

# Análisis de sentimientos para Twitter con Vader y TextBlob

*Fecha de recepción: 2021-08-02 • Fecha de aceptación: 2021-09-10 • Fecha de publicación: 2021-10-10*

**Sofía Belén Alemán Viteri**

Universidad Central del Ecuador

sobealvi@hotmail.com

<https://orcid.org/0000-0003-3287-8590>

## RESUMEN

El análisis de sentimientos constituye una herramienta fundamental para el éxito de las actividades orientadas al público. Las redes sociales se han consagrado como un escenario válido para realizar este análisis, especialmente Twitter, que ofrece un API libre para la obtención de datos. El proceso para análisis de sentimientos incluye etapas de descarga, utilizando la librería Tweepy, depuración implementando métodos para eliminar símbolos que no aporten en el sentimiento del tuit y análisis con dos librerías: Vader y TextBlob. Estas devuelven un porcentaje que define si el tuit es positivo o negativo; sin embargo, cada una funciona con un algoritmo y entrenamiento diferente que ocasiona discrepancia en los resultados, TextBlob presentó mayor precisión. La parte final del análisis constituyen en el cálculo de métricas: precisión, exactitud, sensibilidad, especificidad y matriz de confusión.

**PALABRAS CLAVE:** twitter, sentimientos, algoritmos, exactitud, precisión, python.

## ABSTRACT

Sentiment analysis is a fundamental tool for the success of audience-oriented activities. Social networks have established themselves as a valid scenario to perform this analysis, especially Twitter, which offers a free API for data collection. The process for sentiment analysis includes download stages, using the Tweepy library, debugging by implementing methods to remove symbols that do not

contribute to the sentiment of the tweet and analysis with two libraries: Vader and TextBlob. These return a percentage that defines whether the tweet is positive or negative; however, each one works with a different algorithm and training that causes discrepancy in the results, TextBlob presented greater accuracy. The final part of the analysis is the calculation of metrics: precision, accuracy, sensitivity, specificity and confusion matrix.

**KEYWORDS:** twitter, sentiment, algorithms, accuracy, precision, python.

## Introducción

En los últimos años, el auge de redes sociales ha permitido que los usuarios utilicen estos recursos para expresar su opinión sobre cualquier tema o producto, considerándose ésta más auténtica y fiel a lo que resultaría aplicando una encuesta. El contar con una fuente tan rica de información representa una oportunidad para los interesados; sin embargo, también es una complicación, debido a los procesos que implica su análisis y la confiabilidad que brindan las herramientas dedicadas a ello, demandando así, soluciones que caractericen de mejor manera las opiniones, las cuantifiquen de forma precisa y que, el aplicarlas, no represente una mayor complicación.

De esta necesidad surgió el análisis de sentimientos definido por Cambria y Hussain (2020) como el “conjunto de técnicas computacionales para la extracción, clasificación, comprensión y evaluación de opiniones expresadas en fuentes publicadas en Internet, comentarios en portales web y en otros contenidos generados por usuarios”.

Se observa que la definición implica todos los procesos necesarios para determinar si un texto, en la presente investigación, en cuanto al contenido de un tuit es positivo, negativo o neutro, cuantificándolo en porcentaje. Esto se logra a través de algoritmos de aprendizaje supervisado que son entrenados para textos orientados a algún tema específico, por ejemplo, opiniones sobre películas. Aún no se puede definir un entrenamiento universal, ya que las frases de opinión dependen del contexto (Pauli, 2019).

El éxito de un modelo de aprendizaje supervisado se basa en dos factores: el algoritmo de clasificación que utiliza y las características elegidas para el conjunto de entrenamiento, llamado corpus. Existen varios algoritmos de clasificación, se mencionan los 4 más relevantes que serán objeto de estudio: eegresión logística, máquinas de soporte vectorial, Naive Bayes y aprendizaje por reglas. Elegir el algoritmo correcto para el conjunto de datos a analizar constituirá el elemento esencial para obtener resultados favorables. En la presente investigación se aplicarán 2 de los 4 algoritmos pre entrenados a través de librerías disponibles para python y se utilizará para determinar si son positivos o negativos los tuits de una muestra.

Repasando la literatura previa acerca del tema, se revisó la tesis titulada “Análisis de Sentimientos en Twitter para Descubrir Contenido Xenófobo hacia los Inmigrantes Venezolanos en Ecuador” publicada en el año 2021, donde al probar los 3 algoritmos, para determinar el índice de violencia en Twitter, obtienen la siguiente media de precisión, regresión logística 92,5%, máquinas de soporte vectorial 93,4% y Naive Bayes 88,4% (Romero, 2021), por lo cual determina que el mejor algoritmo son las máquinas de soporte vectorial.

También se revisó el proyecto final titulado: “Análisis de sentimientos – Comparación de algoritmos predictivos y métodos utilizando un lexicón español”, en este se hace énfasis en las métricas necesarias para determinar la efectividad del modelo, éstas son: “Exactitud, Precisión, Exhaustividad y Valor-F” (Pauli, 2019). De igual forma, en el trabajo de Vélez (2020), se selecciona el mejor modelo resultado del entrenamiento y la clasificación de tuits, donde los algoritmos propuestos en esta investigación quedan en el siguiente orden, de peor a mejor: regresión



logística, Naive Bayes y Máquinas de soporte vectorial, llegando a ser esta una alternativa a la opción determinada por el autor, redes neuronales.

Como es evidente por la bibliografía revisada, la respuesta al mejor algoritmo es divergente, esta investigación pretende aportar a la resolución de esa pregunta.

En este sentido, se utiliza la red social de Twitter como caso de estudio, consolidándose esta como uno de los medios más grandes para expresar opiniones, en enero del año 2021 cuenta con 353 millones de usuarios activos al mes (Mejía Llano, 2021), lo cual la transforma en una fuente confiable para análisis de datos, minería de opiniones. Twitter se ha ido adaptando a las necesidades de sus usuarios empresariales y favoreciendo la investigación, liberando los datos con pequeñas limitaciones.

Twitter trabaja con proyectos, permite organizar el trabajo para tener acceso de forma efectiva al monitor de datos (Twitter, 2021). También permite crear aplicaciones externas a proyectos, para esta investigación se utilizará un proyecto, donde, con tokens proporcionados por Twitter, permiten la descarga del contenido.

Según varias encuestas y análisis, los dos lenguajes más utilizados para el análisis de datos son: R seguido de Python (Universidad de Alcalá, s.f). La principal ventaja que ofrece Python es la versatilidad, permite realizar todas las etapas del análisis, desde la obtención de los tuits, depuración y análisis de información, de forma accesible y priorizando la velocidad de ejecución.

### 1.1 Análisis de sentimientos

Existen varios tipos de algoritmos que se pueden aplicar, los supervisados y no supervisados. Los algoritmos supervisados, también llamado *machine learning*, necesitan un conjunto de datos previamente clasificado, están basados en relaciones matemáticas formadas entre los elementos en el entrenamiento, generando un modelo estadístico que luego se compara con el texto a determinar para caracterizar el grupo al que pertenece, en general, se dividen en dos grandes grupos: regresión y clasificación, la diferencia radica en el resultado, los primeros retornan un valor numérico y los otros, un identificador de grupo al que pertenecen, ambos sirven para realizar el análisis de sentimiento. Por otro lado, los algoritmos de aprendizaje no supervisado permiten determinar patrones y anomalías en conjuntos de datos.

La librería NLTK (*Natural Language Toolkit*) calcula el sentimiento a partir de reglas creadas con anterioridad, por ejemplo, contar la cantidad de apariciones de una determinada palabra o analizar la estructura de la oración. Para aplicar este algoritmo, se utiliza NLTK, con su implementación Vader (*Valence Aware Dictionary and sEntiment Reasoner*), modelo que posee aproximadamente 7520 elementos (Hutto & Gilbert, 2014).

TextBlob utiliza el algoritmo Naive Bayes de aprendizaje supervisado que se basa en el teorema de Bayes sobre la probabilidad condicional en la que se quiere calcular la probabilidad (*Figura 1*) de que ocurra un evento C, sabiendo que ha ocurrido X (Pauli, 2019). Toma cada clase como

variable independiente y predice la probabilidad de que pertenezca a un grupo, sea este positivo, negativo o neutro.

Para aplicar el análisis de sentimientos con este algoritmo, se utiliza la librería TextBlob, devuelve una dupla de la forma (Sentimiento, clasificación, porcentaje positivo, porcentaje negativo).

**Figura 1**

*Probabilidad condicional*

$$P(A|R) = \frac{P(R|A)P(A)}{P(R)}$$

{
   
 P(A): Probabilidad de A
   
 P(R|A): Probabilidad de que se de R dado A
   
 P(R): Probabilidad de R
   
 P(A|R): Probabilidad posterior de que se de A dado R

**Fuente:** Roman (2019)

## Metodología

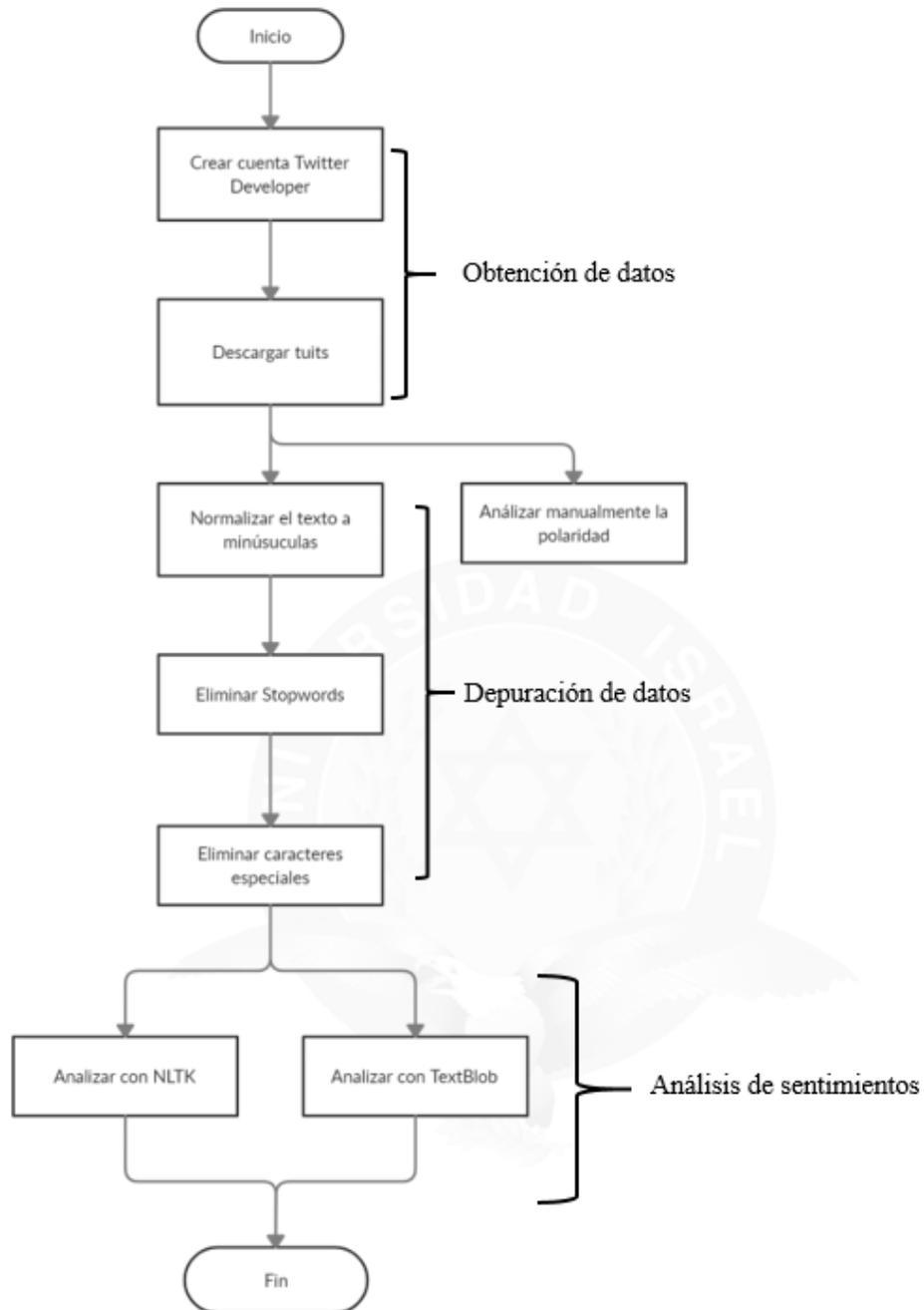
La metodología aplicada en esta investigación es la observación directa, esto significa que “el investigador está ahí, en el lugar donde se desarrolla la acción y está preparado para registrar lo que está ocurriendo” (Mendoza, 1994) interpretando este concepto al contexto en que se desarrolla la presente investigación, significa que los datos se obtienen directamente desde la fuente, la red social Twitter. El estudio también es de carácter cuantitativo, ya que evaluará los porcentajes de precisión que tiene cada librería de entrenamiento.

### 2.1 Proceso del desarrollo

El proceso realizado para esta investigación se muestra en el diagrama de flujo presentado en la *Figura 2*, consta de 3 etapas principales: obtención de datos, depuración de datos y análisis de sentimientos.

**Figura 2**

*Diagrama de flujo con actividades realizadas*

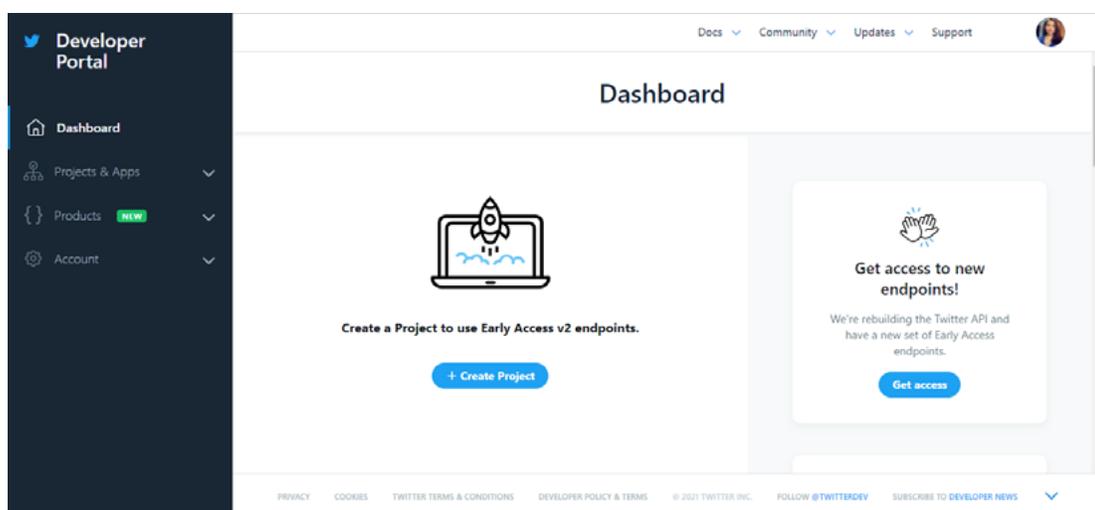


Los datos recopilados para la presente investigación fueron obtenidos a partir de la API de Twitter, como incentivo a la investigación en redes sociales, Twitter permite obtener los tuits, usuarios, seguidores, reacciones y mensajes internos con fin académico.

El único requisito para acceder a esos datos es poseer una cuenta de Desarrollador en Twitter, esta se crea a partir de una cuenta normal, completando la verificación del número telefónico y declarando las intenciones para el uso de la información en el portal <https://developer.twitter.com/en>, posteriormente se presenta la pantalla de la *Figura 3*. Existen dos tipos de cuenta, una estándar, orientada a no profesionales y, otra de investigación académica, orientada a profesionales y estudiantes de posgrado que buscan información para sus trabajos finales de titulación y/o investigación. Se utilizó una cuenta estándar, donde la cuota de tweets a descargar cada mes es de 500 000 con un tiempo histórico de 7 días, también permite descargar tweets en tiempo real; sin embargo, para la presente investigación se descargará el histórico de tweets.

**Figura 3**

*Pantalla principal de Twitter developer*



Para obtener los tuits, se empleó el lenguaje de programación Python. A través de la librería Tweepy, esta librería es de código abierto y permite conectarse e interactuar con la API de Twitter, la principal ventaja que ofrece es la facilidad al momento de gestionar la descarga de tuits y la posibilidad de depurar los datos en el mismo lenguaje con las demás librerías. Para esto fue necesario contar con las credenciales que Twitter entrega al usuario al momento de crear el proyecto: llave y llave secreta. Para esta investigación se utilizaron tuits que contenían palabras relacionadas a una película de ciencia ficción, ya que el modelo será entrenado para analizar reseñas sobre películas. Se optó por manejar los tuits como objetos json, donde el contenido se encuentra en la propiedad "full\_text", posteriormente, se lo almacena en un archivo .txt. Se obtuvieron 102 tuits para analizar.

Para poder empezar a analizar los tuits obtenidos se leen los datos almacenados en el archivo de texto plano y se ubican los resultados en un arreglo, para que éste sea útil es necesario someter este texto a procesos de normalización para evitar que determinados datos puedan influir de manera negativa en el resultado final (Pauli, 2019), se realizarán las reglas propuestas por Pauli (2019), éstas son:

- Normalización de mayúsculas y minúsculas
- Eliminación de *stopwords*
- Eliminación de caracteres especiales

Para todo el proceso de depuración de los datos se utilizan métodos de elaboración propia, la librería NLTK (Natural Language Toolkit) que según su documentación oficial es: “una plataforma líder para que los programas en Python trabajen con lenguaje humano, que provee interfaces fáciles de usar para más de 50 corpus y resultados lexicales como WordNet, junto con conjuntos de librerías de procesamiento de texto para la clasificación, tokenización, etc.”

Proporciona una lista con *stopwords* comunes en el idioma inglés, a la cual se agregó “rt” que permitirá eliminar esta notación que identifica a los retuits. Ejemplo de *stopwords*:

“the”, “I”, “most”, “rt”

Para iniciar con la normalización de mayúsculas y minúsculas se utilizó el método *lower()*, propio del tipo de dato, cadena de texto: Str en Python, por si solo permite transformar cadenas de caracteres a su notación en minúscula. Se obtiene el mismo texto completamente en minúscula, listo para ser separado en tokens, proceso que se realizó a continuación. Python, ofrece el método Split, que permite dividir el texto por palabras.

Por ejemplo:

Texto: ‘I like the rainy days because they are cold’

Texto tokenizado: [‘I’, ‘like’, ‘the’, ‘rainy’, ‘days’, ‘because’, ‘they’, ‘are’, ‘cold’]

El contar con el texto dividido permite navegar entre cada palabra para determinar qué va a pesar en el sentimiento del tuit y qué no, de ahí, diferenciar las palabras que no son de utilidad son llamadas “*stopwords*” o en español “palabras vacías”, por ejemplo, artículos, conectores o preposiciones. Utilizando el mismo ejemplo anterior:

Texto sin palabras vacías: [‘like’, ‘rainy’, ‘days’, ‘cold’]

Finalmente, se pasa el texto por un método, donde se reconocen los símbolos que no son letras y se los elimina, se elaboró una lista con los signos de puntuación más comunes, utilizados en la actualidad y se analiza carácter a carácter para eliminar lo que no sirva.

Para determinar la eficacia de cada algoritmo, los tuits se clasificaron por la autora de esta investigación, tomando en cuenta, el contexto del tuit, el contenido y la escritura, se utilizó el texto original, y se asignaron 3 valores, positivo cuando un tuit expresaba una reseña positiva sobre la película en cuestión, neutro si no expresaba una opinión y negativo si denotaba disgusto por parte del usuario.

## Resultados

A continuación, se muestra el resumen de los resultados obtenidos en la clasificación general de tuits.

**Tabla 1**

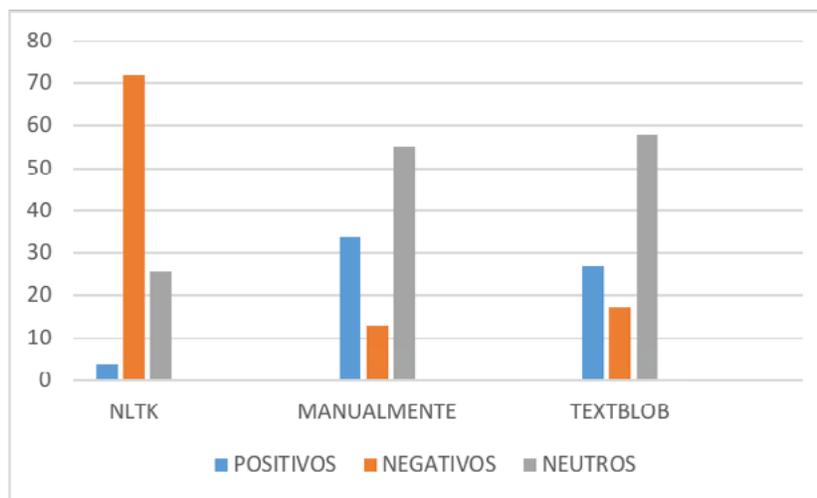
*Resumen de resultados*

	NLTK	MANUALMENTE	TEXTBLOB
POSITIVOS	4	34	27
NEGATIVOS	72	13	17
NEUTROS	26	55	58
Total	102	102	102

De forma manual se contabilizaron 34 tuits que pertenecen a la categoría positivos (*Tabla 1*), 13 a la categoría negativos y 55 a neutros. Para el análisis de los resultados se utilizaron exclusivamente las categorías positivos y negativos, asemejando la situación a una utilización real que puede tener este conjunto de resultados.

**Figura 4**

*Diagrama de barras tuits clasificados*



### 3.1 Matrices de confusión

Una matriz de confusión “es una herramienta que permite visualizar el desempeño de un algoritmo de aprendizaje supervisado (...) permite ver qué tipo de aciertos y errores está teniendo nuestro modelo a la hora de pasar por el proceso de aprendizaje con los datos” (Barrios, 2019). Cada posición contiene las siguientes métricas:

**Figura 4**

*Modelo matriz de confusión*



Fuente: Barrios (2019)

De acuerdo con la *Figura 4* se establece que:

- **Verdaderos positivos:** son aquellos textos que el algoritmo determina positivos de forma correcta, se ubica a la esquina superior izquierda de la matriz.
- **Falsos positivos:** el algoritmo determina como positivos, siendo estos falsos, se ubica en la esquina superior derecha de la matriz.
- **Verdaderos negativos:** textos que el algoritmo determina negativos de forma correcta, se ubica en la parte inferior derecha de la matriz.
- **Falsos negativos:** de forma análoga, son los textos que el algoritmo determina negativos siendo que éstos son positivos, se encuentra en la parte inferior izquierda de la matriz.

Obteniendo la matriz de confusión para los tuits analizados por los algoritmos, en contraste con lo obtenido por los algoritmos, se obtuvieron las siguientes matrices de confusión:

Para realizar esta matriz se tomaron en cuenta los valores con calificación positiva o negativa, si se observa en la *Tabla 1*, este algoritmo clasificó 76 tuits dentro de esta categoría, de los cuales, 4 fueron acertadamente catalogados como positivos, 13 negativos y existió un error de 59 tuits que se identificaron como negativos, siendo positivos, tal como se puede ver en la *Figura 5*.

**Figura 5***Matriz de confusión para análisis de sentimientos basado en reglas*

**MATRIZ DE CONFUSIÓN**

4	0
59	13

En el caso de la matriz de confusión para TextBlob se obtuvieron 44 tuits (*Figura 6*) en las categoría positivo y negativo, se observa que existen pocos tuits clasificados como negativos, siendo positivos y ninguno clasificado como positivo, siendo negativo.

**Figura 6***Matriz de confusión para TextBlob*

27	0
4	13

### 3.2 Métricas

Se va a obtener la exactitud, precisión y sensibilidad del modelo, son fórmulas aplicadas a los valores obtenidos en la matriz de confusión. Se utilizaron las siguientes notaciones:

VP: valores positivos

FP: falsos positivos

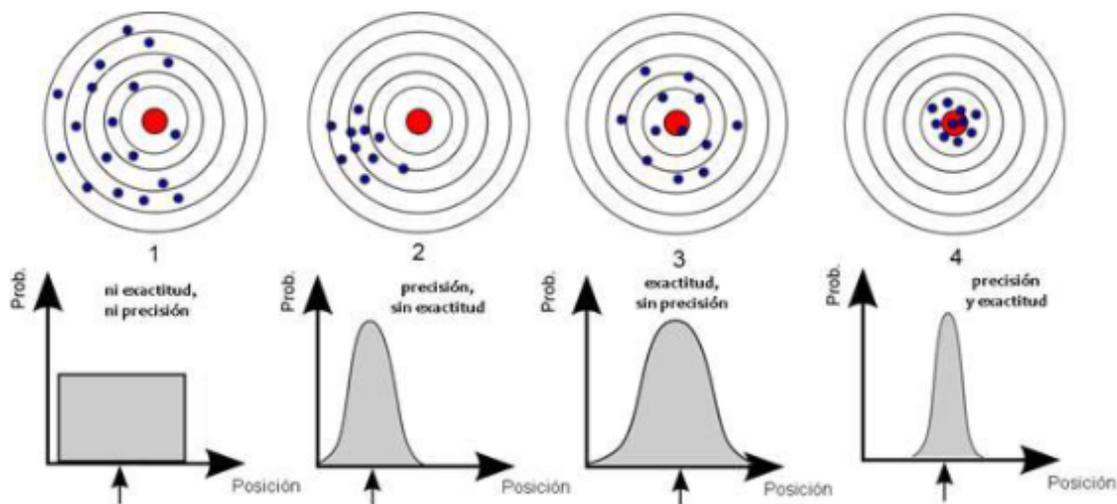
VN: verdaderos negativos

FN: falsos negativos

Para entender de mejor forma qué representan las métricas evaluadas, la *Figura 7* muestra cómo se ven los datos en relación con las métricas más importantes: exactitud y precisión. Si un conjunto de datos no posee exactitud ni precisión, se tiene un conjunto de datos dispersos a lo largo del plano. Si se tiene precisión, pero no exactitud, los datos no se encuentran dispersos; sin embargo, no se acercan al objetivo, si existe exactitud, pero no precisión, los datos se encuentran dispersos, pero es evidente que responden al objetivo, si existe precisión y exactitud, el conjunto de datos está concentrado en torno al objetivo.

**Figura 7**

*Comparación precisión y exactitud*



Fuente: Barrios (2019)

En cuanto a la exactitud, esta se refiere a que tan cerca está el valor calculado al valor verdadero (*Tabla 2*), es la proporción ente los valores verdaderos, ya sean negativos o positivos, entre el total de casos estudiados. La fórmula aplicada se muestra en la *Figura 8*.

**Figura 8**

*Fórmula para la exactitud*

$$(VP+VN)/(VP+FP+FN+VN)$$

Fuente: Barrios (2019)

**Tabla 2***Exactitud*

	NLTK	TEXTBLOB
<b>Exactitud</b>	0,223684211	0,909090909

Es evidente la diferencia obtenida en cuanto a exactitud para ambos modelos, utilizando Textblob, existe una exactitud del 90%, mientras que usando NLTK la exactitud decae al 22%, esta métrica tiene relación con la existencia de 59 falsos negativos.

Por su parte, la precisión hace referencia a la dispersión de los valores, se representa por la relación entre verdaderos positivos dividido entre todos los valores positivo (*Tabla 3*). La fórmula empleada se muestra en la *Figura 9*:

**Figura 9***Fórmula para la precisión*

**Se calcula como:  $VP/(VP+FP)$**

**Fuente:** Barrios (2019)

**Tabla 3***Precisión*

	NLTK	TEXTBLOB
<b>Precisión</b>	1	1

En ambos casos, la precisión es de un 100%, ya que no existen valores falsos positivos en ningún caso.

La sensibilidad, llamada tasa de verdaderos positivos, hace referencia a la proporción de casos positivos correctamente identificados dentro del conjunto de datos (*Tabla 4*). Se calcula dividiendo los casos verdaderos positivos sobre total. La fórmula aplicada se muestra en la *Figura 10*:

**Figura 10***Fórmula para la sensibilidad*

**Se calcula así:  $VP/(VP+FN)$ ,**

**Fuente:** Barrios (2019)

**Tabla 4**

*Sensibilidad*

	NLTK	TEXTBLOB
<b>Sensibilidad</b>	0,063492063	0,870967742

La sensibilidad más alta se obtuvo utilizando textBlob, con un 87% de sensibilidad, caso contrario a NLTK que obtuvo 6%, de aquí, se intuye que TextBlob puede determinar la polaridad de un tuit positivo dentro de los positivos.

La especificidad es muy similar a la sensibilidad, teniendo como objeto de estudio a los tuits negativos (*Tabla 5*). Se define como la tasa de verdaderos negativos y hace referencia a la tasa de textos negativos que el algoritmo ha clasificado de forma correcta. La fórmula de cálculo es la que se observa en la *Figura 11*.

**Figura 11**

*Fórmula para la especificidad*

$$VN/(VN+FP),$$

Fuente: Barrios (2019)

**Tabla 5**

*Especificidad*

	NLTK	TEXTBLOB
<b>Especificidad</b>	1	1

En ambos casos la especificidad es del 100%, predicen de forma correcta la cantidad de tuits negativos dentro el conjunto de datos.

## Conclusiones

En conclusión, el algoritmo Naive Bayes, implementado con la librería TextBlob, identifica de mejor forma el sentimiento de un tuit sobre una película, presentando en general, mayor exactitud, precisión, sensibilidad y especificidad, la proporción de tuits clasificados en cada categoría es similar a la clasificada de forma manual, como se observó en la *Figura 3*.

Ambos algoritmos presentan gran precisión y especificidad, métrica directamente relacionada con la cantidad de falsos positivos, denotando que es más sencillo acertar la clasificación cuando un texto es positivo, de forma contraria, para el algoritmo basado en reglas fue mucho más

complicado determinar si un tuit era negativo, desembocando en una cantidad amplia de falsos negativos.

Como trabajo futuro se plantea realizar la investigación incluyendo más algoritmos y con un conjunto de datos más grande, planteando distintos escenarios que modifiquen las métricas para determinar las condiciones, bajo las cuales funciona mejor cada algoritmo y aprovechar sus potencialidades.



## Referencias

- Barrios Arce, J. (26 de Julio de 2019). La matriz de confusión y sus métricas. *Health Big Data* <https://www.juan-barrios.com/la-matriz-de-confusion-y-sus-metricas/>
- Cambria, E., & Hussain, A. (2012). Sentic computing. *marketing*, 59(2), 557-577. [https://doi.org/10.1007/978-94-007-5070-8\\_4](https://doi.org/10.1007/978-94-007-5070-8_4)
- Hutto, C., & Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the International AAI Conference on Web and Social Media*, 8(1).
- Mejía Llano, J. (17 de marzo de 2021). Estadísticas de redes sociales 2021: usuarios de Facebook, Instagram, YouTube, LinkedIn, Twitter, TikTok y otros. <https://www.juancmejia.com/marketing-digital/estadisticas-de-redes-sociales-usuarios-de-facebook-instagram-linkedin-twitter-whatsapp-y-otros-infografia/#4>  
[Usuarios activos de Twitter](#)
- Mendoza, M. (1994). Técnicas de observación directa para estudiar interacciones sociales infantiles entre los Toba. *RUNA, archivo para las ciencias del hombre*, 21(1), 241-262. <https://doi.org/10.34096/runa.v21i1.1400>
- NLTK Project. (20 de abril de 2021). NLTK 3.6.2 documentation. <https://www.nltk.org/>
- Pauli, P. A. (2019). *Análisis de sentimiento - Comparación de algoritmos predictivos y métodos utilizando un lexicon español*. [Tesis de grado, Instituto Tecnológico Buenos Aires] Repositorio ITBA. <http://ri.itba.edu.ar/handle/123456789/1782>
- Roman, V. (25 de abril de 2019). Algoritmos Naive Bayes: Fundamentos e Implementación. *Medium*. <https://medium.com/datos-y-ciencia/algoritmos-naive-bayes-fudamentos-e-implementaci%C3%B3n-4bcb24b307f>
- Romero Vega, R. (2021). *Análisis de sentimientos en twitter para descubrir contenido xenófobo hacia los inmigrantes venezolanos en Ecuador*. [Tesis de grado, Universidad Nacional de Loja] Repositorio UNL. <https://dspace.unl.edu.ec/jspui/handle/123456789/23796>
- Twitter. (2021). Developer Platform. <https://developer.twitter.com/en/docs/projects/overview>
- Universidad de Alcalá. (s.f). Lenguajes de programación para data science. <https://www.master-data-scientist.com/lenguajes-programacion-data-science/>
- Vélez Bravo, (2020). *Aplicativo para detectar conductas violentas en Twitter, a través de análisis de sentimientos*. [Tesis de grado, Universidad Central del Ecuador]. Repositorio digital UCE. <http://www.dspace.uce.edu.ec/handle/25000/20926>

Copyright (c) 2021 Sofía Belén Alemán Viteri



Este texto está protegido bajo una licencia internacional [Creative Commons](#) 4.0.

Usted es libre para Compartir—copiar y redistribuir el material en cualquier medio o formato — y Adaptar el documento — remezclar, transformar y crear a partir del material—para cualquier propósito, incluso para fines comerciales, siempre que cumpla las condiciones de Atribución. Usted debe dar crédito a la obra original de manera adecuada, proporcionar un enlace a la licencia, e indicar si se han realizado cambios.

Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que tiene el apoyo del licenciante o lo recibe por el uso que hace de la obra.

[Resumen de licencia](#) – [Texto completo de la licencia](#)