

# 2020-1

Implementación de modelos supervisados y no supervisados para la identificación de comentarios tóxicos en la red social Twitter.



Carlos Alberto Murillo, Juan David Zapata, Luz Stella Flórez, Cindy Paola Guerra, Diana Carolina Benjumea, Mónica Zapata.  
UNIVERSIDAD EAFIT  
2020-1

# **Implementación de modelos supervisados y no supervisados para la identificación de comentarios tóxicos en la red social Twitter**

**JIGSAW Clasificación multilingüe de comentarios tóxicos  
Competencia de Kaggle  
Proyecto Integrador 1 (PI1) – 2020/I**

Carlos Alberto Murillo, [cmurill5@eafit.edu.co](mailto:cmurill5@eafit.edu.co)  
Juan David Zapata, [jzapata6@eafit.edu.co](mailto:jzapata6@eafit.edu.co)  
Luz Stella Flórez, [lflorezs@eafit.edu.co](mailto:lflorezs@eafit.edu.co)  
Cindy Paola Guerra [cpguerram@eafit.edu.co](mailto:cpguerram@eafit.edu.co)  
Diana Carolina Benjumea, [dcbenjumeh@eafit.edu.co](mailto:dcbenjumeh@eafit.edu.co)  
Mónica Zapata, [mzapat29@eafit.edu.co](mailto:mzapat29@eafit.edu.co)

## **Abstract**

Cada vez el dominio del internet y su influencia en la vida de las personas es más grande. Las “redes sociales” permiten conectividad y facilitan la comunicación bidireccional entre las personas a través de un medio virtual que con el pasar del tiempo cobra más fuerza. Pero, con cada oportunidad se generan nuevas amenazas, en este caso la amenaza de abuso y acoso en línea se hace más fuerte, significando que las personas dejan de expresarse. Las plataformas luchan para facilitar de manera efectiva las conversaciones, lo que lleva a muchas comunidades a limitar o cerrar por completo los comentarios de los usuarios.

Esta iniciativa está trabajando en herramientas para ayudar a mejorar la conversación en línea, comprendiendo que un área de enfoque es el estudio de los comportamientos negativos en línea, como los comentarios tóxicos (es decir, los comentarios que son groseros, irrespetuosos o que de otra manera pueden hacer que alguien deje una discusión). En este proyecto se tiene el desafío de construir un modelo, que sea capaz de detectar diferentes tipos de toxicidad como amenazas, obscenidades, insultos y odio. Utilizando un conjunto de datos de comentarios de las ediciones de la página de discusión de Wikipedia. Se espera que las mejoras en el modelo actual ayuden a que la discusión en línea sea más productiva y respetuosa.

**Key words:** Comentarios, toxicos, redes sociales.

## Abstract

Every time the dominance of the internet and its influence on people's lives is greater. The "social networks" allow connectivity and facilitate two-way communication between people through a virtual medium that over time becomes more powerful. But, with every opportunity there are new threats, in this case the threat of abuse and harassment online means that many people stop expressing themselves and give up on seeking different opinions. Platforms struggle to effectively facilitate conversations, leading many communities to limit or completely shut down user comments.

This initiative is working on tools to help improve online conversation. One area of focus is the study of negative online behaviors, like toxic comments (i.e. comments that are rude, disrespectful, or otherwise likely to make someone leave a discussion). In this project, the challenge is to build a multi-headed model that's capable of detecting different types of toxicity like threats, obscenity, insults, and identity-based hate. Using a dataset of comments from Wikipedia's talk page edits. Improvements to the current model will hopefully help online discussion become more productive and respectful.

**Key words:** Comments, toxic, social networks.

# Contenido

INTRODUCCIÓN.....	8
La red social Twitter .....	8
Contexto y descripción del Problema .....	8
El reto de Kaggle.....	9
El Natural Language Processing .....	9
Metodologías utilizadas .....	10
Generalidades de los datos.....	11
Modelos propuestos por el Reto de Kaggle .....	12
Arquitecturas CNN .....	12
Arquitecturas RNN .....	12
Tipos de RNN.....	12
Modelos utilizados .....	13
Naive Bayes Bernoulli.....	13
Naïve Bayes Multinomial .....	14
K-Means .....	15
Logistic Regression .....	16
Arquitectura .....	16
Desarrollo .....	21
Extracción de datos de Twitter .....	21
Implementación .....	22
Publicación de archivo a s3 .....	22
Estructura archivo de salida .....	23
Modelo 1.1 - Modelo de Regresión Logística Multinomial para cada clase.....	24
Modelos 1.2 y 1.3.....	25
Multinomial NB para selección de unica clases .....	26
Modelo Multinomial NB para múltiples clases .....	26
Bernoulli NB para cada clase .....	27
Criterio de selección del modelo.....	28
Modelo no supervisado de clústerización (K-means) .....	30
Identificación del perfil toxico para crear la variable de clasificación.....	30
Modelo supervisado de clasificación .....	40
Modelo de Regresión logística multinomial.....	42

Modelo Árbol de clasificación .....	45
Evaluación de los modelos .....	47
Conclusiones .....	50
Referencias.....	52



## Ilustraciones

Ilustración 1. Ciclo de vida de CRISP DM tomado de IBM, 2020 .....	11
Ilustración 2. Arquitectura de la Solución .....	17
Ilustración 3. Buckets S3 .....	18
Ilustración 4. Instancia Amazon SageMaker .....	19
Ilustración 5. Catalogo Amazon Glue .....	20
Ilustración 6. Amazon Athena .....	20
Ilustración 7. Extracción de datos de Twitter.....	21
Ilustración 8. Extracción de datos de Twitter.....	21
Ilustración 9. Código para extracción de datos de Twitter .....	22
Ilustración 10. Publicación de archivo a s3 .....	23
Ilustración 11. Publicación de archivo a s3 .....	23
Ilustración 12. Calificación datos twitter.....	26
Ilustración 13 - Muestra datos entrenamiento .....	30
Ilustración 14 - Tipos de datos herramienta IBM SPSS Modeler.....	31
Ilustración 15 - Muestra de información a analizar .....	31
Ilustración 16 - Resumen modelo K-medias.....	32
Ilustración 17 - distribución de clústers .....	33
Ilustración 18 - comparación de clústers .....	38
Ilustración 19 - kmeans con k = 5 .....	38
Ilustración 20 - Kmeans con K=3 .....	39
Ilustración 21 - Nube de palabras más tóxicas para datos de entrenamiento .....	40
Ilustración 22. Base de datos de twitter. ....	41
Ilustración 23. Correlaciones de Pearson.....	41
Ilustración 24. Selección del modelo a implementar .....	42
Ilustración 25. Número condición de los datos de entrenamiento.....	43
Ilustración 26. Parámetros de las ecuaciones de la regresión .....	43
Ilustración 27. Matriz de confusión para la muestra de entrenamiento. ....	44
Ilustración 28. Matriz de confusión para la muestra de comprobación .....	44
Ilustración 29. Ajuste para la selección del modelo.....	44
Ilustración 30. Residuales SSE del modelo. ....	45
Ilustración 31. Impureza de Gini .....	46
Ilustración 32. Calificación del modelo. ....	46
Ilustración 33. Árbol de decisión generado.....	47
Ilustración 34. Eficiencia de los modelos implementados. ....	47
Ilustración 35. Datos de twitter clasificados bajo el modelo 1. ....	48
Ilustración 36. Datos de twitter clasificados bajo el modelo 1 .....	48
Ilustración 37. Aplicación del modelo de árbol de decisión .....	49
Ilustración 38. Clasificación de texto de twitter.....	49
Ilustración 39. Palabras más representativas.....	50

## Tablas

Tabla 1. Estructura archivo de salida comentarios twitter .....	23
Tabla 2. Calificación del modelo de regresión logística multinomial .....	25
Tabla 3. Definición de la clase .....	26
Tabla 4. Calificación del modelo multinomial para múltiples clases .....	27
Tabla 5. Calificación del modelo de Bernoulli para múltiples clases .....	27
Tabla 6. Comparación de calificaciones de precisión datos de testeo de todos los modelos implementados .....	29
Tabla 7. Comparación de calificaciones de validación cruzada de todos los modelos implementados .....	29
Tabla 8. Comparación de calificaciones de precisión datos de prueba de todos los modelos implementados .....	29
Tabla 9 - Variables a utilizar .....	30

## INTRODUCCIÓN

### La red social Twitter

Twitter fue creado por Jack Dorsey, Noah Glass, Biz Stone y Evan Williams en marzo de 2006 y se lanzó en julio de ese año. Para 2012, más de 100 millones de usuarios publicaron 340 millones de tweets al día.

Twitter es un sistema de 'microblogging' que permite enviar y recibir publicaciones cortas llamadas tweets, que pueden tener hasta 140 caracteres e incluir enlaces a sitios web y recursos relevantes.

Se pueden crear tweets propios o retuitear información que otros tuitearon. Retweetear significa que la información se puede compartir de manera rápida y eficiente con un gran número de personas (Economic and Social Research Council, 2020).

### Contexto y descripción del Problema

Hoy, la discusión en línea se ha convertido en una parte integral de las interacciones de las personas, por ejemplo, comentando interfaces, discutiendo en las diferentes plataformas populares de redes sociales tales como Facebook o Twitter. Por desgracia, estos canales también sufren abusos en la forma de acoso en línea a través de los llamados comentarios tóxicos (Sterckx, 2017).

Un informe de Pew 2014 destaca que el 73% de los usuarios adultos de Internet han visto a alguien acosado en línea, y el 40% lo ha experimentado personalmente (Duggan, 2014 en Sterckx, 2017). La amenaza de abuso, acoso y hostigamiento en línea implica que las personas dejan de expresarse y dejan de buscar opiniones diferentes. Los comentarios tóxicos son comentarios groseros, irrespetuosos o que pueden hacer que alguien deje una discusión. En casos extremos, el hostigamiento lleva a problemas emocionales y psicológicos.

Actualmente, las plataformas en línea luchan por monitorear efectivamente las conversaciones en busca de comportamientos tóxicos, en la medida en que muchas comunidades limitan o cierran por completo los comentarios de los usuarios. Por lo tanto, la identificación rápida de los comentarios tóxicos y la predicción en tiempo real es de suma importancia, para evitar los efectos perjudiciales de dicho comportamiento tóxico en los usuarios de Internet bien informados (Sterckx, 2017).

La clasificación de texto en su forma general, es un tema clásico para el procesamiento del lenguaje natural y un componente esencial en muchas aplicaciones, como la búsqueda web, el filtrado de información, la categorización de temas y el análisis de sentimientos (Sterckx, 2017).

Como resultado, se ha aplicado una amplia gama de metodologías de aprendizaje automático para la clasificación de texto. La investigación reciente en el área, se ha centrado principalmente en la aplicación de arquitecturas de redes neuronales (NN). La abundancia de arquitecturas NN para la clasificación de texto incluye dos grupos dominantes: redes neuronales convolucionales (CNN) y redes neuronales recurrentes (RNN) (Sterckx, 2017).

### El reto de Kaggle

Kaggle, una subsidiaria de Google LLC, es una comunidad en línea de científicos de datos y profesionales del aprendizaje automático que permite a los usuarios encontrar y publicar conjuntos de datos, explorar y construir modelos en un entorno de ciencia de datos basado en la web, trabajar con otros científicos de datos e ingenieros de aprendizaje automático y participar en concursos para resolver desafíos de ciencia de datos.

Kaggle comenzó en 2010 ofreciendo competencias de aprendizaje automático y ahora también ofrece una plataforma de datos pública, un banco de trabajo basado en la nube para la ciencia de datos y educación en inteligencia artificial (Kaggle, 2020).

Para esta competencia, el equipo de conversación AI, una iniciativa de investigación fundada por Jigsaw y Google, desarrolla tecnología para proteger las voces en la conversación online. Un área principal de enfoque son los modelos de aprendizaje automático que pueden identificar la toxicidad en las conversaciones en línea, donde la toxicidad se define como algo grosero, irrespetuoso o que de otra manera pueda hacer que alguien deje una discusión. Si se pueden identificar estas contribuciones tóxicas, podríamos tener un Internet más seguro y colaborativo.

A medida que los recursos informáticos y capacidades de modelado crecen, también lo hace el potencial para apoyar conversaciones saludables en todo el mundo. Esta competencia pretende que equipos de diversas partes del mundo (y dentro de los cuales nos encontramos nosotros), desarrollen estrategias para construir modelos multilingües eficaces para ayudar a la AI de conversación.

El desafío plantea la creación de un modelo de clasificación de etiquetas múltiples que sea capaz de detectar diferentes tipos de toxicidad como amenazas, obscenidades, insultos y odio basado en la identidad, proporcionando estimaciones de probabilidad para cada subtipo.

### Natural Language Processing

El procesamiento del lenguaje natural (PNL) es una rama de la inteligencia artificial que ayuda a las computadoras a comprender, interpretar y manipular el lenguaje humano. PNL

se basa en muchas disciplinas, incluidas las ciencias de la computación y la lingüística computacional, en su búsqueda para llenar el vacío entre la comunicación humana y la comprensión de la computadora ([https://www.sas.com/en\\_us/insights/analytics/what-is-natural-language-processing-nlp.html](https://www.sas.com/en_us/insights/analytics/what-is-natural-language-processing-nlp.html))

El procesamiento del lenguaje natural ayuda a las computadoras a comunicarse con los humanos en su propio idioma y escala otras tareas relacionadas con el lenguaje. Por ejemplo, la PNL hace posible que las computadoras lean texto, escuchen el discurso, lo interpreten, midan el sentimiento y determinen qué partes son importantes. El procesamiento del lenguaje natural incluye muchas técnicas diferentes para interpretar el lenguaje humano, desde métodos estadísticos y de aprendizaje automático hasta enfoques algorítmicos y basados en reglas. Necesitamos una amplia gama de enfoques porque los datos basados en texto y voz varían ampliamente, al igual que las aplicaciones prácticas.

Las tareas básicas de PNL incluyen tokenización y análisis, lematización / derivación, etiquetado de parte del discurso, detección de lenguaje e identificación de relaciones semánticas. Si alguna vez diagramaste oraciones en la escuela primaria, ya has realizado estas tareas manualmente antes.

En términos generales, las tareas de PNL dividen el lenguaje en piezas elementales más cortas, intentan comprender las relaciones entre las piezas y exploran cómo funcionan juntas para crear significado. Estas tareas subyacentes a menudo se usan en capacidades de nivel superior de PNL, como: Categorización de contenido. Un resumen de documento basado en la lingüística, que incluye búsqueda e indexación, alertas de contenido y detección de duplicación. Análisis de los sentimientos. Identificar el estado de ánimo u opiniones subjetivas dentro de grandes cantidades de texto, incluido el sentimiento promedio y la minería de opinión.

En todos estos casos, el objetivo general es tomar una entrada de lenguaje sin procesar y usar lingüística y algoritmos para transformar o enriquecer el texto de tal manera que ofrezca un mayor valor ([https://www.sas.com/en\\_us/insights/analytics/what-is-natural-language-processing-nlp.html](https://www.sas.com/en_us/insights/analytics/what-is-natural-language-processing-nlp.html))

## Metodologías utilizadas

### CRISP-DM:

Denota el “Proceso estándar de la industria cruzada para la minería de datos” por sus siglas en inglés, es una forma probada en la industria para guiar los esfuerzos de minería de datos (IBM, 2020).

- Como metodología, incluye descripciones de las fases típicas de un proyecto, las tareas involucradas en cada fase y una explicación de las relaciones entre estas tareas.

- Como modelo de proceso, CRISP-DM proporciona una visión general del ciclo de vida de la minería de datos.

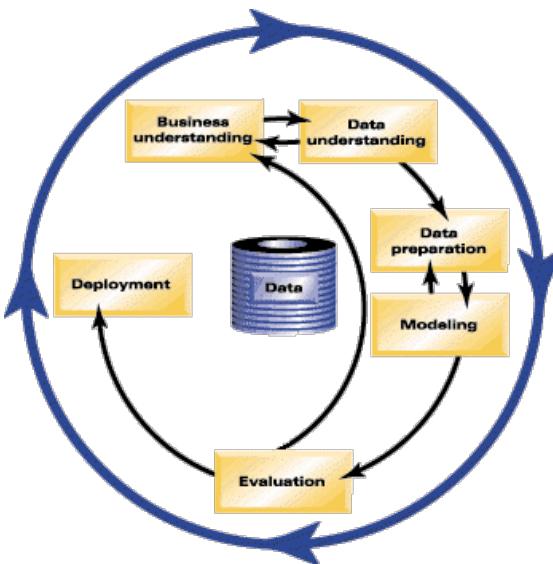


Ilustración 1. Ciclo de vida de CRISP DM tomado de IBM, 2020.

El modelo del ciclo de vida consta de seis fases con flechas que indican las dependencias más importantes y frecuentes entre las fases. La secuencia de las fases no es estricta. De hecho, la mayoría de los proyectos avanzan y retroceden entre fases según sea necesario.

El modelo CRISP-DM es flexible y se puede personalizar fácilmente. Por ejemplo, si su organización tiene como objetivo detectar el lavado de dinero, es probable que analice grandes cantidades de datos sin un objetivo de modelado específico. En lugar de modelar, su trabajo se centrará en la exploración y visualización de datos para descubrir patrones sospechosos en los datos financieros. CRISP-DM le permite crear un modelo de minería de datos que se adapte a sus necesidades particulares (IBM, 2020).

## Fuentes de datos

Los datos fueron proporcionados por Kaggle (<https://www.kaggle.com/c/jigsaw-multilingual-toxic-comment-classification/data>), constan de 9 archivos csv con un peso total de 4.69 GB.

## Generalidades de los datos

Los datos principales para la competencia son, en cada archivo proporcionado, la columna comment\_text, la cual contiene el texto de un comentario que se ha clasificado como tóxico

o no tóxico (0 ... 1 en la columna de tóxicos). Los comentarios del conjunto de entrenamiento están totalmente en inglés y provienen de comentarios civiles o ediciones de la página de discusión de Wikipedia. La columna `comment_text` de los datos de prueba, se componen de varios idiomas que no están en inglés.

## Modelos propuestos por el Reto de Kaggle

### Arquitecturas CNN

Redes neuronales convolucionales (CNN), originalmente propuestas por Lecun et al. (1998) son redes neuronales que modelan la estructura interna de datos como la estructura bidimensional de los datos de imagen a través de capas convolucionales, donde cada unidad de cálculo responde a una pequeña región de datos de entrada (por ejemplo, un cuadrado pequeño de una imagen grande). La esencia de las CNN es convertir pequeñas regiones de datos en vectores de características para su uso en las capas superiores: una capa convolucional aprende a transformar pequeñas regiones de datos en vectores de características de nivel superior.

La arquitectura propuesta por Kim et al. (Kim, 2014) se ha convertido en uno de los métodos más aplicados para aplicar CNNs al texto y ha demostrado ser efectivo en muchas ocasiones, logrando resultados de última generación en la clasificación de sentimientos.

### Arquitecturas RNN

Las redes neuronales recurrentes (RNN) (Elman, 1998) procesan secuencias de longitud arbitraria aplicando recursivamente una función de transición no lineal a un vector de estado oculto interno  $h_t$  y al elemento de entrada actual  $x_t$ .

## Tipos de RNN

### LSTM

La red de memoria a largo plazo (LSTM) fue propuesta por primera vez por Hochreiter et al. (1997) para abordar específicamente este problema del aprendizaje de dependencias a largo plazo en RNN de vainilla. El LSTM mantiene una celda de memoria separada en su interior que actualiza y expone su contenido solo cuando es necesario.  $h_t$  es un estado oculto "candidato" que se calcula en función de la entrada actual y el estado oculto anterior  $h_{t-1}$ . El LSTM contiene tres puertas: las puertas de entrada, de olvido y de salida.  $c_t$  es la memoria interna o el vector de contexto de la unidad. Es una combinación de la memoria anterior  $c_{t-1}$  multiplicada por la puerta de olvido y el estado oculto recién calculado  $h_t$ , multiplicado por la puerta de entrada.

### *GRU*

Una unidad recurrente cerrada (GRU) (Cho et al., 2014) tiene solo dos puertas, una puerta de reinicio y una puerta de actualización, y por lo tanto tiene una menor cantidad de parámetros.

La puerta de reinicio determina cómo combinar la nueva entrada con la memoria anterior, y la puerta de actualización define la cantidad de memoria anterior que se debe retener. Las GRU no poseen estados de contexto interno ( $ct$ ) que difieren del estado oculto expuesto, y no tienen la puerta de salida que está presente en los LSTM. Las puertas de entrada y de olvido están acopladas por una puerta de actualización y se aplica la puerta de reinicio directamente al estado oculto anterior. Por lo tanto, en un GRU, la funcionalidad de la puerta de reinicio LSTM se divide realmente en la puerta de reinicio y actualización.

### *BiRNN*

Los RNN bidireccionales (BiRNN) codifican la secuencia considerada (es decir, el comentario) después de procesarla en una dirección hacia adelante y hacia atrás por un RNN diferente y concatenar los estados ocultos separados.

## Modelos utilizados

Decidimos utilizar algunos modelos vistos en clase y explorar otras variaciones. Según la Universidad de Standford (2011), los métodos de clasificación de texto más comunes dentro del aprendizaje automático supervisado son:

- Naïve Bayes
- Logistic regression
- Support-vector machines
- k-Nearest Neighbors

De los cuáles escogimos Naïve Bayes, Logistic regression y K-Means, adicionando además un Naïve Bayes Multinomial y un Naïve Bayes Bernoulli.

### Naive Bayes Bernoulli

En el modelo de evento de Bernoulli multivariado, un documento es un vector binario sobre el espacio de las palabras. Dado un vocabulario  $V$ , cada dimensión del espacio  $t$ ,  $t \in \{1, \dots, |V|\}$ , corresponde a la palabra  $wt$  del vocabulario. La dimensión  $t$  del vector para el documento  $d$  se escribe  $Bit$  y es 0 o 1, lo que indica si la palabra  $wt$  ocurre al menos una vez en el documento. Con esta representación documental, se toman los supuestos del Naïve Bayes: que la probabilidad de que cada palabra aparezca en un documento es independiente de la aparición de otras palabras en un documento. Entonces, la probabilidad de un documento dado su clase de la ecuación 1 es simplemente el producto

de la probabilidad de los valores de atributo sobre todos los atributos de palabras (McCallum& Nigam, 2002):

$$P(d_i|c_j; \theta) = \prod_{t=1}^{|V|} (B_{it} P(w_t|c_j; \theta) + (1 - B_{it})(1 - P(w_t|c_j; \theta))).$$

### Naïve Bayes Multinomial

El término general **Naïve Bayes** se refiere a los fuertes supuestos de independencia en el modelo, más que a la distribución particular de cada característica (Russell&Norvig, 2003).

Un modelo de Naïve Bayes supone que cada una de las características que utiliza son condicionalmente independientes entre sí dada alguna clase. Más formalmente, si se desea calcular la probabilidad de observar características  $f_1 f_1 \dots f_n f_n$ , dada alguna clase  $c$ , bajo el supuesto Naïve Bayes, se debe cumplir:

$$p(f_1, \dots, f_n | c) = \prod_{i=1}^n p(f_i | c)$$

En la práctica los modelos Naive Bayes se han desempeñado sorprendentemente bien, incluso en tareas complejas donde está claro que los supuestos de independencia son falsos (Russell&Norvig, 2003).

En contraste con el modelo de evento de Bernoulli multivariado, el modelo multinomial captura información de frecuencia de palabras en documentos (McCallum& Nigam, 2002).

El término **Multinomial Naive Bayes** simplemente permite saber que cada  $p(f_i | c)$  es una distribución multinomial, en lugar de alguna otra distribución. Esto funciona bien para los datos que pueden convertirse fácilmente en recuentos, como el recuento de palabras en un texto.

En el modelo multinomial, un documento es una secuencia ordenada de eventos de palabras, extraída del mismo vocabulario  $V$ . Asumimos que la longitud de los documentos es independiente de la clase.<sup>2</sup> Nuevamente hacemos una suposición ingenua similar de Bayes: que la probabilidad de cada evento de palabra en un documento es independiente del contexto y la posición de la palabra en el documento. Por lo tanto, cada documento  $d_i$  se extrae de una distribución multinomial de palabras con tantos ensayos independientes como la longitud de  $d_i$ . Esto produce la representación familiar de "bolsa de palabras" para los documentos. Defina  $N_{it}$  como el recuento del número de veces que la palabra  $w_t$  aparece en el documento  $d_i$  (McCallum& Nigam, 2002).

Entonces, la probabilidad de un documento dada su clase es simplemente la distribución multinomial (McCallum & Nigam, 2002):

$$P(d_i|c_j; \theta) = P(|d_i|)|d_i|! \prod_{t=1}^{|V|} \frac{P(w_t|c_j; \theta)^{N_{it}}}{N_{it}!}.$$

En resumen, el clasificador Naive Bayes es un término general que se refiere a la independencia condicional de cada una de las características del modelo, mientras que el clasificador Multinomial Naive Bayes es una instancia específica de un clasificador Naive Bayes que utiliza una distribución multinomial para cada una de las características (Russell & Norvig, 2003).

### K-Means

Los algoritmos de agrupamiento son herramientas útiles para la minería de datos, la compresión, la estimación de densidad de probabilidad y muchas otras tareas importantes, entre ellos se encuentra el algoritmo k-means, el cuál es conocido por su eficiencia en la agrupación de grandes conjuntos de datos.

Este algoritmo es de carácter iterativo, e intenta dividir el conjunto de datos en K subgrupos distintos no superpuestos (clusters) donde cada punto de datos pertenece a un solo grupo. K-means intenta hacer que los puntos de datos dentro del clúster sean lo más similares posible, al tiempo que mantiene los clústeres lo más diferentes o lo más lejos posible. Asigna puntos de datos a un grupo de modo que la suma de la distancia al cuadrado entre los puntos de datos y el centroide del grupo (media aritmética de todos los puntos de datos que pertenecen a ese grupo) es mínima. Cuanta menos variación tengamos dentro de los grupos, más homogéneos (es decir, similares) serán los puntos de datos dentro del mismo grupo (Towards Data Science, 2020).

La forma en que funciona el algoritmo kmeans es la siguiente:

1. Se especifica el número de grupos K.
2. Se inician los centroides, primero barajando el conjunto de datos y luego seleccionando aleatoriamente K puntos de datos para los centroides sin reemplazo.
3. Se continúa iterando hasta que no haya cambios en los centroides. Es decir, la asignación de puntos de datos a grupos no está cambiando.
4. Se calcula la suma de la distancia al cuadrado entre los puntos de datos y todos los centroides.
5. Se asigna cada punto de datos al grupo más cercano (centroide).
6. Se calculan los centroides para los grupos tomando el promedio de todos los puntos de datos que pertenecen a cada grupo.

El enfoque que kmeans sigue para resolver el problema se llama Expectation-Maximization. El E-step consiste en asignar los puntos de datos al clúster más cercano. El paso M calcula el centroide de cada grupo (Towards Data Science, 2020).

## Logistic Regression

La regresión logística es un algoritmo de clasificación utilizado para asignar observaciones a un conjunto discreto de clases. Algunos de los ejemplos de problemas de clasificación son correo electrónico no deseado o no, transacciones en línea fraude o no fraude, tumores malignos o benignos. La regresión logística transforma su salida usando la función sigmoide logística para devolver un valor de probabilidad (Towards Data Scienceb, 2020).

Hay diferentes tipos de regresiones logísticas:

- Binario (por ejemplo, Tumor maligno o benigno)
- Las funciones multilineales (por ejemplo, gatos, perros u ovejas)

La regresión logística es un algoritmo de aprendizaje automático que se utiliza para los problemas de clasificación, es un algoritmo de análisis predictivo y se basa en el concepto de probabilidad.

Podemos llamar a una regresión logística un modelo de regresión lineal, pero la regresión logística utiliza una función de costo más compleja, esta función de costo puede definirse como la "función sigmoidea" o también conocida como la "función logística" en lugar de una función lineal.

La hipótesis de la regresión logística tiende a limitar la función de costo entre 0 y 1. Por lo tanto, las funciones lineales no lo representan, ya que puede tener un valor mayor que 1 o menor que 0, lo que no es posible según la hipótesis de la regresión logística.

La función sigmoide asigna cualquier valor real a otro valor entre 0 y 1. En el aprendizaje automático, utilizamos sigmoide para asignar predicciones a probabilidades (Towards Data Scienceb, 2020).

## Arquitectura

Como arquitectura para la realización del proyecto se utilizó Amazon AWS, Buckets de S3 para la ingesta de datos y almacenamiento de modelos, Amazon Glue para la catalogación de resultado, Amazon Athena para búsquedas sobre los catálogos, y Amazon SageMaker como máquina de computo encargada de ejecutar Jupyter Notebooks y ejecutar los modelos.

La siguiente imagen ilustra la arquitectura descrita:

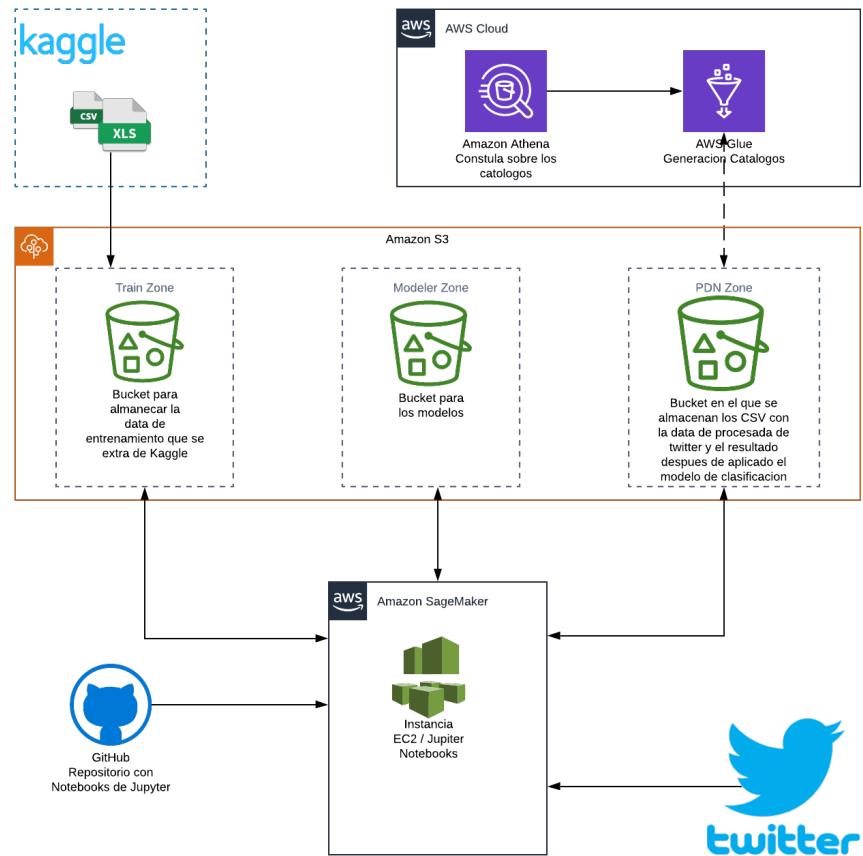


Ilustración 2. Arquitectura de la Solución

Como se aprecia en la *Ilustración 2*, se cuenta con 3 buckets en S3, los cuales cumplen los siguientes propósitos:

- Train Zone - Bucket pyint-train-zone ([clic aquí para acceder](#)): En este bucket se almacena toda la data que proporcionada por Kaggle para entrenar y probar los modelos de clasificación.
- Modeler Zone – Bucket pyint-pyint-modeler-zone ([clic aquí para acceder](#)): En este bucket se almacenan los modelos de clasificación que luego se usaran para clasificar los comentarios que se extraen de twitter.
- PDN Zone – Bucket pyint-pdn-zone ([clic aquí para acceder](#)): En este bucket se almacena la bases de datos de twitter y el posterior resultado de su clasificación.

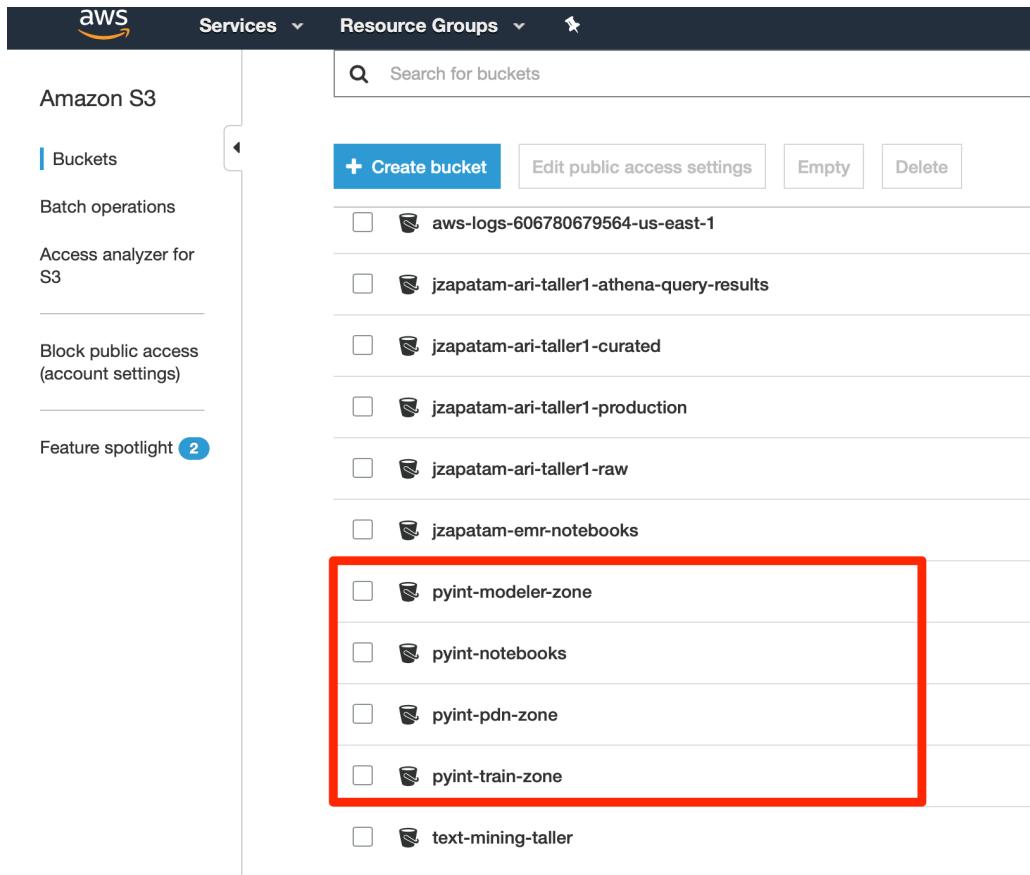


Ilustración 3. Buckets S3

Como instancia de computo se optó por utilizar Amazon SageMaker ([clic aquí para acceder a la instancia de Jupyter](#)) ya que nos permite en una instancia de EC2 contar con una instancia de JupyterLab que nos permite trabajar de forma colaborativa, crear notebooks de jupyter y ejecutar y analizar el resultado de dichos modelos a través de los notebooks.

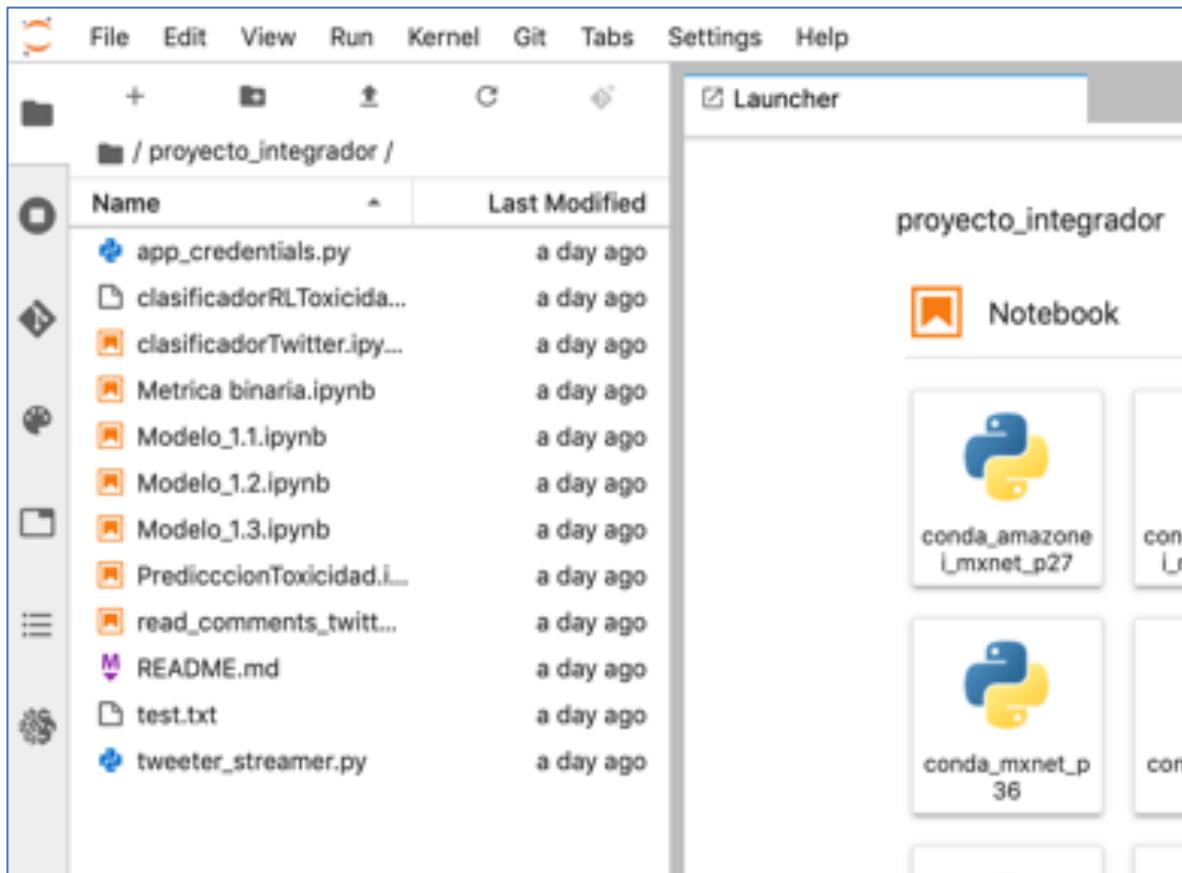


Ilustración 4. Instancia Amazon SageMaker

Adicional, dicha instancia de Amazon SageMaker se encuentra conectada al repositorio Git ([clic acá para acceder al repositorio de GitHub](#)) en el cual se encuentran todos los notebooks y códigos del proyecto, de manera que la instancia ya cuenta con los notebook mencionados

\*\* Para clonar el repositorio use la siguiente url:

[https://github.com/<repo>/proyecto\\_integrador.git](https://github.com/<repo>/proyecto_integrador.git)

Mediante la ejecución de los notebooks de Jupyter, la instancia de SageMaker se conecta con twitter y extraer los comentarios, lo procesa y genera una base de datos (archivo csv), lo almacena en el bucket de S3 pyint-pdn, dicha base de datos es el insumo para que el modelo de clasificación analice los tweets y se identifiquen los comentarios tóxicos.

Como componentes finales de esta arquitectura, tenemos Amazon Glue y Amazon Athena. Amazon Glue se utiliza para catalogo base de datos de los tweets y su clasificación por nivel de toxicidad, y Amazon Athena como el componente que permitirá realizar queries SQL sobre estos catálogos.

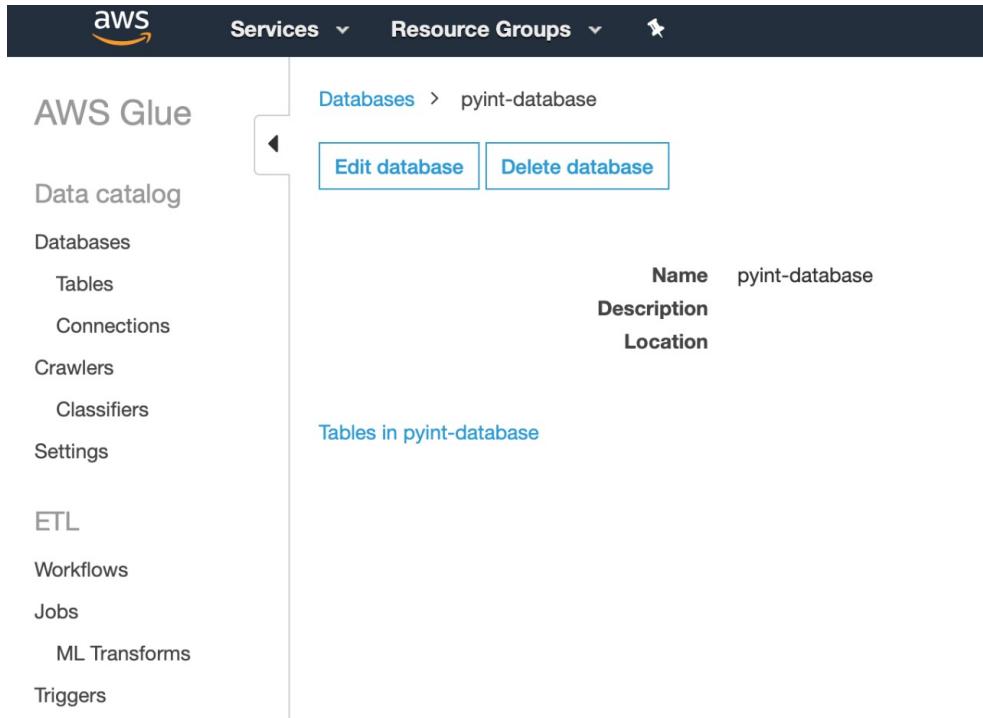


Ilustración 5. Catalogo Amazon Glue

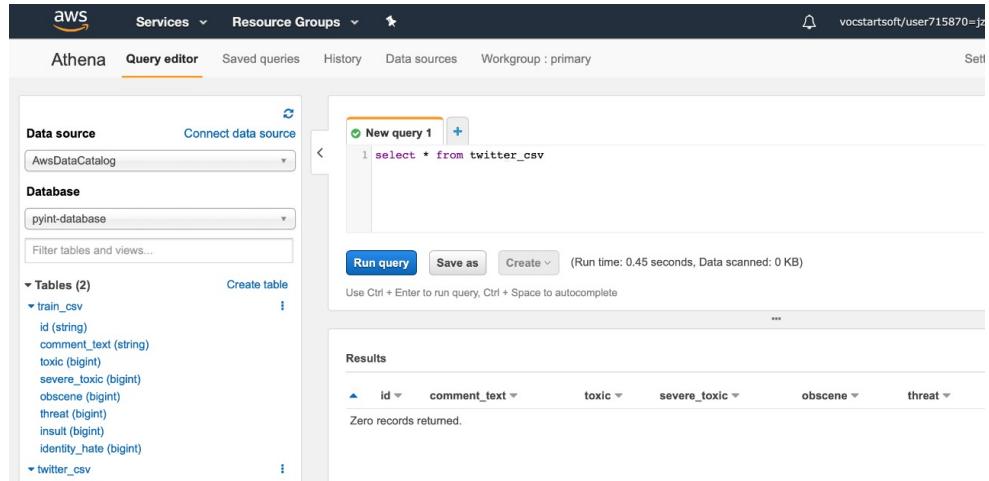
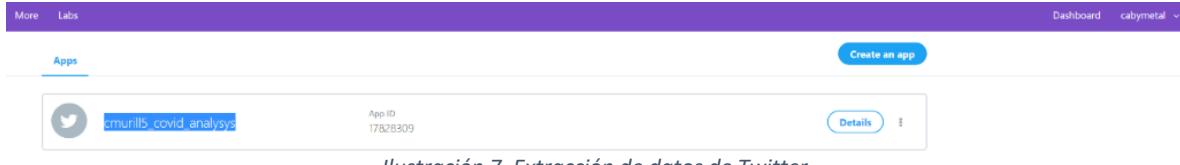


Ilustración 6. Amazon Athena

# Desarrollo

## Extracción de datos de Twitter

Lo primero que se pide es estar registrado como desarrollador en una cuenta de Twitter en: <https://developer.twitter.com>, una vez se realiza la inscripción procedemos a registrar una aplicación, mostramos imagen de una aplicación ya registrada.



La aplicación nos da unas llaves y tokens de acceso estos se deben almacenar localmente en un archivo que luego usaremos para la extracción:

A screenshot of the 'Keys and tokens' section of the Twitter Developer Dashboard. It shows two tabs: 'Keys and tokens' (which is selected) and 'Permissions'. Under 'Keys and tokens', there's a heading 'Consumer API keys'. It displays two API keys: 'API key: Abli3MJtzLyPQE7uer4HTaMsC' and 'API secret key: wOXjB3kZCN3dOYgtMnlq5eDJ1InnCW0cSYKgUDM1U82xzmCk2a'. There are 'Regenerate' buttons next to each key. Below this, there's a section for 'Access token &amp; access token secret'. It contains the text: 'We only show your access token and secret when you first generate it in order to make your account more secure. You can revoke or regenerate them at any time, which will invalidate your existing tokens.' It shows an 'Access token: xxxxxxxxxxxxxxxxxxxxxxxxx' and an 'Access token secret: xxx'. The 'Access level' is listed as 'Read and write'. There are 'Revoke' and 'Regenerate' buttons for this section. A note at the bottom right says 'Last generated: May 1, 2020'. A blue banner at the bottom of the page reads 'Illustración 8. Extracción de datos de Twitter'.

Una vez realizamos estos pasos procedemos con la implementación del código.

## Implementación

Para nuestra aplicación utilizamos Python para capturar los tweets utilizamos la librería Tweepy. El código funciona de una manera sencilla, crea dos hilos: Un hilo que se encarga de controlar el tiempo de ejecución y otro que se encarga de leer y escribir tweets en un archivo de texto.

Cuando el hilo principal que se encarga de controlar el tiempo de ejecución finaliza el tiempo de ejecución, detiene la captura de tweets y se procede a publicar los datos a S3.

```
def background(stream):
    stream.filter(track = ['George Floyd', 'Donald Trump', 'BlackLivesMatters'],
                  )

def wait(minutes):
    for i in tqdm(range(60*minutes)):#five minutes
        time.sleep(1) #update each second

if __name__ == '__main__':
    MINUTES = 30
    FILE_PATH = './'
    FILE_NAME = 'comments.csv'

    listener = StdOutListener(FILE_PATH, FILE_NAME,60*MINUTES)
    #listener = StdoutListener()
    auth = OAuthHandler(app_credentials.CONSUMER_KEY, app_credentials.CONSUMER_SECRET)
    auth.set_access_token(app_credentials.ACCESS_TOKEN, app_credentials.ACCESS_TOKEN)
    api = API(auth)

    stream = Stream(api.auth, listener)

    try:
        background(stream)
        #tweet_capturer = threading.Thread(name = 'background', target = background)
        #tweet_capturer.start()
        wait(MINUTES)
    except:
        stream.disconnect()

    #after creation of file publish to s3
    print("Publicando a bucket")
    S3_BUCKET = "cmurill5tmp"
    s3_client = boto3.client('s3')

    s3_client.upload_file(FILE_PATH+FILE_NAME,S3_BUCKET,FILE_NAME)

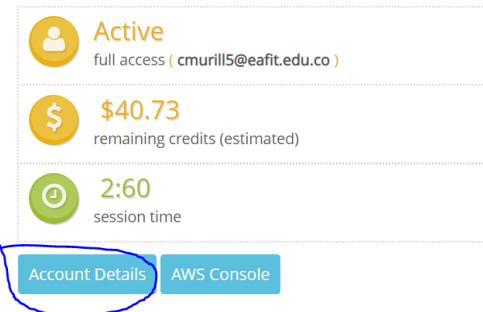
#finaliza ejecucion
```

Ilustración 9. Código para extracción de datos de Twitter

## Publicación de archivo a s3

Para que se pueda publicar este archivo a S3 se configura un archivo con las credenciales de acceso que obtenemos en la consola de AWS Educate y guardarlas de manera local en la carpeta. aws, con el nombre de credentials

## Your AWS Account Status



Please use AWS Educate Account responsibly. Remember to use it to make the best use of your credits. And, do work!

Ilustración 10. Publicación de archivo a s3

Esto permite realizar la publicación del archivo de resultado en S3 como se observa en la imagen:



Ilustración 11. Publicación de archivo a s3

## Estructura archivo de salida

El archivo de resultado está en formato csv y cuenta con la siguiente información:

Tabla 1. Estructura archivo de salida comentarios twitter

id	comment	creation_time	source	tweet_id	user	user_id
0	RT @NoSurrenderHK: @guardiannews The truth is HK now #HKPoliceState....	13/06/2020 16:07	Twitter for Android	1,2718E+18	currentec alamo	1,0059E+18
1	RT @MarisaKabas: Content warning: police brutality Video ...	13/06/2020 16:07	Twitter for Android	1,2718E+18	Donald Dire	1,0883E+18
2	RT @shahmiruk: It is absolutely unfair	13/06/2020 16:07	Twitter for iPhone	1,2718E+18	tanya cochrane 🕷 #FBPE	25872176

	& disingenuous ...				
--	-----------------------	--	--	--	--

Un id que marca un consecutivo de comentario, el texto del comentario, fecha de creación, origen o plataforma de la que se crea el tweet, usuario y el id de usuario.

Este archivo es que deberemos catalogar en pasos futuros.

### Modelo 1.1 - Modelo de Regresión Logística Multinomial para cada clase

Partiendo de datos procedentes de Wikipedia y referenciados en retos de Kaggle , que se encuentran previamente clasificados bajo las categorías “toxic”, “severe\_toxic”, “obscene”, “threat”, “insult”, “identity\_attack”. Se entrena un modelo de regresión logística multinomial con el objetivo de clasificar comentarios extraídos de twitter.

Como primer paso se leen los datos desde la librería **pandas** y se realiza un proceso de preparación de texto simple con el módulo **re**. Donde se unifica el texto bajo formato minúscula, se remueven caracteres especiales y signos de puntuación. Se opta por no efectuar lematización ni remoción de palabras de parada debido a que se pretende conservar el sentido de las oraciones.

Posterior a efectuar la preparación y limpieza del texto, se implementa una vectorización TF-IDF con el apoyo de la librería **sklear** de Python. Donde se definen los conjuntos de testeo y entrenamiento.

Finalmente, aplicando la librería **sklearn** se genera un ciclo para correr el modelo de regresión logística multinomial con el objetivo de lograr clasificaciones con más de dos posibles resultados discretos y se generan las calificaciones de validación cruzada (cross-validation “CV” score) y calificaciones de predicción (prediction score) para el modelo. Obteniendo los siguientes resultados.

Tabla 2. Calificación del modelo de regresión logística multinomial

	Class_Name	Roc_loss	Accuracy	Accuracy_vs_Test
0	toxic	0.969129	0.942029	0.793841
1	severe_toxic	0.983446	0.635575	0.610257
2	obscene	0.982496	0.936630	0.798781
3	threat	0.985455	0.824561	0.570396
4	insult	0.975673	0.863213	0.749949
5	identity_attack	0.971404	0.771654	0.573629

Es importante resaltar que se genera un ciclo, ya que el modelo implementado en realidad consta de seis modelos diferentes. Esto se debe a que para generar el grado de pertenencia del comentario a cada una de las seis clases definidas, es requerido correr un modelo independiente para cada clase.

Para efectos prácticos, se almacena el modelo en un archivo .pkl de python y el resultado de la clasificación del texto, se almacena en un archivo .csv

**Notebook: Modelo\_1\_1.ipynb**

## Modelos 1.2 y 1.3

En estos modelos la intención es asignarle una serie de características al comentario que nos permitan definir un grado de toxicidad, las variables que queremos clasificar son:

***toxic, severe\_toxic, insult, identity\_attack, threat***

Estas variables tienen valores de verdadero o falso {1, 0}. Con estos valores debemos clasificar la base de datos de Twitter uno de los primeros acercamientos que tuvimos para realizar un único modelo que clasificara un comentario en cada una de estas clases fue:

## Multinomial NB para selección de única clase

1. Crear una clase como la concatenación de todos los valores del conjunto de entrenamiento, de esta manera tenemos una clase que funciona como un número binario ej:

Tabla 3. Definición de la clase

numero	binario	significado
32	100000	tóxico
34	100010	tóxico, insult
40	101000	tóxico,obscene
42	101010	tóxico,obscene, insult

2. Donde 0 significa que no es para nada toxic y el número 63 es el mayor grado de toxicidad presente en la base.
3. Después procedemos a crear un modelo Multinomial Naive Bayes, dividiendo el conjunto de datos de entrenamiento en grupos de test y train
4. Probamos la precisión del modelo con un conjunto de datos de entrenamiento
5. Procedemos a calificar datos de twitter:

```
In [20]: df_test_merged.loc[df_test_merged['number'] == 32, 'comment_text'].values[1]  
Out[20]: 'gay fag fag fag'
```

Ilustración 12. Calificación datos twitter

El motivo por el cual descartamos este modelo a pesar de los buenos resultados fue, porque requeríamos que la base de predicción tuviera relacionada la probabilidad de pertenecer a cada clase y con este acercamiento solo se obtiene la probabilidad general. No se podría discriminar por clase.

Notebook: [Modelo\\_1\\_2.ipynb](#)

## Modelo Multinomial NB para múltiples clases

En el Modelo anterior generamos un único modelo para la predicción, basado en los buenos resultados obtenidos por el Modelo anterior, decidimos intentar este modelo para cada clase por separado en un ciclo esperando encontrar resultados muy similares.

El modelo Multinomial es más preciso para documentos de gran tamaño y la lógica de su funcionamiento se explica en otra sección del documento, pero aplicando de manera general el mismo procedimiento que el anterior obtuvimos los siguientes resultados:

*Tabla 4. Calificación del modelo multinomial para múltiples clases*

df_classification_report				
	Class_Name	Roc_MN	Accuracy_MN	Accuracy_MN_vs_Test
0	toxic	0.928747	0.948737	0.700372
1	severe_toxic	0.916472	0.420455	0.507131
2	obscene	0.929168	0.901398	0.680530
3	threat	0.803929	0.086207	0.506554
4	insult	0.921501	0.830539	0.630679
5	identity_hate	0.840786	0.186275	0.500863

Los resultados contra el conjunto de entrenamiento fueron muy poco precisos para calificar threat y severe\_toxic y contra el conjunto de test también por este motivo este modelo fue descartado.

## Bernoulli NB para cada clase

Realizando un procedimiento similar con Bernoulli en el mismo ciclo obtenemos:

*Tabla 5. Calificación del modelo de Bernoulli para múltiples clases*

Roc_BN	Accuracy_BN	Accuracy_BN_vs_Test
0.921285	0.211122	0.765914
0.940403	0.276540	0.548347
0.947607	0.187398	0.836873
0.659383	0.004228	0.500865
0.941699	0.182151	0.833506
0.844139	0.109932	0.520863

Observamos que los resultados son aún más bajos, en las mismas categorías por este motivo se decide descartar este modelo podemos ver el procedimiento más detallado en el Notebook

### Notebook: **Modelo\_1\_3.ipynb**

#### Criterio de selección del modelo.

Para entrenar y evaluar el proceso de clasificación bajo los modelos anteriores, se acude a separar el conjunto de datos iniciales ya clasificados, en un conjunto de datos de entrenamiento (train) y otro de datos de prueba (test). Para lo cual se cuenta con los siguientes grupos:

- X\_train: Registros de entrenamiento
- y\_train: Etiquetas de los resultados esperados de X\_train
- X\_test: Registros para efectuar la prueba
- y\_test: Etiquetas de los resultados esperados de X\_test

Es así que, de acuerdo a la variación en términos de exactitud alcanzada entre los resultados obtenidos y los resultados esperados en el conjunto de entrenamiento al correr los diferentes modelos, se obtiene una calificación de precisión que varía entre cero y uno [0-1]. La cual nombramos **Accuracy**.

**Roc\_loss:** De forma similar, con el fin de medir el comportamiento de los modelos implementados y apoyar en la selección de más adecuado, se acude a la técnica de validación cruzada. Donde, se evalúan los resultados, garantizando la independencia de la partición entre datos de entrenamiento y prueba.

En este proceso se repite y calcula la media aritmética obtenida de las medidas de evaluaciones sobre diferentes conjuntos (prueba, test), empleando una métrica ROC (Receiver Operating Characteristics).

Se selecciona la métrica ROC debido a que es ideal para maximizar la tasa positiva verdadera y minimizar la tasa de falsos positivos. En este proceso se calcula la curva ROC del modelo a partir de validación cruzada y se obtiene el área media bajo la curva, generando la varianza de la curva cuando el conjunto de entrenamiento se divide en sub-conjuntos. Esto muestra como la salida del clasificador puede verse afectada por los cambios en los datos de entrenamiento y cuan diferentes son las divisiones generadas por el proceso de validación cruzada entre sí.

**Accuracy\_vs\_test:** Esta calificación indica la exactitud entre los resultados obtenidos y los resultados esperados en el conjunto de datos de prueba tomado a partir de los datos iniciales ya clasificados.

Tabla 6. Comparación de calificaciones de precisión datos de testeo de todos los modelos implementados

Accuracy	Modelo			
Clase	Regresión logística multinomial para cada clase	Multinomial NB para selección de múltiples clases con único modelo	Multinomial NB para cada clase	Bernoulli NB para cada clase
<b>toxic</b>	0.942029	<b>0.948737</b>	0.9025	0.211122
<b>severe_toxic</b>	<b>0.634199</b>	0.420455		0.276540
<b>obscene</b>	<b>0.936645</b>	0.901398		0.187398
<b>threat</b>	<b>0.824561</b>	0.086207		0.004228
<b>insult</b>	<b>0.863213</b>	0.830539		0.182151
<b>Identity_attack</b>	<b>0.771654</b>	0.186275		0.109932

Tabla 7. Comparación de calificaciones de validación cruzada de todos los modelos implementados

Roc	Modelo		
Clase	Regresión logística multinomial para cada clase	Multinomial NB para selección de múltiples clases con único modelo	Bernoulli NB para cada clase
<b>toxic</b>	<b>0.969129</b>	0.928747	0.921285
<b>severe_toxic</b>	<b>0.983446</b>	0.916472	0.940403
<b>obscene</b>	<b>0.982496</b>	0.929168	0.947607
<b>threat</b>	<b>0.985455</b>	0.803929	0.659383
<b>insult</b>	<b>0.975673</b>	0.921501	0.941699
<b>Identity_attack</b>	<b>0.971404</b>	0.840786	0.844139

Tabla 8. Comparación de calificaciones de precisión datos de prueba de todos los modelos implementados

Accuracy_vs_test	Modelo		
Clase	Regresión logística multinomial para cada clase	Multinomial NB para selección de múltiples clases con único modelo	Bernoulli NB para cada clase
<b>toxic</b>	<b>0.793841</b>	0.700372	0.765941
<b>severe_toxic</b>	<b>0.610257</b>	0.507131	0.548347
<b>obscene</b>	<b>0.798781</b>	0.680530	0.836873
<b>threat</b>	<b>0.570396</b>	0.506554	0.500865
<b>insult</b>	0.749949	0.630679	<b>0.833506</b>
<b>Identity_attack</b>	<b>0.573629</b>	0.500863	0.520863

## Modelo no supervisado de clústerización (K-means)

### Identificación del perfil tóxico para crear la variable de clasificación

La base de datos jigsaw-unintended-bias-train.csv es la base seleccionada para realizar el entrenamiento y confirmación de los modelos de clasificación y no supervisado. En la base de datos contamos con las siguientes variables y 1'902.194 comentarios de Twitter:

Vista de los datos originales:

	id	comment_text	toxic	severe_toxicity	obscene	identity_attack	insult	threat	asian	atheist	...	article_id	rating	funny	wow
0	59848	This is so cool. It's like, 'would you want yo...	0.000000	0.000000	0.0	0.000000	0.000000	0.0	NaN	NaN	...	2006	rejected	0	0
1	59849	Thank you!! This would make my life a lot less...	0.000000	0.000000	0.0	0.000000	0.000000	0.0	NaN	NaN	...	2006	rejected	0	0
2	59852	This is such an urgent design problem; kudos to...	0.000000	0.000000	0.0	0.000000	0.000000	0.0	NaN	NaN	...	2006	rejected	0	0
3	59855	Is this something I'll be able to install on m...	0.000000	0.000000	0.0	0.000000	0.000000	0.0	NaN	NaN	...	2006	rejected	0	0
4	59856	haha you guys are a bunch of losers.	0.893617	0.021277	0.0	0.021277	0.87234	0.0	0.0	0.0	...	2006	rejected	0	0

Ilustración 13 - Muestra datos entrenamiento

Las variables seleccionadas a trabajar en el análisis no supervisado y de clasificación son:

Tabla 9 - Variables a utilizar

id	Descripción
comment_text	comentario texto
toxic	tóxico
severe_toxicity	toxicidad severa
obscene	obsceno
identity_attack	identidad
insult	insulto
threat	amenaza

Tipo de variables o medida:

Campo	Medida
id	Continuo
comment_text	Nominal
toxic	Continuo
severe_toxicity	Continuo
obscene	Continuo
identity_attack	Continuo
insult	Continuo
threat	Continuo

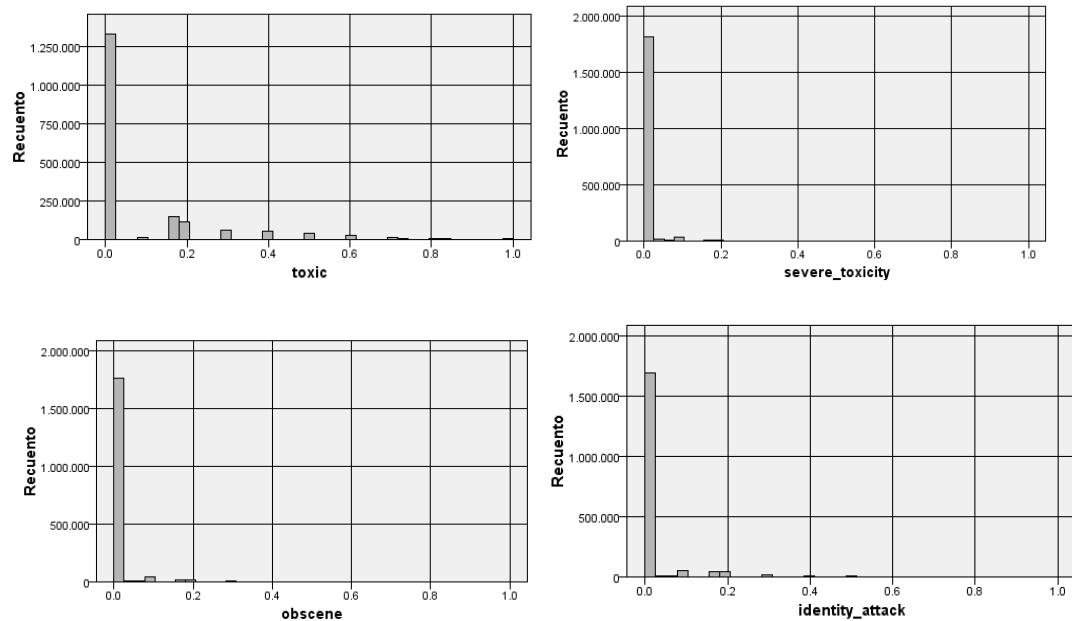
Ilustración 14 - Tipos de datos herramienta IBM SPSS Modeler

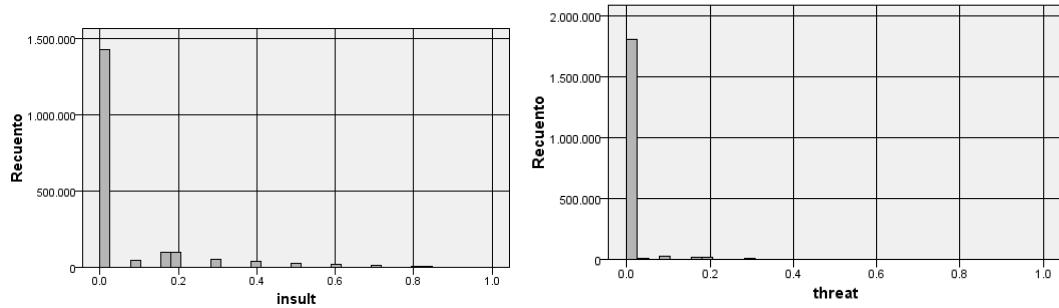
A continuación, se visualiza una muestra de la información que se trabajará en el análisis:

	comment_text	toxic	severe_toxicity	obscene	identity_attack	insult	threat
0	This is so cool. It's like, 'would you want yo...	0.000000	0.000000	0.0	0.000000	0.000000	0.0
1	Thank you!! This would make my life a lot less...	0.000000	0.000000	0.0	0.000000	0.000000	0.0
2	This is such an urgent design problem; kudos t...	0.000000	0.000000	0.0	0.000000	0.000000	0.0
3	Is this something I'll be able to install on m...	0.000000	0.000000	0.0	0.000000	0.000000	0.0
4	haha you guys are a bunch of losers.	0.893617	0.021277	0.0	0.021277	0.87234	0.0

Ilustración 15 - Muestra de información a analizar

Distribución de las variables a trabajar en los modelos estadísticos:





**Análisis de Conglomerados o clúster:** Agrupa aquellos objetos que reúnen idénticas características, es decir, se convierte en una técnica de análisis exploratorio diseñada para revelar las agrupaciones naturales dentro de una colección de datos.

**Método No Jerárquico: Conglomerado K-Medias:** Este procedimiento estadístico procura identificar grupos relativamente homogéneos de casos basados en características seleccionadas, usando un algoritmo que pueda manejar una gran cantidad de casos a través de la distancia Euclidea; donde cada punto es asignado con esta dictación a algún conglomerado, y las medias del conglomerado (centroides) son actualizadas como consecuencia de las observaciones. Sin embargo, el algoritmo le requiere especificar el número de grupos o k óptimos a realizar.

Se hace la ejecución del modelo con 5 clúster y 3 clúster encontrando las siguientes agrupaciones (para identificar de manera más rápida los perfiles se hace los modelos no supervisados en IBM SPSS Modeler):

- Modelo con 3 clúster

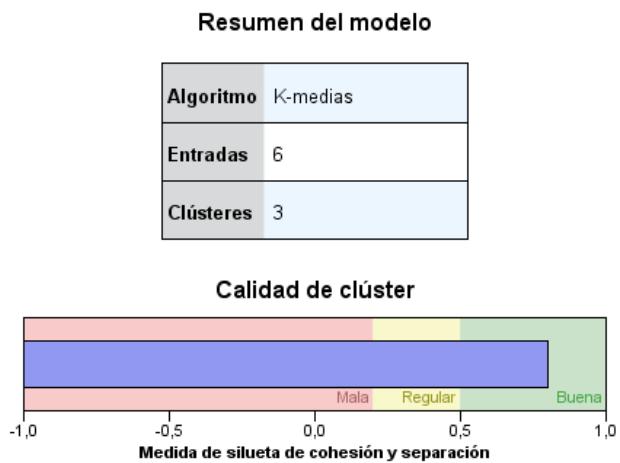
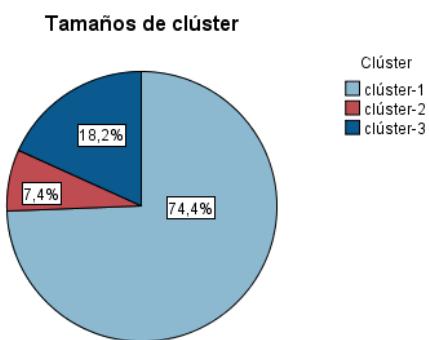


Ilustración 16 - Resumen modelo K-medias

La medida de silueta de cohesión y separación de clúster (basado en el trabajo de Kaufman y Rousseeuw (1990)) indican resultados pobres o buenos. Esta instantánea permite verificar rápidamente si la calidad de los clústeres conformados es deficiente, para este caso tenemos un indicador de 0.8, el cual significa que tenemos una eficiencia de este modelo para 3 grupos del 80%.

La silueta mide promedios, sobre todos los registros,  $(B - A) / \max(A, B)$ , donde  $A$  es la distancia del registro a su centro de grupo y  $B$  es la distancia del registro al centro de grupo más cercano al que no pertenece. Un coeficiente de silueta de 1 significaría que todos los casos están ubicados directamente en sus centros de agrupación. Un valor de -1 significaría que todos los casos están ubicados en los centros de clúster de algún otro clúster. Un valor de 0 significa, en promedio, que los casos son equidistantes entre su propio centro de grupo y el otro grupo más cercano.

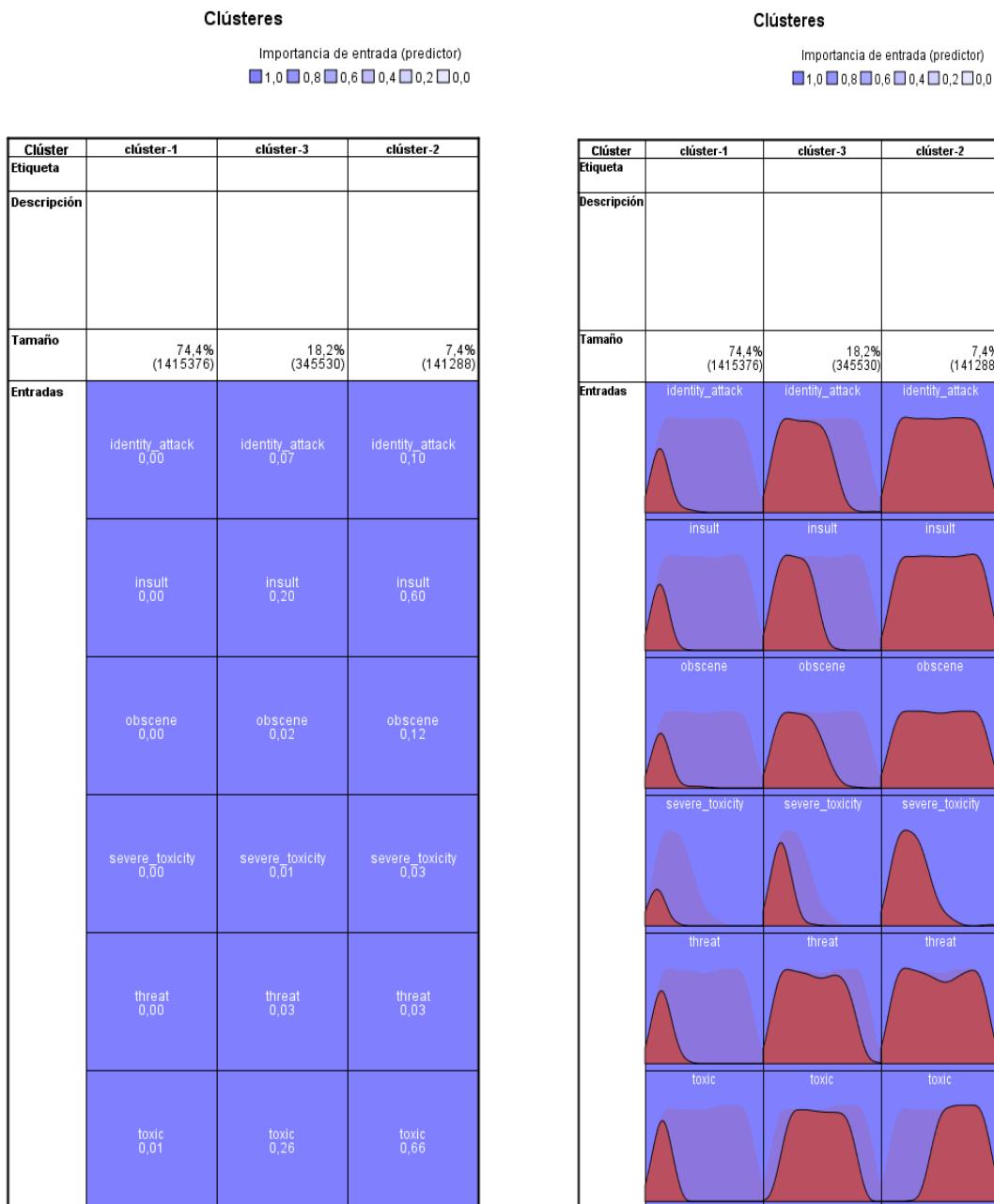
Conformación de los grupos:



Tamaño del clúster más pequeño	141288 (7,4%)
Tamaño del clúster más grande	1415376 (74,4%)
Cociente de tamaños: De clúster más grande a clúster más pequeño	10,02

Ilustración 17 - distribución de clústers

Tenemos que el grupo más grande con el 74,4% de los Twitter son aquellos que pueden tener un perfil más general o que normalmente se presenta en término de toxicidad, el tercer clúster con la población más pequeña que representa el 7,4% de los Twitter, para verificar los perfiles de cada clúster e identificar si estos se ajustan a los criterios de la investigación se calculan los valores de centroide de cada variable en cada grupo, con el fin de analizar cómo queda la distribución de las variables dentro de cada grupo.

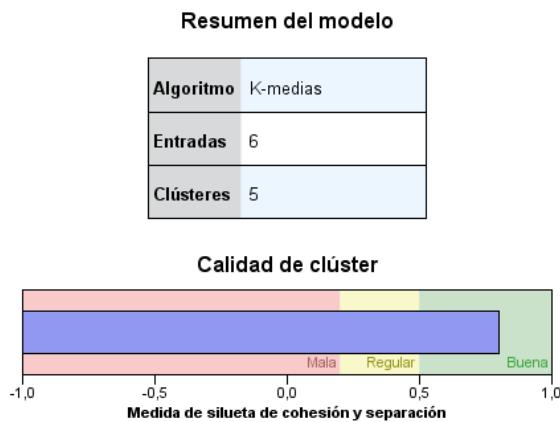


Se evidencia que el clúster-1 contiene la población de Twitter que su comportamiento en las variables es expresado con bajo (0,00) Nivel de tóxico, toxicidad severa, obsceno, identidad, insulto y amenaza, lo que significa que este frupo de personas (que son la mayoría) tiene unas reacciones o expresiones en su Twitter sin toxicidad.

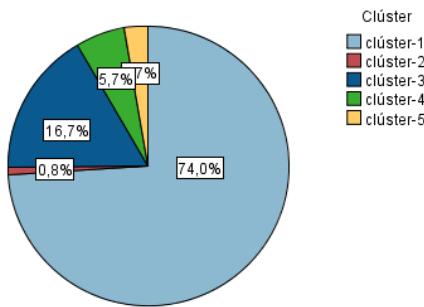
Para el clúster-3 se evidencia que para los niveles de toxicidad expresadas por las variables pasa de 0,00 a un intervalo de 0,02-0,20 lo que significa una identificación de una población con un nivel medio de toxicidad expresada en las variables tóxico, toxicidad severa,

obsceno, identidad, insulto y amenaza; y por ultimo el cluster 2 con la población menor que son 141,288 tweets, con las puntuaciones más elevadas en las 6 componentes analizadas.

- Modelo con 5 clúster



Al aumentar la cantidad de clúster a 5, la eficiencia de las agrupaciones no disminuye ni aumenta, lo que significa que se mantiene en 80% de eficiencia, y quedan con lo siguiente cantidad de tweets en cada grupo:



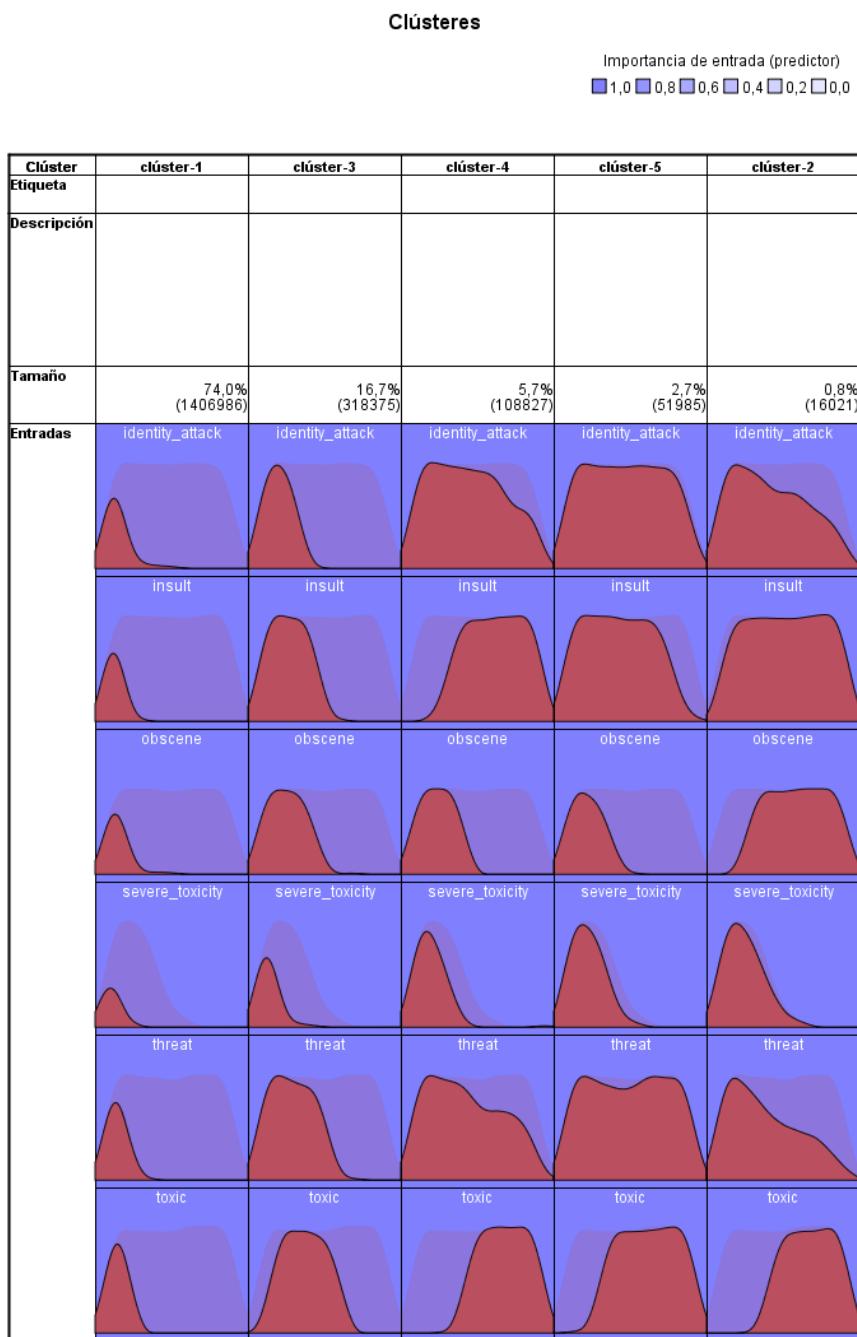
Tamaño del clúster más pequeño	16021 (0,8%)
Tamaño del clúster más grande	1406986 (74%)
Cociente de tamaños: De clúster más grande a clúster más pequeño	87,82

Se evidencia que los dos nuevos grupos salen del cluster tres y dos creados con un k=3, donde se mantiene la misma cantidad de tweets para el clúster 1, con el 74% de la población.

### Clústeres

Importancia de entrada (predictor)  
■ 1,0 ■ 0,8 ■ 0,6 ■ 0,4 ■ 0,2 ■ 0,0

Clúster	clúster-1	clúster-3	clúster-4	clúster-5	clúster-2
Etiqueta					
Descripción					
Tamaño	74,0% (1406986)	16,7% (318375)	5,7% (108827)	2,7% (51985)	0,8% (16021)
Entradas	identity_attack 0,00	identity_attack 0,04	identity_attack 0,05	identity_attack 0,35	identity_attack 0,07
	insult 0,00	insult 0,19	insult 0,64	insult 0,27	insult 0,49
	obscene 0,00	obscene 0,02	obscene 0,07	obscene 0,03	obscene 0,55
	severe_toxicity 0,00	severe_toxicity 0,01	severe_toxicity 0,02	severe_toxicity 0,03	severe_toxicity 0,06
	threat 0,00	threat 0,02	threat 0,02	threat 0,08	threat 0,03
	toxic 0,01	toxic 0,24	toxic 0,66	toxic 0,49	toxic 0,70



Al analizar el perfil, identificamos que varios grupos tiene un comportamiento muy parecido y que estos se pueden unir para conformar un único clúster. Se verifica si a nivel de distribución por diagrama de cajas se traslanan los clúster en cada una de las variables con relación a un perfil general de toda la masa de tweets:

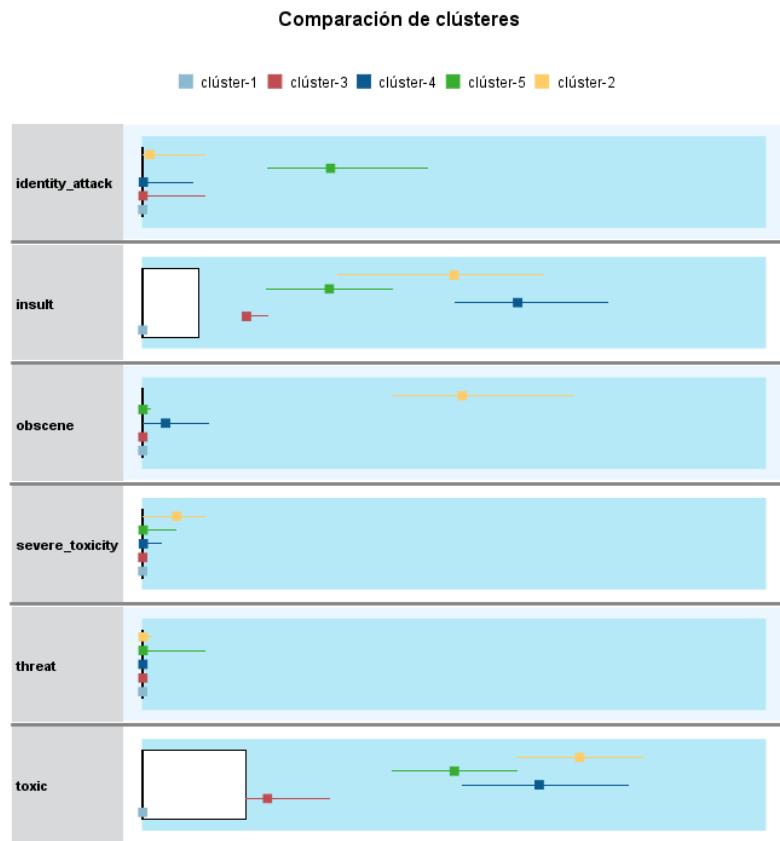


Ilustración 18 - comparación de clústeres

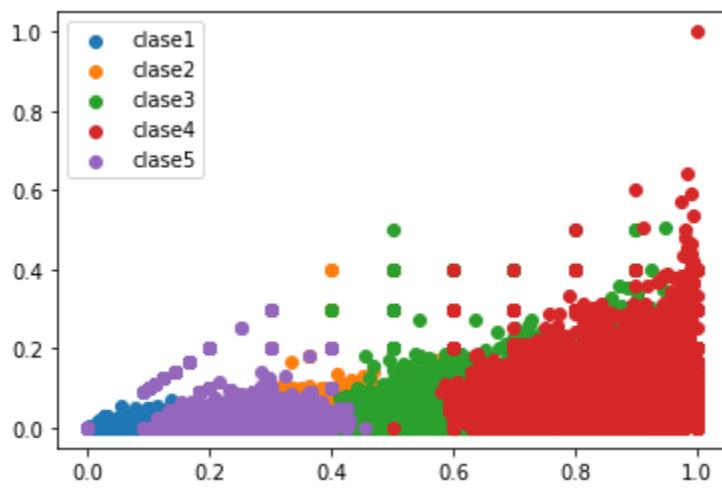


Ilustración 19 - kmeans con  $k = 5$

Se identifica que muchos grupos tienen un comportamiento parecido y vemos que no es necesario tener una mayor cantidad de grupos, así como muchas personas quedan clasificadas en grupos que no le pertenecen, **por tal motivo se decide con el equipo de**

trabajo generar la variable objetivo con k=3, la cual quedará con la siguiente marcación en cada uno de los tweets:

Valor	Proporción	%	Recuento
Alto		7.43	141288
Bajo		74.41	1415376
Medio		18.16	345530

Muestra de la base de datos 0= bajo, 1=medio y 2= alto:

	toxic	severe_toxicity	obscene	identity_attack	insult	threat	toxicidad
0	0.000000	0.000000	0.0	0.000000	0.000000	0.0	0
1	0.000000	0.000000	0.0	0.000000	0.000000	0.0	0
2	0.000000	0.000000	0.0	0.000000	0.000000	0.0	0
3	0.000000	0.000000	0.0	0.000000	0.000000	0.0	0
4	0.893617	0.021277	0.0	0.021277	0.872340	0.0	2
...	...	...	...	...	...	...	...
1902189	0.000000	0.000000	0.0	0.000000	0.000000	0.0	0
1902190	0.166667	0.000000	0.0	0.166667	0.166667	0.0	1
1902191	0.400000	0.000000	0.0	0.100000	0.400000	0.0	1
1902192	0.000000	0.000000	0.0	0.000000	0.000000	0.0	0
1902193	0.000000	0.000000	0.0	0.000000	0.000000	0.0	0

1902194 rows × 7 columns

A partir de aquí ya se cuenta con la variable objetivo lista para ingresar a un modelo supervisado que nos permita clasificar de la población actual de tweets, a continuación, se puede visualizar la identificación de cada tweet a su grupo (relación de las variables: toxic y severe\_toxicity con las clase).

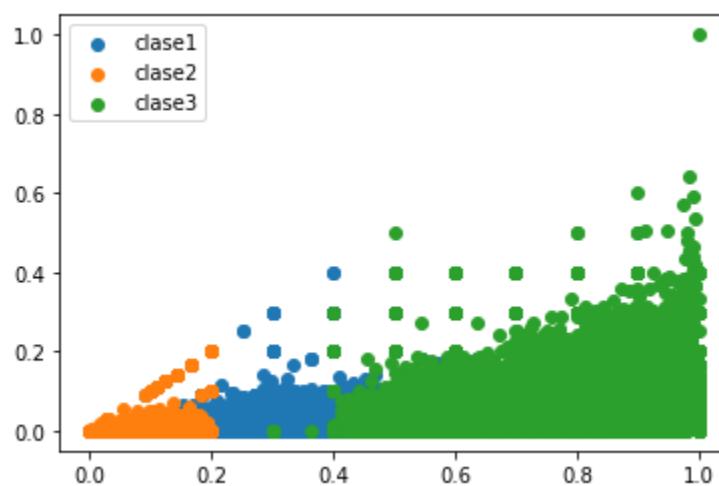


Ilustración 20 - Kmeans con K=3



	toxic	severe_toxicity	obscene	identity_attack	insult	threat	toxicidad
0	0.000000	0.000000	0.0	0.000000	0.000000	0.0	0
1	0.000000	0.000000	0.0	0.000000	0.000000	0.0	0
2	0.000000	0.000000	0.0	0.000000	0.000000	0.0	0
3	0.000000	0.000000	0.0	0.000000	0.000000	0.0	0
4	0.893617	0.021277	0.0	0.021277	0.872340	0.0	2
--	--	--	--	--	--	--	--
1902189	0.000000	0.000000	0.0	0.000000	0.000000	0.0	0
1902190	0.166667	0.000000	0.0	0.166667	0.166667	0.0	1
1902191	0.400000	0.000000	0.0	0.100000	0.400000	0.0	1
1902192	0.000000	0.000000	0.0	0.000000	0.000000	0.0	0
1902193	0.000000	0.000000	0.0	0.000000	0.000000	0.0	0
1902194	rows × 7 columns						

Ilustración 22. Base de datos de Twitter.

Para el inicio del modelo se hace una verificación de la correlación de las variables que se ingresaran al modelo supervisado:

Correlaciones de Pearson	
severe_toxicity	0.394
obscene	0.493
identity_attack	0.450
insult	0.928
threat	0.288

Ilustración 23. Correlaciones de Pearson

**La Correlación** mide la fuerza de la relación entre las variables continuas o numéricas; tomando valores entre -1.0 y 1.0. Los valores cercanos a +1.0 indican una asociación positiva fuerte de modo que los valores altos en un campo están asociados con valores altos en el otro y los valores bajos están asociados con valores bajos. Los valores cercanos a -1.0 indican una fuerte asociación negativa, de modo que los valores altos para un campo están asociados con valores bajos para el otro, y viceversa. Los valores cercanos a 0.0 indican una asociación débil, de modo que los valores para los dos campos son más o menos independientes.

Para esta correlación se realizó con el estadístico de Pearson busca la correlación lineal y su covarianza estandarizada. Se identifica que todas las variables están relacionadas encontrando con un impacto mayor la variable insultos, lo que significa que con las variables podemos clasificar de forma eficiente el nivel de toxicidad de los contenidos. Antes del modelo se realiza la separación de la muestra del entrenamiento y comprobación con una distribución de 70% - 30%.

Se utiliza en IBM SPSS Modeler de forma automática los tres modelos más significativos que podríamos utilizar para clasificar la toxicidad del contenido de los twits, con la finalidad de tener una guía del posible modelo a implementar:

¿Utilizar?	Gráfico	Modelo	Tiempo de generación	Precisión general (%)	Número de campos
<input checked="" type="checkbox"/>	 	 Regresión logística 1	9	100,0	6
<input checked="" type="checkbox"/>	 	 C5 1	9	99,992	6
<input checked="" type="checkbox"/>	 	 LSVM 1	< 1	99,973	6

Ilustración 24. Selección del modelo a implementar

Se identifica que el mejor modelo a utilizar sería la regresión logística Multinomial, y en segunda posición un árbol de decisión.

## Modelo de Regresión logística multinomial

Es una técnica estadística supervisada utilizada para clasificar registros donde su variable objetivo tiene más de dos opciones. La regresión logística funciona construyendo un conjunto de ecuaciones que relacionan los valores del campo de entrada con las probabilidades asociadas con cada una de las categorías de campo de salida. Una vez creado el modelo, se puede usar para estimar las probabilidades de los nuevos datos de comprobación o cuando se pone en producción para la calificación de la masa de datos de los clientes nuevos a calificar. Para cada registro, se calcula una probabilidad de pertenencia para cada categoría de salida posible. La categoría objetivo con la probabilidad más alta se asigna como el valor de salida previsto para ese registro.

Se utiliza la librería **sklearn**, el modelo logístico multinomial utiliza la estimación de máxima verosimilitud. La primera iteración es un modelo sin regresor, solo la intercepción. La siguiente iteración incluye regresores o parámetros de la variable independiente en el modelo. Los parámetros se cambian en cada iteración, y las iteraciones continúan hasta que se dice que el modelo ha convergido.

Usando los datos de entrenamiento, ajustamos el modelo de regresión logística multinomial y luego se implementa en el conjunto de datos de prueba para rectificar la eficiencia del modelo. Se calcula el número de condición de los datos de entrenamiento, el número condición nos indica que el sistema de ecuaciones tiene inversa y, por lo tanto, tiene solución.

```
#Obtener el numero condición de nuestros datos de entrenamiento
matriz = np.asmatrix(X_train)
np.linalg.cond(matriz)

14.887105139780772
```

Ilustración 25. Número condición de los datos de entrenamiento

A continuación, se estiman los parámetros de las ecuaciones de la regresión:

```
resultLRmodel.intercept_
array([ 27.35363357, -35.45002626,   8.0963927 ])

resultLRmodel.coef_
array([[-68.41817831, -5.46480882, -10.58718915, -19.28254251,
       -62.75654212, -7.30781881],
       [ 59.70272265,  3.11161091,   9.89172711,  10.39047953,
      55.92743458,  3.50276046],
      [ 8.71545566,  2.35319791,   0.69546203,   8.89206298,
     6.82910754,  3.80505835]])
```

Ilustración 26. Parámetros de las ecuaciones de la regresión

La primera matriz contiene tres intersecciones y la segunda matriz contiene tres conjuntos de coeficientes de regresión, esto es por la forma de cálculo de la librería, la cual difiere de los cálculos normales que se realizan en plataforma como R, SAS y SPSS. La solución tiene una ecuación para cada clase. Estos actúan como modelos independientes de regresión logística binaria. La salida real es  $\log(p(y=c) / 1 - p(y=c))$ , que son coeficientes logísticos multinomiales, de ahí las tres ecuaciones. Después de exponer cada coeficiente obtenemos cocientes de probabilidades. La interpretación de los coeficientes es para un cambio de una sola unidad en la variable predictora, el registro de probabilidades cambiará por un factor indicado por el coeficiente beta, dado que todas las demás variables se mantienen constantes.

Podemos concluir que los parámetros para la clasificación baja significa que las variables tóxico, toxicidad severa, obsceno, identidad, insulto y amenaza intervienen en una afectación negativa, caso contrario para las categorías de medio y alta toxicidad que sus betas tiene la afectación positiva en aumento.

Se calcula la matriz de confusión para la muestra de entrenamiento y de comprobación:

## Entrenamiento

```
In [35]: pred_yLRtrain = LRmodel.predict(X_train)
mostrar_resultados(y_train, pred_yLRtrain)
```

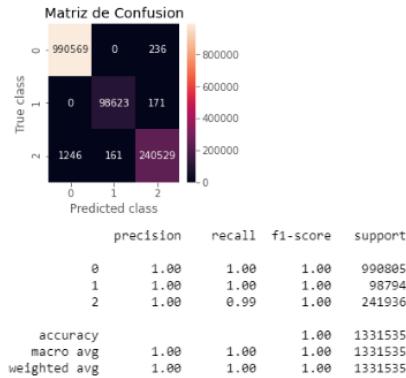


Ilustración 27. Matriz de confusión para la muestra de entrenamiento.

Verificamos que la eficiencia de este modelo para las tres categorías está por encima del 80%

## Comprobación:

```
: pred_yLR = LRmodel.predict(X_test)
mostrar_resultados(y_test, pred_yLR)
```

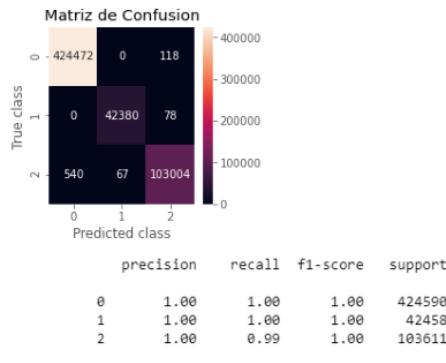


Ilustración 28. Matriz de confusión para la muestra de comprobación

Se identifica que los errores disminuyen en el conjunto de datos de comprobación y se mantiene un nivel de significancia mayor al 80% del modelo, lo cual nos da una un buen ajuste para seleccionar el modelo. Verificar que no se presente sobreajuste (overfitting):

```
#Verificamos si el modelo está sobreajustado
print('Score para entrenamiento :',LRmodel.score(X_train, y_train), ' y testeo: ',LRmodel.score(X_test, pred_yLR))
```

Score para entrenamiento : 0.9987691842642212 y testeo: 1.0

Ilustración 29. Ajuste para la selección del modelo.

Como otro método de verificación el modelo se calcula el valor de los residuales SSE, encontrando los siguientes valores:

```
: residuo_del_entrenamiento = y_train - LRmodel.predict(X_train)
sse_train = SSE(residuo_del_entrenamiento)
sse_train
```

```
: 1873
```

```
: residuo_del_testeo = y_test - LRmodel.predict(X_test)
sse_train = SSE(residuo_del_testeo)
sse_train
```

```
: 406
```

*Ilustración 30. Residuales SSE del modelo.*

## Modelo Árbol de clasificación

Los árboles de clasificación, también llamados de decisión o de identificación son métodos de aprendizaje inductivo supervisado no paramétrico. La clasificación de la variable objetivo se realiza en base a una serie de preguntas sobre los valores de sus atributos, empezando por el nodo raíz y siguiendo el camino determinado por las respuestas a las preguntas de los nodos internos, hasta llegar a un nodo hoja. La etiqueta asignada a esta hoja es la que se asignará al patrón a clasificar.

El algoritmo utilizado es `sklearn.tree.DecisionTreeClassifier`, clasificador es un algoritmo de aprendizaje supervisado. Se puede usar tanto para clasificación como para regresión. También es el algoritmo más flexible y fácil de usar, es computacionalmente rápido, se dice que cuantos más árboles tenga, más robusto el modelo, es robusto frente a datos atípicos u observaciones mal etiquetadas y Detecta de forma automática estructuras complejas entre variables. El criterio utilizado para tomar las decisiones es minimizando una función llamada impureza que evalúa la calidad de la división realizada. Se manejan tres funciones de impureza que son las más utilizadas: índice de Gini, la entropía y el error de clasificación, en general, una impureza de 0 significa orden absoluto y una impureza de 1 significa lo contrario: desorden absoluto.

**Entropía:** El nodo raíz divide los datos utilizando la función que proporciona la mayor ganancia de información. La ganancia de información nos dice lo importante que es un atributo dado de los vectores de características.

Información obtenida = entropía (padre) - [entropía promedio (niños)]

$$\text{Entropy} = \sum_i -p_i \log_2 p_i$$

**Impureza de Gini:** es computacionalmente más rápida, ya que no requiere el cálculo de funciones logarítmicas, a impureza de Gini es otra medida de impureza y se calcula de la siguiente manera:

$$Gini = 1 - \sum_i p_i^2$$

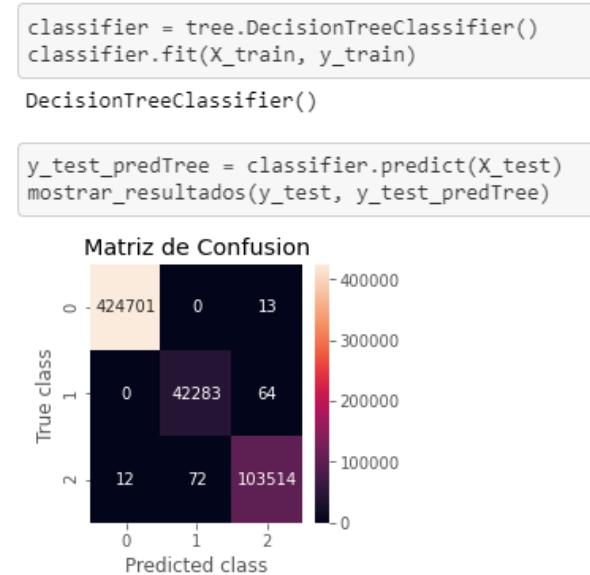


Ilustración 31. Impureza de Gini

	precision	recall	f1-score	support
0	1.00	1.00	1.00	424714
1	1.00	1.00	1.00	42347
2	1.00	1.00	1.00	103598
accuracy			1.00	570659
macro avg	1.00	1.00	1.00	570659
weighted avg	1.00	1.00	1.00	570659

Ilustración 32. Calificación del modelo.

Se verifica la tabla de clasificación la cual evidencia una mejor precisión por encima del 90% y con una reducción significativa de los errores cometidos en cada una de las categorías, las cuales se pueden comparar con el resultado de la regresión multinomial.

A continuación, se presenta la gráfica del árbol generado, el cual presenta muchas reglas y es difícil de visualizar debido a que su desprendimiento es binario:



Ilustración 33. Árbol de decisión generado.

## Evaluación de los modelos

Para definir el modelo a seleccionar se implementa una métrica basada en Rogers y Tanimoto (1960), el cual asignara un peso de 2 veces más importante a los errores cometidos en la clasificación:

```
#Definir una metrica basada en Rogers y Tanimoto (1960)
def obtener_eficiencia(pred_y):
    conf_matrix = confusion_matrix(y_test, pred_y)

    traza = np.trace(conf_matrix)
    total = np.sum(conf_matrix)

    eficiencia = traza / ((total-traza)*2 + traza) #castigamos lo que está por fuera de la diagonal
    return eficiencia

#Eficiencia de los modelos
print("Regresión logística: ",obtener_eficiencia(pred_yLR))
print("Regresión logística con balanceo: ",obtener_eficiencia(pred_ybalancedRL))
print("Regresión árbol de decisión: ",obtener_eficiencia(y_test_predTree))

Regresión logística:  0.9977140336628673
Regresión logística con balanceo:  0.9967214606120407
Regresión árbol de decisión:  0.9994358992326828
```

Ilustración 34. Eficiencia de los modelos implementados.

Se define como mejor modelo el árbol de decisiones con una eficiencia del 99,99% para hacer la clasificación de la toxicidad de los comentarios de la red social **Twitter**, teniendo en cuenta la matriz de confusión de cada modelo y el indicador de evaluación.

### Modelo de clasificación de Twitter

Se toma la base de datos submission.csv, el cual tiene los valores de Twitter extraídos:

<b>id</b>	<b>comment_text</b>	<b>creation_time</b>	<b>source</b>	<b>tweet_id</b>	<b>user</b>	<b>user_id</b>	<b>toxic</b>	<b>severe_toxic</b>	<b>obscene</b>	<b>threat</b>	<b>insult</b>
0	rt nosurrenderhk guardiannews the truth is hk ...	2020-06-13 16:07:02	Twitter for Android	1271836487663808513	currentecalamo	1005852785609326592	0.028525	0.001610	0.008574	0.000802	0.00910
1	rt marisakabas content warning police brutalit...	2020-06-13 16:07:02	Twitter for Android	1271836487663775744	Donald Dire	1088300096666591232	0.006022	0.002773	0.013770	0.001169	0.01665
2	rt shahmiruk it is absolutely unfair amp disint...	2020-06-13 16:07:02	Twitter for iPhone	1271836487693275144	tanya cochrane C#FBPE	25872176	0.581241	0.007748	0.068670	0.001182	0.11974
3	rt autototheonqueen terfs police the boundaries...	2020-06-13 16:07:02	Twitter for iPhone	1271836487747862529	Michael Birmingham	59031350	0.024132	0.001585	0.008654	0.000426	0.00931
4	looks like a scary demon witch monster to me	2020-06-13 16:07:02	Twitter for iPhone	1271836487676579841	untossable chum	2771192143	0.274693	0.002532	0.029018	0.000760	0.05014

Ilustración 35. Datos de Twitter clasificados bajo el modelo 1.

Se seleccionan las variables con las que se configuró el modelo de clasificación:

	<b>toxic</b>	<b>severe_toxic</b>	<b>obscene</b>	<b>identity_attack</b>	<b>insult</b>	<b>threat</b>
0	0.028525	0.001610	0.008574	0.003459	0.009100	0.000802
1	0.006022	0.002773	0.013770	0.003206	0.016658	0.001169
2	0.581241	0.007748	0.068670	0.061484	0.119748	0.001182
3	0.024132	0.001585	0.008654	0.013586	0.009315	0.000426
4	0.274693	0.002532	0.029018	0.006064	0.050144	0.000760
***	***	***	***	***	***	***
46320	0.377025	0.006002	0.040397	0.004014	0.080982	0.001909
46321	0.043127	0.001917	0.003345	0.005009	0.011429	0.004729
46322	0.285102	0.008395	0.025220	0.021252	0.051000	0.050829
46323	0.063165	0.002518	0.008595	0.001974	0.021814	0.000706
46324	0.050074	0.005939	0.013102	0.004785	0.021646	0.002224

46325 rows x 6 columns

Ilustración 36. Datos de Twitter clasificados bajo el modelo 1

Se aplica el modelo de árbol de decisión:

```
#Cargar el modelo
#model = pickle.load( open( "clasificadorRLToxicidad.pkl", "rb" ) )
model = pickle.load( open( "clasificadorArbolToxicidad.pkl", "rb" ) )
```

<code>pred_y = model.predict(X)</code>
<code>comentariosTwitter[“toxicidad”] = pred_y</code>
<code>comentariosTwitter.head()</code>
<code>1_time source tweet_id user user_id toxic severe_toxic obscene threat insult identity_attack toxicidad</code>
-13 2 Twitter for Android 1271836487663808513 currentecalamo 1005852785609326592 0.026525 0.001610 0.008574 0.000802 0.009100 0.003459 0
-13 2 Twitter for Android 1271836487663775744 Donald Diré 1068300096666591232 0.006022 0.002773 0.013770 0.001169 0.016658 0.003206 0
-13 2 Twitter for iPhone 1271836487693275144 tanya cochrane @FBPE 25872176 0.581241 0.007748 0.068670 0.001182 0.119748 0.061484 2
-13 2 Twitter for iPhone 1271836487747862529 Michael Bermingham 59031350 0.024132 0.001585 0.008654 0.000426 0.009315 0.013586 0
-13 2 Twitter for iPhone 1271836487676579841 untosable chum 2771192143 0.274693 0.002532 0.029018 0.000760 0.050144 0.006064 2

Ilustración 37. Aplicación del modelo de árbol de decisión

A continuación, se visualiza la cantidad de twits clasificados en cada categoría:

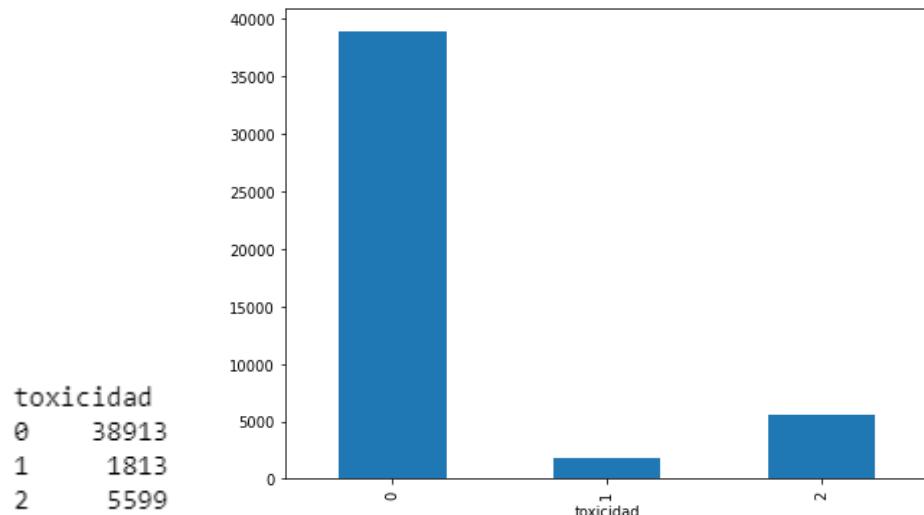


Ilustración 38. Clasificación de texto de Twitter.

Se encuentran 5.599 comentarios con alta toxicidad de la red social Twitter y las palabras más representativas:

```

2      rt shahmiruk it is absolutely unfair amp disin...
4          looks like a scary demon witch monster to me
22     blm amp supporters protesting peacefullyon the...
26     rt richiek143 forgot about the idiots causing ...
27     rt sooksthe so the police seem to be yet agai...
...
46293   rt fobbsmagazine tw black death y'all were lou...
46310   shes a complete disgrace to the position and u...
46313   rt escapedmatrix how is this acceptable this i...
46320   rt gatorfun1 watch what a hypocrite gov whitm...
46322   piersmorgan arm the police shoot to kill the...
Name: comment_text, Length: 5599, dtype: object

```

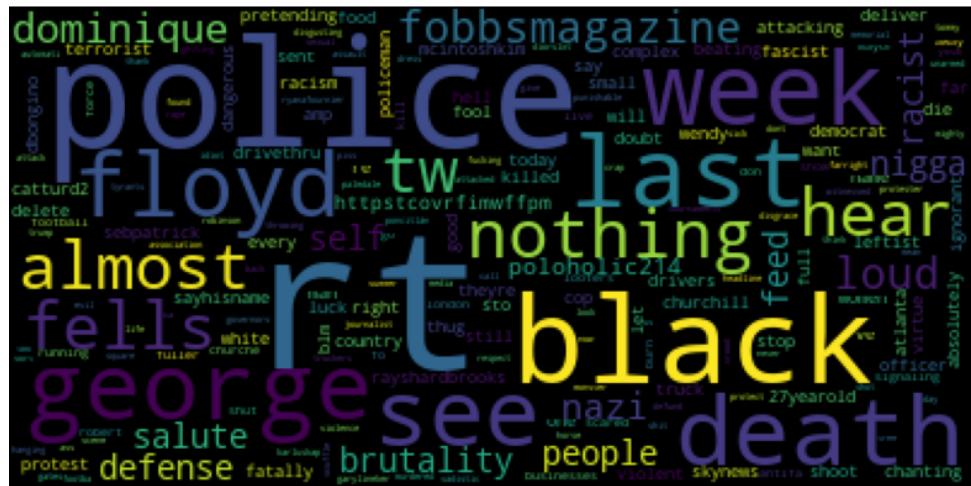


Ilustración 39. Palabras más representativas.

## Conclusiones

El aprendizaje más significativo obtenido de este proyecto que integra los contenidos vistos en los cursos de Álgebra en Ciencia de Datos, Estadística en analítica y Almacenamiento y Recuperación de Información; correspondientes al primer semestre de maestría en ciencia de datos y analítica de la universidad EAFIT, es el diseño de estructuras de clasificación y consumo de modelos en la nube, creando un ambiente de trabajo colaborativo.

Como dificultades del proyecto, se menciona que debido al acceso limitado a ciertas herramientas de AWS no se pudo realizar la puntuación de tweets en tiempo real que requiere una arquitectura distinta.

Para conservar el sentido de las oraciones y evaluar la toxicidad, se realiza una preparación de texto simple, es decir: sin remoción de stopwords y aplicar lemmatizer.

Para implementar modelos supervisados y no supervisados se hace necesario realizar una exploración, entendimiento y preparación de los datos previamente. Utilizando estrategias como CRISP-DM.

## Referencias

- Cárdenas, Santiago. Quiceno, Juan Diego. 2020. El Colombiano. Recuperado de:  
<https://www.elcolombiano.com/antioquia/nuevos-casos-de-coronavirus-en-medellin-y-antioquia-KF12602282>
- Economic and Social Research Council. 2020. What is Twitter and why should you use it? Recuperado de: <https://esrc.ukri.org/research/impact-toolkit/social-media/twitter/what-is-twitter/>
- IBM. 2020. IBM Knowledge Center. Recuperado de:  
[https://www.ibm.com/support/knowledgecenter/SS3RA7\\_15.0.0/com.ibm.spss.crispdm.help/crisp\\_business\\_understanding\\_phase.htm](https://www.ibm.com/support/knowledgecenter/SS3RA7_15.0.0/com.ibm.spss.crispdm.help/crisp_business_understanding_phase.htm).
- Gobierno Metropolitano de Medellin. 2020. Recuperado de:  
<https://www.metropol.gov.co/ambientales/aire/red-aire>
- Kaggle. 2020. Jigsaw Multilingual Toxic Comment Classification. Recuperado de:  
<https://www.kaggle.com/getting-started/44916>
- McCallum, Andrew. Nigam, Kamal. 2002. A Comparison of Event Models for Naïve Bayes Text Classification. Recuperado de:  
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.65.9324&rep=rep1&type=pdf>
- Maeve Duggan. 2017. Online harassment. Pew Research Center. Recuperado de:  
<https://www.pewresearch.org/internet/2018/07/03/the-negatives-of-digital-life/>
- Paz, Antonio José. 2020. Recuperado de: <https://es.mongabay.com/2020/04/calidad-del-aire-y-coronavirus-incendios-en-colombia/>
- Revista Semana. 2017.** Recuperado de: <https://sostenibilidad.semana.com/medio-ambiente/articulo/contaminacion-del-aire-en-medellin-es-un-problema-cronico/38650>
- Russell, Stuart J. Norvig , Peter. 2003. Artificial Intelligence: A Modern Approach (2 ed.). Pearson Education. See p. 499 for reference to the general definition of the Naïve Bayes model and its independence assumptions.
- Standford University. 2011. Text Classification and Naïve Bayes. Recuperado de:  
[https://web.stanford.edu/~jurafsky/slp3/slides/7\\_NB.pdf](https://web.stanford.edu/~jurafsky/slp3/slides/7_NB.pdf)
- Sterckx, Lukas. 2017. An Evaluation of Neural Network Models for Toxic Comment Classification. Recuperado de: <http://lusterck.github.io/papers/sterckx2017toxic.pdf>
- Towards Data Science. 2020. K-means: clustering algorithm. Recuperado de:  
<https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a>
- Towards Data Scienceb. 20202. Logistic Regression. Recuperado de:  
<https://towardsdatascience.com/introduction-to-logistic-regression->

[66248243c148#:~:text=Logistic%20Regression,on%20the%20concept%20of%20probability.&text=The%20hypothesis%20of%20logistic%20regression,function%20between%200%20and%201%20.](#)

IBM SPSS Modeler:

[https://www.ibm.com/support/knowledgecenter/SS3RA7\\_18.1.0/modeler\\_mainhelp\\_client\\_ddita/clementine/statistics\\_correlationlabels.html](https://www.ibm.com/support/knowledgecenter/SS3RA7_18.1.0/modeler_mainhelp_client_ddita/clementine/statistics_correlationlabels.html)

<ftp://public.dhe.ibm.com/software/analytics/spss/documentation/modeler/18.0/es/ModelerUserGuide.pdf>

Arboles de decisión:

<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>

<https://scikit-learn.org/0.17/modules/generated/sklearn.tree.DecisionTreeClassifier.html>

<https://benalexkeen.com/decision-tree-classifier-in-python-using-scikit-learn/>

Regresión logística Multinomial

<https://www.datasklr.com/logistic-regression/multinomial-logistic-regression>

<https://www.kaggle.com>thisisnic/multinomial-logistic-regression-0-02983>

Adicionales:

<https://www.aprendemachinelearning.com/k-means-en-python-paso-a-paso/>

<https://seaborn.pydata.org/generated/seaborn.pairplot.html>

<https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>

<https://www.aprendemachinelearning.com/clasificacion-con-datos-desbalanceados/>

<https://www.kaggle.com/tarunpaparaju/jigsaw-multilingual-toxicity-eda-models>