SAMSUNG

Big Data Course

Memoria final (MOD) - HATE: discurso de odio

For students (instructor review required)

SAMSUNG

Course	Big Data Course
Team Name	MOD
Team Leader/ Members	Alejandro Leo Ramírez/ Javier Gil Rodríguez, Víctor Tavara Pérez, Javier Garrido Sola
Project Title	HATE: Discurso de odio

Goal

Predecir si comentarios de diferentes redes sociales son hate o no.
Mostrar en tiempo real si se están produciendo comentarios de hate.
Analizar qué días se produce más hate en diferentes redes sociales.
Prevenir el hate en redes sociales.
Construir un sistema Big Data real con almacenamiento y predicción en tiempo real, mediante el uso de las herramientas vistas en el curso, y otras adicionales.

Abstract

El hate afecta de forma negativa a las personas. Con la llegada de las redes sociales, el hate está en el día a día de todo el mundo de una forma mucho más directa. Lo que antes podía reducirse a un entorno o a un colectivo, ahora afecta de forma pública a cualquier persona. Aunque las causas del hate pueden ser muchas, se quiere comprobar si este puede estar relacionado con el día de la semana y otras causas (día laboral, festivo, etc.). De este modo, podría predecirse si un día se va a producir más hate, de modo que una persona evite meterse en redes sociales. Además, se incluye un sistema que analiza en tiempo real los comentarios que recibe un vídeo que se emite en directo y muestra si estos son comentarios de hate o no.

Method

Para obtener datos, utilizaremos la metodología explicada en la sección de datos. Los datos se extraerán de forma masiva, por lo que será necesario filtrar estos datos por columna.
Durante la fase de obtención de datos para entrenar un modelo, filtraremos las tablas para únicamente quedarnos con columnas de: texto de la publicación, boolean de si la publicación es hate o no.
Con los datos obtenidos y con el modelo entrenado, utilizaremos nuevas publicaciones con timestamp y las filtraremos para quedarnos únicamente con la publicación y el timestamp. Se usará el modelo entrenado para incluir información de la columna de si la publicación es hate o no.
Se utilizará el timestamp para obtener el día de la semana, si era festivo o no, etc,

de modo que se puedan utilizar estos datos para predecir datos futuros en base a la información que obtengamos.

Data

Los datos se van a obtener de distintas fuentes: YouTube, YouTube Live, TikTok, Reddit, 4chan e Instagram. En primer lugar, obtuvimos 3 datasets de múltiples fuentes como Kaggle, Google Dataset Search o bases de datos científicas como IEEE BigData (más de 160000 datos en total). Estos datos permitieron entrenar un modelo (desarrollado con PySpark) que predice si un comentario de redes sociales es hate o no.

Para obtener datos en tiempo real se han diseñado y desarrollado scripts que, mediante web scraping y llamadas a APIs, obtienen publicaciones en tiempo real que se generan en las diferentes redes sociales. A estos datos se les aplica el modelo entrenado y se les etiqueta como comentarios de hate o comentarios sin hate.

Todos los datos de entrenamiento del modelo y que se obtienen en tiempo real son almacenados en la base de datos en MariaDB y en HDFS con sus correspondientes etiquetas de hate. Del mismo modo, los .csv que se generan a partir de los datos obtenidos de las redes sociales son almacenados en un clúster HDFS compuesto por tres dispositivos (para ser consultado a través de Hive), gestionado este flujo gracias a Apache NiFi.

Expected Outcome

Los resultados obtenidos son varios:

1. Modelo entrenado que detecta si un comentario es hate.
2. Ficheros .csv con los datos obtenidos de las redes sociales etiquetados.
3. Ficheros .csv en un sistema HDFS formado por tres dispositivos.
4. Datos en una base de datos MariaDB.
5. Sistema de consulta con Hive en HDFS.
6. Tableros de PowerBI con información sobre el histórico de hate y datos en tiempo real.
7. Página web con datos en directo de hate de un vídeo de YouTube.
8. Analíticas de hate en redes sociales por día.

SAMSUNG

Role by Member

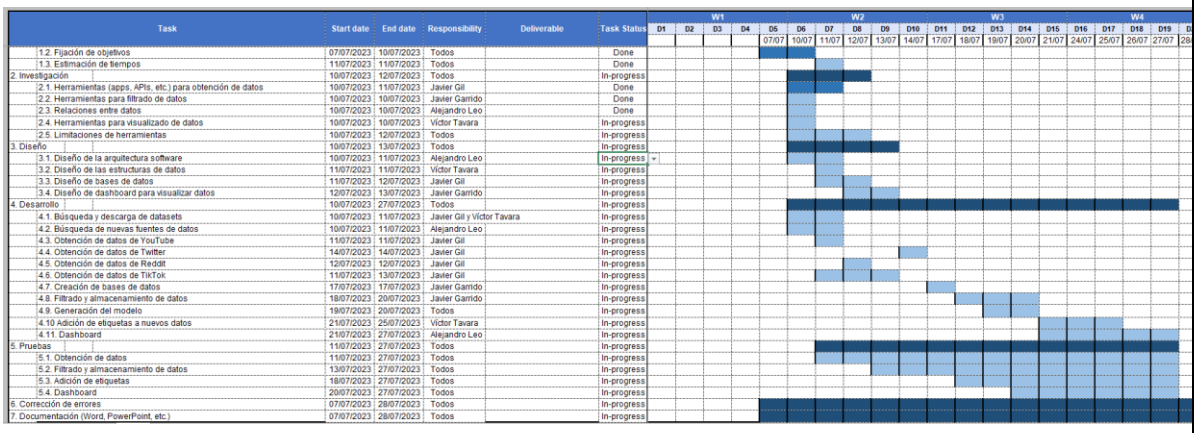
Alejandro Leo: jefe de grupo; gestión y administración de tareas y tiempo; portavoz de grupo frente al profesor; diseño de la arquitectura del sistema; diseño de las bases de datos; soporte al resto de compañeros; redacción y presentaciones.

Javier Gil: búsqueda de datasets válidos; desarrollo de scripts para la obtención de datos de las diferentes redes sociales; generación de modelo con PySpark; desarrollo de scripts de filtrado y etiquetado de los nuevos datos; diseño y desarrollo de la página web; redacción y presentaciones.

Víctor Tavara: búsqueda de datasets válidos; diseño de desarrollo de scripts en Apache NiFi; diseño y desarrollo de dashboard en PowerBI; redacción y presentaciones.

Javier Garrido: configuración y administración del sistema operativo; configuración y administración de la red virtual privada; configuración y administración de los nodos del clúster HDFS; configuración y administración de Hive; redacción y presentaciones.

Schedule Summary



Task	Start date	End date	Responsibility	Deliverable	Task Status
1.2. Fijación de objetivos	07/07/2023	10/07/2023	Todos		Done
1.3. Estimación de tiempos	11/07/2023	11/07/2023	Todos		Done
2. Investigación	10/07/2023	12/07/2023	Todos		In-progress
2.1. Herramientas (apps, APIs, etc.) para obtención de datos	10/07/2023	11/07/2023	Javier Gil		Done
2.2. Herramientas para filtrado de datos	10/07/2023	10/07/2023	Javier Garrido		Done
2.3. Relaciones entre datos	10/07/2023	10/07/2023	Alejandro Leo		Done
2.4. Herramientas para visualizado de datos	10/07/2023	10/07/2023	Víctor Tavara		In-progress
2.5. Limitaciones de herramientas	10/07/2023	12/07/2023	Todos		In-progress
3. Diseño	10/07/2023	13/07/2023	Todos		In-progress
3.1. Diseño de la arquitectura software	10/07/2023	11/07/2023	Alejandro Leo		In-progress
3.2. Diseño de las estructuras de datos	11/07/2023	11/07/2023	Víctor Tavara		In-progress
3.3. Diseño de bases de datos	11/07/2023	12/07/2023	Javier Gil		In-progress
3.4. Diseño de dashboard para visualizar datos	12/07/2023	13/07/2023	Javier Garrido		In-progress
4. Desarrollo	10/07/2023	27/07/2023	Todos		In-progress
4.1. Búsqueda y descarga de datasets	10/07/2023	11/07/2023	Javier Gil y Víctor Tavara		In-progress
4.2. Búsqueda de nuevas fuentes de datos	10/07/2023	11/07/2023	Alejandro Leo		In-progress
4.3. Obtención de datos de YouTube	11/07/2023	11/07/2023	Javier Gil		In-progress
4.4. Obtención de datos de Twitter	14/07/2023	14/07/2023	Javier Gil		In-progress
4.5. Obtención de datos de Reddit	12/07/2023	12/07/2023	Javier Gil		In-progress
4.6. Obtención de datos de TikTok	11/07/2023	13/07/2023	Javier Gil		In-progress
4.7. Creación de bases de datos	17/07/2023	17/07/2023	Javier Garrido		In-progress
4.8. Filtrado y almacenamiento de datos	18/07/2023	20/07/2023	Javier Garrido		In-progress
4.9. Generación del modelo	19/07/2023	20/07/2023	Todos		In-progress
4.10 Adición de etiquetas a nuevos datos	21/07/2023	25/07/2023	Víctor Tavara		In-progress
4.11. Dashboard	21/07/2023	27/07/2023	Alejandro Leo		In-progress
5. Pruebas	11/07/2023	27/07/2023	Todos		In-progress
5.1. Obtención de datos	11/07/2023	27/07/2023	Todos		In-progress
5.2. Filtrado y almacenamiento de datos	13/07/2023	27/07/2023	Todos		In-progress
5.3. Adición de etiquetas	18/07/2023	27/07/2023	Todos		In-progress
5.4. Dashboard	20/07/2023	27/07/2023	Todos		In-progress
6. Corrección de errores	07/07/2023	28/07/2023	Todos		In-progress
7. Documentación (Word, PowerPoint, etc.)	07/07/2023	28/07/2023	Todos		In-progress

1. Arquitectura, software y datos

En esta sección se explicarán tanto la arquitectura del sistema, como el software y los datos que forman parte de él. En cada subsección se hará un repaso de todos ellos. El código [1] y la documentación técnica [2] pueden consultarse de forma pública.

1.1. Arquitectura y software

La arquitectura del proyecto se ha basado en diferentes tecnologías que hemos utilizado a lo largo del curso y otras tecnologías que ya se habían utilizado en otros cursos de nuestros respectivos grados. Por ello, el número de tecnologías y herramientas utilizados en este proyecto es bastante amplio, desde HDFS hasta HTML, pasando por Hive, PySpark o Apache NiFi. La arquitectura queda resumida en la Ilustración 1.

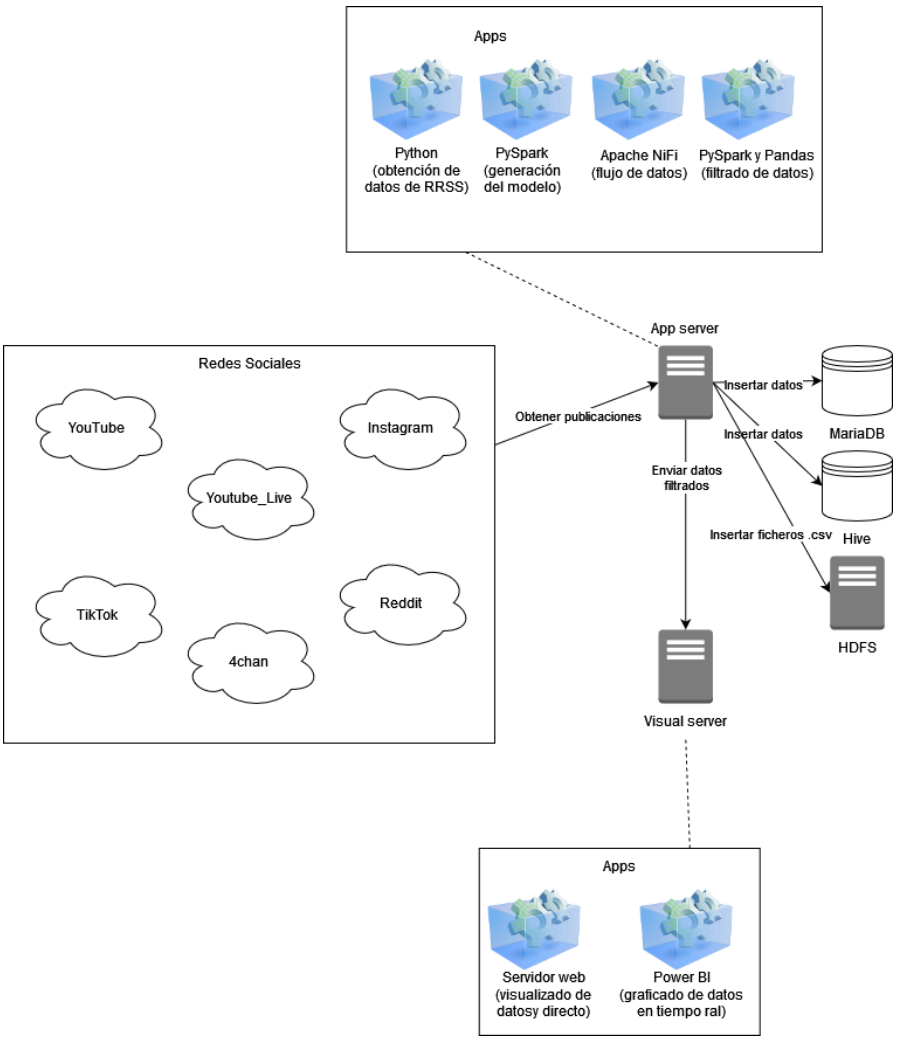


Ilustración 1. Arquitectura completa.

La Ilustración 1 muestra el conjunto de elementos hardware y software que componen el sistema del proyecto.

En primer lugar, se tienen las diferentes redes sociales: YouTube, YouTube Live, TikTok, Instagram, Reddit y 4chan. Todas estas redes conforman el conjunto "Redes sociales", de las cuales *App server* obtiene la obtiene los datos (publicaciones en todas estas redes sociales).

App server es el servidor que contiene la aplicación que obtiene los datos, los filtra y los inserta en las diferentes bases de datos. La obtención de datos ha sido desarrollada en Python, se ha generado el modelo de predicción de hate en PySpark, usa Apache NiFi para realizar el flujo de datos hasta HDFS y filtra los datos mediante PySpark y Pandas. Este *App server* (Ilustración 2) coincide con el *namenode* de HDFS, decisión tomada para simplificar el diseño.

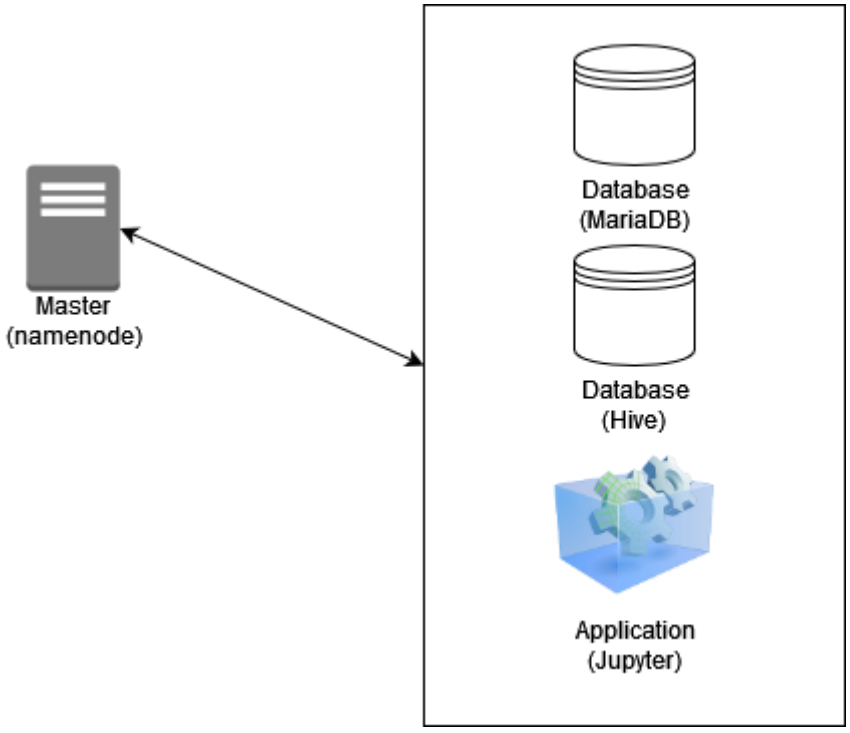


Ilustración 2. Master (namenode).

Los datos filtrados se agrupan y almacenan en tres fuentes distintas. En primer lugar, ficheros .csv que se acaban almacenando en el clúster HDFS. En segundo lugar, los datos se almacenan en una base de datos de MariaDB. Por último, se envían los datos a un servidor que tiene aplicaciones que permiten visualizar los

datos históricos y los datos en tiempo real (*visual server*).

Visual server tiene dos aplicaciones que permiten la visualización de datos. Una de ellas es PowerBI, donde se grafican los datos en tiempo real y, por otro lado, se ha desarrollado una web donde se puede visualizar vídeos en directo, los comentarios y la analítica de ellos. Además, estas herramientas muestran la analítica de hate por día o el hate en diferentes redes sociales.

Para conseguir la comunicación entre los diferentes dispositivos que conforman el sistema se ha diseñado y configurado una red virtual privada a través de Zerotier One, de modo que la comunicación de los diferentes dispositivos (y sus respectivas aplicaciones) puedan comunicarse de forma remota. El esquema queda simplificado a través de la Ilustración 3.

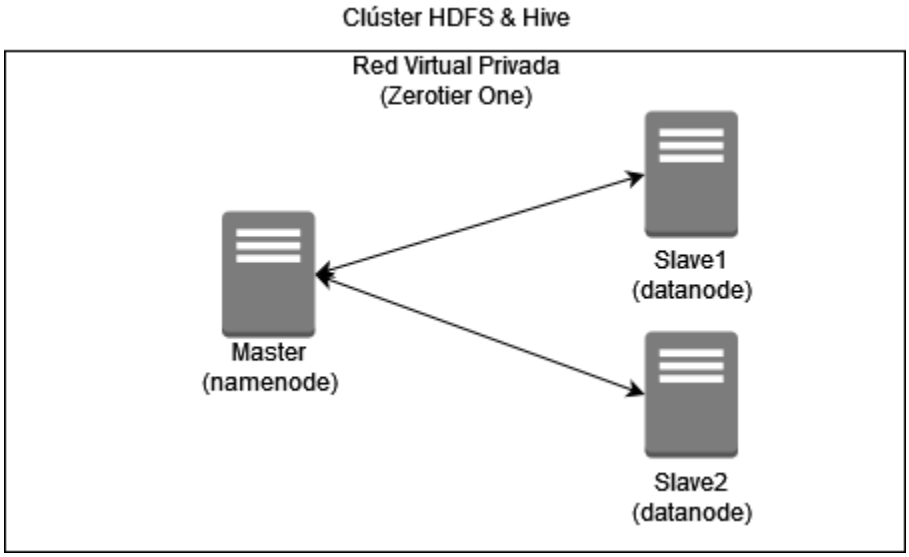


Ilustración 3. Red virtual privada.

1.2. Datos

El flujo de los datos se muestra en la Ilustración 4.

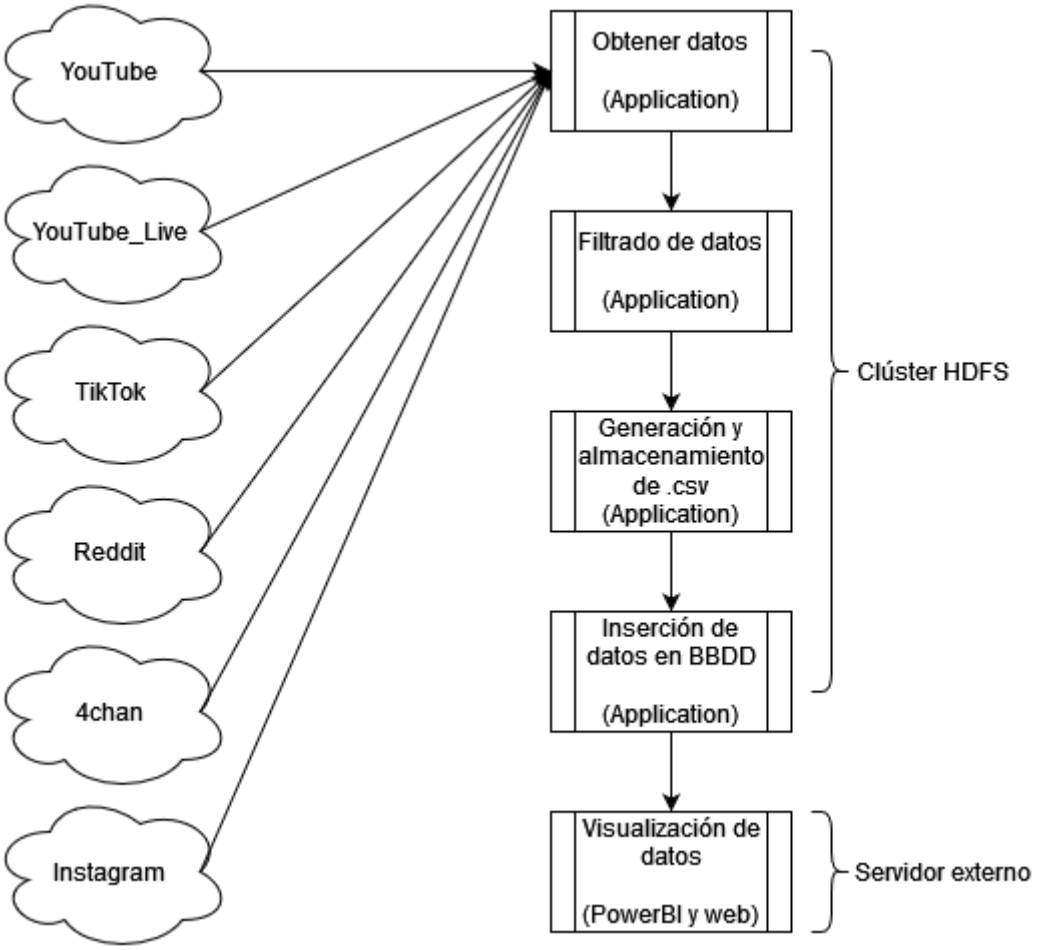


Ilustración 4. Flujo de datos.

Los datos se recolectan de las diferentes redes sociales ya mencionadas. Esta información se obtiene a través de técnicas de *web scrapping* y llamadas a las APIs de las redes sociales. Esta información se obtiene "en crudo" y se procesa a través de Pandas y PySpark, de modo que procesamos únicamente la información relevante.

Esta información ya filtrada consta de diferentes tipos de datos, la cual se almacena en bases de datos en MariaDB. Las bases de datos se dividen en dos: la de los datos de entrenamiento del modelo (llamada 'modelo') y la de predicción de hate (llamada 'prediccion'). La base de datos con los datos del entrenamiento la compone una única tabla llamada 'train' que almacena los datos en tres columnas: id (bigint), comentario (text) y hate (boolean).

La segunda base de datos, 'modelo', se ha diseñado de modo que sea simple e intuitiva. Por ello consta únicamente de dos tablas: 'sources' y 'comments'. La tabla 'sources' contiene el id de la red social (int) y el nombre de esta (string). Por otro lado, la tabla 'comments' posee la información mostrada en la Ilustración 5.

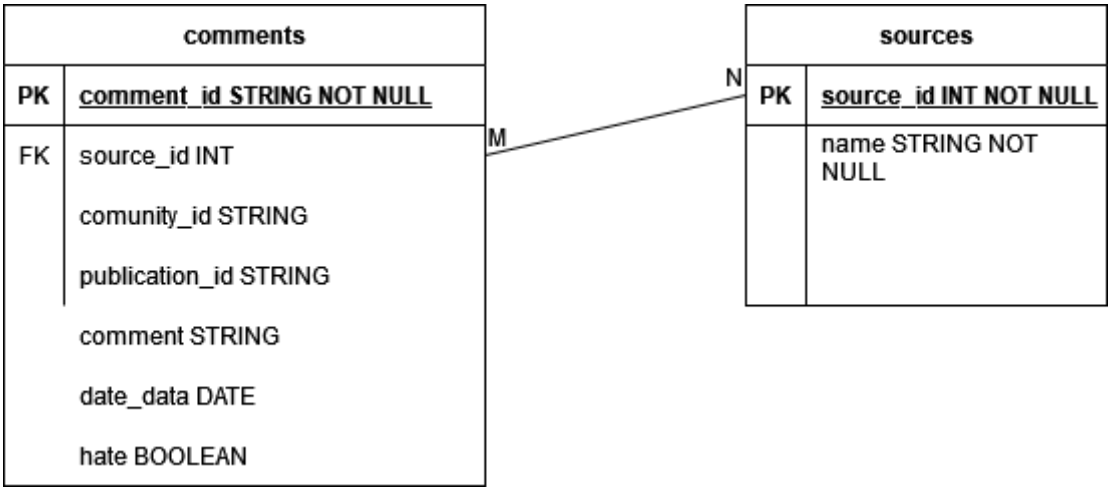


Ilustración 5. Base de datos 'prediccion'.

Se debe tener en cuenta que los datos almacenados no son únicamente históricos, sino que también se recogen en tiempo real, de modo que esta recolección, filtrado y procesado de datos se produce de manera constante.

1.3. Modelo y predicciones

El modelo nos permite etiquetar datos nuevos a partir de un conjunto de datos de entrenamiento sobre comentarios de hate. Para construir ese modelo, realmente se entrenaron dos modelos. El primero de ellos se hizo a través de PySpark para intentar hacer un trabajo realista en el ámbito de Big Data. Sin embargo, debido a un mayor dominio de Python, este modelo acabó desarrollándose también en este lenguaje mediante el uso de redes neuronales. El entramiento del modelo se ha llevado a cabo con más de 160000 datos y esto ha resultado en un sistema que etiqueta nuevas publicaciones de redes sociales.

2. Resultados obtenidos

Los resultados obtenidos en este proyecto se pueden observar en el *visual server*. Este servidor contiene y muestra los datos recogidos y diferentes analíticas de los datos a través de dos aplicaciones: un servidor web y dashboard de PowerBI [3].

El dashboard de PowerBI muestra datos a diferentes niveles. Desde la analítica de

hate por días hasta los datos que se están recogiendo en tiempo real por el sistema. Los paneles se han diseñado de modo que sean usables e intuitivos, intentando optimizar al máximo la información que aquí se muestra.

El servidor web, por otro lado, muestra los dashboard de PowerBI y permite consultar directos de YouTube. De estos directos se toma información en tiempo real de los comentarios que se producen en el chat y se analiza la cantidad de hate que se está produciendo en ese momento.

Por otro lado, en Jupyter se han extraído diferentes gráficas que permiten diferenciar la cantidad de hate que se produce en diferentes comunidades de las múltiples redes sociales. Además, se han extraído gráficas que permiten analizar la cantidad de hate que se produce por día, por día festivo o de cada red social.

3. Conclusiones

Este proyecto ha surgido de la necesidad de prevenir problemas de salud mental provocados por el hate. Para ello, se ha desarrollado un sistema que analiza diferentes redes sociales y predice qué comentarios son de hate. Además, el sistema genera analíticas con la cantidad de hate por día.

Para diseñar y desarrollar este sistema se han tenido en cuenta y utilizado herramientas que se han visto en el curso de Big Data de SAMSUNG. Además, con el objetivo de dotar de realismo al sistema, se han diseñado y configurado herramientas como un clúster HDFS y Hive.

Los datos son obtenidos a través de técnicas de web scrapping y llamadas a APIs de las diferentes redes sociales (YouTube, YouTube Live, TikTok, Reddit, 4chan e Instagram) y son analizados y procesados mediante diferentes herramientas que hemos visto a lo largo del curso (PySpark, Pandas, Apache NiFi, ...).

Además, este sistema incluye un sistema de visualización de los datos tanto históricos como de aquellos que se están tomando en tiempo real. Esto permite a una persona, empresa, o colectivo analizar en qué días es más factible hacer publicaciones con el objetivo de recibir menos hate, ayudando a prevenir problemas de salud mental.

De este proyecto se pueden obtener diferentes conclusiones, como la importancia de realizar una buena planificación o de repartir tareas de modo lógico para que

todos los integrantes del grupo estén motivados mientras desarrollan este proyecto. Sin embargo, debemos destacar que el apartado de la salud mental es muy sensible y que los proyectos de esta índole deberían desarrollarse con apoyo de psicólogos y profesionales en la materia, basándose en evidencias científicas.

Debido a ello, aunque este proyecto sea un prototipo, podría suponer un punto de partida muy interesante de cara a realizar estudios que eviten que las personas sufran hate, teniendo en cuenta qué días se produce más hate y por qué. Es un proyecto que podría servir de base para avanzar más en el campo de la salud mental.

Anexo A (Enlaces de interés)

[1] Enlace al repositorio público de GitHub: https://github.com/aleo-phd/MOD-Samsung

[2] Documentación técnica del proyecto: https://github.com/aleo-phd/MOD-Samsung#readme

[3] Dashboard de PowerBI: