Análisis de sentimientos de datos de redes sociales usando técnicas de Machine Learning

Gerard Ramos Gambús

Resumen– Hoy en día hay mucha diversidad de comentarios en las redes sociales y no hay un control sobre qué tipo de opiniones se permiten publicar. Es por eso que este documento resume la aplicación de un sistema de análisis de sentimientos a las redes sociales de Twitter y Facebook. Este proyecto empieza con el uso de un algoritmo que, habiendo sido previamente entrenado con un Corpus, sea capaz de descargar un conjunto de *posts* de dichas redes sociales y asignar un valor de polaridad junto con una etiqueta positiva, negativa o neutra a cada texto descargado. Como objetivo final se diseñará un entorno web donde el usuario podrá indicar sobre qué red social y qué tipo de análisis quiere hacer.

Palabras clave– Python, Flask, Big Data, Machine Learning, Naive Bayes, Análisis de sentimientos, Twitter, Facebook, Web, API, Librerías

Abstract– Nowadays there is a lot of comment diversity on social media and there is no control about what type of opinions can be posted. This is why this paper summarizes the application of a sentiment analysis system to the social media of Twitter and Facebook. This project begins with the use of an algorithm that, having been trained with a Corpus, is able to download a set of *posts* of said networks and assign a polarity value and a positive, negative or neutral tag to each text downloaded. As a final goal, a web environment will be set up in which the user will be able to indicate on which social media and what type of analysis they want to carry out.

Keywords– Python, Flask, Big Data, Machine Learning, Naive Bayes, Sentiment Analysis, Twitter, Facebook, Web, API, Libraries

1 Introducción

En los últimos años el uso de las redes sociales ha tenido un incremento exponencial. Este hecho ha ido acompañado de la situación epidemiológica en la que nos encontramos, la cual ha provocado que, debido al aislamiento, muchas personas utilicen las redes sociales como pasatiempo.

La figura 1 muestra el incremento comentado previamente 1:

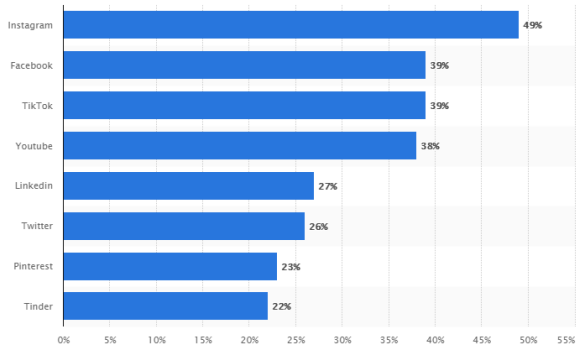

Fig. 1: Incremento del uso de las redes sociales durante la cuarentena en España en 2020.

Viendo esto, se debería tener en cuenta qué tipo de información o comentarios escriben los usuarios de las redes sociales, ya que hay mucha diversidad de opiniones, y lo que no se busca es fomentar el odio o cualquier otro tipo de

• E-mail de contacte: gambusrg@gmail.com
• Menció realitzada: Tecnologies de la Informació
• Treball tutoritzat per: Diego Mauricio Freire Bastidas (TIC)
• Curs 2020/21

conducta hiriente hacia personas, etnias, religiones o razas.

El propósito general de este trabajo es evidenciar un análisis de comentarios y opiniones que escriben usuarios en diversas redes sociales, poniendo como objetivo Twitter y Facebook.

Para conseguir realizar este análisis se utilizará el lenguaje de programación Python, utilizando un conjunto de librerías que se explicarán posteriormente. Dependiendo de la red social podremos acceder a los datos a través de la API (Application Programming Interface) o una librería.

2 ESTADO DEL ARTE

Actualmente el Machine Learning es una de las disciplinas científicas con mayor uso. ¿Pero realmente qué entendemos por Machine Learning? Aprender en el contexto de la Inteligencia Artificial quiere decir identificar y prever patrones futuros en conjuntos de datos. Esto implica que los sistemas mejoran de forma autónoma sin la intervención humana. Existen varios tipos de aprendizajes:

- Aprendizaje supervisado: Se entrena al algoritmo a través de conjuntos de datos previamente etiquetados con la respuesta correcta. Cuanto mayor sean estos conjuntos de datos, el algoritmo más aprenderá sobre el tema.
- Aprendizaje no supervisado: Se entrena al algoritmo con datos sin etiquetar. El objetivo es que el mismo encuentre patrones que le ayuden a entender el conjunto de datos.
- Aprendizaje por refuerzo: El algoritmo se entrena observando el mundo que le rodea. Su información de entrada es el feedback o retroalimentación que obtiene del mundo exterior como respuesta a sus acciones. Por lo tanto, el sistema aprende a base de ensayo-error.

Dentro del aprendizaje supervisado hay que destacar el término: clasificador. Los sistemas clasificadores predicen la categoría de unos datos de entrada gracias al aprendizaje previo. Hoy en día hay varios modelos clasificadores que son usados para el análisis de sentimientos de textos. Estos modelos utilizan algoritmos para asignar un valor o una etiqueta a un texto en función de su polaridad, entendiendo como polaridad el valor que determina como de positivo o negativo es el texto analizado.

Durante los últimos años varios autores han realizado modelos de análisis de sentimientos en redes sociales. En 2018, Lukas Kohorst, después de crear un bot de Twitter[2] utilizando Tweepy (una librería de Python) quiso ver cómo la gente percibía dicho bot, así que creó un programa que analizaba los tweets de los usuarios. Para asignar una polaridad a cada tweet, utilizó la librería TextBlob [3], la cual proporciona una API para realizar tareas de procesamiento de lenguaje natural como extraer nombres de un texto, asignar tags a las palabras, realizar análisis de sentimientos, clasificar y traducir. Para determinar dicha polaridad, TextBlob usa el algoritmo *Naive Bayes*.

En 2019, Shuamik Daityari creó otro programa capaz de analizar los sentimientos de un conjunto de tweets [4]. Lo hizo ya que hoy día se generan muchos datos no estructurados, por lo que se necesita un procesamiento para generar información. A diferencia de Lukas Kohorst, él usó la conocida librería de Python NLTK (Natural Language Toolkit), otra librería de procesamiento de lenguaje natural. Ésta también incorpora el algoritmo de *Naive Bayes* que es el que usó Daityari para poder clasificar y polarizar los tweets de los usuarios. En ese mismo año, Anas Al-Masri publicó un artículo donde exponía su analizador de sentimientos escrito en Python usando un clasificador, concretamente el de *Naive Bayes*[5]. Las librerías que utilizó fueron **Twitter, NLTK y re** (regular expresión operations, librería usada para encontrar patrones en el texto o para realizar operaciones sobre él). El autor se centró en cinco fases para obtener un correcto clasificador:

1. Preparar el entorno obteniendo las credenciales necesarias y tener la autenticación del script de Python para poder descargar la información a través de la API de Twitter.
2. Preparar el conjunto de entrenamiento descargando un corpus que contiene cinco-mil tweets clasificados manualmente como positivo, negativo, neutro o irrelevante.
3. Preprocesar los tweets en los DataSets. Para poder utilizar el algoritmo Naive Bayes necesitaba tener los tweets procesados de forma que el algoritmo entienda qué recibe.
4. Utilizar el algoritmo Naive Bayes para entrenar el modelo.
5. Testear el modelo

Con todo esto que tenemos en cuenta, hemos decidido implementar nuestro modelo clasificador, basado en aprendizaje supervisado. En las siguientes secciones se presentará la Selección del clasificador, donde se expondrá qué clasificador y librerías utilizaremos. Seguidamente se expondrán los objetivos y el desarrollo del proyecto. Después la polaridad, donde se argumentará qué se tiene en cuenta para etiquetar los datos para su clasificación. Los siguientes apartados harán referencia a cómo se estructuran, guardan y visualizan los datos. Seguidamente se presentarán las restricciones que han aparecido a lo largo del proyecto. Por último, se hará una breve explicación sobre la virtualización del proyecto, la metodología y la planificación, además de las conclusiones y la bibliografía.

3 METODOLOGÍA

La realización de este proyecto se ha llevado a cabo siguiendo la metodología agile Scrum. El proyecto se ha dividido en Sprints, los cuales tienen una duración de dos semanas cada uno. Cada semana se hace una breve reunión (Sprint meeting) para poner puntos en común y asegurarnos que se sigue un buen ritmo de trabajo. En estas reuniones se expone la evolución del proyecto, dando a conocer los avances y problemas que han ido apareciendo. Estas reuniones se han realizado a través de videollamada. Al final de cada Sprint se realiza una retrospectiva para comprobar que el grado de madurez del proyecto ha aumentado y se define el siguiente Sprint.

Para el control de versiones del código se ha utilizado Git.

4 OBJETIVOS

El objetivo global del proyecto es obtener un conjunto de textos de varias redes sociales para posteriormente generar unos *dashboards* recopilando la información extraída y presentándola de una forma clara y simplificada. Dicha información a representar será un análisis de sentimiento de los textos extraídos de las redes sociales de Twitter y Facebook. Los objetivos principales del proyecto son:

- Seleccionar un entorno y un lenguaje de programación adecuados. Dada la amplia variedad de lenguajes, se ha escogido Python ya que es el más completo y el que tiene mayor documentación, junto con una comunidad muy activa.
- Obtener las APIs de las redes sociales para poder acceder a través de ellas y recuperar el conjunto de *posts* a analizar.
- Realizar la extracción de textos de ambas redes sociales dada una palabra o usuario en el caso de Twitter y una página pública en el de Facebook. Gracias a la API obtenida o la librería será posible obtener el DataSet de textos.
- Depurar del contenido extraído de las redes sociales, es decir, eliminar los caracteres sobrantes y especiales para que dicho texto quede bien estructurado para que sea analizado. Para ello se utilizarán expresiones regulares.
- Escoger un algoritmo de clasificación. Se dispone de varios algoritmos para crear el modelo los cuales son *Naive Bayes, Support-Vector Machine, K Nearest Neighbours y Decision Trees*, pero se ha seleccionado *Naive Bayes* porque se ha estudiado previamente en la carrera y ha generado un particular interés y es el que mejor se adapta a las estructuras de datos que se han pensado utilizar.
- Crear un modelo capaz de determinar si un texto es positivo, negativo o neutro. Para ello se tendrá que hacer un *Train y Test* del modelo con un conjunto de textos para determinar la precisión de su clasificación y posteriormente proceder al análisis de la información descargada. Para clasificar los textos se usará la polaridad. Esta se refleja en un valor que determina cómo de positivo o negativo es el texto analizado. Según dicho valor se le asignará una etiqueta de sentimiento, la cual será Positivo, Negativo o Neutro.
- Mostrar la información en un conjunto de *dashboards* en una página web. En dicha web se debe poder seleccionar la red social sobre la que se quiere hacer el análisis e ingresar un término. Seguidamente se deben visualizar un conjunto de gráficos sintetizando el análisis de sentimientos y mostrándolo de una forma simplificada.

5 DESARROLLO

Expuesto lo anterior, este proyecto busca integrar las redes sociales de Twitter y Facebook, realizar un análisis de sentimientos de textos contenidos en dichas redes sociales y posteriormente hacer una visualización simplificada de los resultados. Para realizar la clasificación se necesita un Corpus de datos de entrenamiento en español. El inconveniente es que hay pocos. Para ello se considerarán varias opciones:

1. La primera de ellas será utilizar un Corpus en inglés y traducir los textos para realizar el entrenamiento, ya que los DataSets en inglés son mucho más completos que los que están en español. Esta traducción se haría mediante la librería TextBlob.

2. Como segunda consideración se plantea utilizar un DataSet en español, teniendo en cuenta que no es la opción más fiable ya que no son demasiado extensos y no son muy completos. Por otro lado, se ha contemplado realizar directamente nuestro propio DataSet de entrenamiento en español, aunque esto ocuparía una gran parte de tiempo debido a que hay que clasificar manualmente un conjunto amplio de textos.

3. Como última opción se tendrá en cuenta la API de Google de análisis de sentimientos, ya que es la más actualizada, debido al volumen de información de la que dispone, y aporta una seguridad indiscutible.

5.1. Librerías

En el caso de Twitter, las librerías que se usarán en el proyecto serán **Tweepy**[6] y **Pandas**[7] como las más relevantes. Tweepy es necesaria para poder acceder y utilizar la API de dicha red social y Pandas es la librería con más documentación y más utilizada para la manipulación de Dataframes, que son unas estructuras de datos que se comentarán más adelante.

En el caso de Facebook se utilizará la librería **Facebook-scraper**[8], la cual permite extraer información sin necesidad de usar la API.

Ambas redes sociales compartirán el uso de las librerías **NLTK**, y **re**, entre otras, ya que son las más completas. Además, NLTK incorpora demostraciones gráficas y datos de muestra. Por otro lado, **re** es la librería con mayor documentación y con la comunidad más activa sobre el uso de patrones en textos.
Para realizar el clasificador, se tienen que tener claros dos conceptos, *Train y Test*.

5.2. Traducción de textos

Como se ha comentado previamente, desde un inicio del proyecto se ha buscado realizar un clasificador en español. Dada la inexistencia de un Corpus en español con las condiciones adecuadas, se ha añadido una fase de traducción de textos.

- En primera instancia se intentó traducir los textos mediante la librería TextBlob, la cual permite realizar tareas de Procesamiento del Lenguaje Natural como análisis morfológico, extracción de entidades, análisis de opinión, traducción automática, etc. Esta opción no dio buenos resultados debido a problemas de formato y codificación de los textos.

- Como segunda opción se realizó una búsqueda exhaustiva de conjuntos de datos de entrenamiento en español, pero en comparación con la gran cantidad de datos en inglés que había no era la mejor forma de conseguir crear un clasificador eficaz, fiable y consistente.

- Por último, se volvió a intentar realizar la traducción de textos mediante una API de Google llamada **`google_trans_new`**[9], la cual funciona de la forma esperada. A pesar de su correcto funcionamiento, surgieron algunos problemas, por ejemplo, que dicha API traduce un conjunto máximo de 5000 caracteres a la vez, y nuestro conjunto de entrenamiento dispone de muchos más. Para solventar esto, la ejecución de la traducción de los textos se ha tenido que realizar texto por texto, lo cual ha supuesto otro problema como es el bloqueo de la IP durante unas horas, esto sucede ya que se realizan muchas peticiones a través de la API. Para lidiar con esto, se ha utilizado al librería **time** [10], la cual nos permite en cada iteración de cada texto realizar un time.sleep (suspender la ejecución del hilo de llamada durante el número de segundos especificado) para no saturar la API.

5.3. Train

La parte de Train se realiza procesando el conjunto de datos de entrenamiento en inglés que hemos descargado y posteriormente traducido, y luego con los datos preparados, utilizar el algoritmo para que nuestro clasificador se entrene. La figura 2 muestra la parte simplificada del *Train*:

Fig. 2: Training del clasificador

Como se puede ver, la fase de *Train* consta de cinco subfases que se indagará más sobre ellas en los siguientes apartados.

El preprocesamiento de textos obtenidos del Corpus ha sido realizado como muestra la figura 3.

Fig. 3: Preprocesamiento de datos

A continuación, la figura 4 muestra en detalle el preprocesamiento de los textos. Este algoritmo recibe como ***input*** los textos sin procesar, tanto para el entrenamiento del modelo o para calcular la polaridad. Una vez se inicie el proceso de limpiar los textos, obtendremos como ***output*** dichos textos sin ruido y ni caracteres especiales.

Algorithm 2: Preprocesamiento de datos

```
   Input  : Data en crudo
   Output: Data limpia y sin ruido
   /* Leer data en crudo                                   */
1  begin Limpiar data
2      Eliminar caracteres especiales;
3      Eliminar hyperlinks;
4      Eliminar usernames;
5      Tokenize texto;
6      Eliminar stopwords y signos de puntuacion;
7      Get stem;
       /* stem : Simplifica la palabra a su raiz          */
8  end
```

Fig. 4: Preprocesamiento de datos

En cuanto se tengan los datos preprocesados se podrá realizar el entrenamiento del modelo. La figura 5 muestra cómo se ha realizado. Este proceso tiene como ***input*** un Corpus con datos preprocesados los cuales han sido obtenidos de un repositorio, se han traducido y posteriormente se han preprocesado para esta fase. Como ***output*** del proceso se obtendrá el modelo entrenado.

Algorithm 1: Entrenamiento Naive Bayes Classifier

```
   Input  : Corpus con datos preprocesados
   Output: Naive Bayes Classifier entrenado
1  begin Entrenamiento
2      Calcular palabras únicas;
3      Calcular positividad y negatividad de las palabras;
4      Calcular frecuencia de palabras;
5      Calcular variables de probabilidad;
6  end
```

Fig. 5: Entrenamiento del modelo

Una vez se haya entrenado el modelo se procederá a realizar el *Test*.

5.4. Test

La fase *Test* consiste en descargar un conjunto de textos para que sean analizados y, posteriormente, dar un valor de polaridad calculando el sentimiento de cada uno de ellos. Este proceso se realizará gracias a la API de Twitter y a la librería comentada previamente que pemitirá obtener la data de Facebook sin el uso de su API.

Esta parte de *Test* se ha realizado de una forma mucho más simple. El procesamiento de los textos es el mismo que en el *Train* para ambas redes sociales (excluyendo la traducción, ya que la búsqueda de tweets y *posts* de Facebook se realiza en español), pero la fuente de estos son los usuarios de Twitter y páginas públicas de Facebook. La ejemplificación de esta fase se muestra en la figura 6.

Para comprender mejor cómo se ha implementado el *Test* se ha incorporado la figura 7. Este algoritmo recibe como ***input*** un término que introducirá el usuario. Según la red social sobre la que queramos realizar el análisis será una palabra, hashtag, un usuario (en caso de Twitter) o una página pública de Facebook. Y como **output** se obtendrá un conjunto de DataFrames con textos ya clasificados.

Si la red social sobre la cual se realizará el *Test* es Twitter, es imprescindible que antes se obtengan de las credenciales

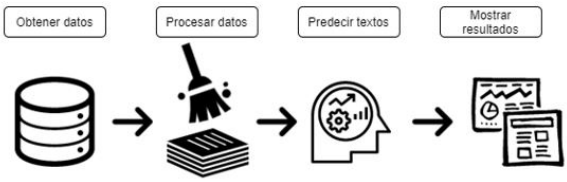

Fig. 6: Test de los textos descargados

Algorithm 3: Testing

Input : Palabra de interés
Output: DataFrames con textos clasificados

```
begin Testing
    if Twitter then
        Obtener crednciales de la API de Twitter;
        Descargar tweets;
    else
        Preparar librería Facebook;
        Descargar posts de Facebook;
    end
    Preprocesamiento de textos;
    Predecir textos;
    Generar DataFrames;
end
```

Fig. 7: Test del clasificador

de la API ya que sino no se podrá utilizar. Para obtener estas credenciales previamente se ha tenido que crear una cuenta de tipo desarrollador de Twitter y realizar la autenticación mediante OAuth[11] y el access token, y así poder tener acceso a los datos públicos de Twitter.

Una vez se descarguen los datos se tendrán que preprocesar, como se ha mostrado en la figura 4. Posteriormente, cuando los datos estén limpios estarán listos para ser clasificados.

En el momento en que se disponga de los textos clasificados y agrupados en DataFrames estarán listos para que sean mostrados en la página web.

5.4.1. Obtención de datos de Twitter

Cuando se realiza la extracción de datos de Twitter, previamente se ha tenido que crear una cuenta con permisos de desarrollador. Esto es algo obligatorio ya que gracias a esa cuenta será posible obtener las credenciales necesarias para poder utilizar la API de Twitter. El fragmento Code 1 muestra cómo a través de la API se extraen los datos, en este caso referentes a la búsqueda por usuario.

```
import tweepy as tw
api = getApi()
tweets_fetched = api.user_timeline(
    screen_name=search_keyword,
    since=start_date, until=end_date,
    count=new_num, include_rts=False)
```

Code 1: Extracción de *posts* de un usuario de Twitter

5.4.2. Obtención de datos de Facebook.

La forma de extraer datos de Facebook es parecida a la de Twitter a excepción del uso de la API. En este caso se utiliza la librería `facebook_scrapper`. Una vez se hayan obtenido los textos ya se podrá proceder a relizar la limpieza y la predicción. La ejemplificación del método se muestra en el Code 2.

```
from facebook_scraper import get_posts
return [{"text": status['text'],
    "label": None, "likes":
    status['likes'], 'comments':
    status['comments_full'],
    'date':None, 'id': '2F' +
    status['post_id']} for status in
    get_\textit{posts} (search_term,
    pages=num, options={"comments":
    True})]
```

Code 2: Extracción de *posts* de Facebook.

6 POLARIDAD

Para determinar la polaridad de un texto se usará la probabilidad de *Naive Bayes Classifier*, mostrada en la figura 8.

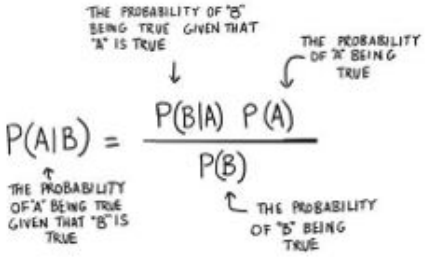

Fig. 8: Naive Bayes Classifier

La idea general es determinar las palabras claves de un texto y comprobar si cada una de estas tienen una connotación negativa o positiva. Por ejemplo con el texto “Hoy es un buen día”:

- P(texto sea positivo | [‘hoy’,’buen’,’día’])
- P(texto sea negativo | [‘hoy’,’buen’,’día’])

Para determinar dicha probabilidad se usará un diccionario de frecuencias que contendrá las palabras clave que aparecen en el Corpus de entrenamiento. Cada palabra tendrá asociado el número de veces total que aparece y la connotación de ella según los textos en los que se encuentren. Por ejemplo, si la palabra ”buen” aparece dos veces en textos positivos y en el diccionario hay cuatro palabras, se podrá calcular la probabilidad de que esta palabra clave esté en un texto positivo o negativo, dividiendo el recuento negativo o positivo del diccionario entre las veces que aparece en los textos a clasificar.

- P(‘buen’ en un texto positivo) = 2/4 = 50 %
- P(‘buen’ en un texto negativo) = 0/4 = 0 %

Podemos concluir en este caso que la palabra ”buen” tiene una probabilidad más alta de estar en un texto postivo que en uno negativo. A partir de este punto, solo se tendrá que tener en cuenta la probabilidad de todas las palabras de un texto para determinar si este es positivo o negativo.

7 Estructuras de datos

Para realizar el procesamiento de los textos se han utilizado listas, las cuales contienen los *posts* de entrenamiento. Cuando se acabe de realizar esta fase, dicha lista contendrá las palabras claves de cada *post*, sin ningún tipo de ruido ni caracteres no necesarios. Para entrenar el modelo se requiere tener dos tipos de datos:

- La data separada en dos partes, las palabras clave de cada texto y su correspondiente *label*.
- El diccionario de frecuencias comentado previamente. Dicho diccionario es una estructura de dos tuplas, una de (clave-valor)-valor, la cual contiene la palabra y el número de veces que aparece y el valor asociado a la polaridad del texto en la que aparece.

Para hacer el *Test* del modelo se utilizan listas, pero a la hora de mostrar los textos se necesitará operar estructuras que contengan dos columnas, el *post* descargado y su clasificación asignada. Esta estructura se ha guardado en formato DataFrame utilizando la librería Pandas.

8 Uso de base de datos

En un principio se usó una base de datos como método de almacenamiento. Hubo dos factores clave en el momento de determinar su uso:

- Conjunto de datos de entrenamiento.
- Resultados de los análisis

Para realizar el entrenamiento del clasificador se utiliza un conjunto amplio de textos los cuales se descargan de un repositorio de Github y posteriormente se traducen. El planteamiento inicial fue alojar el Corpus de entrenamiento en la base de datos, pero debido a que este Corpus se descargaba en formato JSON [12] y solo se tenía que realizar una vez para tener el modelo entrenado, no se vio necesidad de seguir con el almacenamiento a través de una base de datos. La figura 9 muestra la conversión de datos JSON.

Fig. 9: Procesamiento de datos JSON a DataFrame

Por otro lado, debido a que la extracción y la clasificación se realiza a tiempo real y no es un objetivo del proyecto almacenar las extracciones de datos sobre cada uno de los diferentes análisis realizados, no es lo más óptimo ya que la presentación de resultados se realiza sobre el término, usuario o página pública introducida en aquel momento, y no sobre datos que ya previamente han sido clasificados.
El diseño del proyecto, permitirá en caso de que el clasificador sea ampliado y se quieran implementar funciones de generación de *dashboards* a partir de un histórico de datos, una integración ágil de una base de datos.

9 Restricciones

Como se ha comentando, durante el desarrollo del proyecto han aparecido varios problemas relacionados con la traducción, pero estos no han sido los únicos con los que se ha tenido que lidiar.
En Twitter y Facebook varias restricciones han provocado la modificación de la extracción de los datos, entre los cuales podemos destacar:

- **Twitter:** En un principio realizaron pruebas incrementales de extracciones de textos y se vio que a partir de 900 peticiones se restringe el acceso a la API. Este problema es debido a que se usa la versión gratuita la cual tiene un límite establecido de 900 peticiones cada 15 minutos. Dada esta situación se ha tenido que implementar la búsqueda por intervalos teniendo en cuenta el último ID del tweet extraído para poder empezar el siguiente intervalo de extracción a partir de dicho ID.
- **Facebook:** Debido a que en 2018 entró en vigor la GDPR (General Data Protection Regulation) [13], solo ha sido posible extraer datos utilizando páginas públicas. Esta ley es la más completa jamás introducida y cambia por completo la forma en que se pueden utilizar los datos personales de los usuarios. En referencia al web scraping, que es el proceso por el cual se extrae información de usuarios de Facebook, es necesario un documento por escrito con el permiso para extraer los datos.

El hecho de tener que extraer información de páginas públicas no ha simplificado las cosas ya que Facebook, al igual que Twitter, si hay un exceso de peticiones en un cierto periodo de tiempo bloquea la IP.

10 Prueba de concepto

Uno de los puntos clave que se ha querido implementar en este proyecto ha sido el poder alojarlo en un servidor para que el clasificador pueda ser utilizado desde cualquier máquina. Para ello se ha optado por una prueba de concepto la cual requiere una virtualización. Aunque es un paso a realizar, la virtualización va fuera del foco del proyecto por lo cual no se hace un análisis profundo de las ventajas y desventajas que conlleva.
La figura 10 muestra el fundamento básico de una virtualización, poder alojar diferentes máquinas virtuales con diferentes sistemas operativos en un mismo ordenador.

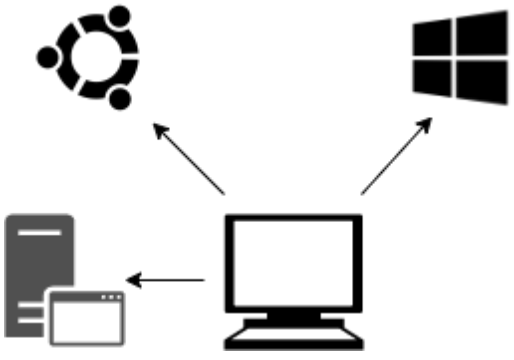

Fig. 10: Virtualización del servidor.

En este caso se virtualizará el servidor y la conexión remota. De esta forma cuando se contrate una nube(e.g. VPS) en la web se contratará un servicio virtualizado.
La virtualización del proyecto se realizará con **Vmware**. Se utilizará una distribución Linux, concretamente una distribución **Debian 10** con las siguientes especificaciones:

- 8GB de RAM.
- 3 procesadores.
- 50 GB de disco SCSI.

Llegado el momento de alojar el proyecto en la máquina virtual, se utilizó el programa **WinSCP**[14], el cual funciona como cliente SFTP e implementa SSH. La función principal de este programa es la transferencia de archivos entre dos sistemas informáticos, uno local y uno remoto que ofrezca servicios SSH. Para poder realizar la transferencia se tiene que preparar la máquina virtual, ya sea habilitando SSH en caso que no lo tenga o abriendo el puerto correspondiente.
Dicho lo anterior, para este proyecto se pretende utilizar un proveedor de servicios Cloud. Se ha escogido el servidor de VPS para alojar nuestra máquina virtual. De esta forma se podrá acceder desde donde sea, ya que la idea de este proyecto es que sea funcional.

11 PÁGINA WEB

El objetivo final del proyecto es poder visualizar las clasificaciones realizadas de forma simplificada y agradable.
Para ello se mostrarán los datos en una página web en la que, debido a la naturaleza de cada red social, el usuario tendrá diferentes opciones para realizar el análisis de sentimientos. En Twitter el usuario podrá realizar análisis sobre un término, hashtag o usuario, indicando el número de *posts* a analizar. Si se quiere realizar el análisis sobre Facebook se podrá indicar el número de páginas a analizar. En ambas redes sociales se podrá establecer un rango de fechas sobre el que se realizará el análisis.
Cabe destacar que el proyecto se ha desarrollado de forma modular, lo que permite realizar una escalabilidad a otras redes sociales en un futuro.

11.1. Visualización de los datos

Los datos se presentarán de la siguiente manera:

- Los *posts* analizados estarán en el formato de su red social. Además, se ha hecho una divisón al mostrar los resultados para apreciar la diferencia de polaridad entre los *posts* . Se van a mostrar los dos más negativos y los dos más positivos.
  Cada *post* irá acompañado de su polaridad junto con una etiqueta. Además, en el caso de Facebook se mostrará el porcentaje de comentarios positivos, negativos y neutros en relación al *post* al que pertenecen.
- Si el análisis se realiza sobre un hashtag o palabra de Twitter se mostrará un *WordCloud* y un gráfico de sectores resumiendo los datos. En caso que sea un usuario, el *WordCloud* será substituido por un histograma sobre *posts* publicados en los últimos dias.
- Si el análisis se realiza sobre una página pública de Facebook se mostrará un gráfico de sectores y un *WordCloud* pero sobre los comentarios del *post* que se indique.
- En ambas redes sociales se podrá descargar un informe con todos los *posts* analizados y, si se precisa, un pdf del resumen del análisis en formato de tabla.

11.2. Implementación

Para el desarrollo del apartado gráfico se ha utilizado Flask[15], un microframework basado en Werkzeug, Jinja 2. Este incluye un servidor de desarrollo integrado, compatibilidad con pruebas de unidades y está totalmente habilitado para Unicode con envío de solicitudes RESTful y cumplimiento de WSGI(Web Server Gateway Interface).
Se ha creado un entorno el cual permite visualizar los datos referentes a *posts*, usuarios o páginas públicas de ambas redes sociales de una forma clara y entendedora, poniendo como prioridad la polaridad de los *posts* a través de un gráfico de sectores, el cual va acompañado de una leyenda para facilitar su comprensión. Como ejemplo, la figura 11 muestra el gráfico de sectores resumiendo la polaridad de cincuenta *posts* de Twitter sobre el reciente estreno de la película Cruella.

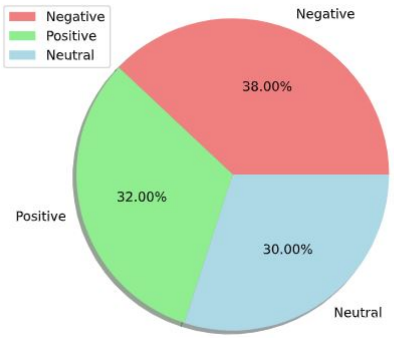

Fig. 11: Gráfico de sectores sobre la polaridad de un término.

Lo que se busca en este proyecto es el entendimiento rápido de un análisis, así que también se ha incorporado un *WordCloud* que acompaña y completa el gráfico de sectores. La figura 12 es una ejemplificación de un *WordCloud* que resume un conjunto de *posts* de Twitter sobre la temática Cruella.
Dado que no se logra visualizar bien la información referente a la polaridad, se ha decidido implementar el *WordCloud* en un formato con el que podamos distinguir la polaridad de las palabras en los diferentes tipos de comentarios. Este *WordCloud* estará formado por tres sectores; negativo, neutro y positivo, donde cada uno de ellos se relacionará con el gráfico de sectores en base al color de las palabras.
En base a las pruebas realizadas, cuando se haga una búsqueda de *posts* de un usuario de Twitter, un *WordCloud* no proporciona información concisa ya que se trata de una búsqueda genérica para ver el comportamiento del usuario. Este hecho ha supuesto la incorporación de un histograma

Fig. 12: *WordCloud* sobre el término Cruella.

que refleja el número de *posts* en los últimos días. En caso de que la busqueda no incluya la opción de fechas, el histograma se creará con los ultimos cinco días. La figura 13 muestra dicho histograma.

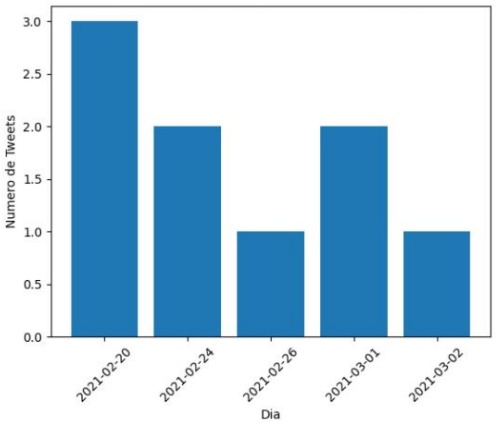

Fig. 13: Histograma de los últimos cinco dias con *posts* de un usuario de Twitter.

Atendiendo a la red social de Facebook, la implementación se ha llevado a cabo de otra forma. Dado que el análisis solo se ha podido realizar sobre *posts* de páginas públicas, se decidio implementar la análisis de polaridad sobre los comentarios de estos *posts* , por lo que se permite al usuario seleccionar el *post*, sobre cuál quiere visualizar el análisis. En este caso también se mostrará un gráfico de sectores y un *WordCloud*. Para hacer más completa la predicción, también se permite al usuario descargar un informe en formato xlxs donde podrá visualizar el análisis de todos los *posts* sin procesar. Además, tendrá la posibilidad de descargar una versión simplificada del análisis en PDF.

12 Futuros trabajos

Este proyecto tiene la intención de proponer mejoras para aumentar la calidad y el uso de nuevas funciones en distintas partes del software. El elevado crecimiento de las redes sociales y el BigData, juntamente con el análisis de sentimientos ha provocado que muchas grandes empresas indaguen en este aspecto e incorporen dicha tecnología en sus proyectos. Existen una gran variedad de proyectos y ámbitos donde se puede aplicar el trabajo y ampliarlo:

- **Análisis de sentimientos:** El análisis de sentimientos tiene un gran margen de mejora, desde analizar *posts* en una red social hasta calificar el futuro *post* antes de ser enviado. Por eso se propone implementar este proyecto al foro de la universidad y en las encuestas con el fin de advertir al usuario de la magnitud de su opinión antes de que sea publicada. Además, habría una mejora considerable en la precisión del clasificador si se utilizara un Corpus en catalán dado que la mayoría de comentarios en las encuestas son en catalán o si se implementara un Corpus clasificado manualmente.

- **Análisis de sentimientos en imágenes:** En este proyecto, la polaridad de un *post* se ha basado únicamente en los caracteres escritos. Sin embargo, hay muchas otras formas de expresar una opinión. Una de ellas son las imágenes. Muchos *posts* van acompañados de una imagen la cual complementa el texto o viceversa, y no es necesario leer su descripción para llegar a entender lo que quiere transmitir. Por ello, es importante determinar qué tipo de imágenes son válidas para según el sitio donde vayan a ser publicadas.

  Para poder determinar el análisis de sentimiento sobre una imagen se propone el uso de la red social Instagram juntamente con la tecnología **Microsoft Cognitive Servicies**.

- **Seguimiento del comportamiento de una cuenta:** Dar opinión en un *post* de una red social es algo muy común que puede realizar cualquier persona que disponga de una cuenta. Según que *post* se publique, se pueden recibir muchas críticas, ya sea positivas o negativas. Ciertas empresas podrían estar interesadas en saber el comportamiento de sus publicaciones, por lo tanto se ve un potencial de uso el extender el proyecto hacia un seguimiento del comportamiento de una cuenta, con la posibilidad de notificar al propietario de ésta si un *post* está recibiendo muchas reacciones negativas o positivas.

- **Seguridad y privacidad:** Realizar una extracción de datos para posteriormente mostrarlos en una página web es un proceso el cual puede conllevar problemas debido a la inclusión de datos sensibles en algún *post*. Sería interesante incluir en el proyecto una autenticación de usuario junto con incorporar seguridad en los datos utilizando encriptación.

13 Conclusiones

La primera conclusión a la que se ha llegado es que las redes sociales han irrumpido en el ámbito de la opinión social, siendo un canal de comunicación capaz de canalizar las ideas, opiniones y los sentimientos de usuarios sobre temas. Clasificar los *posts* como positivos, negativos y neutros, puede aportar información relevante sobre un tema o usuario, y hacerlo de forma manual sería una tarea ineficiente debido a la cantidad de contenido generado en las redes sociales.
La segunda conclusión obtenida es que un clasificador con resultados consistentes no requiere el uso de un Corpus en español, además, tampoco es necesario recurrir a versiones de pago de APIs para obtener buenos resultados.

Para conseguir estos resultados, entre la recopilación de los post y la clasificación hay una fase fundamental para el éxito del proyecto que es la limpieza de los datos, la cual debe mantener un balance entre conservar la esencia del post y el retirar los elementos que generen ruido en el clasificador, ya sean *stopwords* o signos de puntuación.
Como tercera y última conclusión, las pruebas han mostrado que una parte de la comunidad utiliza las redes sociales como fuente informativa o como medio para expresar su opinión acerca de un tema, pero también hay un amplio número de comentarios agresivos los cuales tendrían que ser moderados. Dado que los clasificadores están en evolución y el hecho de tener un clasificador entrenado es un proceso cíclico, este proyecto tiene el potencial de informar a un usuario si las opiniones que está a punto difundir pueden incluir temas de odio, irrespeto o difamación.
El análisis de sentimientos en redes sociales tiene aún retos por superar, uno de estos retos está asociado a la dialéctica de la comunicación, pues el sarcasmo y el doble sentido no siempre consiguen ser clasificados de forma adecuada.

14 AUTOCRÍTICA

Atendiendo a las diferentes pruebas que se han realizado a lo largo del proyecto podemos concluir que el nivel de precisión del clasificador podría ser considerablemente mejorado utilizando un Corpus en español, sin necesidad de tener que traducir los textos. Como se ha comentado en apartados anteriores, el hacer nuestro propio Corpus fue algo que se tuvo en cuenta pero por razones de tiempo se dejó de lado.

AGRADECIMIENTOS

En primer lugar me gustaría agradecer a mi tutor Diego Mauricio Freire Bastidas todo el apoyo que me ha dado a lo largo del proyecto, por a pesar de las horas a las que le escribía siempre estar dispuesto a revisar lo que hiciese falta, resolver mis dudas o animarme en los momentos más precarios. Además de guiarme y enseñarme no solo a hacer un proyecto, sino a ver desde otros enfoques la forma de trabajar. En segundo lugar agradecerle a mi familia y amigos el darme el soporte externo necesario para seguir avanzando en los momentos más difíciles.

REFERENCIAS

[1] Redes sociales con mayor incremento de nuevos usuarios durante la cuarentena por coronavirus en España en 2020. [En línea]. Recuperado de https://es.statista.com/estadisticas/1118907/covid-19-redes-sociales-con-mayor-numero-de-nuevos-perfiles-espana/ [Último acceso: 27 de Febrero 2021]

[2] Kohorst, L. Basic data analysis on Twitter with Python. [En línea]. Recuperado de https://www.freecodecamp.org/news/basic-data-analysis-on-twitter-with-python-251c2a85062e/ [Último acceso: 28 de Febrero 2021]

[3] TextBlob: Simplified Text Processing. [En línea]. Recuperado de https://textblob.readthedocs.io/en/dev/ [Último acceso: 09 de Marzo 2021]

[4] Daityari, S. How To Perform Sentiment Analysis in Python 3 Using the Natural Language Toolkit(NLTK). [En línea]. Recuperado de https://www.digitalocean.com/community/tutorials/how-to-perform-sentiment-analysis-in-python-3-using-the-natural-language-toolkit-nltk [Último acceso: 09 de Marzo 2021]

[5] Al-Masri, A. Creating The Twitter Sentiment Analysis Program in Python with Naive Bayes Classification. [En línea]. Recuperado de https://towardsdatascience.com/creating-the-twitter-sentiment-analysis-program-in-python-with-naive-bayes-classification-672e5589a7ed [Último acceso: 09 de Marzo 2021]

[6] Roesslein, J. Tweepy Documentation. [En línea]. Recuperado de: https://docs.tweepy.org/en/laTest/ [Último acceso: 10 de Marzo 2021]

[7] Pandas Documentation. [En línea]. Recuperado de: https://pandas.pydata.org/docs/index.html [Último acceso: 12 de Marzo 2021]

[8] Zúñiga, K. Facebook Scrapper. [En línea]. Recuperado de https://pypi.org/project/facebook-scraper/ [Último acceso: 12 de Marzo 2021]

[9] LuShan. `google_trans_new` Documentation. [En línea]. Recuperado de https://pypi.org/project/google-trans-new/ [Último acceso: 12 de Marzo 2021]

[10] Time access and conversions. [En línea]. Recuperado de https://docs.python.org/3/library/time.html [Último acceso: 25 de Marzo 2021]

[11] Authentication. [En línea]. Recuperado de: https://developer.twitter.com/en/docs/authentication/oauth-1-0a [Último acceso: 28 de Abril 2021]

[12] Introducing JSON. [En línea]. Recuperado de: https://www.json.org/json-en.html [Último acceso: 28 de Abril 2021]

[13] Ganesan, M. Is It Legal to Scrape Facebook Data. [En línea]. Recuperado de https://proxiesapi-com.medium.com/is-it-legal-to-scrape-facebook-data-4788962dd136 [Último acceso: 19 de Mayo 2021]

[14] WinSCP Documentation. [En línea]. Recuperado de: https://winscp.net/eng/index.php [Último acceso: 19 de Mayo 2021]

[15] Flask Documentation. [En línea]. Recuperado de: https://flask.palletsprojects.com/en/2.0.x/ [Último acceso: 19 de Mayo 2021]

APÉNDICE

A.1. Análisis de Twitter por usuario

Se procede a mostrar los resultados de un análisis de un usuario el cual se ha anonimizado su nombre e imagen para preservar su identidad. No se ha introducido ningún parámetro de búsqueda, por lo que se realizará una predicción de los cincuenta tweets más recientes. Los resultados en el formato de Twitter se reflejan en la figura 14:

Fig. 14: Tweet positivo.

El tweet no contiene palabras que puedan hacer decaer el rating asociado, la mayoría son neutrales, pero la inclusión del ”jeje” decanta hacia positivo el *post*.
El siguiente resultado obtenido es un tweet negativo que se muestra en la figura 15:

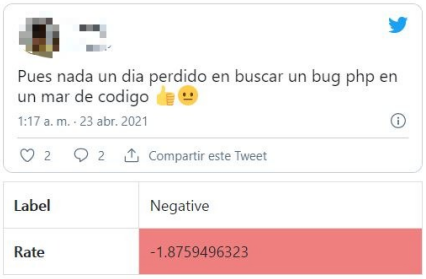

Fig. 15: Tweet negativo.

En este caso los emoticonos tienen un peso importante a la hora de valorar el tweet, ya que el Corpus de entrenamiento del modelo también incorporaba emoticonos.
A partir de estos tweets y cuarenta y ocho más se ha obtenido un gráfico de sectores que refleja la polaridad global en la figura 16.

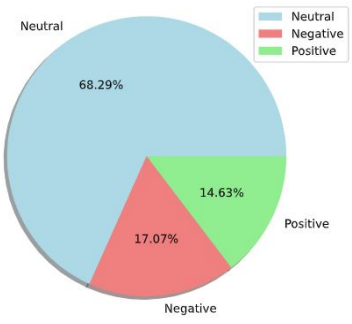

Fig. 16: Gráfico de sectores.

Atendiendo al gráfico de sectores se puede apreciar que la mayoría de *posts* del usuario tienen una valoración neutral.
Además también se ha generado un histograma donde se muestra su actividad en los últimos cinco días que ha publicado algo en Twitter.
La figura 17 indica dicha actividad.

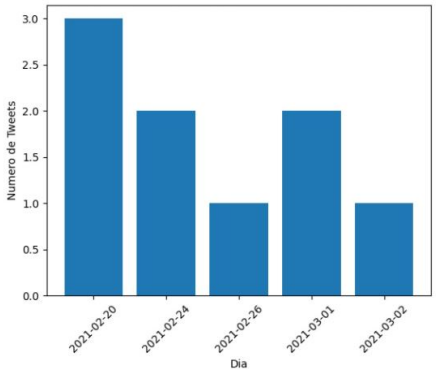

Fig. 17: Histograma del usuario.

B.2. Análisis de Twitter por término

Se procede a realizar una búsqueda por término o hashtag en la red social de Twitter, esta vez el término será ”Barça B” y se indicará como opciones avanzadas analizar un conjunto de trescientos tweets. Como en el caso anterior, la figura 18 muestra un ejemplo de un comentario negativo mostrado en la página web:

Fig. 18: Tweet negativo sobre la temática Barça b.

Se puede apreciar que cuando hay palabras más agresivas, en este caso ”basuras”, o la inclusión de varios ”no” hace que el rating baje considerablemente.
Otro ejemplo, pero esta vez de un tweet positivo sería el que muestra la figura 19.
Como se puede apreciar este tweet tiene un rating bastante positivo. Esto es debido al uso de términos como ”ojalá”, ”apoyo” o ”ídolo”.
Completando el análisis de los *posts* se ha obtenido el gráfico de sectores que refleja la figura 20.
Para comprender sobre qué se habla en el conjunto de comentarios de cada tipo de *post* sobre la temática ”Barça b” se ha desarrollado el *WordCloud* de la figura 21.
Como se puede apreciar hay una división de polaridad por grupos de palabras. Esto se ha realizado para distinguir de

Fig. 19: Tweet positivo sobre la temática Barça b.

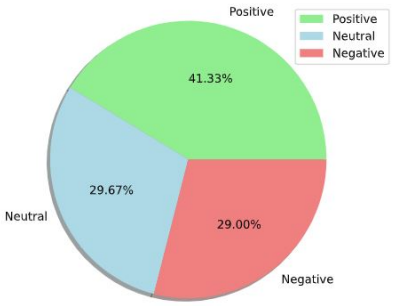

Fig. 20: Gráfico de sectores sobre la temática Barça b.

Fig. 21: *WordCloud* sobre la temática Barça b.

una forma clara sobre qué hablan los usuarios. Las palabras de cada color hacen referencia a la leyenda del gráfico de sectores.

C.3. Análisis de Facebook

En el caso de Facebook podremos seleccionar cuántas páginas queremos analizar. En este caso será una y se predecirá un *post* de la página pública de NetflixElite.
La figura 22 muestra el análisis del *post*.
De cada *post* se diferencia el rating y el label, además de los porcentajes de comentarios positivos, negativos y neutros. Atendiendo a la polaridad del *post*, se puede afirmar que no ha sido bien clasificado, el valor debería estar en el rango de neutro. Como se ha comentado en el apartado de autocrítica, si el entrenamiento se hiciese con un Corpus en español los resultados serían mas precisos y habría menos margen de error.
A los comentarios del *post* se les puede realizar un análisis y ver el gráfico de sectores mostrado en la figura 23.
Además, el *WordCloud* complementario al gráfico de sectores también hace referencia a los comentarios del *post*. La figura 24 representa el *WordCloud*.

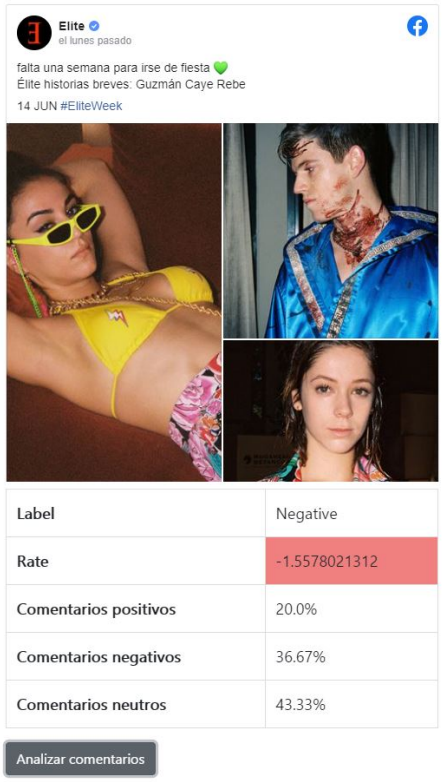

Fig. 22: *post* de Facebook.

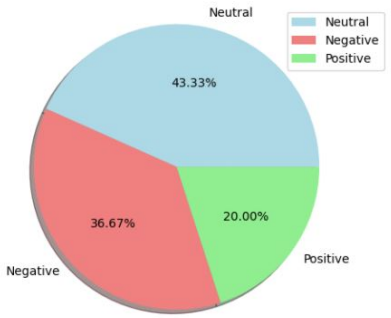

Fig. 23: Gráfico de sectores de los comentarios de un *post* de Facebook.

Fig. 24: *WordCloud* de los comentarios de un *post* de Facebook.

En el caso que un *post* tenga cero o insuficientes comentarios se mostrará un aviso de que no hay datos a analizar.

D.4. Análisis de encuestas UAB

Se ha decidido realizar una prueba de concepto utilizando los resultados de las encuestas obtenidos este año, por razones de privacidad los nombres de profesores, asignaturas y facultades se encuentran anonimizados. El conjunto que se utilizará será de cuarenta y cuatro opiniones, veintidós escritas en el cuadro de texto de opiniones positivas y veintidós en el de negativas. Como opiniones más relevantes se han obtenido las que muestra la figura 25:

1	No se podía ver la pizarra, ni al profesor correctamente, no era fácil seguir las clases online y el profesor no ha hecho ningún esfuerzo para que no fuera así. No se ha podido aprender todo lo esperado y aún y así los exámenes eran demasiado largos y difíciles.	Negative	-8.3528136921
2	-Excesivo material para hacer en casa. -Practicas muy largas y difíciles de entender en algunas ocasiones. -Poca adaptación para principiantes en la programación. -Examen de practicas muy injusto (ya que en este cuando lo realice salían cosas de la practica que tenia que empezar a realizar ese mismo día). -Primer parcial bastante difícil para personas que nunca habían programado antes de entrara a la carrera. -Lentitud de corrección del primer parcial.	Negative	-5.2837308165
3	Temario interesante, veo que es una asignatura base que sirve para ampliar conocimientos. Con la actual situación se han intentado adaptar de la mejor manera posible.	Positive	5.3377594463
4	En general, todo está prácticamente perfecto. Buen profesorado, buenas clases tanto de teoría como de problemas, ejercicios útiles y pruebas avaluables coherentes con el temario dado.	Positive	5.7380574767

Fig. 25: Conjunto de opiniones clasificadas.

Sobre el conjunto de encuestas se ha obtenido el gráfico de sectores de la figura 26:

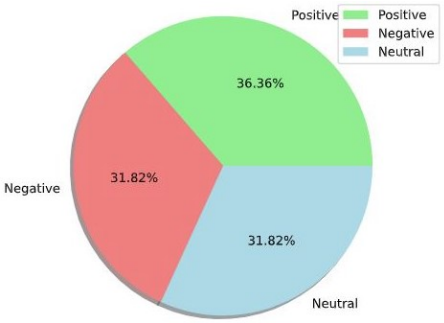

Fig. 26: Gráfico de sectores de las encuestas.

Fig. 27: *WordCloud* de las encuestas.

El gráfico refleja lo esperado, un porcentaje prácticamente idéntico de opiniones positivas y negativas, a excepción de las neutras. Las neutras muestran opiniones positivas o negativas no tan extremas. Para determinar sobre qué se habla en cada tipo de opiniones se ha incorporado el *WordCloud* de la figura 27.
Como se ha comentado previamente, sería interesante entrenar el clasificador con un Corpus en español y más completo e incorporar el clasificador a las encuestas para determinar si los comentarios publicados exceden lo que se busca en relación a dar la opinión sobre la asignatura. Por otro lado, como se puede apreciar en la figura 27 el *WordCloud* refleja que hay opiniones escritas en castellano y catalán, y como se ha mencionado en el apartado 11, sería interesante utilizar un Corpus en catalán o bilingüe que nos permitiese ajustar mejor las clasificaciones y la muestra de resultados.