Autores: Mat. Cipriano Callejas Hernández, Mat. Athziri Padilla Medina & Mat. Erika Rivadeneira Pérez

1. Abstract

A medida que el contenido en línea sigue creciendo, también lo hace la propagación del discurso de odio (HS, por sus siglas en ingles). Uno de los grandes desafíos de la detección automática del HS es la diferenciación entre HS y varias estancias de lenguaje ofensivo ya que existen dificultades como país o locación, sutilezas en el lenguaje, diferentes definiciones sobre lo que constituye el HS y por tanto sesgo en la base de datos. En el presente trabajo se propone utilizar técnicas de procesamiento de lenguaje natural (NLP, por sus siglas en ingles) para procesar tweets extraídos del repositorio de Github de Davidson, et. al., [Dav+17] y se aplican métodos de clasificación multiclase como: árboles de decisión, regresión logística, máquinas de vectores soporte y perceptron multicapa para clasificar los tweets en las clases: discurso de odio, lenguaje ofensivo y ninguna. Posteriormente se comparan y se analizan los resultados.

2. Introducción

No existe una definición formal de discurso de odio, pues este depende usualmente del país o locación, pero existe un consenso que manifiesta que es el discurso dirigido hacia los grupos sociales desfavorecidos de una manera que es potencialmente dañina para ellos, este discurso puede promover la violencia o el desorden social [Jac02]. En muchos países, incluyendo el Reino Unido, Canadá y Francia existen leyes que prohíben el discurso de odio. Estas leyes se extienden a Internet y las redes sociales, lo que lleva a muchos sitios a crear sus propias disposiciones contra el discurso de odio. Tanto Facebook como Twitter han respondido a las críticas, por no hacer lo suficiente para prevenir el discurso de odio en sus sitios, instituyendo políticas para prohibir el uso de sus plataformas por ataques a personas por sus características como: raza, etnia, género, orientación sexual, amenazas de violencia hacia otros, etc.

Los conjuntos de datos existentes difieren en su definición de discurso de odio, lo que lleva a datos que no solo provienen de diferentes fuentes, sino que también capturan infor-

mación diferente [Mac+19]. Además, es importante mencionar que, nuestra definición no incluye todos los casos de lenguaje ofensivo porque las personas a menudo usan términos que son muy ofensivos para ciertos grupos pero de una manera cualitativamente diferente. Por ejemplo, algunos afroamericanos a menudo usan el término n*gga en el lenguaje cotidiano. Además, existen personas que usan términos como b*tch cuando citan letras de rap, y los adolescentes usan insultos homofóbicos como h*e cuando juegan videojuegos [WH12]. Esta clase de lenguaje prevalece en redes sociales, haciendo que esta condición de límite sea crucial para cualquier sistema de detección de discursos de odio utilizable. Por esto, tanto Facebook como Twitter deben reforzar sus acciones para evitar que existan situaciones donde se presenten discursos de odio mal clasificados. Estos aspectos mencionados hacen que la detección automática de discurso de odio sea todo un desafío. Por otro lado, la forma en que se detecta lenguaje de odio de manera manual es mediante el uso de monitores humanos, desafortunadamente son usualmente empleos mal pagados, por tanto, el automatizar y mejorar la detección de discursos de odio permitiría que sea cada vez menos necesario el uso de monitores humanos.

En el presente trabajo se clasifican tweets, extraidos del repositorio de gitgub de [Dav+17], cada tweet puede estar clasificado en una de las tres categorías: discurso de odio, lenguaje ofensivo o ninguno. Esta clasificación manual la realizaron usuarios de CrowdFlower (CF) y se consideran solamente los tweets codificados por al menos 3 usuarios de CF. Se entrenaron varios modelos, entre ellos: árboles de decisión, regresión logística, máquina de vector soporte (SVM) y perceptrón multicapa (MLP), con el objetivo de diferenciar entre estas tres categorías para cada tweet. Por último se analizan y se comparan los resultados.

3. Metodología

Se proponen modelos de clasificación basados en características de última generación que incorporan características de distribución TF-IDF, etiquetas de parte del discurso (POS, por sus siglas en inglés de *parts of speech*), donde se etiquetan a las palabras en su tipo (verbo, proverbio, artículo, etc.) y otras características lingüísticas como el análisis de sentimiento (positivo, negativo o neutro), número de sílabas, número de caracteres, etc. La incorporación de estas características lingüísticas ayuda a identificar el discurso de odio al distinguir entre los diferentes usos de los términos, pero aún adolece de algunas sutilezas, como cuando los términos típicamente ofensivos se usan en un sentido positivo (por ejemplo, Queer en “He’s a damn good actor. As a gay man, it’s awesome to see

an openly queer actor given the lead role for a major film.", del conjunto de datos de HatebaseTwitter [Dav+17]).

3.1. Características de distribución TF-IDF

TF-IDF, del inglés Term frequency−Inverse document frequency, es una medida numérica que expresa cuán relevante es una palabra para un documento en una colección. El valor TF-IDF aumenta proporcionalmente al número de veces que una palabra aparece en el documento, pero es compensada por la frecuencia de la palabra en la colección de documentos, lo que permite manejar el hecho de que algunas palabras son generalmente más comunes que otras.

TF-IDF es el producto de dos medidas, frecuencia de término y frecuencia inversa de documento. La frecuencia de término, $\mathrm{tf}(t,d)$, es la frecuencia del término t en el documento d,

$$\mathrm{tf}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}.$$

Por otro lado, la frecuencia inversa del documento (IDF) es una medida de cuánta información proporciona la palabra, es decir, si es común o rara en todos los documentos y está definida como:

$$\mathrm{idf}(t,D) = \log \frac{N}{|\{d \in D : t \in d\}|}$$

donde N es el número total de documentos en el corpus $N = |D|$, $|\{d \in D : t \in d\}|$: es el número de documentos donde aparece el término t (i.e., $\mathrm{tf}(t,d) \neq 0$). Si el término no está en el corpus se llega a una división por cero. Por tanto, es común ajustar el denominador a $1 + |\{d \in D : t \in d\}|$.
Luego, el TF-IDF está definido por el producto

$$\mathrm{tfidf}(t,d,D) = \mathrm{tf}(t,d) \cdot \mathrm{idf}(t,D).$$

3.2. Árboles de decisión

Árbol de decisión es un modelo de predicción utilizado en diversos ámbitos que van desde la inteligencia artificial hasta la Economía. Dado un conjunto de datos se fabrican diagramas de construcciones lógicas, muy similares a los sistemas de predicción basados en reglas, que sirven para representar y categorizar una serie de condiciones que ocurren de forma sucesiva para la resolución de un problema [RM14].

Los árboles de decisión están formados por nodos, vectores de números, flechas y etiquetas. Cada nodo se puede definir como el momento en el que se ha de tomar una decisión de entre varias posibles, lo que va haciendo que a medida que aumenta el número de nodos aumente el número de posibles finales a los que puede llegar el individuo. Los vectores de números son la solución final a la que se llega en función de las diversas posibilidades que se tienen, dan las utilidades en esa solución. Las flechas son las uniones entre un nodo y otro y representan cada acción distinta y finalmente, las etiquetas se encuentran en cada nodo y cada flecha y dan nombre a cada acción, ver figura (1).

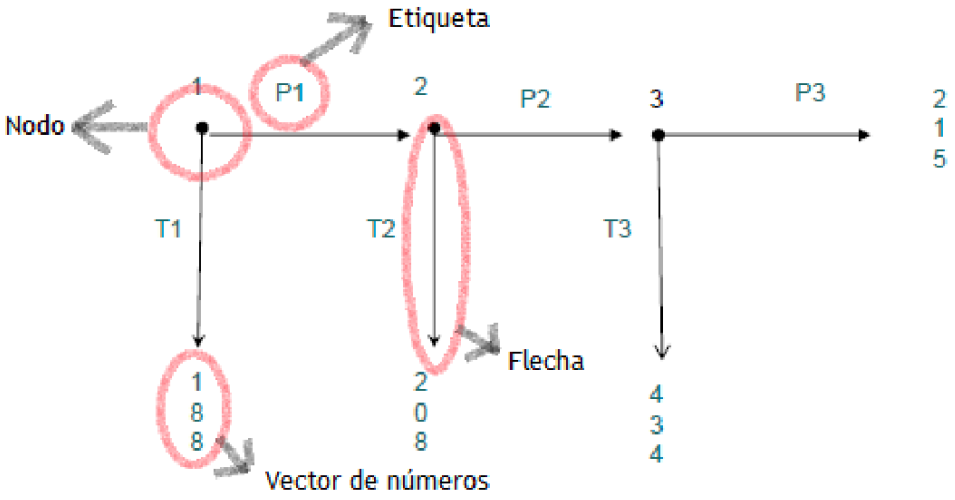


Figura 1: Estructura de un árbol de decisión

El aprendizaje basado en árboles de decisión utiliza un árbol de decisión como un modelo predictivo que mapea observaciones sobre un artículo a conclusiones sobre el valor objetivo del artículo. Es uno de los enfoques de modelado predictivo utilizadas en estadísticas, minería de datos y aprendizaje automático [Wu+07].

Los criterios para calcular la información de costo son el índice de Gini y la entropía. Los algoritmos de árboles de decisión utilizan la información de costo para dividir un nodo.
Tanto Gini como entropía son medidas de impureza de un nodo. Un nodo que tiene varias clases es impuro, mientras que un nodo que sólo tiene una clase es puro. La entropía en estadística es análoga a la entropía en termodinámica, donde significa desorden. Si hay varias clases en un nodo, hay desorden en ese nodo.

3.3. Regresión Logística

Es un tipo de análisis de regresión utilizado para predecir el resultado de una variable categórica en función de las variables independientes o predictoras. La regresión logística estima una función de regresión lineal múltiple definida como:

$$\ln\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_i x_i,$$

para $i = 1, ..., n$. El objetivo es encontrar los mejores valores p_i para cada muestra i, maximizando la función log-verosimilitud del modelo sobre los datos observados. La función log-verosimilitud de la variable respuesta y_i (binaria) se la denota por $\mathrm{L}_{\log}$.
Una manera de evitar sobreajuste es agregando el término de penalización $\lambda \sum {\beta_j}^2$ a la función de costo. Este término de penalización se lo conoce como **penalización L2**. Entonces, la nueva función de costo viene dada por

$$L_{log} + \lambda \sum_{j=1}^{p} {\beta_j}^2.$$

Esta función de costo penalizada se la conoce como *regresión Ridge* [HK70]. Por otro lado, una pequeña modificación a la penalización es usando los valores absolutos de B_j en vez de sus cuadrados. A esta penalización se la conoce como penalización L1. El método de regresión que usa la penalización L1 se la conoce como *regresión Lasso* [Tib96]. Entonces, la nueva función de costo vendría dada por

$$L_{log} + \lambda \sum_{j=1}^{p} |\beta_j| .$$

La penalización L1 tiende a elegir una variable al azar cuando las variables predictoras están correlacionadas. En este caso, parece que una de las variables no es importante, aunque aún podría tener poder predictivo. La regresión de Ridge, por otro lado, reduce los coeficientes de las variables correlacionadas entre sí, manteniéndolas todas. Un nuevo método, denominado como *red elástica*, usa ambas penalizaciones, L1 y L2 [ZH05]. La función de costo de la red elástica viene dada por

$$L_{\log} + \lambda \sum_{j=1}^{p} \left(\alpha \beta_j^2 + (1-\alpha) |\beta_j|\right) .$$

3.4. Máquinas con vectores soporte (SVM)

SVM es un algoritmo de aprendizaje automático supervisado que ayuda en problemas de clasificación o regresión. La finalidad es encontrar un límite óptimo entre los posibles resultados. En otras palabras, SVM realiza transformaciones de datos complejos según la función del kernel seleccionada y, en función de esas transformaciones, intenta maximizar los límites de separación entre sus puntos de datos según las etiquetas o clases definidas.

El objetivo es encontrar un hiperplano que maximice la separación de los puntos de datos de sus clases potenciales en un espacio n-dimensional. Los puntos de datos con la distancia mínima al hiperplano se denominan vectores de soporte. En la Figura (2) los vectores de soporte son los 3 puntos (2 azules y 1 verde) que se encuentran en las líneas dispersas, y el hiperplano de separación es la línea roja sólida

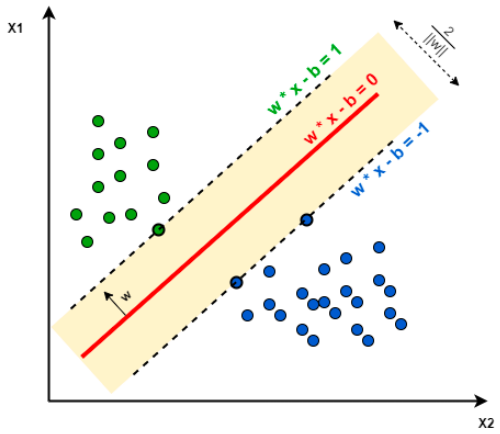


Figura 2: Hiperplano que separa puntos de datos de dos clases.

En su tipo más simple, SVM no admite la clasificación multiclase ya que es compatible con la clasificación binaria. Para la clasificación multiclase, se utiliza el mismo principio después de dividir el problema de multiclasificación en varios problemas de clasificación binaria.

La idea es mapear puntos de datos en un espacio de gran dimensión para obtener una separación lineal mutua entre cada dos clases. Esto se denomina enfoque uno a uno, que desglosa el problema multiclase en varios problemas de clasificación binaria. Un clasificador binario por cada par de clases. Otro enfoque que se puede utilizar es One-to-Rest. En ese enfoque, el desglose se establece en un clasificador binario por cada clase [Ben+01].

Una sola SVM realiza una clasificación binaria y puede diferenciar entre dos clases. En resúmen, los dos enfoques de desglose para clasificar los puntos de datos del conjunto

de datos de m clases son:

- El enfoque One-to-Rest, el clasificador puede usar m SVMs. Cada SVM predeciría la pertenencia a una de las clases m.
- El enfoque uno a uno, el clasificador puede usar $\frac{m(m-1)}{2}$ SVMs.

Consideremos el siguiente ejemplo de clasificación de tres clases: rojo, azul y verde (ver Figura (3)).

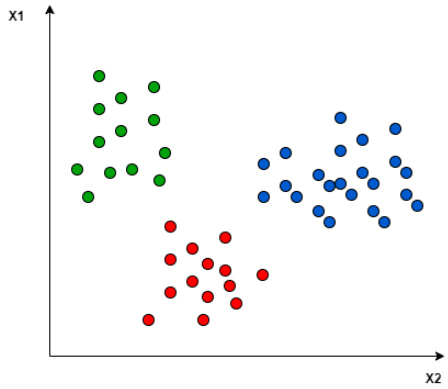


Figura 3: Ejemplo de clasificación multiclase.

En el enfoque uno a uno, se necesita un hiperplano para separar cada dos clases, descuidando los puntos de la tercera clase. Por ejemplo, la línea rojo-azul intenta maximizar la separación solo entre los puntos azul y rojo, ver figura (4). En el enfoque One-to-Rest,

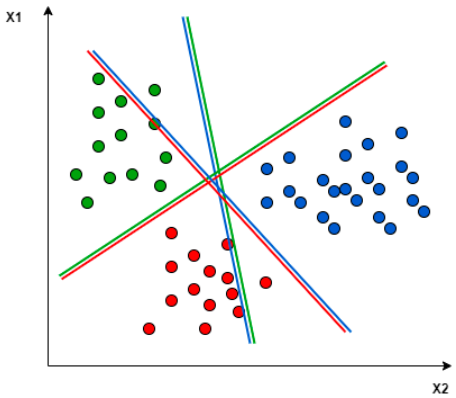


Figura 4: Enfoque uno a uno de SVM multiclase.

se necesita un hiperplano para separar una clase y todas las demás a la vez. Esto significa que la separación tiene en cuenta todos los puntos, dividiéndolos en dos grupos; un grupo para los puntos de la clase y un grupo para todos los demás puntos [Ben+01]. Por ejemplo, la línea verde intenta maximizar la separación entre los puntos verdes y todos los demás puntos a la vez, ver Figura (5).

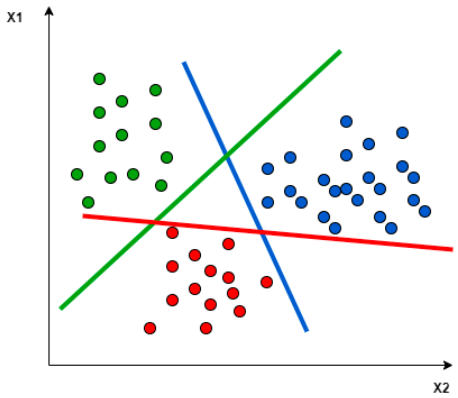


Figura 5: Enfoque uno a uno de SVM multiclase.

3.5. Perceptrón Multicapa (MLP)

Se sabe que la unidad básica de una red neuronal es una red que tiene un solo nodo, y esto se conoce como perceptrón. El perceptrón está formado por entradas $x_1, x_2, ..., x_n$ y sus correspondientes pesos $w_1, w_2, ..., w_n$. Luego una función conocida como función de activación toma estas entradas, las multiplica con sus pesos correspondientes y produce una salida y.

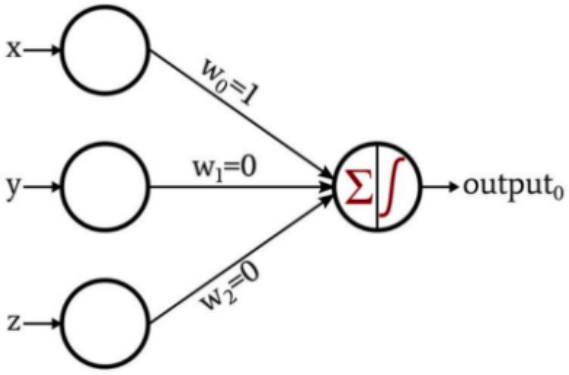


Figura 6: Un perceptrón

Por otro lado, un perceptrón multicapa es una clase de red neuronal que se compone de al menos 3 nodos. Además, cada uno de los nodos del perceptrón multicapa, excepto el nodo de entrada, es una neurona que utiliza una función de activación no lineal. Los nodos del perceptrón multicapa están compuestos en capa de entrada, capa de salida y capas ocultas (capas entre la entrada y la salida). El algoritmo de aprendizaje para el perceptrón multicapa se conoce como backpropagation [HTF09].

La función de activación mapea las entradas ponderadas a la salida de la neurona. Una de esas funciones de activación es la función sigmoide. Un ejemplo de función sigmoide

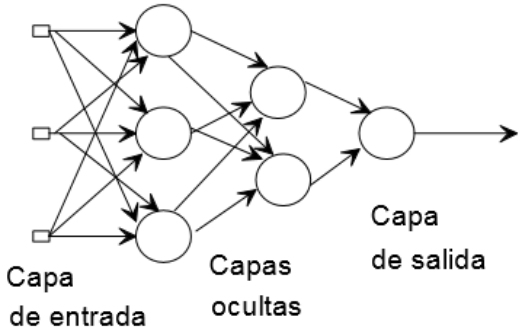


Figura 7: Perceptrón multicapa compuesto por varias neuronas, donde cada neurona es descrita por una función sigmoide.

es la función logística definida por

$$f(x) = \frac{1}{1 + e^{-\beta x}}.$$

Otro ejemplo de función sigmoide es la función de activación tangente hiperbólica, $f(x) = \tanh(x)$, la cual produce un output en el rango de -1 y 1.

Con la función de activación se puede calcular la salida de cualquier nuerona en el MLP [HTF09]. Asumiendo que $\mathbf{w}$ denota al vector de pesos, $\mathbf{x}$ al vector de inputs, b a los sesgos y φ denota a la función de activación. Para la i-ésima neurona el output y está dado por:

$$y = \left(\sum_{i=1}^{n} w_i x_i + b \right) = \varphi\left(\boldsymbol{w}^T \mathbf{x} + b\right).$$

3.6. Preparación de datos

Los datos utilizados se encuentran almacenados en un CSV como un data frame de pandas (Python 2.7). En el cuadro (1) se presentan las características del conjunto de datos.

Los datos contienen las siguientes 5 columnas (variables):

- count = número de usuarios de *CrowdFlower* que codificaron cada tweet (el mínimo es 3).
- hate_speech = número de usuarios de CF que juzgaron que el tweet era un discurso de odio.

Dataset	Características y porcentajes en el dataset	Fuente de Origen	Lenguaje
HatebaseTwitter [Dav+17]	Lenguaje de Odio: 5 % Lenguaje Ofensivo: 76 % Ninguno: 17 %	Twitter	Inglés

Cuadro 1: Características del conjunto de datos relacionados con el lenguaje de odio.

- offensive_language = número de usuarios de CF que juzgaron que el tweet era ofensivo.
- none = número de usuarios de CF que consideraron que el tweet no era ni ofensivo ni no ofensivo.
- class = etiqueta de clase para la mayoría de los usuarios de CF. 0 - discurso de odio 1 - lenguaje ofensivo 2 - ninguno.

En la figura (8) se muestra el resumen estadístico de los datos de [Dav+17].

	count	hate_speech	offensive_language	neither
count	24783.000000	24783.000000	24783.000000	24783.000000
mean	3.243473	0.280515	2.413711	0.549247
std	0.883060	0.631851	1.399459	1.113299
min	3.000000	0.000000	0.000000	0.000000
25%	3.000000	0.000000	2.000000	0.000000
50%	3.000000	0.000000	3.000000	0.000000
75%	3.000000	0.000000	3.000000	0.000000
max	9.000000	7.000000	9.000000	9.000000

Figura 8: Descripción de los datos

Los tweets del conjunto de datos contienen: direcciones URL, menciones o muchos espacios en blanco. Estas cadenas de texto son innecesarias ya que no aportan información relevante para el transfondo del tweet.A continuación se muestran algunos de estos sin preproceso:

```
!!!!! RT @mleew17: boy dats cold...tyga dwn bad for cuffin dat hoe in the
1st place!!
```

```
!!!!!!"@__BrighterDays: I can not just sit up and HATE on another bitch .. I got
too much shit going on!"

!!!!&#8220;@selfiequeenbri: cause I'm tired of you big bitches coming for us
skinny girls!!&#8221;

" @rhythmixx_ :hobbies include: fighting Mariam" bitch

" So hoes that smoke are losers ? " yea ... go on IG
```

Para el preprocesamiento del texto, primero se cambiaron todos los caracteres a minúsculas, después se eliminaron URLs, menciones y espacios en blanco. Una vez preprocesado el texto se obtuvieron tweets modificados sin carácteres innecesarios. Algunos ejemplos se muestran a continuación:

```
Tweet original:
    !!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby4life: You ever fuck a
    bitch and she start to cry? You be confused as shit
Tweet preprocesado:
    !!!!!!! RT Dawg!!!! RT : You ever fuck a bitch and she start to cry?
    You be confused as shit

Tweet original:
    !!!!&#8220;@selfiequeenbri: cause I'm tired of you big bitches coming
    for us skinny girls!!&#8221;
Tweet preprocesado:
    !!!!&#8220;: cause I'm tired of you big bitches coming for us skinny
    girls!!&#8221

Tweet original:
    " pussy is a powerful drug " &#128517; #HappyHumpDay http://t.co/R8jsymiB5b
Tweet preprocesado:
    " pussy is a powerful drug " &#128517; #HappyHumpDay
```

Por último, para proceder con el procesamiento de texto se tokenizaron los tweets por, a lo más, triadas de caracteres. La tokenización es una forma de separar un fragmento de texto en unidades más pequeñas llamadas tokens. Aquí, los tokens pueden ser palabras, caracteres o subpalabras.

4. Resultados

Primeramente veamos como están clasificados los tweets de la base de datos, así como también la nuble de palabras, es decir las palabras que más se utilizan para nuestras tres categorías, discurso de odio, lenguaje ofensivo y ninguno:

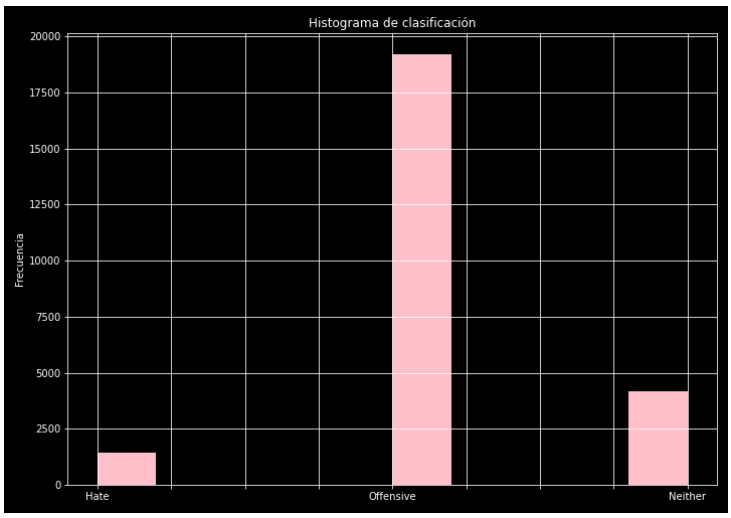
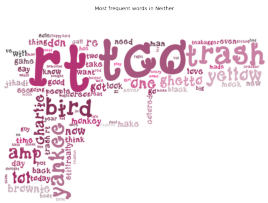


Figura 9: Palabras más frecuentes en tweets clasificados en el grupo “offensive language”

(a) Palabras más frecuentes en tweets clasificados en el grupo “offensive language”

(b) Palabras más frecuentes en tweets clasificados en el grupo “hate speech”



(c) Palabras más frecuentes en tweets clasificadas en el grupo “neither”

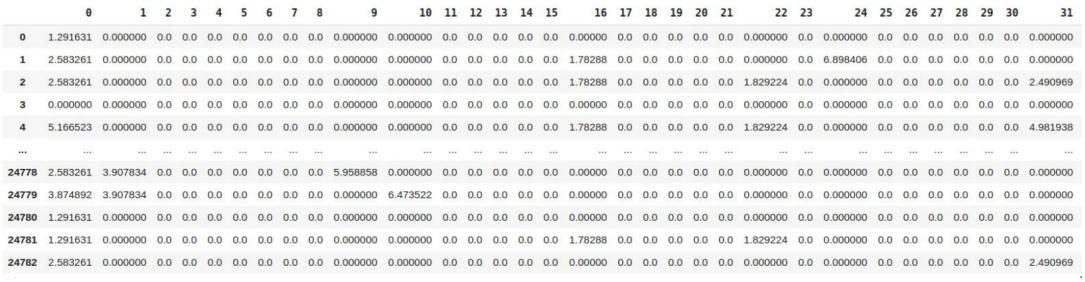
Figura 10: Three simple graphs

Podemos observar de la figura (9) que la mayoria de los tweets fueron clasificados como lenguaje ofensivo.

Las matrices que vamos a utilizar fueron descritas en la sección 3.1, y tienen la siguiente forma:

	and	antagonistic	are	cats	dogs	four	hate	have	he	legs
vector1	0.000000	0.000000	0.000000	0.402040	0.000000	0.528635	0.000000	0.528635	0.000000	0.528635
vector2	0.490479	0.490479	0.490479	0.373022	0.373022	0.000000	0.000000	0.000000	0.000000	0.000000
vector3	0.000000	0.000000	0.000000	0.000000	0.473630	0.000000	0.622766	0.000000	0.622766	0.000000

Figura 11: Matriz TF-IDF



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	1.291631	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	0.00000	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.000000
1	2.583261	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	1.78288	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	6.898406	0.0	0.0	0.0	0.0	0.0	0.0	0.000000
2	2.583261	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	1.78288	0.0	0.0	0.0	0.0	0.0	1.829224	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	2.490969
3	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	0.00000	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.000000
4	5.166523	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	1.78288	0.0	0.0	0.0	0.0	0.0	1.829224	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	4.981938
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
24778	2.583261	3.907834	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.958858	0.000000	0.0	0.0	0.0	0.0	0.0	0.00000	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.000000
24779	3.874892	3.907834	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	6.473522	0.0	0.0	0.0	0.0	0.0	0.00000	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.000000
24780	1.291631	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	0.00000	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.000000
24781	1.291631	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	1.78288	0.0	0.0	0.0	0.0	0.0	1.829224	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.000000
24782	2.583261	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.000000	0.0	0.0	0.0	0.0	0.0	0.00000	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	2.490969

Figura 12: Matriz TF-IDF + Part of speech + other features

Donde las filas son los tweets y las columnas son las características. Para ambas matrices utilizamos los siguientes modelos:

- Árboles de decisión.
  - Gini.
  - Entropy.
- Regresión Logística.
  - Penalización L1.
  - Penalización L1 con datos estandarizados.
  - Penalización L2.
  - Penalización Red Elástica.
  - Penalización Red Elástica con datos estandarizados.
- Máquinas de vector soporte (SVM).
- Perceptrón Multicapa (MLP).

Además, la precisión (accuracy) mostrado para ambas matrices se obtiene de la siguiente manera:

$$\text{Precisión} = \frac{\text{VN} + \text{VP}}{\text{VN} + \text{FN} + \text{VP} + \text{FP}},$$

donde VN y VP son los verdaderos negativos y verdaderos positivos respectivamente, i.e., las observaciones clasificadas correctamente. FN y FP son los falsos negativos y falsos positivos respectivamente. En otras palabras, la precisión está dada por el cociente de las observaciones bien clasificadas sobre todas las observaciones.

4.1. Matriz TF-IDF

A continuación se muestran los diferentes histogramas para cada una de las categorías de nuestros clasificadores

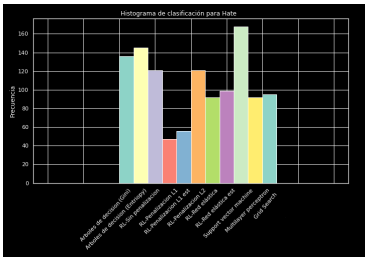


Figura 13: Hate Speech

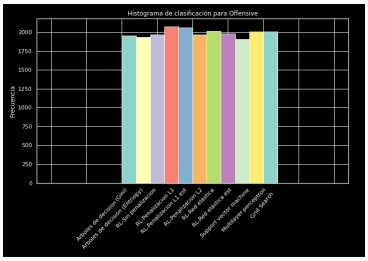


Figura 14: Offensive

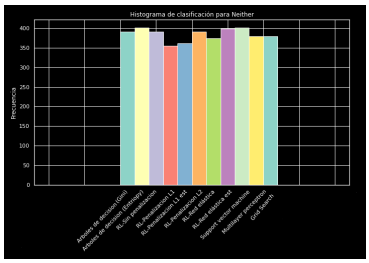


Figura 15: Neither

El precisió global de cada método es:

Metodo	Precisión
Arboles de decision (Gini)	0.79911
Arboles de decision (Entriopy)	0.78338
RL-Sin penalizacion	0.82049
RL-Penalizacion L1	0.84509
RL-Penalizacion L1 est	0.83541
RL-Penalizacion L2	0.82049
RL-Red elática	0.83582
RL-Red elástica est	0.81686
Máquinas de vector soporte	0.78781
Perceptrón Multicapa	0.82936
Grid Search	0.83582

Cuadro 2: Accuracy de todos los métodos

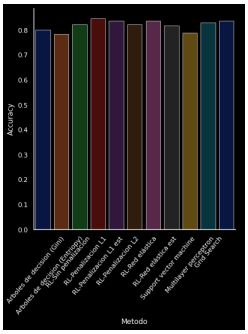


Figura 16: Accuracy de todos los métodos

Como podemos observar de la tabla [2], los métodos que mejor precisión global tienen son:

- Regresión Logistica (L1).
- Red Elástica.
- Perceptrón Multicapa.

Sin embargo, al analizar sus matrices de confución notamos que:

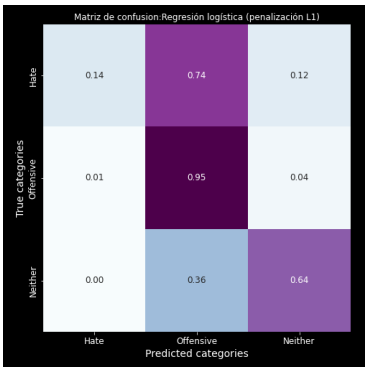


Figura 17: Regresión Logistica (L1)

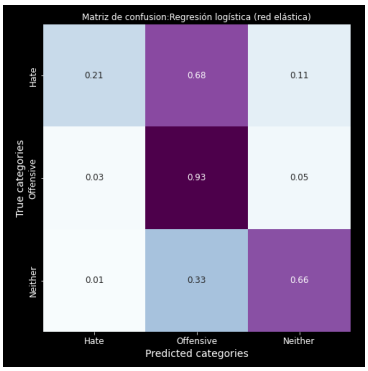


Figura 18: Red Elástica.

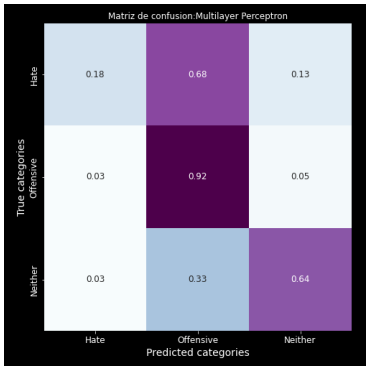


Figura 19: Perceptrón Multicapa.

Ninguno de estos modelos clasifica bien el discurso de odio, ya que por un lado no se diferencia el discurso de odio respecto al lenguaje ofensivo, y por otro lado se puede considerar erróneamente un discurso de odio.

Si bien los modelos mencionados obtuvieron mayor porcentaje de precisión los modelos que *mejor* clasifican el discurso de odio son:

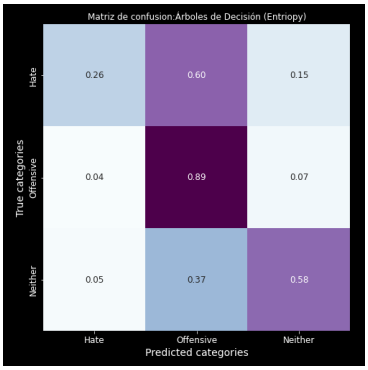


Figura 20: Arboles de decision (Gini).

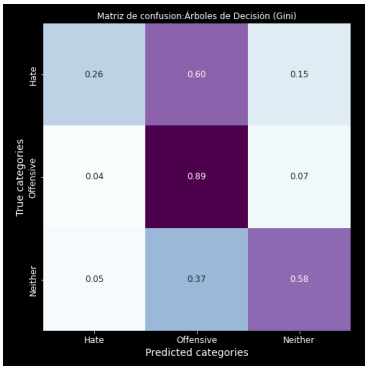


Figura 21: Arboles de decision (Entropy).

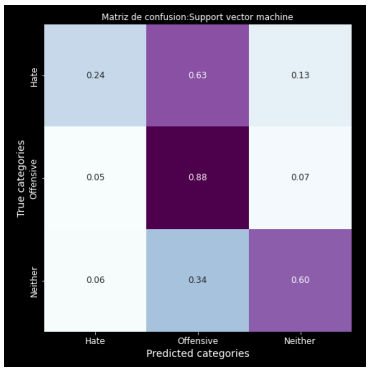


Figura 22: Máquinas de vector soporte.

A pesar de ello, los procentajes de clasificación que arrojan siguen siendo bajos (26 %). Ahora bien, procedemos a utilizar nuesto conjunto de datos de validación para examinar los resultados de nuestros clasificadores obtenidos.

Validación

- Regresión logística con penalización L1

```
tweet: @SchulzGrayson @Dswizzle3 oh woulda coulda shoulda ass niggas bitch
       ass nigga, what u bout oh ban wagging ass nigga catch your phase
True:  2
Pred:  1
```

```
tweet: I want my bitch to answer if my ex calling, with my dick in her mouth
True:  2
Pred:  1
```

Notemos que en este último tweet se aprecia que se trata de lenguaje ofensivo, ya que no va dirigido a un grupo de personas. En este caso, el clasificador lo categorizó correctamente.

- Regresión logística con penalización de red elástica

```
tweet: @MoenKiana fine walk bitch
True:  0
Pred:  2

tweet: @_Saltlife13 Long story lol but I've missed you faggot
True:  0
Pred:  1
```

- MLP

```
tweet: These bitches boringggggg
True:  2
Pred:  1

tweet: RT @Noworriezzzz: Don't lie to me bitch I'm giving you my heart
True:  0
Pred:  1
```

Notemos que el clasificador MLP clasificó correctamente los tweets mientras que los moderadores los clasificaron erroneamente.

4.2. Matriz TFIDF + Part of Speech + Other features

Los histogramas para cada una de las categorias fueron los siguientes:

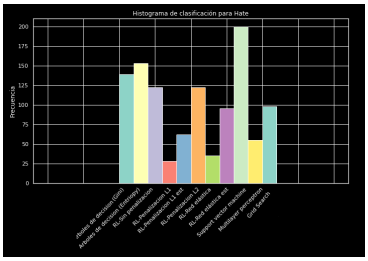


Figura 23: Hate Speech

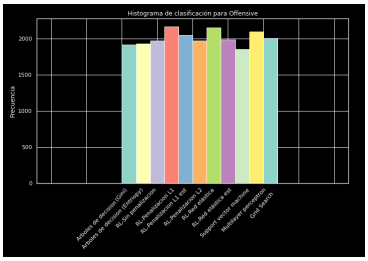


Figura 24: Offensive

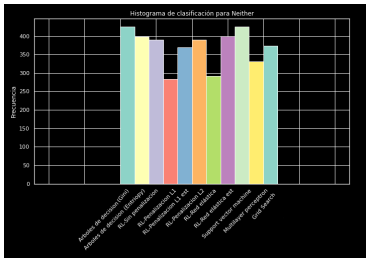


Figura 25: Neither

El accuracy global de cada método es:

Método	Accuracy
Arboles de decisión (Gini)	0.80072
Arboles de decisión (Entriopy)	0.78580
RL-Sin penalización	0.82815
RL-Penalización L1	0.83945
RL-Penalización L1 est	0.84751
RL-Penalización L2	0.82815
RL-Red elática	0.84429
RL-Red elástica est	0.82170
Máquinas de vector soporte	0.77450
Perceptrón Multicapa	0.83743
Grid Search	0.83743

Cuadro 3: Accuracy de todos los métodos

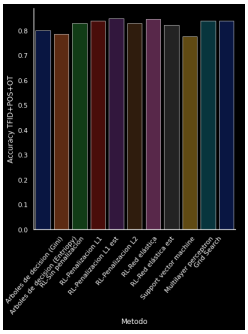


Figura 26: Accuracy de todos los métodos

Los métodos que mejor precisión global tienen son:

- Regresión Logistica con penalización L1.
- Regresión Logistica con penalización L1 y dátos estandarizados.
- Red Elástica.

Sin embargo, ocurre algo similar que con los resultados TF-IDF ya que al analizar sus matrices de confusión notamos que:

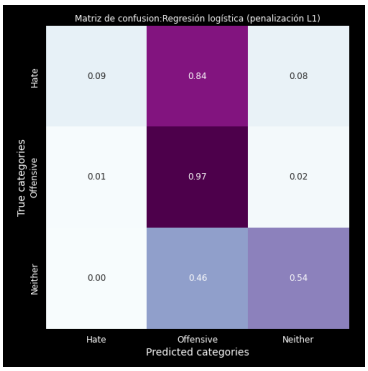


Figura 27: Regresión Logistica (L1)

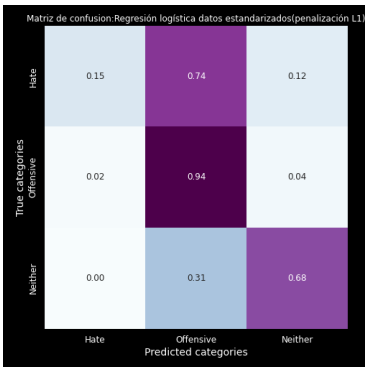


Figura 28: Regresión Logistica (L1), datos estandarizados.

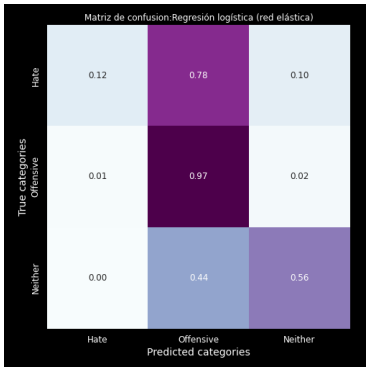


Figura 29: Red Elástica.

Ninguno de estos modelos clasifica bien el discurso de odio, ya que por un lado no se diferencia el discurso de odio respecto al lenguaje ofensivo, y por otro lado se puede considerar erróneamente un discurso de odio.

En cambio,los modelos que *mejor* clasifican el discurso de odio son:

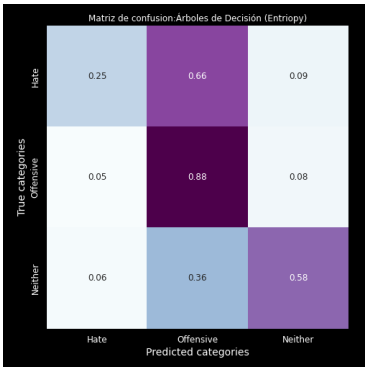


Figura 30: Arboles de decision (Gini).

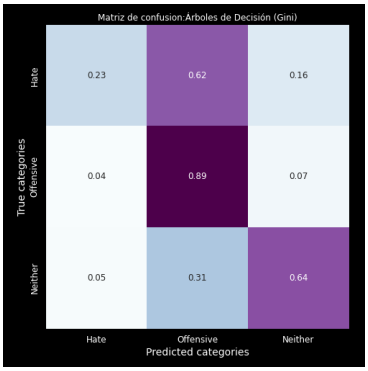


Figura 31: Arboles de decision (Entropy).

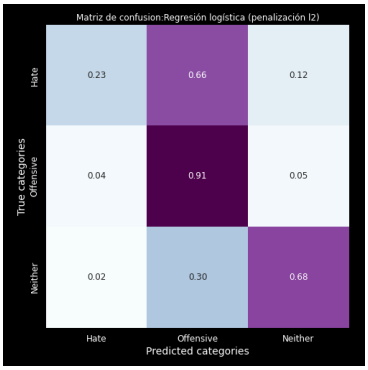


Figura 32: Regresión Logistica (L2).

A pesar de ello, ocurre algo muy singular, los porcentajes de clasificación que arrojan son aún más bajos que los obtenidos por TF-IDF (25 %).
Ahora bien, procedemos a utilizar nuesto conjunto de datos de validación para examinar los resultados de nuestros clasificadores obtenidos.

Validación

- Regresión logística con penalización L1

```
tweet: @realDonaldTrump he looks like reg. Memphis,tn. trash! we got
       them everywhere.
True:  2
Pred:  1
```

```
tweet: @_Saltlife13 Long story lol but I've missed you faggot
True:  0
Pred:  1
```

- Regresión logística con penalización de red elástica

```
tweet: @_Saltlife13 Long story lol but I've missed you faggot
True:  0
Pred:  1

tweet: No time for bitch niggas.
True:  2
Pred:  1
```

En este último tweet se equivocaron los métodos de regresión logística con penalizaciones L1 y de red elástica. Este tweet en realidad no es discurso de odio ni lenguaje ofensivo pero por tener la palabra "faggot"lo categorizó como ofensivo.

- MLP

```
tweet: @HollygroveShawn word!! school flow on a dumb hoe...
True:  0
Pred:  1

tweet: No time for bitch niggas.
True:  2
Pred:  1
```

5. Discusión

De este trabajo tuvimos varias cosas que están a discusión, como por ejemplo que los tweets con groserías tendían más a ser considerados discurso de odio que lenguaje de odio. También, los comentarios sexistas no se clasificaban como discursos de odio, cuando en realidad deberían de serlo ya que se dirigen a un grupo de personas.

Un dato curioso que notamos en este trabajo es que con el segundo análisis realizado el modelo de regresión lineal obtenía mejores resultados que los que dió la red neuronal de MLP, esto pudo ser así por el sobre ajuste de la red o por como estaban dispuestos los datos de la matriz, aún así fue una observación interesante.

Y por último pero no menos importante, nos llamo la atención que aunque la diferencia no fue mucha, se tuvo mayor clasificación para el discurso de odio con los datos obtenidos de solo utilizar la matriz de TF-IDF que con la matriz que consideraba TF-IDF, POS y el análisis de sentimiento.

6. Conclusiones

Descubrimos que independientemente del modelo, la detección de discurso de odio sigue siendo un tema complicado, y por tanto, aún hay mucho por hacer. En cuanto al sesgo inherente en las bases de datos Davidson et., al., [Dav+17] se lograron identificar que ciertas bases de datos ocupadas (como baseline) en diversos artículos están sesgadas, esto usualmente es a causa de la elección del protocolo de anotación, es decir, quién y cómo se va a llevar a cabo la clasificación manual de los tweets.

Bibliografía

[HK70] Arthur E. Hoerl y Robert W. Kennard. "Ridge Regression: Biased Estimation for Nonorthogonal Problems". En: *Technometrics* 12.1 (1970), págs. 55-67. DOI: 10.1080/00401706.1970.10488634. eprint: https://www.tandfonline.com/doi/pdf/10.1080/00401706.1970.10488634. URL: https://www.tandfonline.com/doi/abs/10.1080/00401706.1970.10488634.

[Tib96] R. Tibshirani. "Regression Shrinkage and Selection via the Lasso". En: *Journal of the Royal Statistical Society (Series B)* 58 (1996), págs. 267-288.

[Ben+01] Asa Ben-Hur y col. "Support Vector Clustering". En: *Journal of Machine Learning Research* 2 (nov. de 2001), págs. 125-137. DOI: 10.1162/15324430260185565.

[Jac02] James B. Jacobs. "Hate Crime: Criminal Law and Identity Politics: Authorâs summary". En: *Theoretical Criminology* 6.4 (2002), págs. 481-484. DOI: 10.1177/136248060200600406. eprint: https://doi.org/10.1177/136248060200600406. URL: https://doi.org/10.1177/136248060200600406.

[ZH05] Hui Zou y Trevor Hastie. "Regularization and variable selection via the Elastic Net". En: *Journal of the Royal Statistical Society, Series B* 67 (2005), págs. 301-320.

[Wu+07] Xindong Wu y col. "Top 10 Algorithms in Data Mining". En: *Knowl. Inf. Syst.* 14.1 (dic. de 2007), 1â37. ISSN: 0219-1377. DOI: 10.1007/s10115-007-0114-2. URL: https://doi.org/10.1007/s10115-007-0114-2.

[HTF09] T. Hastie, R. Tibshirani y J.H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer, 2009. ISBN: 9780387848846. URL: https://books.google.com.mx/books?id=eBSgoAEACAAJ.

[WH12] William Warner y Julia Hirschberg. "Detecting hate speech on the world wide web". En: jun. de 2012, págs. 19-26.

[RM14] Lior Rokach y Oded Maimon. *Data Mining With Decision Trees: Theory and Applications*. 2nd. USA: World Scientific Publishing Co., Inc., 2014. ISBN: 9789814590075.

[Dav+17] Thomas Davidson y col. "Automated Hate Speech Detection and the Problem of Offensive Language". En: *Proceedings of the International AAAI Conference on Web and Social Media* 11.1 (mayo de 2017), págs. 512-515. URL: https://ojs.aaai.org/index.php/ICWSM/article/view/14955.

[Mac+19] Sean MacAvaney y col. "Hate speech detection: Challenges and solutions". En: *PLoS ONE* 14 (2019).