

## Aplicación de algoritmos de aprendizaje automático sobre un corpus depresivo digital

César-Jesús Núñez-Prado<sup>2</sup>, Claudia Talavera Ortega<sup>1</sup>,  
Liliana Chanona-Hernández<sup>1</sup>, Grigori Sidorov<sup>2</sup>

<sup>1</sup> Instituto Politécnico Nacional,  
Escuela Superior de Ingeniería Mecánica y Eléctrica,  
México

<sup>2</sup> Instituto Politécnico Nacional,  
Centro de Investigación en Computación,  
México

{cesar.jnprado, claudiatalaveraor, lchanona}@gmail.com,  
sidorov@cic.ipn.mx

**Resumen.** Uno de los principales objetivos que estimula al avance tecnológico; sin importar el campo de desarrollo, es el de hacer más sencilla y cómoda la vida de las personas. Estos avances en el área de la inteligencia artificial han impulsado a los algoritmos de aprendizaje automático para realizar la clasificación de ciertos objetos gracias al reconocimiento de patrones que descubren durante las fases del entrenamiento. Dichos algoritmos en conjunto con técnicas de procesamiento de lenguaje natural están capacitados para realizar la detección de emociones en el discurso escrito. Una de estas emociones es la depresión, la cual está catalogada como uno de los trastornos más comunes y además más letales que puede afectar a la población mundial y ello nos motiva a dirigir nuestra investigación a aplicar algunos modelos de aprendizaje automático a la detección del posible discurso de depresión sobre un corpus lingüístico digital en español construido a partir de mensajes publicados en la red social Twitter y evaluado por profesionales en el campo de la psicología.

**Palabras clave:** Inteligencia artificial, reconocimiento de patrones, aprendizaje automático, procesamiento de lenguaje natural, depresión.

### Application of Machine Learning Algorithms on a Digital Depressive Corpus

**Abstract.** One of the main objectives that stimulates technological progress; regardless of the field of development, is to make people's lives simpler and more comfortable. These advances in the area of artificial intelligence have driven machine learning algorithms to perform the classification of certain objects thanks to the recognition of patterns that they discover during the training phases. These algorithms in conjunction with natural language processing techniques are capable of detecting emotions in written speech. One of these emotions is depression, which is ranked as one of the most common and also one of the most lethal disorders that can affect the world population and this motivates us to direct our research to apply some machine learning models to the detection of possible

depression discourse on a digital linguistic corpus in Spanish built from messages posted on the social network Twitter and evaluated by professionals in the field of psychology.

**Keywords:** Artificial intelligence, pattern recognition, machine learning, natural language processing, depression.

## 1. Introducción

De acuerdo con el Instituto Nacional de Estadística y Geografía en México (INEGI)<sup>1</sup>, entre el 2016 y el 2019 se emitieron más de 140 millones de publicaciones en las redes sociales, por otra parte; la estadística<sup>2</sup> estima que entre las aplicaciones más populares en personas con un rango de edad entre los 16 y los 64 años, se encuentran: Facebook, WhatsApp, Instagram, TikTok, Twitter y Telegram.

Toda esta información respalda que el uso de las redes sociales es tan común que se ha convertido en un aspecto muy importante en la vida de muchos adolescentes y adultos jóvenes, los cuales utilizan este tipo de comunicación para compartir con otros usuarios aspectos relevantes y no relevantes de su vida diaria.

Principalmente, las redes sociales están enfocadas a compartir videos, imágenes o texto con cualquier usuario que tenga una cuenta activa dentro de la misma red social, (Aunque es posible que, en algunas redes sociales, las publicaciones solo estén disponibles para un grupo de personas en específico), a cualquier hora y en cualquier parte del mundo.

La mayoría de estas redes no poseen un filtro eficaz para validar si la información con la cual se aperturan las cuentas es verídica o falsa, por tal motivo; la identidad de muchos de los usuarios se puede conservar en un completo anonimato. Este tipo de acciones puede repercutir de manera negativa, ya que detrás del perfil creado se pueden esparcir noticias falsas, se pueden ocultar acosadores, etc.

En el anonimato no todo es negativo ya que cuando el usuario no se siente observado y enjuiciado todo el tiempo, se puede sentir con la libertad y confianza de expresar sus propios sentimientos reales y ello da la pauta para poder realizar el análisis de emociones en las publicaciones dentro de las redes sociales.

En los últimos años, el análisis de los sentimientos en las redes sociales ha ido en incremento y no sólo con fines de mercadotecnia sino con fines de índole social, ya que se analizan los sentimientos negativos, como la depresión, y la finalidad de ello es intentar reducir el índice de consecuencias letales que se pueden presentar en personas con este tipo de trastorno.

Esta investigación está enfocada a aplicar técnicas de procesamiento de lenguaje natural y algunos algoritmos de aprendizaje automático sobre un corpus digital en español desarrollado a partir de publicaciones en *Twitter* con la finalidad de analizar los mensajes y detectar de manera automática si el mensaje puede ser clasificado como un mensaje depresivo.

---

<sup>1</sup>[www.inegi.org.mx/inegi/sociales.html](http://www.inegi.org.mx/inegi/sociales.html)

<sup>2</sup>[es.statista.com/estadisticas/1035031/mexico-porcentaje-de-usuarios-por-red-social/](https://es.statista.com/estadisticas/1035031/mexico-porcentaje-de-usuarios-por-red-social/)

Vector de vocabulario							
hartar	morir	muerte	odiar	...	olvidar	querer	vida
Vector de presencia y ausencia							
1	0	1	1	...	0	1	1

Fig. 1. Ejemplo del vector de vocabulario y el vector de presencia y ausencia.

70 - 30				
80 - 20				
90 - 10				

Fig. 2. Ejemplo de las métricas de validación.

La estructura de este artículo será la siguiente, se presentarán algunos trabajos relacionados, la aplicación de la metodología y finalmente se mostrarán los resultados obtenidos y las conclusiones finales de la investigación.

## 2. Trabajos relacionados

Los avances tecnológicos a nivel mundial forzaron la evolución de los medios de comunicación y con ello se modificó la manera tradicional de transmitir información. El alto uso de las redes sociales permite que toda la información publicada en ellas pueda ser analizada tanto con fines lucrativos como con fines de índole social.

En el análisis de texto extraído de Twitter encontramos a [1] en donde aplican técnicas de procesamiento de lenguaje natural para realizar un estudio social entre las publicaciones de los usuarios. Aplicaron una metodología que es capaz de rastrear<sup>3</sup> publicaciones semejantes, de acuerdo a ciertos dominios de información y los códigos fuentes fueron desarrollados en el lenguaje de programación python.

Entre las investigaciones de índole social se encuentra [2] en donde aplican minería de datos en el campo de la sicología para detectar depresión en usuarios de redes sociales de China. Su trabajo reporta una precisión aproximada del 80 % y tuvieron apoyo de la comunidad médica (sicólogos especializados en el tema).

El análisis de las emociones en las publicaciones en usuarios de redes sociales en China ha ido incrementando año con año debido al alza en las estadísticas de la tasa de suicidio, por ello en [3] aplican técnicas de análisis de sentimientos para detectar casos potenciales de depresión entre usuarios menores de edad.

<sup>3</sup> El término en inglés se conoce como «crawling»

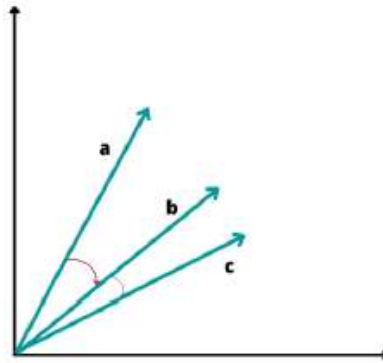


Fig. 3. Representación de 3 vectores.

Crearon una base de conocimiento de recursos encontrados en internet enfocados a la misma tarea y con el apoyo de psicólogos generaron una lista de vocabulario del sentimiento depresivo. Una vez que realizan el procesamiento de la información y obtienen la clasificación de los mensajes, se pondera si es necesario enviar una notificación de alarma a trabajadores sociales o a los padres de los usuarios.

Por último, en [4] indican que existe una correlación entre el alto uso de las redes sociales y el incremento de la depresión en las personas. En su investigación aplicaron los clasificadores: Máquinas de vectores de soporte (Support Vector Machines SVM), Bayes ingenuo (Naive Bayes) y árboles de decisión (Decision Trees). Compararon el historial de las publicaciones y generaron una métrica para calificar si era posible o no, que el usuario sufriera de depresión de acuerdo a los cambios de comportamiento en sus publicaciones en el último periodo de tiempo.

### 3. Aplicación de la metodología

En esta sección se explicará de manera detallada el proceso que se siguió para la aplicación de los algoritmos de aprendizaje automático sobre el corpus lingüístico digital en español enfocado a la depresión.

#### 3.1. Preprocesamiento del corpus

El corpus lingüístico digital en español enfocado a la depresión cuenta con 1,623 mensajes y se encuentra etiquetado con sólo dos opciones: 1 (clase uno) si el mensaje es depresivo y 0 (clase cero) si el mensaje es no depresivo. Este corpus fue evaluado por personal profesional de la salud en el campo de la psicología con experiencia en la atención de pacientes en el área de la depresión. El procesamiento que se realizó sobre cada uno de los mensajes que componen el corpus se desarrolló en Python<sup>4</sup> y fue el siguiente:

---

<sup>4</sup> Se utilizó la versión 3.9.12 de Python.

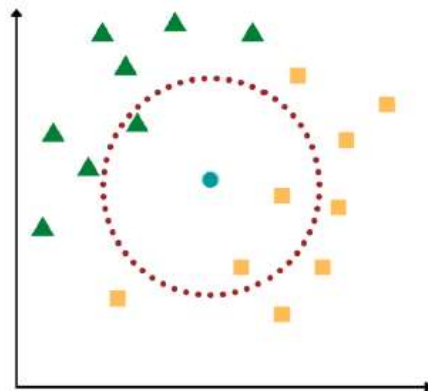


Fig. 4. Representación gráfica de k-vecinos más cercanos.

- Tokenización (es el proceso mediante el cual se obtienen las unidades mínimas de las oraciones, es decir; se separan las oraciones en palabras).
- Lematización (es la acción de encontrar la palabra raíz sin flexionar, por ejemplo: para el verbo «corriendo» el lema es «correr»).
- Etiquetado de las palabras (part of speech tagging) (un ejemplo de etiquetado: si aparece la palabra «correr» la etiqueta asociada es «verbo»).
- Eliminación de stop-words (es el conjunto de palabras que incluyen a los artículos definidos, indefinidos, preposiciones, etc).

Para la tokenización y la eliminación de stop-words se utilizó la biblioteca de NLTK (por sus siglas en inglés, Natural Language ToolKit)<sup>5</sup> y para el proceso de lematización y el etiquetado de las palabras se usó la biblioteca Stanza<sup>6</sup>.

### 3.2. Vectorización

Debido a que no es posible realizar operaciones matemáticas sobre las palabras considerándolas sólo cadenas de caracteres, se aplicó la vectorización de los mensajes utilizando la técnica one hot encoding, la cual es capaz de transformar un mensaje en un vector numérico de presencia y ausencia.

El procedimiento inicial en esta técnica es formar un vector de vocabulario, este se consigue guardando en un arreglo cada una de las palabras del corpus sin repetir, es decir; contendrá sólo palabras únicas (por ello es importante el proceso de realizar la lematización de las entradas, ya que en diferentes mensajes se podrían encontrar las palabras «odié» y «odiado» y sólo se agregarían representaciones flexionadas de la misma raíz «odiar»). Generalmente, este vector ordena de manera alfabética todas las entradas.

Una vez que se cuenta con el vector de vocabulario, se procesa cada mensaje de manera individual y se genera un vector con sólo unos y ceros, en donde el uno

<sup>5</sup> Se utilizó la versión 3.7 de NLTK.

<sup>6</sup> Se utilizó la versión 1.4.2 de Stanza.

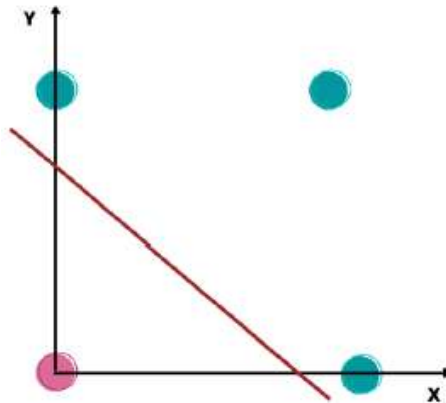


Fig. 5. Representación de clasificación con una red neuronal.

representa presencia y el cero ausencia. Este nuevo vector tiene exactamente la misma longitud que el vector de vocabulario. En la Figura 1 se visualiza un ejemplo de vectorización en ausencia y presencia.

### 3.3. División del corpus

Ya que se realizó la transformación de los mensajes a vectores numéricos, se realizó la segmentación del corpus con la finalidad de disponer de un rango de información para que los algoritmos escogidos puedan aprender y otra sección para examinar el funcionamiento de los mismos.

Se escogieron los métodos de validación 70 – 30, 80 – 20 y 90 -10, en donde la primera cifra corresponde al conjunto de entrenamiento y la segunda cifra al conjunto de prueba. De la figura anterior, los recuadros color naranja en cada métrica de validación representan al conjunto de entrenamiento y los recuadros azules son el conjunto sobre el cual se realizarán las pruebas.

### 3.4. Clasificación

Para realizar la etapa de la clasificación se escogieron 3 algoritmos de clasificación supervisada<sup>7</sup>: similitud coseno, k-vecinos más cercanos y perceptrón multicapa y a continuación se explicará de manera general el funcionamiento de cada uno de ellos.

#### 3.4.1. Similitud coseno

Este algoritmo de clasificación compara dos vectores con las mismas dimensiones y obtiene un resultado que se encuentra entre cero y uno; cuando el cálculo obtenido tiende hacia cero<sup>8</sup> implica que los vectores casi no se parecen entre sí y por el contrario,

<sup>7</sup> En el paradigma supervisado se cuenta con las entradas etiquetadas, es decir; desde un inicio se conoce el resultado esperado y con ello se puede verificar de manera automática si los resultados obtenidos son correctos o incorrectos.

<sup>8</sup> Cuando los vectores tienen una similitud de cero se dice que los vectores son ortogonales entre sí.

**Tabla 1.** Precisión similitud coseno.

Algoritmo	Métrica	Precisión
Similitud coseno	90 – 10	71.62 %
	80 – 20	73.43 %
	70 – 30	75.42 %

cuando el resultado tiende hacia uno, se dice que los vectores se parecen demasiado entre sí.

Supongamos que la Figura 3 representa la gráfica en dos dimensiones de 3 vectores y se desea clasificar al vector «b» aplicando similitud coseno, se obtendrían dos resultados, el resultado entre «b y c» sería mayor que el resultado entre «b y a» por lo cual, al vector «b» se le asignaría la clase del vector «c».

Al aplicar este algoritmo sobre el corpus vectorizado, se debe calcular la similitud coseno del vector que se desea clasificar con todos los vectores del conjunto de entrenamiento, después buscar cual fue el resultado mayor, identificar su clase y asignarla al vector objetivo.

### 3.4.2. k-vecinos más cercanos

Este algoritmo de clasificación busca encontrar un número finito de elementos cercanos al elemento a clasificar y después realizar un conteo de clases entre dichos elementos, la clase con el número mayor de votos en este conteo es la clase que se asigna al vector objetivo.

Cabe mencionar, que cuando se trata de clasificación bi-clase (sólo 2 posibles clases a asignar), se escoge un número impar entre los elementos más cercanos a buscar con la finalidad de evitar empates entre los conteos.

De la Figura 4 supongamos que deseamos clasificar al círculo azul y que tenemos solo dos posibles clases «triángulo verde» y «cuadro amarillo», para evitar empates en el conteo elegimos  $k = 3$ , esto implica que el algoritmo deberá identificar los 3 elementos más cercanos<sup>9</sup> al elemento a clasificar y realizar el conteo.

En dicho conteo habría «triángulo verde = 1» y «cuadro amarillo = 2» por lo que la asignación de la clase al círculo azul sería cuadro amarillo. Los parámetros escogidos para este algoritmo con el corpus de depresión fue calcular la distancia euclidiana entre los vectores y el número de vecinos se varió con  $k = 3, k = 5, k = 7, k = 9$  y  $k = 11$ .

### 3.4.3. Perceptrón multicapa

Este algoritmo pertenece a un conjunto de algoritmos conocidos como redes neuronales artificiales en donde se busca replicar (reproducir) el funcionamiento de las redes neuronales biológicas. De manera general, este tipo de redes están compuestas de 3 capas: la capa de entrada, la capa oculta y la capa de salida y cada capa puede contener «n» número de neuronas; cada neurona puede o no, estar enlazada con las demás capas por medio de aristas (pesos) que generalmente se inicializan con valores aleatorios.

<sup>9</sup> Generalmente se calcula la distancia euclidiana entre los elementos.

**Tabla 2.** Precisión k-vecinos más cercanos.

Algoritmo	Métrica	<i>k</i>	Precisión
k-vecinos más cercanos	90 – 10	3	76.68 %
	70 – 30	5	79.70 %
	90 – 10	7	81.75 %
	90 – 10	9	82.09 %
	70 – 30	11	82.29 %

Desde que se presenta información a clasificar en la capa de entrada hasta que se obtiene un valor en la capa de salida se le conoce como época y las neuronas de la capa oculta actúan a través de funciones de activación.

El resultado obtenido en la capa de salida se compara con el resultado esperado y de ser diferente se realiza un cambio en todos los pesos de la red neuronal. Este último proceso se suspende de acuerdo a dos condiciones: que se cumpla el número de épocas máximo establecido o que se alcance la precisión correcta esperada.

Consideremos que se desea clasificar de la Figura 5 al círculo color rosa a través de aplicar una red neuronal artificial, de manera gráfica la capa oculta de la red deberá dibujar una línea (llamado hiperplano en más de dos dimensiones) que sea capaz de dividir los elementos de las clases.

Cuando se compara la salida con el resultado esperado y se verifica que no se clasificó de manera correcta, los pesos de la red se modifican y con ello se modifica la ubicación de la línea.

Este proceso finaliza cuando la línea ha podido dividir correctamente ambas clases, es decir; los elementos de una clase se encuentran por encima de la línea y los elementos de la otra clase se encuentra por debajo<sup>10</sup>. Dentro de los parámetros que se eligieron para examinar el corpus fueron los siguientes:

- Número de épocas = 300.
- Taza de aprendizaje = 0.001.
- Funciones de activación:
  - Identity,
  - Logistic,
  - Relu,
  - Tanh.
- Optimizadores:
  - Adam,
  - Lbfgs,
  - Sdg.

<sup>10</sup> Lo cual sólo sucede cuando las clases son linealmente separables.



**Tabla 3.** Precisión perceptrón multicapa.

Algoritmo	Métrica	Función activación	Optimizador	Precisión
Perceptrón multicapa	90 – 10	Identity	Sgd	83.78 %
	80 – 20	Logistic	Sgd	84.12 %
	90 – 10	Relu	Sgd	84.22 %
	70 – 30	Tanh	Sgd	83.78 %

Se realizó una combinación completa entre todas las funciones de activación y los optimizadores conservando el número de épocas y la tasa de aprendizaje para buscar los mejores resultados.

### 3.5. Resultados

En esta sección se muestran los mejores resultados obtenidos, de acuerdo con la métrica de validación aplicada en cada uno de los algoritmos de clasificación. En la Tabla 1 se muestran los resultados obtenidos con el algoritmo de similitud coseno, el mejor resultado se obtuvo con la métrica 70 – 30 con una precisión del 75.42 % y el promedio entre los resultados es de 73.15 %.

En la Tabla 2 se muestran los resultados aplicando el algoritmo de k-vecinos más cercanos considerando valores de 3, 5, 7, 9 y 11 para el valor de «k»; el mejor resultado se obtuvo con la métrica 70 – 30, con k = 11 con una precisión del 82.29 % y el promedio entre los resultados es de 80.5 %.

En la Tabla 3 se visualizan los mejores resultados obtenidos con el perceptrón multicapa, el mejor resultado se obtuvo con la combinación entre la métrica de validación 90 – 10, la función de activación «Relu» y el optimizador «Sgd» con una precisión del 84.22 % y el promedio entre los resultados mostrados en la Tabla 3 es de 83.97 %.

## 4. Conclusiones y trabajo a futuro

Se realizaron con éxito las pruebas sobre el corpus lingüístico digital en español con 3 algoritmos de aprendizaje automático con la finalidad de detectar el posible discurso depresivo. El promedio obtenido entre los mejores resultados fue de 73.15 % para similitud coseno, 80.5 % para k-vecinos más cercanos y del 83.97 % para el perceptrón multicapa.

El perceptrón multicapa fue el que dio los mejores resultados en las clasificaciones, pero se debe destacar que también fue el algoritmo que más tiempo y recursos computacionales requirió. Para el trabajo a futuro se plantea continuar con las descargas de los mensajes en Twitter para poder hacer más grande el corpus lingüístico digital y que los mensajes no sólo sean en español sino también incluir mensajes en el idioma inglés.

Respecto a las métricas de validación, buscaremos aplicar otras métricas tal como k-fold cross validation así como también nos gustaría poder aplicar otros algoritmos de aprendizaje automático y algoritmos de aprendizaje profundo. La depresión es un

perseguidor silencioso, por lo que consideramos de suma importancia enlazar este tipo de investigaciones con estudiantes de psicología, para ahondar en el estudio de trastornos mentales con apoyo de la tecnología y con ello estar atentos para evitar cualquier tipo de desgracia.

## Referencias

1. Mendivelso, J. D., Baron, M. J. : Análisis social aplicando técnicas de lenguaje natural a información extraída de twitter. *Scientia et Technica*, vol. 24, no. 3, pp. 496–503 (2019) doi: 10.22517/23447214.21731
2. Wang, X., Zhang, C., Ji, Y., Sun, L., Wu, L., Bao, Z.: A depression detection model based on sentiment analysis in micro-blog social network, trends and applications in knowledge discovery and data mining. *Lecture Notes in Computer Science*, vol. 7867, pp. 201–213 (2013) doi: 10.1007/978-3-642-40319-4\_18
3. Babu, N. V., Kanaga, E. G.: Sentiment analysis in social media data for depression detection using artificial intelligence: a review. *SN Computer Science*, vol. 3, no. 1 (2021) doi: 10.1007/s42979-021-00958-1
4. Alsagri, H. S., Ykhlef, M.: Machine learning-based approach for depression detection in twitter using content and activity features. *IEICE Transactions on Information and Systems*, vol. E103.D, no. 8, pp. 1825–1832 (2020) doi: 10.1587/transinf.2020edp7023
5. Cremades, S. Z., Gómez, J. M., Colorado, B. N.: Diseño, compilación y anotación de un corpus para la detección de mensajes suicidas en redes sociales. *Procesamiento del Lenguaje Natural*, no. 59, pp. 65–72 (2017) <http://hdl.handle.net/10045/69092>
6. Ameer, I., Arif, M., Sidorov, G., Gómez-Adorno, H., Gelbukh, A.: Mental illness classification on social media texts using deep learning and transfer learning (2022) doi: 10.48550/ARXIV.2207.01012
7. Bird, S., Klein, E., Loper, E.: *Natural language processing with python: Analyzing text with the natural language toolkit*. O'Reilly Media, 1st Edition (2009)
8. Sidorov, G.: *Construcción no lineal de n-gramas en la lingüística computacional: N-gramas sintácticos, filtrados y generalizados*. Sociedad mexicana de inteligencia artificial (2013)
9. Qi, P., Zhang, Y., Zhang, Y., Bolton, J., Manning, C.: Stanza: A python natural language processing toolkit for many human languages. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 101–108 (2020)
10. Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., Vanderplas, J., Joly, A., Holt, B., Varoquaux, G.: API design for machine learning software: experiences from the scikit-learn project. In: *European Conference on Machine Learning and Principles and Practices of Knowledge Discovery in Databases* (2013) doi: 10.48550/ARXIV.1309.0238