecido al de los medios de comunicación tradicionales. En [18] plantean la hipótesis de que es más probable que se reenvíe contenido de noticias negativas mientras que, en el resto de tweets los sentimientos positivos aumentan la probabilidad de retweet o viralidad.

En [19] proponen un método de clustering para la detección de memes que llaman "*proto-memes"* basado en múltiples medidas de similitud para obtener memes como grupos con una cohesión de tweets que reflejan conceptos o temas de discusión. Entre las características del clustering hay métricas como las descritas en el apartado anterior y procesamiento del lenguaje natural. Defienden este método justificando la viralidad de los memes ya que no es práctico tener un sistema de aprendizaje supervisado para

detectar tweets con los datasets creciendo de manera infinita día a día. Por ello proponen el modelo de stream clustering.

El modelo de stream clustering es un modelo muy grande que irá creciendo en datos que serán muy costoso de almacenar en memoria. En la literatura existen diferentes modelos:

- **Landmark window:** Contiene todos los datos del stream desde el instante t=0 hasta el instante actual t=T Actual.

- **Damped window:** Asigna pesos a cada punto en el stream de datos. Proporciona pesos utilizando una función de decaimiento temporal (decay function), dando mayor peso a los datos más recientes.

  $$w(t) = 2^{-\lambda(T-t)}, \text{ donde } \lambda > 0$$

  Un mayor valor de λ implica mayor importancia/peso de dato más reciente.

- **Sliding window:** Tiene una longitud de *l* pasos en cada momento T, y contiene solamente los puntos recibidos los últimos *l* pasos. La ventana intervalo se define como sigue:

  $$W = (T - l \cdot \Delta t, T]$$

  Los algoritmos de análisis propuestos desestiman todos los puntos temporales más antiguos que $T - l\Delta t$. Este modelo es el más simple, y el que han seleccionado en el artículo [19].

En este trabajo también se utiliza una *sliding window* de longitud de un mes para la agregación de métricas de usuario y de tweets. Sin embargo, en lugar de utilizar el método de clustering propuesto en este trabajo directamente utilizamos un método supervisado sobre los tweets entrantes, teniendo en cuenta sus contenidos textuales y métricas.

2.3 Procesamiento del Lenguaje Natural

Para poder procesar el texto publicado por los usuarios y enriquecer la información obtenida de las métricas, hay que utilizar técnicas de Procesamiento de lenguaje natural o NLP por sus siglas en inglés. Una técnica para realizarlo es mediante lo que se conoce como el paradigma de "Word embedding", que es el que se ha adoptado en este trabajo. La principal razón para ello es su independencia del idioma de los textos y la omisión de cualquier tipo de ingeniería de características en la representación de los documentos.

Word embedding es el nombre de un conjunto de lenguajes de modelado, donde las palabras o frases se convierten a vectores de números reales, consiguiendo así tener una representación matemática en un espacio de N dimensiones de las palabras o diccionarios.

En la figura 4 se ha realizado una representación gráfica de los métodos de clasificación automática de textos supervisado. En esta figura se muestran los métodos clásicos separados de los métodos modernos que aparecieron a partir de Mikolov et al [21,23].

Los métodos considerados clásicos están basados en la representación de Bolsa de Palabras o Bag of Words (BoW). Este método de procesado de lenguaje trata de representar documentos ignorando el orden en el que aparecen las palabras en el texto, considerando solo sus estadísticas en el documento y en la colección.

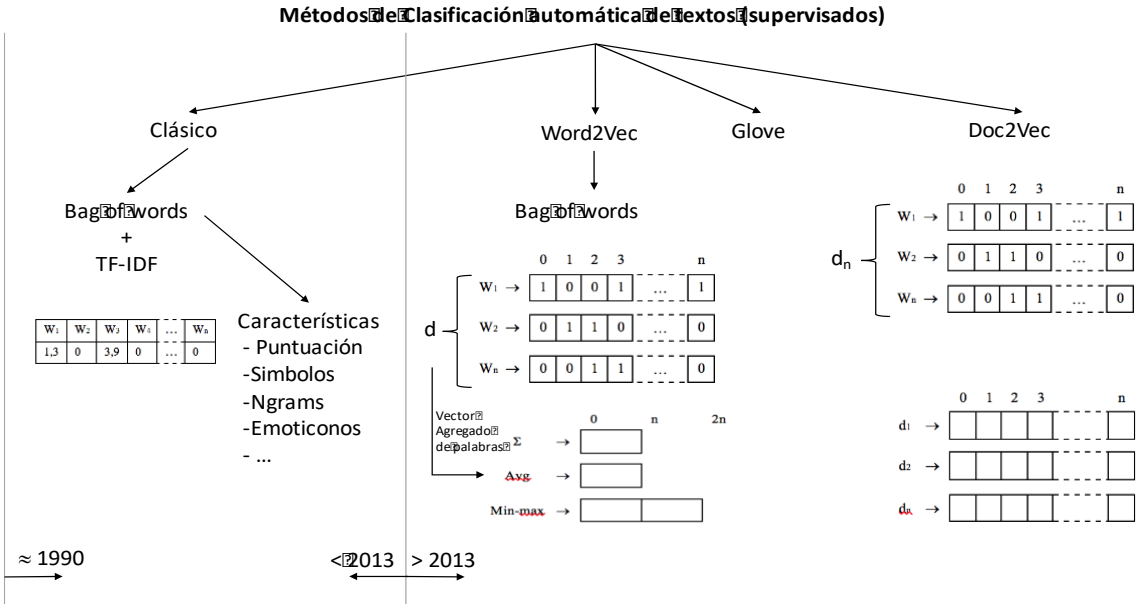


Figura 4: Métodos de clasificación automática de textos

La representación BoW de documentos consiste en asignar un vector a cada documento de dimensión igual al número de palabras que conforman el vocabulario de la colección. El primer paso que realiza el método es construir el diccionario representando cada palabra con una posición del vector. Por cada documento o frase, se ponen a "1" las posiciones que corresponden con las palabras del vocabulario. Después se calculan los valores de cada posición del vector mediante el producto de las dos medias TF e IDF, que se definen como sigue:

- **TF:** Es la cantidad de veces que aparece cada palabra en el documento corresponde con el termino en inglés "Term Frequency".

- **IDF:** Es la puntuación correspondiente a como de rara es la palabra en todos los documentos, y se representa con el logaritmo de la frecuencia inversa de la palabra en toda la colección.

El método BoW ha sido ampliamente utilizado para el análisis de sentimientos de los textos de los tweets mediante la extracción de características mostradas en la figura 4, tales como signos de puntuación, símbolos, emoticonos, etc.

En 2013, Mikolov et al. introducen la literatura un método novedoso [20]. Este método convierte los documentos a un espacio vectorial de N-dimensiones sin depender del

número de palabras contenidas en el diccionario. El método es mostrado mediante un ejemplo ilustrativo donde las palabras se ubican en un espacio métrico que permite calcular analogías mediante operaciones sobre sus vectores.

King - Man + Woman = Queen

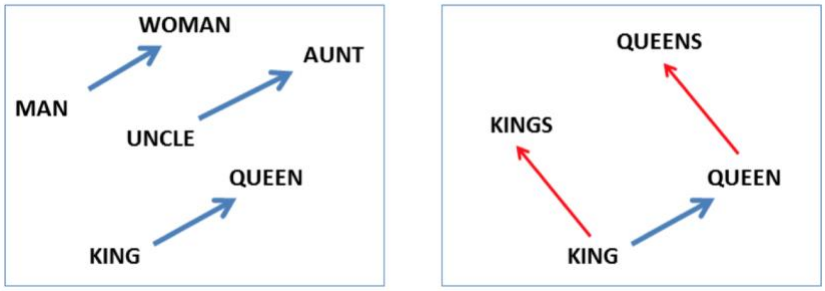


Figura 5: Ilustración método vectores de Mikolov et al [23]

Mikolov et al. basan la similitud de vectores mediante el método basado en la distancia del coseno (al igual que en los métodos clásicos), aunque son más populares otras medidas basadas en distancias, como las propuestas en [25] y Word Mover's Distance. [26]

En el uso de representación de palabras mediante vectores, GloVe [21], por sus siglas en inglés, identifican dos modelos de familias para aprender vectores densos (embeddings):

1. Métodos globales de factorización matricial como LSA, "Latent Semantic Analysis"

2. Métodos locales de ventana de contexto como Skip-gram model de [20].

La ventaja de los métodos LSA es que aprovechan eficientemente la información estadística, sin embargo, no son muy buenos en la analogía de palabras. En este sentido, el método Skip-gram es mejor en la analogía de palabras, pero utiliza de una manera pobre las estadísticas del contexto puesto que entrenan en ventanas de contexto local separadas en vez de en conteos globales de co-ocurrencias.

Mikolov et al presentan dos nuevas arquitecturas para procesar representación de palabras mediante vectores. Esta es una mejora del modelo Neural Net Language Model, NNLM que fue propuesto por Bengio et al. [22]. Los modelos son los siguientes:

- Continuous Bag-of- Words Model (CBOW): Modelo de Bolsa de palabras, en el cual no influye el orden de las palabras aprendidas. Este modelo utiliza palabras históricas de entrada y palabras futuras, donde el criterio de entrenamiento es clasificar correctamente la palabra actual

- Skip-gram Model: Esta arquitectura es similar a CBOW pero predice la palabra tratando de maximizar la clasificación basándose en otra palabra de la misma oración, en lugar de realizarlo basándose en el contexto como hace el CBOW.

Word2vec entrena una red neuronal simple, con una sola capa oculta para una tarea que después se utiliza para otra. Los pesos de la capa oculta son en realidad los "vectores de palabras" que estamos tratando de aprender.

La tarea falsa para la que se entrena la red neuronal es para que, dada una palabra dentro de una oración, mire las palabras cercanas y elija una al azar. La red dirá cuál es la probabilidad de cada una de las palabras del vocabulario y se elegirá la más cercana dentro de la ventana configurada.

El resultado esperado de la red es que dada una palabra de entrada nos de probabilidades mayores para palabras de contexto similar con probabilidades altas y probabilidades bajas para palabras que tengan otro contexto diferente. Por ejemplo, si la palabra de entrada a la red manzana, tendrá que dar mayores probabilidades para fruta y árbol que para coche o Rusia.

La red aprenderá la cantidad de veces que aparece cada emparejamiento de palabras, por eso cuando finalice el entrenamiento generará una probabilidad mayor dada la palabra manzana con fruta o árbol que con coche o Rusia.

Una pregunta que debemos responder antes de seguir, es: ¿Como se procederá a construir la red? La red neuronal no se puede alimentar con palabras. Lo primero que hay que hacer es crear un diccionario donde cada palabra corresponderá con un índice del vocabulario. La entrada a la red será un vector de la misma longitud que el vocabulario y cuando se quiera introducir una palabra a la red se pondrá un "1" en el índice del vector de entrada correspondiente a la posición en el vocabulario, y "0" en todas las demás posiciones.

La salida de la red se corresponderá con un vector de la misma longitud que el vocabulario, una posición por cada palabra del vocabulario. La red rellenará cada posición del vector de salida con una distribución de probabilidad de que, dada la palabra de entrada, la salida corresponda con cada una de las palabras.

Como se ha dicho, el objetivo final de todo esto es entrenar la matriz de pesos de la capa oculta, y no que la red prediga las palabras que pueden ir a continuación de una palabra de entrada, por ello, si nos centramos en el entrenamiento de la red, dada una oración de entrada, y en función del tamaño de la ventana configurado, se realizará el entrenamiento de los pesos.

Doc2vec[24] es una extensión de Word2vec que se encarga de representar textos o documentos sin importar la longitud de ellos. Permite representación en espacio denso de los documentos sin necesidad de ingeniería de características como si requiere el modelo Word2vec que agrega vectores realizando agregados mediante suma, promedio o min-max. Este último genera un vector de salida de 2 x N-dimensiones. En ninguno de estos métodos, la dimensión del vector de cada documento no es dependiente de la cantidad de palabras que tenga el diccionario.

Las razones por las que se ha elegido el modelo Doc2vec para la realización de este trabajo son mejor forma de agregar los documentos por sus vectores, tiene representaciones compactas y no sensibles a vocabulario. También podrían entrenarse en función de otras etiquetas semánticas que permitirán ajustar mejor el espacio métrico a las necesidades de la tarea de clasificación.

2.4 Técnicas de aprendizaje automático para tratamiento de texto

En [27,28,29,35] se presenta una comparativa de algoritmos clásicos como Naïve Bayes, Maximum Entropy, SVM con clasificadores compuestos que combinan varios estimadores básicos. Los primeros son ampliamente utilizados para resolver problemas típicos de clasificación, aunque suelen sufrir sesgos hacia la clase mayoritaria cuando se trata de problemas desbalanceados.

En Fernández-Delgado et al [30] hacen una comparativa de clasificadores preguntándose: "Do we need hundreds of Classifiers to solve real world classification problems?". Hacen una comparativa de 179 clasificadores con 121 datasets, de los cuales 112 pertenecen al repositorio de datasets de machine learning UCI. Esta comparativa de análisis de datasets y clasificadores es considerada como una orientación a la hora de seleccionar los algoritmos, ya que no existe un método que siempre sea "el mejor" y dependerá del problema a resolver la calidad que obtendrá ese mismo modelo, así como los parámetros con los que se ajuste.

En [31] han realizado una comparación y evaluación de varios enfoques para resolver su trabajo donde han utilizado técnicas tradicionales de Machine Learning y Deep Learning con diferentes técnicas de emulación de palabras.

Como no existe un modelo con una configuración de parámetros, se deberá evaluar un conjunto de clasificadores con varios parámetros de ajuste para encontrar la mejor solución al problema que se pretende resolver en este trabajo.

Los modelos combinados o combinación de clasificadores son una solución conjunta donde la decisión de la clasificación no es tomada únicamente por un clasificador, si no que es tomada por el conjunto de los clasificadores débiles. Estos pueden agruparse en dos tipos principales:

- Bagging: Consiste en generar de manera aleatoria conjuntos de entrenamiento de uno original para entrenar diferentes clasificadores.
- Boosting: Cada clasificador que conforma el modelo es entrenado en aquellas instancias que otras reglas precedentes no han sabido discriminar.

En [32] se introduce el método de Boosting que en esos momentos se considera como el estado del arte en los últimos benchmarks. Se trata del método "Scalable end-to-end trees boosting system" o "eXtreme Gradient Boosting", abreviado como XGBoost. De aquí en adelante se utilizará el término XGBoost para denominar a este algoritmo de clasificación. Cabe mencionar que este algoritmo es ampliamente utilizado en muchas áreas del aprendizaje automático como detección de fraude y spam.

Entre los factores más importantes para seleccionar este método de clasificación está la escalabilidad en todos los escenarios, es decir que es capaz de aprender en grandes conjuntos de datos, y que utiliza modelos estadísticos capaces de capturar dependencias complejas de los datos. Además, el sistema es capaz de correr 10 veces más rápido que otras soluciones populares existentes.

Los clasificadores elegidos en este trabajo están entre las 6 primeras familias del estudio de Fernández-Delgado et al. Estos han sido Naïve Bayes, Random Forest, Máquinas de Soporte Vectorial [33] y como algoritmo de Boosting, el XGBoost que apareció en la bibliografía después de Fernández-Delgado et al. En dicho estudio, la familia de clasificadores Random Forest es la que mejores resultados han obtenido, seguidos de las máquinas de soporte vectorial, y en sexto lugar la familia de algoritmos Boosting.

Naïve Bayes es uno de los clasificadores más utilizados por ser simple y rápido. Se trata de una técnica de clasificación supervisada, como todas las que se han utilizado en este trabajo. El teorema de Bayes se utiliza para revisar probabilidades previamente calculadas cuando se posee nueva información. El clasificador requiere crear un modelo previamente antes de poderse utilizar para la tarea de clasificación. La fórmula del teorema de Naïve Bayes se representa de la siguiente manera:

$$P(C|x) = \frac{P(x|C) \cdot P(C)}{P(x)}$$

La creación del modelo de clasificación requiere de los siguientes pasos:

1. Calcular la probabilidad a priori de que dada una muestra pertenezca a una clase u otra.
2. Recuento de valores por atributo para cada clase.
3. Normalizar todos los valores de la taba entre 0 y 1.

Después de crear el modelo, dada una muestra nueva, se deberá realizar los siguientes pasos:

1. Determinar los valores de probabilidad de cada atributo de la nueva muestra.
2. Calcular la probabilidad de pertenecer a cada clase, es decir multiplicar el valor de cada atributo de dicha nueva clase, por el valor de probabilidad normalizado en el paso de creación del modelo.
3. Por último, la clase se clasificará según la probabilidad obtenida para cada clase.

El algoritmo Random Forest (RF), como su nombre indica es un algoritmo que consiste en combinar varios árboles de decisión. Este algoritmo pertenece a la familia de métodos de Bagging.

En la fase de creación del modelo, RF crea un número N de árboles, genera de manera aleatoria un conjunto de entrenamiento para cada uno de los N árboles. Cada árbol es entrenado individualmente, utilizando así el resto de conjuntos de datos para evaluar el rendimiento del árbol.

Para cada nodo de partición del árbol elige aleatoriamente K variables en las cuales basará la decisión.

En la fase de clasificación, el modelo recibe una nueva muestra que es introducida en cada uno de los árboles que tiene el algoritmo. Cada uno de ellos produce una clasificación, y la clase más votada por todos los árboles, será la que el clasificador determinará para esa muestra.

Las máquinas de soporte vectorial (Support Vector Machine - SVM) para clasificación realizan una transformación, generalmente a un espacio de una dimensionalidad superior donde poder generar un hiperplano que separe las clases con la ayuda de los denominados vectores soporte.

Los vectores soporte son muestras o puntos del conjunto de entrenamiento que forman dos líneas paralelas al hiperplano. La distancia que hay entre estas las líneas paralelas es el margen. La mejor solución es la que permita mayor margen entre las muestras de las dos clases.

En la figura 6 se muestra un ejemplo de un hiperplano de una máquina de soporte vectorial en un problema de dos clases.

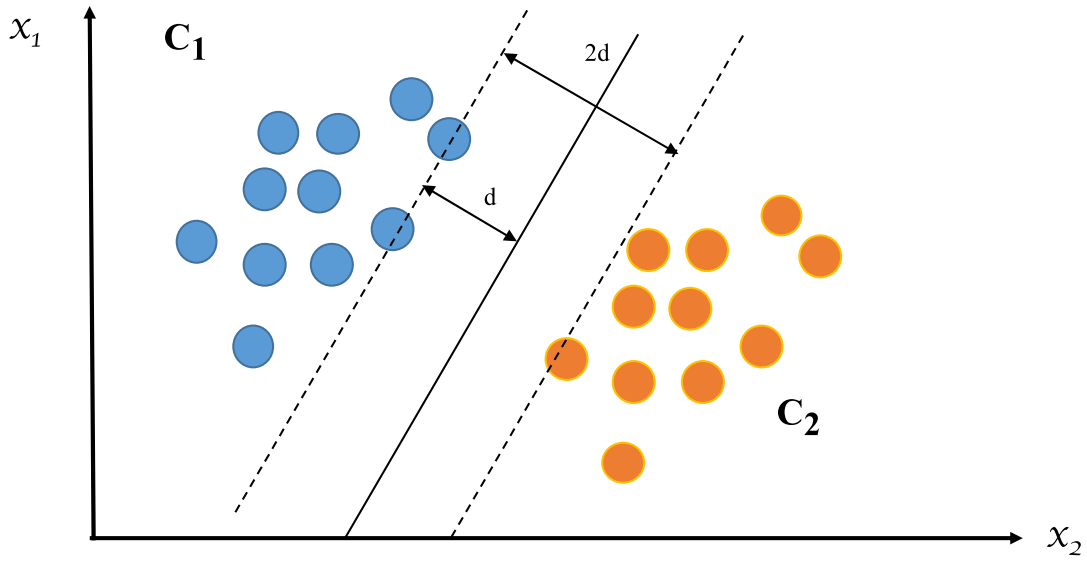


Figura 6: Hiperplano Clasificador SVM

.

Capítulo 3

Metodología usada

La metodología utilizada para conseguir los objetivos de este Trabajo Fin de Máster se ha basado en líneas generales en la realización de las siguientes tareas:

1. En primer lugar, se ha realizado una revisión del estado del arte actual mediante un análisis bibliográfico de lo que se está realizando en estos momentos. Esto se ha descrito en el apartado anterior.

2. Analizar el stream de tweets obtenidos mediante una consulta de dominio través de la Filter realtime API de Twitter.

3. Identificar y seleccionar características del conjunto de datos que nos sirvan para determinar la manera de construir el conjunto de datos que permita identificar o determinar qué tweets son irrelevantes.

4. Seleccionar cuál es el método de clasificación idóneo para el problema en cuestión.

5. Analizar con métricas de evaluación la calidad de la solución aportada.

3.1 Preparación de los datos

El dataset de partida estaba particionado en dos ficheros. El primero de los ficheros contenía la información relativa al Tweet y el segundo de los ficheros contenía la información relativa a las métricas de usuario.

Esto ha sido particionado de esta manera para generar un menor volumen de información a almacenar a priori, ya que la información de usuario estaría repetida en cada uno de los tweets que un usuario hubiera publicado.

La primera parte del dataset constaba de las siguientes columnas:

Dataset Parte I			
TwitterID	User Name	Language	TweetText
1000000448080859136	PeugeotSevilla	es	Todos estos Peugeot están esperando tu visita…
…	…	…	…

Tabla 5: Formato dataset original parte I

Siendo:

- **TwitterID:** Es un número de 19 cifras que identifica al Tweet en la red social Twitter.
- **User Name:** Es un string con el nick o nombre que identifica al usuario.
- **Language:** Es una columna categórica que indica el lenguaje en el que ha sido publicado el Tweet. Puede tomar los valores inglés o castellano.
- **Tweet Text:** Es un string con el texto del Tweet publicado.

La segunda parte del dataset estaba preparada para poder realizar un *join* a través de la clave User Name que comparten los dos datasets y así formar un dataset con mayor número de características.

Dataset Parte II						
User ID	Favourites count	Followers count	Friends count	Statuses count	onDomain tweets	User Name
636864983	23	700	966	1161	44	PeugeotSevilla
…	…	…	…	…	…	…

Tabla 6: Formato dataset original parte II

En esta segunda parte del dataset, la información del usuario contenida era la siguiente:

- **User ID:** Es un número de 9 cifras que identifica al usuario en la red social Twitter.
- **Favourites count:** Cuántas veces ha sido favorito
- **Followers count:** Número de seguidores de la cuenta de twitter
- **Friends count:** Cuentas que son seguidores recíprocamente entre ellas
- **Statuses count:** Número de tweets emitidos por ese usuario
- **OnDomain tweets:** Cantidad total de tweets publicados en el dominio de la consulta
- **User Name:** Es un string con el nombre (nickname) que identifica al usuario.

Una vez analizado el formato del dataset, la siguiente tarea a realizar ha sido añadir una columna más en la que indicar la clase del tweet. Cada tweet puede ser considerado:

- **Tweet relevante:** Es aquel que utiliza las palabras clave de la consulta con el sentido correcto y su contenido está relacionado con la temática de los tweets obtenidos de la consulta de dominio.

- **Tweet irrelevante:** Se consideran irrelevantes los considerados spam o memes, o aquellos que utilizan las palabras clave de la consulta para formar expresiones hechas con un sentido diferente al de la temática.

3.1.1 Etiquetado del dataset

Este punto ha sido uno de los más difíciles de automatizar, ya que se ha partido de cero, con un conjunto de datos de los cuales no se tenía ninguna etiqueta. Este es uno de los principales problemas que tienen los algoritmos de aprendizaje automático supervisados [34], necesitan de un etiquetado previo de clases para poder funcionar, que por lo general son generados manualmente por expertos en la materia.

El proceso de etiquetado de clases ha sido un proceso semi-supervisado e iterativo. Los pasos realizados han consistido en los siguientes puntos:

1. Realizar un análisis de los tweets del dataset en búsqueda de tweets que pudieran considerarse "no relevantes". (Spam o memes repetitivos)

2. Una vez identificados varios tweets de los que se consideran de esta clase, se ha realizado un filtrado de todo el dataset por cadenas de strings y los que han coincido con la consulta, han sido etiquetados en la clase de "no relevantes" (denominada Clase 2).

3. Se ha repetido el proceso realizado en el apartado dos para conseguir varios tweets de los considerados de clase relevantes (denominada Clase 1).

4. Con los tweets etiquetados en los apartados dos y tres, y los no etiquetados hasta el momento se realiza un entrenamiento del modelo Doc2vec para confeccionar un diccionario con el que se podrán inferir vectores.

5. Con el modelo Doc2vec entrenado y el diccionario generado, se infieren los vectores de todos los tweets, etiquetados y sin etiquetar.

6. Con los vectores inferidos de los Tweets etiquetados se entrenan tres algoritmos: Random Forest, Support Vector Machine y Linear Support Vector Machine. Con los algoritmos entrenados con las clases etiquetadas, se realiza una predicción en búsqueda de más tweets de la clase relevante.

7. Con los resultados de los modelos, se realiza una nueva consulta con nuevas cadenas de caracteres para etiquetar más Tweets.

8. El proceso de los puntos 2 a 7 se ha repetido 2 veces.

Este proceso se ha representado en el diagrama de bloques de la figura 7.

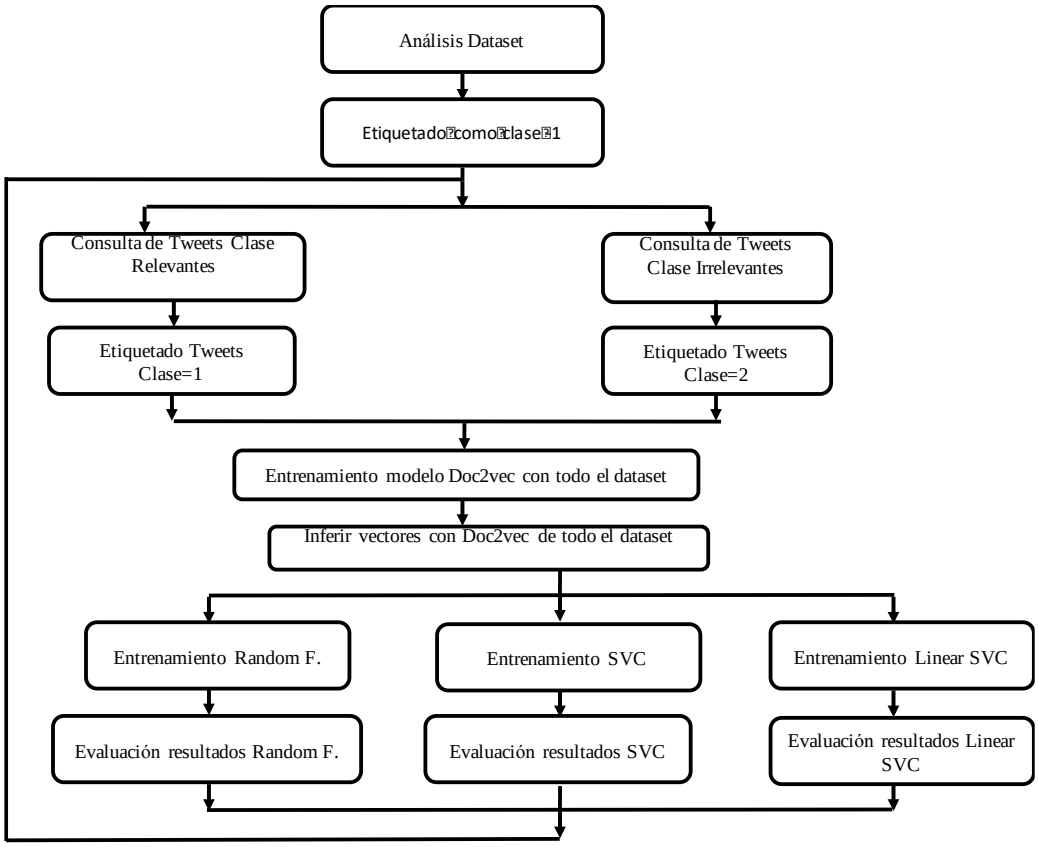


Figura 7: Diagrama para etiquetar Tweets

3.1.2 Generación de los datasets de entrenamiento

Para el entrenamiento de los clasificadores, se han generado tres conjuntos de entrenamiento con los que realizar los ensayos. Los ensayos realizados se presentan en el capítulo 4. Los datasets que se han generado han sido los siguientes:

- Dataset 1: Solo métricas
- Dataset 2: Texto de tweets convertido en vectores
- Dataset 3: Métricas enriquecidas con los vectores inferidos de los textos de los tweets

La generación de los tres datasets ha sido realizada tomando como partida el dataset etiquetado según se ha explicado en el apartado anterior.

Para generar el dataset 1, con solamente las métricas, el proceso realizado ha sido el siguiente:

1. Con el dataset de partida, se le ha añadido las métricas de usuario mediante un merge con las columnas User Name común en ambos datasets.

2. Se han eliminado las columnas siguientes:
    a. TwitterID
    b. TweetText
    c. UserName
3. Se ha reemplazado la columna Language los datos:
    a. Idioma “Spanish” =1
    b. Idioma “English” =0

Tras realizar estos pasos, el dataset generado tiene las columnas que se ven en la tabla 7:

Favourites count	Followers count	Friends count	Statuses count	onDomain tweets	Language	Clase

Tabla 7: Dataset I entrenamiento solo métricas

El dataset 2 ha sido formado por el texto de los tweets convertido en vectores. Tal y como se ha explicado en el capítulo 2, para la obtención de los vectores de los tweets se ha decidido utilizar el modelo Doc2vec que infiere los vectores de los tweets utilizado la librería gensim para Python que implementa dicho modelo.

El modelo doc2vec puede implementar el modelo Continuous Bag-of- Words o el Skip-gram. Se ha utilizado el modelo Skip-gram que, aunque es un poco más costoso en tiempo, proporciona mejores resultados puesto que tiene en cuenta el orden en que las palabras aparecen mientras que CBOW no.

Google propone el modelo con una capa oculta de 300 neuronas en su modelo entrenado (https://code.google.com/archive/p/word2vec/), en este trabajo primero se entrenó el modelo con una capa oculta de 100 neuronas y otro ensayo con 50. Tras evaluar los resultados en una clasificación posterior con el mismo set de clasificadores configurados de la misma manera, y obtener los mismos resultados, se decidió dejar la capa oculta con 50 neuronas para siguientes ensayos y reducir el tamaño del dataset y un ahorro de tiempo de entrenamiento en los siguientes ensayos.

La configuración final con la que se ha generado el modelo ha sido la de la tabla 8:

Parámetro	**Valor**	**Descripción**
Workers	8	Número de trheads para entrenar el modelo
min_count	5	Ignora palabras que aparecen menos que el valor configurado
Sample	1e-5	Umbral para configurar qué palabras se muestrean al azar
Alpha	0,01	Tasa de aprendizaje inicial
Window	10	Máxima distancia entre la palabra actual y la pronosticada dentro de una frase
Epochs	40	Número de iteraciones sobre el corpus
Dm	0	Configura Distributed Bag of Words
dbow_words	1	Se entrena simultáneamente Skip-gram con DBOW
vector_size	50	Dimensión de neuronas de la capa oculta y del vector de salida

Tabla 8: Configuración del modelo Doc2vec

El proceso para generar los vectores para el dataset 2 ha sido:

1. Filtrar los textos procedentes de los tweets.
2. Construir el modelo con los parámetros de la tabla 8.
3. Entrenar el modelo con todo el conjunto de tweets para aprender un vocabulario lo más amplio posible.
4. Inferir los vectores de los textos de los tweets

El dataset 2 está formado por las 50 columnas de los vectores obtenidos de los textos de los tweets más una columna con la etiqueta de la clase.

Para la generación del dataset 3 se ha fusionado el dataset 1 que contiene las métricas con el dataset 2 que contiene los vectores obtenidos del texto de cada uno de los tweets, de modo que el dataset 3 tiene la forma del dataset 1 más las 50 columnas añadidas con los vectores del texto, formando un conjunto de datos de 56 características más una columna adicional con la etiqueta de clase.

3.2 Selección del método de clasificación

Para seleccionar el mejor método de clasificación se han generado los tres datasets descritos en el apartado anterior y se ha realizado una comparación de resultados con el conjunto de algoritmos probados.

En este trabajo se van a evaluar dos puntos que servirán para conseguir el objetivo de determinar cuál es el mejor método para detectar tweets considerados irrelevantes y poder ser filtrados para un análisis de información social. Para ello, como en cualquier problema de minería de datos es tan importante el algoritmo seleccionado como la selección de características para formar el conjunto de datos de entrenamiento.

En el capítulo 4 se describe cómo se han realizado los experimentos para detectar el mejor método de clasificación y la forma de construir el dataset que ayuda a maximizar los resultados en la fase de predicción.

El procedimiento efectuado para esta tarea ha sido el siguiente:

1. Formar los tres conjuntos de datos a trabajar: Uno de solo métricas, otro de solo el texto de los tweets convertido en vectores y el tercero la unión de los dos datasets anteriores.

2. Evaluar con el mismo conjunto de algoritmos cada uno de los tres datasets, con esto se podrá ver con cuál de los tres conjuntos de datos se obtienen mejores resultados de clasificación y ventajas y desventajas de utilizar cada uno de los tres conjuntos de datos.

3. Por último, se seleccionará el algoritmo de clasificación que mejores resultados ofrezca en la detección de tweets pertenecientes a la clase irrelevante. Para ello

se utilizarán las métricas descritas en el siguiente apartado como medida de evaluación comparativa entre los distintos algoritmos.

Una vez seleccionadas las características que conforman el conjunto de datos de trabajo y el clasificador que mejores resultados otorga en la detección de tweets irrelevantes, la siguiente tarea sería validar el método en su totalidad con un conjunto de datos de otro espacio temporal donde se pueda verificar la capacidad de generalizar y detectar posibles memes que existan en ese espacio temporal.

Como se explicó en el capítulo 2, un meme tiene un momento de aparición que se hace viral y muchos usuarios lo comparten y después deja de ser viral. Para el caso de la detección de memes seguramente habría que incluir en la información del tweet, cuál es el time-stamp de publicación del tweet, así como el número de retweets acumulado.

3.3 Métricas de evaluación

En este apartado se definen las métricas que se han utilizado para evaluar este trabajo [12], en primer lugar, se definirán unos conceptos fundamentales con su terminología en inglés:

- **Positives (P):** es el número de eventos.
- **Negatives (N):** es el número de muestras que no son los eventos que se quieren detectar.
- **True Positives(TP):** es el número de eventos de la clase positiva clasificados en la clase positiva.
- **True Negatives (TN):** es el número de muestras de la clase negativa clasificadas como eventos de la clase positiva.
- **False Negatives (FN):** es el número de muestras de la clase negativa que se clasifican como clase positiva.
- **False Positives (FP):** es el número de eventos de la clase positiva que se clasifican como clase negativa.

Matriz de confusión:

La matriz de confusión es una tabla como la siguiente, que se utilizará para representar el rendimiento del clasificador:

Clase de los datos	Clasificados como positivos	Clasificados como negativos	
Positive	True Positive (TP)	False Negative (FN)	P = TP + FN
Negative	False Positive (FP)	True Negative (TN)	N = TN + FP

Tabla 9: Matriz de confusión

Accuracy: Es la medida de rendimiento más intuitiva y simple de interpretar. Es la suma de aciertos dividido del total de elementos.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \; \epsilon \; [0,1]$$

Precisión: Este indicador corresponde al ratio de las predicciones realizadas positivas dividido entre todas las predicciones positivas.

$$Precisión = \frac{TP}{TP + FP} \; \epsilon \; [0,1]$$

Recall o True Positive rate (TPr): Esta ratio ilustra las muestras predichas correctamente entre todas las que hay en la misma clase positiva

$$Recall = \frac{TP}{TP + FN} \; \epsilon \; [0,1]$$

F1-Score: O media armónica entre la precisión y el recall. Es el inverso de la media aritmética

$$F1 - Score = \left(\frac{Recall^{-1} + Precisión^{-1}}{2} \right)^{-1} \; \epsilon \; [0,1]$$

Operando obtenemos:

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \; \epsilon \; [0,1]$$

Support:

Es el número de ocurrencias para cada clase etiquetada correctamente.

Micro-Average y Macro-Average:

En el caso de que las clases del conjunto de datos sean desbalanceadas, tenemos los métodos media Micro (Micro-average) y media Macro (Macro-average) que se puede aplicar para promediar las métricas anteriores.

El método Micro-average consiste en calcular en conjunto para todas las clases cada una de las medidas de rendimiento. Esto favorece a que la clase mayoritaria puntúe más y la clase minoritaria puntúe menos. En este caso, la forma de calcular las métricas de rendimiento para un problema de clasificación binario, es tal cual se ha expuesto con anterioridad, es decir:

$$Precisión\ Micro - Avg = \frac{TP1 + TP2}{TP1 + TP2 + FP1 + FP2} \in [0,1]$$

$$Recall\ Micro - Avg = \frac{TP1 + TP2}{TP1 + TP2 + FN1 + FN2} \in [0,1]$$

Por el contrario, el método Macro-average consiste en calcular cada una de las medidas de rendimiento para cada una de las clases y después darles el mismo peso a ambas clases.

En un problema de clasificación binaria como el de este trabajo, la precisión con Macro-Average se calculará en primer lugar la precisión para cada una de las clases:

$$Precisión\ P1 = \frac{TP1}{TP1 + FP1} \in [0,1]$$

$$Precisión\ P2 = \frac{TP2}{TP2 + FP2} \in [0,1]$$

Una vez calculadas las precisiones por separado, se realiza el promedio de los dos valores, es decir:

$$Precisión\ Macro - Avg\ = \frac{P1 + P2}{2} \in [0,1]$$

De forma análoga, se calcularía el recall. En primer lugar, se calcula para cada clase por separado:

$$Recall\ R1 = \frac{TP1}{TP1 + FN1} \in [0,1]$$

$$Recall\ R2 = \frac{TP2}{TP2 + FN2} \in [0,1]$$

Por último, una vez calculado el Recall para cada una de las clases por separado se realiza el promedio de ambos:

$$Recall\ Macro - Avg\ = \frac{R1 + R2}{2}\ \epsilon\ [0,1]$$

Este método Macro-average, permite dar la misma importancia a ambas clases independientemente de la cantidad de muestras que tenga. Permite mostrar de una manera más justa las métricas del clasificador.

Otro método de evaluar las métricas es mediante la media ponderada, esto significa que el valor de la métrica se calcula clase por clase y después se multiplica por la cantidad de muestras que había de la clase y se divide del total de muestras que han participado, dicho de otra manera, cada media se multiplica por el porcentaje de muestras que tiene.

El método de media ponderada y Micro-average han sido descartados en este trabajo dado a que el conjunto de datos está desbalanceado y los resultados de las métricas podrían verse influenciados a favor de los resultados obtenidos en la clase mayoritaria.

Capítulo 4

Experimentos y resultados

4.1 Software utilizado

El presente trabajo ha sido desarrollado exclusivamente mediante el uso del lenguaje Python, este lenguaje de programación creado por Guido van Rossum a principios de los años 90 y consta de una licencia BSD que permite su uso de manera gratuita (https://www.python.org). Esto junto con la gran cantidad de librerías disponibles y la potencia del lenguaje ha ayudado a su fuerte expansión y que hoy en día sea uno de los lenguajes más utilizados.

Python es un lenguaje de programación interpretado o de scripts, con una sintaxis limpia y pensada para ser un código legible por las personas. Es un lenguaje de programación multiparadigma, ya que permite crear programas de varios estilos de programación, programación orientada a objetos, programación imperativa (como C, Fortran…) o programación funcional basado en el uso de funciones matemáticas [36].

Las librerías que se han utilizado en la realización de este trabajo han sido:

- Pandas: Esta librería ha sido utilizada para la importación de los datasets, así como para la manipulación de los datos contenidos en dichos conjuntos de datos [37].

- Matplotlib: Es una de las librerías más utilizadas para mostrar resultados gráficamente. En este trabajo se han utilizado para mostrar las gráficas de barras y los boxplots que aparecen en este trabajo [38].

- Scikit-learn: Es la librería utilizada para todas las tareas de machine learning, desde el particionado de los datasets, implementación del K-Fold cross validation, implementación de los algoritmos de machine learning. [39]

- Gensim: Es la API utilizada para implementar el algoritmo Doc2vec que convierte documentos de texto, en este caso tweets en vectores de N-dimensiones para poder trabajar con ellos con los algoritmos de clasificación [40]

- NLTK: Es una plataforma para trabajar en python con lenguaje natural, ésta ha sido utilizada para la preparación de los textos de los tweets antes de ser convertidos a vectores para procesarlos mediante los algoritmos de machine learning. [41]

- WordCloud: Es la librería utilizada para la realización de las nubes de palabras más repetidas por cada clase. [42]

4.2 Experimentos realizados

Según se ha explicado en el capítulo 3, la primera tarea a realizar después de realizar una revisión del estado del arte actual ha sido construir el dataset de trabajo. El dataset ha sido realizado tras analizar el stream de tweets obtenidos de la consulta de la API streaming de twitter.

4.2.1 Etiquetado del dataset

La consulta de dominio consiste en una serie de palabras clave , de modo que si aparecen en un tweet, la API devuelve el tweet que contiene alguna de las palabras clave. Las palabras que forman la query en la API de Twitter son las siguientes:

```
entities = ["opel corsa", "vauxhall corsa", "Renault Clio", "Citroen C3", "Peugeot 207", "Seat Ibiza", "Peugeot 208", "Toyota Auris", "Renault Megane", "Peugeot 308", "Citroen C4", "Ford Focus", "Seat Leon", "Toyota Avensis", "opel insignia", "vauxhall insignia", "Citroen C5", "Ford Mondeo", "Peugeot Bipper", "Citroen Nemo", "Renault Kangoo", "Citroen Berlingo", "Peugeot Partner", "Ford Transit", "Volkswagen Caddy", "Nissan NV200", "Citroen Jumpy", "Peugeot Expert", "Opel Vivaro", "Vauxhall Vivaro", "Nissan Pathfinder", "Nissan Qashqai", "Toyota Rav 4", "Toyota Rav4", "Land Rover Freelander", "NISSAN X-TRAIL","Nissan xtrail", "Land Rover Defender", "Mitsubishi Montero", "Mitsubishi Pajero", "Shogun Spirit", "Mitsubishi L200", "Toyota Hilux", "Volkswagen Amarok", "Nissan Cabstar","Seat Arona","Seat Ateca","Renault Talisman"]
```

El proceso realizado para etiquetar y construir el conjunto de datos es el que se ha explicado en el diagrama de bloques de la figura 7. Tras realizar dos iteraciones al proceso definido en el diagrama de bloques, se ha generado el dataset final de trabajo mediante las consultas que se describen en las tablas 10 y 11 que han proporcionado las etiquetas para las dos clases posibles:

En la tabla 10 se muestran todas las consultas de palabras o cadenas de palabras realizadas para etiquetar los tweets de la clase 1, o clase relevante.

Clase Relevante		
#Seat600	CV	Lamentablemente me tocó a mí
experto	Diesel	Los lunes no siempre son grises
seat leon	TDI	Ruego que cualquiera que pasase por la A67 dirección
Peugeot 208	CDTI	Se busca a una amiga. Su vehículo es un Ford focus
Anda un peugeot 207	GTI	chicas de illescas tened cuidado!!!
Margarita Barrientos	#Ford	Para todas las chicas de Torrejón
vendo seat	#Nissan	Figueruelas
Se vende	#Peugeot	PEUGEOT 207 COMPAC NEGRO PATENTE LXJ893
FORD TRANSIT	#Opel	Hoy han vuelto a perseguirme por la Castellana tres coches
renting	#Citroen	Nos acaban de robar en la bomba COPEC
robaron	#Mitsubishi	Pathfinder
#Los40BasicoOpelCorsa	#Subaru	Toyota anunció que aumentaba
Alboran	#Hyundai	TABLÓN de ANUNCIOS
#doncasterisgreat	#RenaultClioCup	Robado
#ROBADO	#luxurycars	Stolen
#URGENT	#Dakar	
Año	Year	

Tabla 10: Consultas para etiquetar clase relevante

En la tabla11, se pueden observar en las tres primeras consultas una búsqueda con dos palabras que podían aparecer separadas y por ello se ha realizado la búsqueda cumpliendo las dos condiciones, al lado del símbolo “&”.

Clase Irrelevante	
ADHD & focus	Opel Corsa blanco mal aparcado
Piso & Torrevieja	-Perdonad, hay un SEAT Ibiza rojo aparcado en la puerta, que vaya el dueño
smart tv & focus	Más tenso que el dueño de una Citroën Berlingo
Vendo Opel Corsa	Scioli pidió ver la letra chica del acuerdo con el FMI'
Se vende opel corsa	Dios es ese primo tuyo que dejó
#Eurovision	El audio de Simeone tendría más repercusión
#Eurovision	Para pedir comida por favor
matricula 6969	Twitter ya ha perdido la poca cordura que le quedaba
#VendoOpelCorsa	nuggets de pollo
lamborgini	@Dellafuente_ Vendo opel corsa
#MpuntoRajoy	Ni que me aprestaran las patas como para
Rajoy	Hoy todos los diputados del PP han abandonado
Puigdemont	#Follar está sobrevalorado
rufian	La procesión de las autopistas resucitadas con dinero público arranca el sábado
Colau	Al dueño del Opel Corsa verde
Irene_Montero	Yo más que un Ferrari soy un Renault
Adolfo Suárez	Tranquilos que cuando Pablo e Irene
demócrata	Virgen Santísima la pinta que tiene Sampaoli de tener un Seat León amarillo
Barajas	Pueden sacar una temporada del Fortnite basada en barrios de España donde puedas conducir un 600
A juzgar por los unfolous	ninja
Pajamei	Jaja

Tabla 11: Consultas para etiquetar clase irrelevante

4.2.2 Estudio de los datos

En este punto se ha tratado de identificar y seleccionar características de las métricas. Para ello en primer lugar se ha realizado un análisis estadístico de las métricas separadas por clase 1 y clase 2. Se ha resumido en las tablas 12 y 13.

Resumen estadístico Métricas Clase 1					
	Favourites count	Followers count	Friends count	Statuses count	OnDomain tweets
count	121129.0	121129.0	121129.0	121129.0	121129.0
mean	7,932E+15	6,291E+15	2,150E+16	3,987E+15	1,728E+15
std	2,637E+16	6,607E+15	1,026E+16	9,272E+15	6,130E+16
min	0.0	0.0	0.0	1.0	1.0
25%	12.0	148.0	174.0	2259.0	2.0
50%	487.0	584.0	513.0	9776.0	64.0
75%	5213.0	2093.0	1469.0	40454.0	794.0
max	1043111.0	6689508.0	346450.0	2923351.0	89436.0

Tabla 12: Resumen estadístico de las métricas de la clase relevante

Las tablas del análisis estadístico tienen ocho filas que representan:

- **Count:** Número total de tweets con datos de cada una de las columnas
- **Mean:** Valor medio de los datos contenidos en cada una de las columnas
- **Std:** Desviación estándar de los datos contenidos en cada columna
- **Min:** Valor mínimo
- **25%, 50%, 75%:** Cantidad de valores en el intervalo 0 a 25, 25 a 50 y 50 a 75
- **Max:** Cantidad de valores en el intervalo 75 a 100

Resumen estadístico Métricas Clase 2					
	Favourites count	Followers count	Friends count	Statuses count	onDomain tweets
count	27717.0	27717.0	27717.0	27717.0	27717.0
mean	1,530E+16	1,289E+16	5,983E+15	2,602E+16	2,973E+16
std	3,121E+16	1,330E+16	1,389E+15	4,709E+15	3,768E+16
min	0.0	0.0	0.0	1.0	1.0
25%	1804.0	151.0	192.0	4246.0	1.0
50%	5694.0	330.0	351.0	12302.0	1.0
75%	16331.0	656.0	642.0	29647.0	2.0
max	843624.0	1365769.0	143793.0	1299921.0	16686.0

Tabla 13: Resumen estadístico de las métricas de la clase irrelevante

En la figura 8 se muestran los *boxplots* de cómo se distribuyen las métricas de los usuarios en función de la clase a la que pertenece cada uno de los tweets publicados. El objetivo del análisis de los boxplots es entender cuanto pueden aportar cada una de las

métricas en la tarea de segmentar las clases. Si comparamos las métricas On_Domain_tweets y Statuses comparision que son el número total de tweets en el dominio de la consulta y el total de tweets publicados por el usuario, se ve como la primera segmentará mejor los tweets que la segunda.

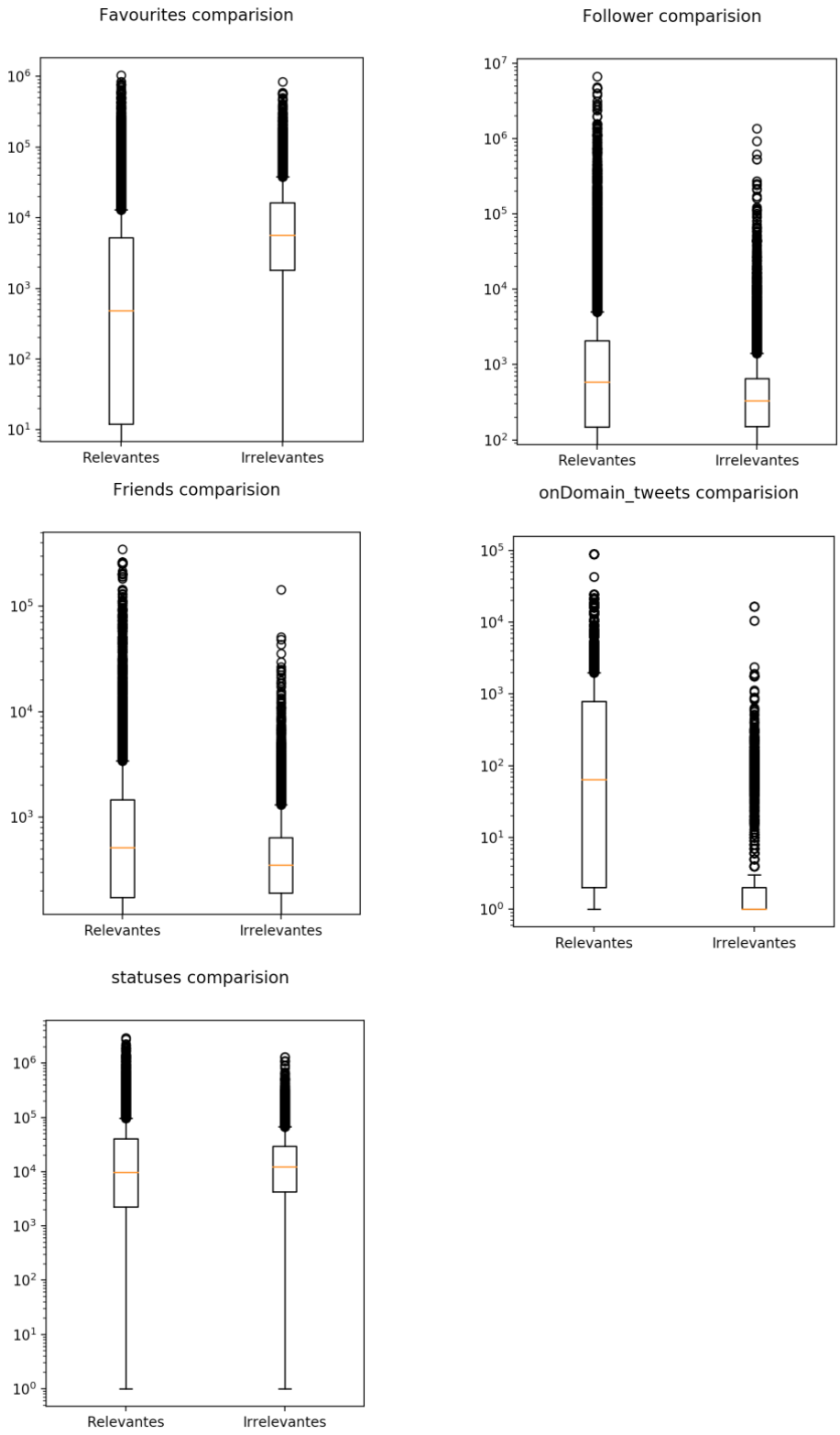


Figura 8: Boxplots de las métricas por clase

Al conjunto de métricas se podría haber añadido la métrica retweets que tiene un tweet, ya que ayudaría en la detección de memes. Como se explicó en el capítulo 2, un tweet considerado meme es un tweet que, a pesar de no haber nacido para ser meme, acaba convirtiéndose en viral. La métrica retweet se ha desestimado su incorporación dado que el número de retweets proporcionará una información de viralidad que no será online si no que será tras repetirse el mismo tweet e ir incrementando el contador de tweets de manera dinámica, por ello se considera una métrica interesante para un análisis a posteriori pero no cumpliría el objetivo de detectar tweets memes en tiempo real.

Continuando con el análisis de los datos se ha realizado un Word Cloud con las 100 palabras más repetidas en cada una de las dos clases. En la figura 9 se pueden ver las palabras más repetidas de la clase relevante y en la figura 10 las palabras más repetidas en los tweets de clase irrelevante.

Observando las palabras que más se repiten en la figura 9, se puede deducir sin mirar el pie de la figura que se trata de las palabras que más tienen que ver con el dominio de la consulta ya que aparecen muchas palabras de modelos de coches, marcas, etc.

Figura 9: Representación de las 100 palabras más repetidas en la clase 1

Sin embargo, al observar las palabras más repetidas en la figura 10 se pueden ver palabras del dominio como Opel corsa, pero las demás no pertenecen al dominio de la consulta realizada. Estas dos palabras que son del dominio están utilizadas fuera del contexto del dominio de la consulta y por ello están dentro de las consideradas de la clase irrelevante coinciden con el famoso meme “Vendo Opel Corsa”.

Dentro del mismo ejemplo de vendo Opel corsa, en mayo del 2018 durante el festival de eurovisión hubo una mutación del típico meme vendo Opel corsa que se hizo muy viral porque un espontáneo saltó al escenario cuando estaba cantando la representante del Reino Unido.

Figura 10: Representación de las 100 palabras más repetidas en la clase 2

4.2.3 Diseño de los experimentos

En este apartado se exponen los ensayos realizados [43] y se justificará el porqué de las decisiones tomadas. En primer lugar, se han realizado dos ensayos con un conjunto de cuatro clasificadores.

El primero de los ensayos se ha realizado conformando un conjunto de datos donde se han utilizado únicamente las métricas, en el segundo los vectores de los textos de los tweets, y el tercero se ha formado de la unión de las métricas de los tweets como en el primer caso y enriqueciéndolo con los vectores inferidos con el algoritmo Doc2vec del segundo. El proceso de generación de los datasets se explica en la sección 3.

Los ensayos han sido realizados mediante una partición inicial de los datos, donde el 70% de las muestras es utilizado para la fase de entrenamiento y el 30% restante se ha utilizado para formar el conjunto de validación del algoritmo. Este último no se utilizará en la fase de entrenamiento y será totalmente desconocido por el algoritmo a la hora de evaluar.

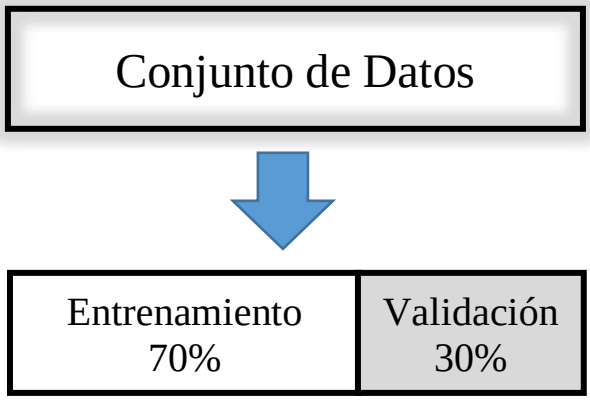


Figura 11: Particionado del conjunto de datos

El conjunto de entrenamiento, tal y como se muestra en la figura 11 se ha utilizado para entrenar cada uno de los algoritmos utilizando un método de remuestreo conocido como

validación cruzada de K bloques (K-fold cross-validation) [44]. El conjunto de datos de entrenamiento de la figura 11, es el utilizado en el método ilustrado en la figura 12.

El método de validación cruzada por bloques consiste en dividir de forma aleatoria el conjunto de datos en K bloques [32,35]. Después se toman K-1 bloques para entrenar y el bloque que no se ha utilizado para el entrenamiento, se utiliza para comprobar el resultado del entrenamiento. Esto se repite para todas las combinaciones de K-1 bloques

Una cuestión crítica suele ser determinar el valor de K, ya que si K es muy grande se tendrá un modelo muy preciso, pero con una varianza alta, mientras que si K es pequeña la varianza será baja, pero el modelo puede resultar sesgado por la aleatoriedad de las muestras elegidas en el particionado. Normalmente los valores de K típicos son K=5 y K=10. En este trabajo todos los ensayos de validación cruzada han sido realizados con un valor de K=10.

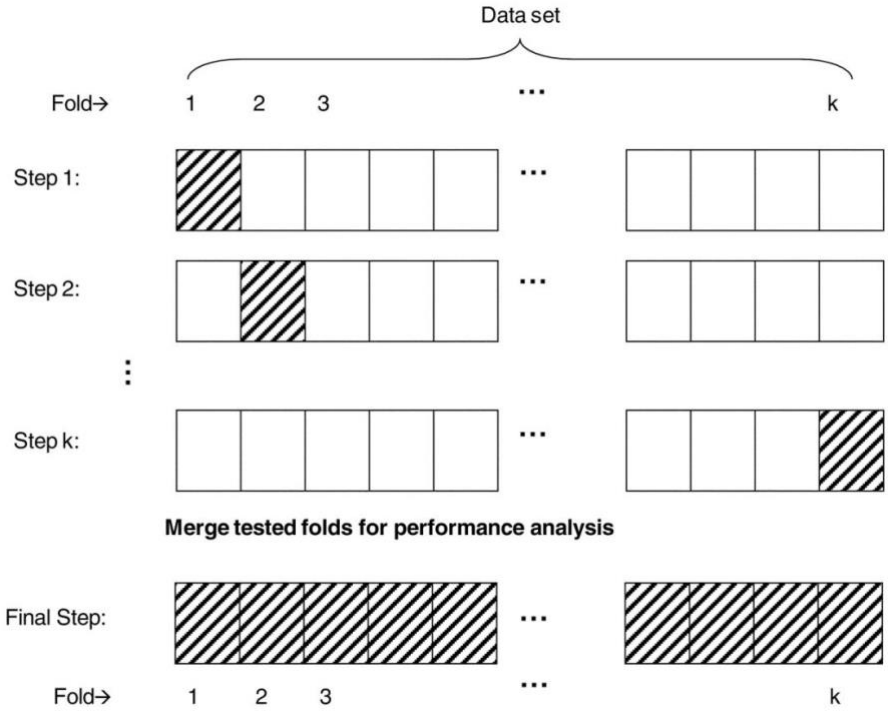


Figura 12: Método de validación cruzada por bloques

El método de validación cruzada implementado en este trabajo ha sido el conocido como Stratified K-Fold cross-validation [44]. En esencia es lo mismo que se ha explicado, solo que ésta configuración garantiza que las proporciones de las clases utilizadas en cada partición sea la misma. Esto se ha realizado así dado que la clase irrelevante está menos representada que la clase de tweets relevantes y el método no Stratified no garantiza las proporciones en las particiones.

El set de clasificadores elegidos y que pueden verse en las tablas de resultados 14 y 15 ha sido formado por: Naïve Bayes como clasificador básico con el que comparar los resultados, Random Forest, Linear Suport Vector Machine, Support Vector Machine con kernel radial denominado en las tablas como SVC y XGBosst.

4.3 Discusión de resultados obtenidos

En la tabla 14 se muestran los resultados de los ensayos realizados con los datasets solo métricas y solo los vectores obtenidos de los textos de los tweets.

			Ensayo solo métricas					Ensayo vectores			
		Naive Bayes	R. Forest	Linear SVC	SVC	XG Boost	Naive Bayes	R. Forest	Linear SVC	SVC	XG Boost
Accuracy Train	Avg	0,41	**0,86**	0,61	0,83	**0,86**	0,80	**0,96**	0,93	0,94	0,95
	Std	0,006	**0,003**	0,187	0,004	**0,002**	0,003	**0,001**	0,003	0,002	0,002
Accuracy Test		0,41	**0,86**	0,71	0,83	**0,86**	0,80	**0,96**	0,93	0,94	0,95
Precision	Clase 1	0,98	**0,92**	0,87	0,83	0,91	0,97	0,95	0,94	0,95	**0,96**
	Clase 2	0,24	0,63	0,33	**0,80**	0,62	0,48	**0,98**	0,86	0,89	0,93
	Avg Micro	0,41	**0,86**	0,71	0,83	**0,86**	0,81	**0,96**	0,93	0,94	0,95
	Avg Macro	0,61	0,78	0,60	**0,82**	0,77	0,72	**0,97**	0,90	0,92	0,94
Recall	Clase 1	0,28	0,91	0,76	**0,99**	0,91	0,79	**1,00**	0,98	0,98	0,99
	Clase 2	0,98	**0,65**	0,52	0,11	0,62	0,89	0,77	0,70	0,77	**0,78**
	Avg Micro	0,41	**0,86**	0,71	0,83	**0,86**	0,81	**0,96**	0,93	0,94	0,95
	Avg Macro	0,63	**0,78**	0,64	0,55	0,77	0,84	**0,89**	0,84	0,88	0,88
F1-score	Clase 1	0,44	**0,92**	0,81	0,90	0,91	0,87	**0,97**	0,96	**0,97**	**0,97**
	Clase 2	0,38	**0,64**	0,41	0,20	0,62	0,62	**0,87**	0,77	0,83	0,85
	Avg Micro	0,41	**0,86**	0,71	0,83	**0,86**	0,81	**0,96**	0,93	0,94	0,95
	Avg Macro	0,41	**0,78**	0,61	0,55	0,77	0,75	**0,92**	0,86	0,90	0,91
Support	Clase 1	36339	36339	36339	36339	36339	38972	38972	38972	38972	38972
	Clase 2	8315	8315	8315	8315	8315	8350	8350	8350	8350	8350
	Avg	44654	44654	44654	44654	44654	47322	47322	47322	47322	47322

Tabla 14: Resultados ensayos solo métricas y solo vectores

En la tabla 15 se muestran los resultados de los ensayos realizados con el dataset formado por métricas y vectores.

			Ensayo métricas + vectores			
		Naive Bayes	R. Forest	Linear SVC	SVC	XG Boost
Accuracy Train	Avg	0,40	0,97	0,75	0,82	**0,98**
	Std	0,009	0,001	0,080	0,003	**0,001**
Accuracy Test		0,40	**0,98**	0,58	0,82	0,97
Precision	Clase 1	0,98	0,97	0,96	0,82	**0,98**
	Clase 2	0,23	**0,99**	0,29	0,99	0,96
	Avg Micro	0,40	**0,98**	0,58	0,82	0,97
	Avg Macro	0,60	**0,98**	0,62	0,90	0,97
Recall	Clase 1	0,27	**1,00**	0,50	**1,00**	0,99
	Clase 2	0,98	**0,88**	0,90	0,01	0,89
	Avg Micro	0,40	**0,98**	0,58	0,82	0,97
	Avg Macro	0,63	**0,94**	0,70	0,50	**0,94**
F1-score	Clase 1	0,43	**0,99**	0,44	0,90	0,98
	Clase 2	0,38	**0,93**	0,44	0,02	**0,93**
	Avg Micro	0,40	**0,98**	0,58	0,82	0,97
	Avg Macro	0,40	**0,96**	0,55	0,46	0,96
Support	Clase 1	36343	36343	36343	36343	36343
	Clase 2	8201	8201	8201	8201	8201
	Avg	44544	44544	44544	44544	44544

Tabla 15: Resultados ensayo métricas y vectores

El objetivo principal de este trabajo como ya se ha comentado es detectar los tweets que son irrelevantes, por ello, dadas las 3 formas de hacer las medias de cada una de las métricas que se han explicado: media ponderada, media micro y media macro [45]; la que penaliza más a la clase que estamos buscando es la media macro, ya que teniendo menos muestras que la clase 1 o clase relevante, de este modo tienen ambas el mismo peso.

Por ello, los métodos de media ponderada y media micro han sido descartados en este trabajo dado a que el conjunto de datos está desbalanceado y los resultados de las métricas podrían verse influenciados a favor de los resultados obtenidos en la clase

mayoritaria, enmascarando el resultado real del clasificador. Por lo general una clase con mayor número muestras representa mejor a una población estadística y el resultado será mejor que la clase menor representada. Esto ha sido corroborado tras obtener los resultados de los ensayos mostrados en las tablas 14 y 15.

En primer lugar, se va a evaluar cuál de los tres datasets es el que mejores resultados de clasificación genera y así elegir la forma óptima de construir el dataset en el caso de querer poner este sistema en producción, u optimizar este trabajo a futuro. Los datasets a evaluar son:

- Dataset 1: Solo métricas.
- Dataset 2: Solo vectores.
- Dataset 3: Métricas y vectores.

Estos tres datasets han sido evaluados mediante un conjunto de algoritmos tal y como se han visto en las tablas 14 y 15. En el primer tipo de ensayo que se ha realizado con el dataset conformado únicamente con métricas, es donde peores resultados han obtenido cada uno de los clasificadores, aunque los resultados que obtienen los algoritmos Random Forest o XGBoost en precisión, recall y F1-score están rondando el 0,8, lo cual es un indicador bueno de calidad de estos dos clasificadores. En la figura 13 se han representado mediante una gráfica de barras los valores de precisión obtenidos por cada clasificador en cada ensayo, en la figura 14 los resultados recall y en la figura 15 el F1-score.

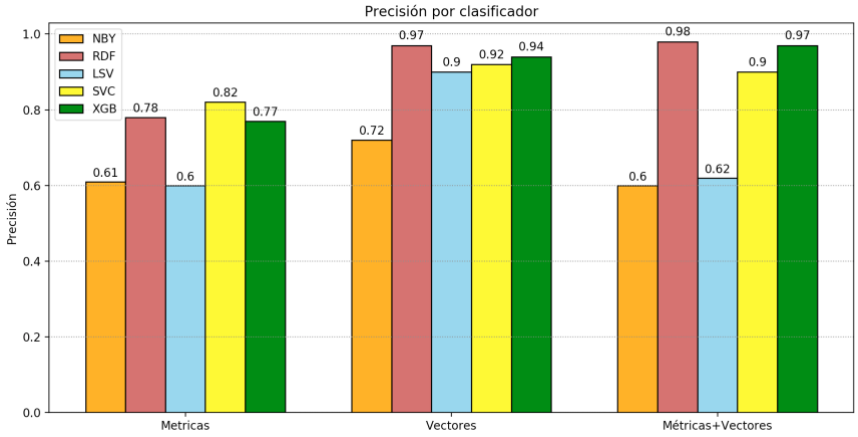


Figura 13: Gráfica precisión por algoritmo vs ensayo

Siguiendo el análisis para conformar el dataset de la forma más óptima, evaluaremos los otros dos ensayos de manera conjunta, ya que no todos los clasificadores probados funcionan mejor con el mismo conjunto de datos.

Los clasificadores Naïve Bayes y las SVM, en el ensayo de solo vectores, es donde mejores puntuaciones obtienen tanto con kernel lineal como con kernel radial. Los valores de precisión, recall y f1-score, de las máquinas de soporte vectorial obtienen sus mejores puntuaciones rondando el 0,9 exceptuando cuando trabaja con kernel lineal que el recall y F1-score se queda por debajo. Para el algoritmo de Naïve Bayes, este ensayo es donde obtiene su mejor puntuación, siendo el que peor puntuación ha obtenido en este ensayo de entre todos los modelos. Siguiendo con el ensayo de solo vectores, los

mejores resultados han sido obtenidos, al igual que en el primero por los modelos Random Forest y XGBoost, aunque con menor diferencia con los clasificadores explicados.

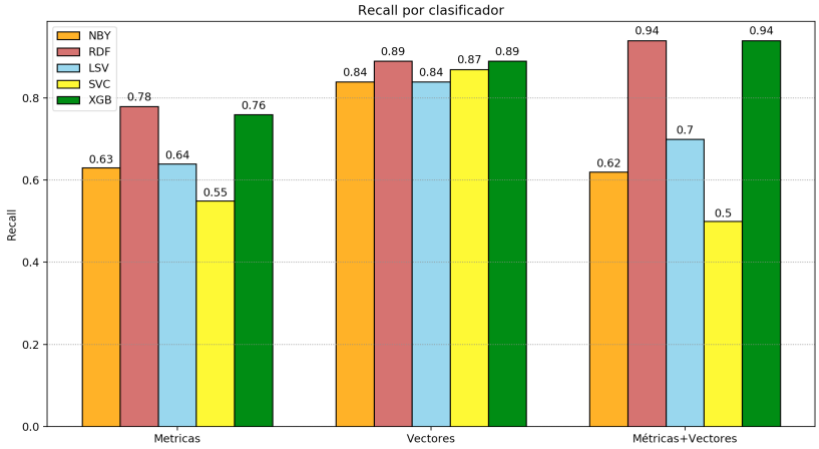


Figura 14: Gráfica Recall de cada clasificador vs ensayo

Por último, si nos centramos en el ensayo del último conjunto de datos y más completo de todos, formado por métricas y vectores, nuevamente los mejores resultados son para los clasificadores Random Forest y XGBoost. En este caso las diferencias con respecto al resto de modelos son mayores.

Una vez observado todo esto, se ve que los resultados obtenidos en el dataset de solo vectores frente al que contiene vectores y métricas, las diferencias en los modelos que tienen mejores puntuaciones en ambos casos son el Random Forest y el XGBoost y que ambos dos clasificadores tienen resultados con diferencias menores a 0,06 en el peor de los casos, es decir, comparando el *recall* del XGBoost en estos dos casos.

Con todo lo expuesto, se concluye que la mejor manera de conformar el dataset entre las tres expuestas, podría ser usando el conjunto de vectores o el de vectores con métricas siempre y cuando se utilicen los clasificadores con mejores puntuaciones. El caso de no utilizar métricas tiene el problema de que conforme el vocabulario contenido en los tweets cambie, será necesario aprender nuevamente el vocabulario, ya que empeorarán los resultados. Por ello la mejor forma, y la elegida en este trabajo, es teniendo en cuenta métricas y vocabulario. Utilizando este formado de dataset se alivia la dependencia a una actualización frecuente del vocabulario asociado a la clase irrelevante, aunque sí es recomendable ya que la forma de expresarse de la gente y los temas sobre los que opina la comunidad de Twitter van evolucionando con el paso del tiempo.

Como trabajo futuro a este trabajo se estudia validar la posibilidad de reducir la dimensionalidad de los vectores de los textos a un valor inferior a 50. En este trabajo se ha visto que pasar de un espacio de 100 dimensiones a 50 no ha comportado rendimientos peores en la etapa final de clasificación, y ha acelerado los procesos de entrenamiento. Sin embargo, una reducción excesiva de la dimensionalidad puede conllevar empeoramientos en los resultados.

Por otro lado, también se debería analizar si las métricas elegidas son las más adecuadas, así como ver la posibilidad de eliminar alguna que proporcione menor separación de clases, por ejemplo, evaluar según el boxplot de la figura 11, si la métrica *statuses count* que indica el número de tweets emitidos por ese usuario selecciona la clase relevante de la que no lo es.

Otra tarea a realizar es el estudio de añadir nuevas métricas como puede ser el *time-stamp* de cada tweet, que puede ayudar a detectar tweets que son repetidos en un espacio de tiempo pequeño como sucede con los memes. A diferencia de incluir el número de retweets, esta métrica podría segmentar que se esté repitiendo un tweet porque los vectores sean iguales o similares, y que la cercanía temporal prediga que se trate de un meme [19].

Una vez seleccionada la manera en la que vamos a confeccionar el dataset, la siguiente tarea es evaluar cuál es el mejor clasificador para resolver el principal objetivo de este trabajo, detectar tweets irrelevantes.

En primer lugar, se ha seleccionado el clasificador de Naïve Bayes como línea base con la que comparar los resultados obtenidos. Como era de esperar según lo visto en la bibliografía no es el mejor clasificador para minería de textos, pero es el más elegido para comparar los resultados de los clasificadores en Machine Learning.

En la figura 13 están representadas las precisiones de todos los algoritmos en cada uno de los ensayos. Se puede ver que la precisión de los clasificadores Random Forest y XGBoost son mejores en los tres ensayos, a excepción del ensayo 1 donde la máquina de Soporte Vectorial con kernel radial supera a estos dos clasificadores. Como se ha visto anteriormente, la forma de construir el dataset con solo métricas no es la mejor, por lo que nos quedaremos con los resultados de los otros dos ensayos, donde ambos clasificadores han obtenido mejor precisión, es decir mejor ratio entre los clasificados correctamente.

La métrica denominada precisión de un clasificador es importante, pero no es la única, ya que la precisión no tiene en cuenta cuantos ha clasificado erróneamente. Como se explicó en el capítulo 3, para eso tenemos el recall. Esta métrica está representada en la figura 14 y se puede ver como en el ensayo de métricas y vectores, los clasificadores SVM obtienen resultados bastante peores (de cuatro a cinco décimas) que los obtenidos por Random Forest y XGBoost.

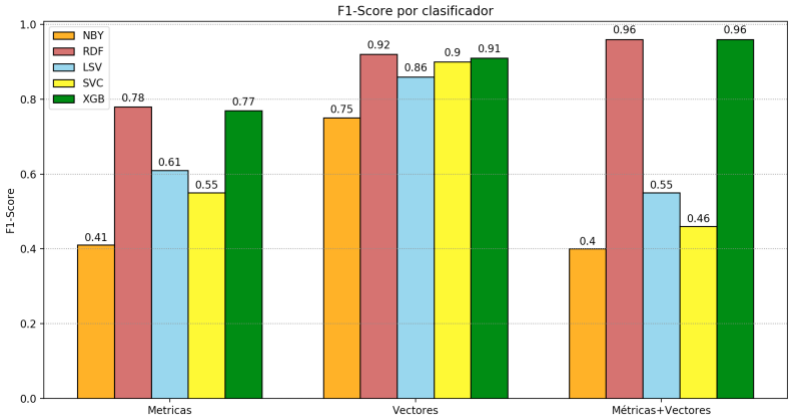


Figura 15: Gráfica F1-Score por clasificador vs ensayo

Con la precisión por un lado y el recall por otro tenemos dos métricas importantes para poder elegir cual es el clasificador que vamos a seleccionar para resolver el problema, pero si se quiere encontrar un equilibrio entre ellas, se puede realizar la media armónica entre ambas métricas y obtenemos la métrica llamada F1-score. Esta métrica está representada para cada clasificador en la figura 15.

De la gráfica de la figura 15 podemos destacar el clasificador SVM con kernel radial, denominado SVC en la leyenda. Este modelo obtiene el peor resultado en el ensayo de métricas y vectores. Este resultado está reforzado por la ratio de Falsos positivos obtenidos en los tres ensayos y que se ilustran en la figura 16. Esta ratio se ha obtenido de realizar la fracción entre los Falsos Positivos de la clase 2, entre las muestras de la clase 2 en cada ensayo.

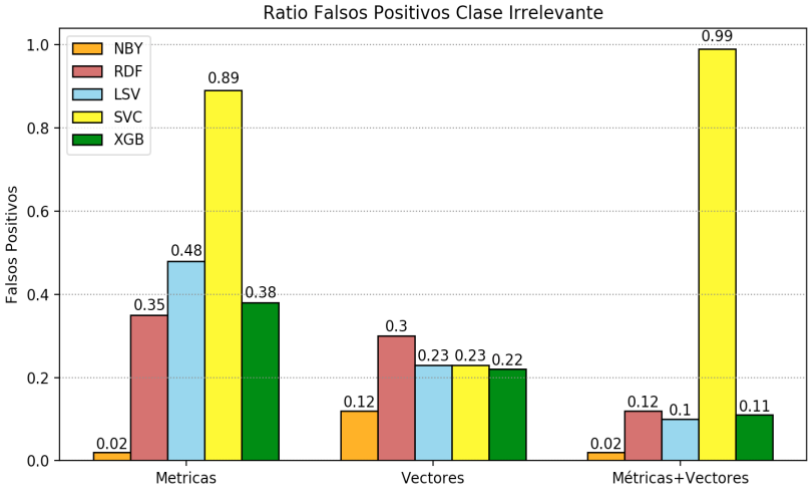


Figura 16: Gráfica ratio Falsos positivos en clase Irrelevante

Como resumen a los resultados expuestos se puede concluir que los métodos combinados dan los mejores resultados. Ambos clasificadores, Random Forest y XGboost, han conseguido valores muy parecidos en todas las métricas, por lo que ambos serían válidos para cumplir con la tarea de clasificación de este trabajo.

Capítulo 5

Conclusiones y Trabajo futuro

5.1 Conclusiones

Para cualquier problema de Machine Learning lo primero que se necesita es disponer de lo que se conoce como "ground truth", o lo que es lo mismo, disponer del dataset etiquetado con la clase a la que pertenece cada una de las muestras.

Una de las tareas más costosas de este trabajo ha sido conseguir etiquetar el conjunto de datos de trabajo, ya que a priori solamente se disponían de los tweets sin etiquetar. El método propuesto en el capítulo 3 para etiquetar el dataset es un proceso semi-supervisado que aumenta proporcionalmente su coste cuanto más grande es el conjunto de datos a etiquetar.

Los resultados obtenidos en los ensayos son bastante buenos, aunque podrían estar influidos por varios factores que pueden hacer dudar de los mismos:

- La forma en la que se ha construido el dataset deja fuera muchos tweets que sean diferentes a los demás, o unitarios en forma y vocabulario. El dataset actual contiene tweets etiquetados por alguna cadena de texto que se ha visto en repetidas ocasiones dentro del dataset.
- Sobre ajuste producido por el vocabulario inferido con la misma colección de testeo. El vocabulario ha sido generado en una etapa anterior a inferir los vectores de todo el conjunto de datos que después sería dividido en un conjunto de entrenamiento y otro de evaluación tal como se ha explicado en el capítulo 4.

Por el contrario, hay otras evidencias que permiten confiar en los resultados obtenidos. En primer lugar, las métricas por si solas ya discriminan bastante bien las clases, tal y como se vio de antemano en el análisis de las métricas, y su posterior ensayo con el dataset de solo métricas. En segundo lugar, el conjunto de entrenamiento ha sido realizado mediante validación cruzada, lo cual debería reducir el problema del sobreajuste anteriormente mencionado.

Como ya se ha dicho, el método propuesto para conseguir el objetivo de este trabajo debe conllevar una tarea de reentrenamiento del algoritmo donde se amplíe también el vocabulario para ir adaptándose a las nuevas formas de expresarse y las nuevas temáticas que se están comentando en la red.

5.2 Trabajo futuro

A corto plazo, sería necesario realizar un estudio más exhaustivo utilizando métodos no supervisados en conjuntos muy grandes de tweets, de forma similar al trabajo de [19].

También sería interesante entrenar y validar los modelos de aprendizaje en periodos temporales muy separados. De esta forma, podría evaluarse cuánto ha generalizado el algoritmo propuesto, a la hora de obtener patrones de las métricas de usuarios y de las relaciones de nuevos memes.

Las opciones para valorar cómo de bien generaliza el algoritmo serían dos. Empezaremos describiendo la primera y más ortodoxa, aunque también más costosa:

1. Repetir el proceso descrito en el capítulo 3 para etiquetar el dataset.
2. En paralelo al proceso de etiquetado se podrían inferir los vectores de los nuevos tweets utilizando el vocabulario aprendido para el conjunto de datos de este trabajo no el del nuevo dataset.
3. Una vez tengamos etiquetado el dataset y los vectores inferidos, construiríamos el nuevo dataset con las nuevas métricas del usuario que ha publicado cada tweet y los vectores de estos tweets.
4. Con el nuevo dataset construido, entrenaríamos el algoritmo con todo el banco de datos del trabajo actual y pasaríamos a predecir las clases del nuevo dataset.
5. Por último, solo quedaría evaluar los resultados con las métricas descritas en el capítulo 3.

La segunda opción, y menos costosa, aunque no completa, sería evaluar cuanto ha generalizado el algoritmo de este trabajo centrándonos únicamente en los tweets detectados como irrelevantes:

1. Inferir los vectores del nuevo dataset con el vocabulario del conjunto de datos del presente trabajo.
2. Construir el nuevo dataset con las métricas y vectores, tal y como se ha explicado en el método anterior.
3. Una vez construido el dataset, proceder a clasificar los nuevos tweets en las clase relevante o irrelevante. Con los tweets de esta última clase que es la clase de interés a detectar en el trabajo, realizaríamos una inspección manual únicamente de los tweets obtenidos en la clase objeto de estudio.

Este último método tiene el inconveniente que vamos a perder todos los tweets que se hayan clasificado erróneamente en la clase 1 (relevante) y pertenezcan a la clase 2 (irrelevenate). Esto significa que perderíamos todos los Falsos Positivos y no podríamos calcular ni la precisión ni el accuracy del clasificador en este nuevo escenario, y solamente podríamos calcular el recall, ya que en las anteriores métricas es necesario disponer de la información de los Falsos Negativos.

Otro punto que se podría considerar es la evaluación de cuánto aporta cada una de las métricas a la clasificación, y ver si otra información puede ayudar a la tarea de clasificación de memes, por ejemplo, añadir el *time-stamp* de cada tweet podría ayudar en la detección de memes como se mencionó anteriormente.

Referencias

[1] R. Berlanga, L. García-Moya, V. Nebot, M. José Aramburu, I. Sanz, D. María Llidó: SLOD-BI: An Open Data Infrastructure for Enabling Social Business Intelligence. IJDWM 11(4): 1-28 2015

[2] https://hootsuite.com/ (Último acceso verificado: 8 de agosto de 2018)

[3] J.Huang, KM. Tohornton EN EFthimiadis Conversational Tagging in Twitter. Proceedings of the 21st ACM conference on Hypertext and hipermedia Jun. 2013

[4] https://developer.twitter.com/en/docs (Último acceso verificado: 8 de septiembre de 2018)

[5] https://developer.twitter.com/en/apply/user (Último acceso verificado: 8 de septiembre de 2018)

[6] https://developer.twitter.com/en/docs/twitter-for-websites/log-in-with-twitter/guides/implementing-sign-in-with-twitter.html
(Último acceso verificado: 8 de septiembre de 2018)

[7] A. Sanzgiri, A. Hughes and S. Upadhyaya. Analysis of Malware Propagation in Twitter. IEEE 32nd International Symposium on Reliable Distributed Systems. Oct. 2013

[8]A. Aggarwal, A. Rajadesingan, P. Kumaraguru. PhishAri: Automatic Realtime Phishing Detection on Twitter. Oct. 2012

[9] Abdullah Talha Kabakus, Resul Kara. A Survey of Spam Detection Methods on Twitter. International Journal of Advanced Computer Science and Applications, Vol. 8, No. 3, 2017

[10] C.D. Gowri, V. Mohanraj, A Survey on Spam Detection in Twitter: A Review, Int. J. Comput. Sci. Bus. Informatics. 14 (2014) 92–102. http://ijcsbi.org/index.php/ijcsbi/article/view/418.

[11] Y. He, H. Saif, Z. Wei, K. Wong Quantising Opinions for Political Tweets Analysis. Aug 2014

[12] C. O'Neil and R. Schutt: Doing Data Science. Straight talk from the frontline. Editorial O'Reilly. Oct. 2013

[13] C. Yang, R. Harkreader, G. Gu: Empirical Evaluation and New Design for Fighting Evolving Twitter Spammers, IEEE Trans. Inf. Forensics Secur. 8 2013

[14] R. Dawkins. El gen egoísta. Editorial: Oxford University Press. Jul. 1976

[15] G. Pérez, A. Aguilar y M.Ernestina. El meme en internet. Usos sociales, reinterpretación y significados, a partir de Harlem Shake. Aug. 2014

[16] D. Kempe, J. Lleinberg, E. Tardos. Maximizing the spread of influence through a social network. ACM SIGKDD international conference on Knowledge discovery and data mining. Aug 2003

[17] M. Coscia. Competition and Success in the Meme Pool: a Case Study on Quickmeme.com (Apr. 2013)

[18] L.K. Hansen, A. Arvidsson, F.A. Nielsens, E. Colleoni, M .Etter: Good Friends, Bad News Affect and Virality in Twitter. 6th International Conference, FutureTech Jun. 2011

[19] M. JafariAsbagh, F. Menczer, E. Ferrara, O. Varol, A. Flammini. Clustering memes in social media. 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013). Aug.2013

[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. CoRR, abs/1301.3781, 2013.

[21] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In EMNLP, pages 1532–1543. ACL, 2014.

[22] Y. Bengio, R. Ducharme, P. Vincent. A neural probabilistic language model. Journal of Machine Learning Research, 3:1137-1155, 2003.

[23] T. Mikolov, W.T. Yih, G. Zweig. Linguistic Regularities in Continuous Space Word Represen- tations. NAACL HLT 2013.

[24] T. Mikolov, Distributed Representations of Sentences and Documents. International Conference on Machine Learning. 2014.

[25] M. Kusner , Y. Sun, N. Kolkin, and K. Weinberger. From word embeddings to document distances. In International Conference on Machine Learning (pp. 957-966).] Jun. 2015.

[26] G. Huang, C.Guo, Y.Kilian, M.J.Kusner and F.Sha. Supervised Word Mover's Distance. Neural Information Processing Systems Conference. Dic. 2016

[27] M.Kanakaraj, R.Mohana and R.Guddeti NLP Based Sentiment Analysis on Twitter Data Using Ensemble Classifiers. 2015 3rd International Conference on Signal Processing, Communication and Networking. Mar. 2015

[28]M.Buuch, I. Lee and T. Wu. NLP-based Approach to Twitter User Classification

[29] A.Hotho, A.Nürnberger, G. Paass. A Brief Survey of Text Mining. Journal for Computational Linguistics and Language Technology. May. 2005

[30] M. Fernández-Delgado, E. Cernadas and S. Barro. Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? Journal of Machine Learning Research. Oct.14

[31] Nebot, V., Rangel, F., Berlanga, R., & Rosso, P. Identifying and Classifying Influencers in Twitter only with Textual Information. In International Conference on Applications of Natural Language to Information Systems (pp. 28-39). Springer, Cham. Jun. 2018

[32] T.Chen and C.Guestrin. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Aug. 2016.

[33] J. VanderPlas. Python Data Sciende Handbook. Essential Tools for working with data. Editorial O'Reilly. Dec. 2016

[34] R. Aznar, R. Del Hoyo and M. del Carmen Rodríguez. Towards a Structured Representation of Results in an Information Retrieval System for Public Examination Calls. Proceedings of the 5th Spanish Conference on Information Retrieval. Jun. 2018

[35] Á.F. Godoy. Técnicas de aprendizaje de máquina utilizadas para la minería de texto. Investig. bibl vol.31 no.71 México Apr. 2017

[36] A. C. Müller and S.Guido. Introduction to Machine Learning with Python. A guide for data scientists. Editorial O'Reilly. Oct. 2016.

[37] https://pandas.pydata.org/ (Último acceso verificado: 8 de septiembre de 2018)

[38] https://matplotlib.org/ (Último acceso verificado: 8 de septiembre de 2018)

[39] http://scikit-learn.org (Último acceso verificado: 8 de septiembre de 2018)

[40] https://radimrehurek.com/gensim/ (Último acceso: 8 de septiembre de 2018)

[41] https://www.nltk.org/ (Último acceso verificado: 8 de septiembre de 2018)

[42] https://www.datacamp.com/community/tutorials/wordcloud-python
(Último acceso verificado: 8 de septiembre de 2018)

[43] O. Irsoy, O. Taner, and E. Alpayd. Design and Analysis of Classifier Learning Experiments in Bioinformatics: Survey and Case Studies. IEEE/ACM Transactions On Computational Biology And Bioinformatics. Nov. 2012

[44] D. Krstajic, L.J. Buturovic, D. E Leahy and S. Thomas. Cross-validation pitfalls when selecting and assessing regression and classification models. Journal of Cheminformatics. Jan. 2014

[45] B.Klimt, Y.Yang. The Enron Corpus: A New Dataset for Email Classification Research. European Conference on Machine Learning. Sep. 2004