ISSN: 1692-7257 - Volumen 1 – Número 43 - 2024

RCTA
Revista Colombiana de Tecnologías de Avanzada
UNIPAMPLONA

Digital Object Identifier: 10.24054/rcta.v1i43.2506

Impacto del preprocesamiento en la clasificación automática de textos usando aprendizaje supervisado y reuters 21578

Impact of preprocessing on automatic text classification using supervised learning and reuters 21578

Ing. José Manuel Arengas Acosta[1], PhD. Rafael Guzmán Cabrera[1]
PhD. Misael López Ramírez[1]

[1] ***Universidad de Guanajuato,*** *Campus Irapuato-Salamanca, Departamento de Estudios Multidisciplinario, Yuriria, Guanajuato, Mexico.*

Correspondencia: *jm.arengasacosta@ugto.mx*

Recibido: *15 octubre 2023.* ***Aceptado:*** *17 diciembre 2023.* ***Publicado:*** *31 marzo 2024.*

Cómo citar: *J. M. Arengas Acosta, M. Lopez Ramirez, y R. Guzman Cabrera, «Impacto del preprocesamiento en la clasificación automática de textos usando aprendizaje supervisado y reuters 21578», RCTA, vol. 1, n.º 43, pp. 110–118, mar. 2024. Recuperado de https://ojs.unipamplona.edu.co/index.php/rcta/article/view/2506*

Resumen: Ante la creciente generación de datos digitales, surgen retos en su gestión y categorización. Este estudio enfatiza en la clasificación automática de textos, poniendo especial énfasis en el impacto del preprocesamiento. Al emplear el conjunto de datos Reuters 21578 y aplicar algoritmos de aprendizaje supervisado como Random Forest, k-Vecinos Más Cercanos y Naïve Bayes, se analizó cómo técnicas como la tokenización y eliminación de palabras vacías influencian la precisión clasificatoria. Los hallazgos resaltan el valor agregado del preprocesamiento, destacando a "Random Forest" como el algoritmo óptimo, alcanzando una precisión del 92.2%. Este trabajo ilustra la potencialidad de combinar técnicas de preprocesamiento y algoritmos para mejorar la categorización de textos en la era digital.

Palabras clave: Clasificación automática de texto, Preprocesamiento, Reuters 21578, aprendizaje automático.

Abstract: Faced with the increasing generation of digital data, challenges emerge in its management and categorization. This study emphasizes automatic text classification, placing special emphasis on the impact of preprocessing. By using the Reuters 21578 dataset and applying supervised learning algorithms such as Random Forest, k-Nearest Neighbors, and Naïve Bayes, we examined how techniques like tokenization and the removal of stop words influence classification accuracy. The findings underscore the added value of preprocessing, singling out "Random Forest" as the optimal algorithm, achieving a precision of 92.2%. This research illustrates the potential of combining preprocessing techniques and machine learning algorithms to enhance text categorization in the digital age.

Keywords: Automatic text classification, Preprocessing, Reuters 21578, machine learning.

1. INTRODUCCIÓN

Cada segundo, cada individuo conectado a internet genera una ingente cantidad de información en formatos digitales como lo son: imagen, audio, video y texto. Una pregunta surge inevitablemente: ¿qué hacer con todo ese volumen de información? Ante esta realidad, empresas de diversos sectores se ven impelidas a desarrollar herramientas para analizar y gestionar dicha información.

En el ámbito de la comunicación, por ejemplo, agencias de noticias necesitan categorizar sus contenidos adecuadamente, asignándolos a secciones como deportes, economía, política o entretenimiento [1]. A su vez, los lectores desean filtrar o hallar noticias centradas en temas o características específicas, con el objetivo de agilizar el proceso y acceder de manera rápida a la información de su interés. No obstante, etiquetar estos contenidos manualmente requiere de considerable esfuerzo humano y tiempo. Aquí es donde la tecnología contemporánea cobra relevancia, impulsando el desarrollo de herramientas y programas que, mediante la inteligencia artificial, automatizan este proceso [2], [3], [4], [5], [6].

De esta manera surgen los sistemas de clasificación automática de textos, creados para emular la tarea humana de categorización.
Para desarrollar un programa que clasifique documentos textuales de manera automática, una forma es disponer de un conjunto de datos ya etiquetados y hacer uso de algoritmos de aprendizaje automático supervisado [7]. Durante el proceso de entrenamiento, estos algoritmos aprenden las características distintivas asociadas a cada categoría basándose en los datos proporcionados. Así, una vez entrenado el modelo, al introducir un documento no examinado anteriormente, el sistema tiene la capacidad de clasificarlo con un grado significativo de precisión. La implementación de este proceso de entrenamiento abarca distintas etapas, que incluyen la selección de datos, el preprocesamiento, la elección de características relevantes, el entrenamiento en sí y, finalmente, la evaluación de los resultados obtenidos [8].

Este estudio lleva a cabo un análisis comparativo con el objetivo de observar el impacto de diversas técnicas de preprocesamiento en la clasificación automática de textos. Dichas técnicas incluyen tokenización, conversión a minúsculas, eliminación de palabras de poca frecuencia, así como la eliminación de palabras vacías, números, signos de puntuación y caracteres especiales. Se pretende entender cómo estas técnicas afectan las métricas de precisión, el puntaje F. Para lograr este propósito, se emplea el conjunto de datos Reuters 21578 y se implementan tres algoritmos de aprendizaje supervisado: Random Forest, k-vecinos más cercanos (k-NN) y Naïve Bayes.

2. ESTADO DEL ARTE

El aprendizaje supervisado se ha consolidado como una de las principales técnicas en la minería de texto y la clasificación automática. La capacidad de procesar y categorizar grandes volúmenes de datos en tiempo real, y de manera eficiente, es esencial en la era actual de la información digital.

Zdrojewska en [9] se adentró en el mundo de los algoritmos de refuerzo, destacando la superioridad de los algoritmos de boosting en comparación con otros métodos tradicionales. Su investigación con el conjunto de datos de Reuters demuestra que la evolución y mejora de las técnicas de clasificación es una necesidad constante. Con la transformación de documentos en vectores mediante la ponderación TF-IDF y la utilización de bibliotecas especializadas como Scikit-learn, es posible alcanzar resultados que superan a estudios previos.

Por otro lado, [10] enfrentaron el desafío de categorizar textos en el idioma Lao, un ámbito lingüístico con recursos limitados. A través de la optimización del algoritmo KNN y la aplicación de técnicas de normalización, lograron una precisión considerable del 69.2%. Esta investigación subraya la importancia de adaptar y optimizar técnicas de aprendizaje supervisado para diferentes idiomas y contextos culturales.

En [11], al analizar extensos volúmenes de texto no estructurado, resaltó la invaluable información que pueden proporcionar los datos textuales, especialmente en la toma de decisiones empresariales y en la prevención de correos no deseados. La necesidad de automatización es palpable, y es aquí donde el Procesamiento del Lenguaje Natural (PLN) toma protagonismo.
En la misma línea, en la investigación [12] subraya la trascendencia del aprendizaje automático supervisado en la categorización de textos. La evolución desde la categorización manual hasta el uso de algoritmos avanzados como Naïve Bayes y SVM muestra el rápido avance y adaptabilidad del aprendizaje supervisado.

En conclusión, estas investigaciones demuestran la creciente importancia del aprendizaje supervisado en la clasificación automática de texto. La capacidad de estas técnicas para adaptarse, evolucionar y ofrecer resultados precisos en diferentes contextos e idiomas refuerza su relevancia en el campo de la minería de texto y el procesamiento del lenguaje natural.

3. ETAPAS EN LA CLASIFICACIÓN AUTOMATICA DE TEXTOS

La clasificación automática de textos: Este proceso implica analizar y agrupar documentos de tipo texto de acuerdo con sus características y contenidos. Está compuesta por distintas etapas, descritas a continuación:

3.1. Conjunto de Datos

Refiere al grupo inicial de documentos que se desea clasificar. Cuando se utiliza el aprendizaje supervisado, es esencial que estos documentos estén debidamente etiquetados. Esta etiquetación facilita la tarea del algoritmo, permitiéndole aprender y distinguir las diferentes categorías de clasificación.

En el contexto de este estudio, se emplea el conjunto de datos "Reuters 21578" [13]. Este es una colección estandarizada de textos, ampliamente usado en pruebas de clasificación. Está integrado por archivos que contienen notas periodísticas y artículos provenientes de la agencia de noticias Reuters.

3.2. Preprocesamiento

El preprocesamiento de texto es una etapa esencial en la clasificación automática de textos. Su propósito es preparar y estructurar la información, simplificando el texto para facilitar tareas posteriores, como el entrenamiento de algoritmos de aprendizaje automático [14]. Este proceso hace que los patrones y características relevantes de cada categoría sean más prominentes, resultando en un texto más coherente, interpretable y manejable para los algoritmos. Durante esta etapa, se establece un "Baseline" o punto de referencia, que permite contrastar los resultados obtenidos en diferentes escenarios estudiados [8]

Las técnicas de preprocesamiento utilizadas en este estudio incluyen:

3.2.1. Selección del Baseline

Esta técnica consiste en preparar los datos con los que se llevará a cabo el estudio. Implica extraer el texto de cada archivo basándose en características específicas relacionadas con la clasificación que se pretende realizar [15].

3.2.2. Tokenización

Esta técnica implica dividir un texto en unidades menores llamadas "tokens". Aunque estos tokens son comúnmente palabras, también pueden ser frases, oraciones o símbolos. La tokenización simplifica y estructura el texto, facilitando su posterior procesamiento y análisis.

3.2.3. Conversión a minúsculas

Esta técnica se refiere al proceso de transformar todas las letras de un texto a su forma en minúscula. El objetivo principal es homogeneizar el texto para evitar duplicaciones y reducir la variabilidad, eliminando las diferencias causadas por el uso de mayúsculas y minúsculas. Por ejemplo, las palabras "Libro", "LIBRO" y "libro" se tratarían como entidades distintas sin esta conversión. Al convertir todo a minúsculas, se asegura que las palabras se reconozcan y procesen como la misma entidad, facilitando tareas posteriores de clasificación.

3.2.4. Eliminación de palabras vacías

Esta técnica implica la eliminación de las "palabras vacías" o "stop words". Estas son palabras que, a pesar de ser comunes en un idioma, carecen de un significado crítico intrínseco. Algunos ejemplos en español son "y", "de" y "la". Al descartar estos términos, se minimiza el ruido en el texto, permitiendo que los algoritmos se enfoquen en palabras de mayor relevancia semántica. Esto facilita el análisis y mejora la adquisición y comprensión de la información.

3.2.5. Eliminación de signos de puntuación, números y caracteres especiales

Durante el procesamiento de textos, se identifican elementos, como signos de puntuación, números y caracteres especiales, que, si bien son cruciales para la interpretación humana, pueden resultar superfluos o problemáticos para los análisis algorítmicos. Al eliminarlos, el texto se purifica, minimizando ambigüedades y potenciales confusiones en su interpretación automática. Esta depuración facilita que los algoritmos focalicen su atención en el núcleo

del contenido: las palabras y su significado contextual.

3.2.6. Eliminación de palabras de poca frecuencia por categoría

Esta técnica se centra en omitir términos que se presentan escasamente en una categoría dada. Dado que estas palabras tienen una frecuencia baja, su impacto en la interpretación o clasificación del contenido es limitado y, ocasionalmente, pueden añadir ruido innecesario al análisis. Al prescindir de ellas, se logra una representación de datos más depurada, optimizando así el enfoque de los algoritmos de aprendizaje automático hacia las características más significativas y predominantes de la categoría.

3.3. Algoritmos de Aprendizaje Supervisado

Estos algoritmos se enmarcan en el aprendizaje automático y funcionan mediante la interpretación de relaciones en datasets etiquetados [5], [12], [16] El entrenamiento requiere suministrar al algoritmo un conjunto donde cada muestra está vinculada a una etiqueta o clase determinada. Durante esta etapa, el algoritmo identifica y modela las características distintivas de cada categoría, estableciendo correlaciones entre las entradas y las etiquetas asociadas. Una vez que el modelo está optimizado, puede predecir etiquetas de muestras inéditas basándose en las estructuras y relaciones previamente identificadas.
En este estudio, se utilizan los siguientes algoritmos de aprendizaje supervisado: k-Vecinos más Cercanos (k-NN), Random Forest (RF) y Naïve Bayes (NB).

3.3.1. k-Vecinos Más Cercanos (k-NN):

Este algoritmo es uno de los más básicos y esenciales dentro del aprendizaje supervisado. Funciona bajo un concepto intuitivo: la similitud. Al introducir una nueva instancia para clasificación, k-NN busca en el conjunto de entrenamiento las 'k' muestras más próximas a dicha instancia. Posteriormente, asigna a la nueva instancia la etiqueta predominante entre esos vecinos. El valor de 'k' es crucial para el desempeño del algoritmo: un 'k' pequeño puede resultar en un modelo susceptible al ruido, mientras que un 'k' grande puede generar decisiones más generalizadas. Aunque k-NN es eficiente para datasets con baja dimensionalidad, su rendimiento tiende a disminuir en espacios de alta dimensionalidad [17].

3.3.2. Random Forest (RF)

Es un algoritmo de aprendizaje de ensambles. Utilizando el enfoque "bagging", genera múltiples árboles de decisión en el entrenamiento y, al recibir una entrada, ofrece una clasificación basada en la moda o una predicción promedio de estos árboles. Frente a la alta dimensionalidad de los datos de texto, Random Forest aborda el desafío seleccionando aleatoriamente subconjuntos de características para cada árbol, incentivando la diversidad y reduciendo el sobreajuste. Al momento de clasificar un texto nuevo, cada árbol en el conjunto aporta su clasificación, y la decisión final se toma a través de una votación mayoritaria entre todos [18], [19].

3.3.3. Naïve Bayes (NB)

es un algoritmo basado en el teorema de Bayes, reconocido por su robustez y eficiencia en la clasificación de textos. Opera estimando la probabilidad de que un documento se asocie a una categoría determinada, fundamentándose en la presencia o ausencia de términos específicos en dicho documento. Para determinar estas probabilidades, se respalda en datos de entrenamiento ya etiquetados. Este algoritmo es comúnmente empleado en conjunción con representaciones como "bolsa de palabras" o TF-IDF. En [20] estos modelos, cada término o palabra del documento se convierte en una característica única, facilitando al algoritmo discernir y categorizar textos basándose tanto en el contenido semántico como en la frecuencia de determinados términos.

3.4. Escenarios Experimentales

Estos se refieren a las configuraciones específicas o parámetros bajo los cuales se entrenan y evalúan los algoritmos. Incluyen aspectos tales como las técnicas de preprocesamiento aplicadas y las estrategias para particionar el dataset, como la división entre datos de entrenamiento y validación. El propósito de definir múltiples escenarios experimentales es examinar el rendimiento de cada algoritmo en diferentes condiciones y, en consecuencia, identificar la estrategia más eficaz para el problema en cuestión. Estos escenarios posibilitan una comparación sistemática de los algoritmos, facilitando la identificación de ventajas y desventajas inherentes a cada método en relación con el problema de clasificación específico.

3.5. Las métricas de evaluación

En [21] se describen como herramientas para cuantificar la eficacia y el rendimiento de modelos y algoritmos en el ámbito del aprendizaje automático. Proporcionan una perspectiva objetiva sobre la capacidad de un modelo, valorando su precisión, confiabilidad y adaptabilidad a contextos reales. Su utilización para validar garantiza la exactitud de las predicciones y de los resultados obtenidos. Las métricas empleadas en este estudio incluyen: precisión, recall y puntaje F.

3.5.1. Precisión

Es una métrica comúnmente utilizada en tareas de clasificación para evaluar la calidad de las predicciones. Mide la proporción de casos que el modelo clasificó en una categoría y que efectivamente pertenecen a ella.

Se define como:

$$\text{Precisión} = \frac{VP}{VP+FP} \quad (1)$$

Donde:

VP: Verdaderos Positivos, corresponden a los casos que el modelo clasifica en una categoría y que realmente pertenecen a ella.

FP: Falsos Positivos, Corresponden a los casos que el modelo clasifica en una categoría, pero que en realidad no pertenecen a ella.

3.5.2. Recall (Sensibilidad)

Es una métrica que mide la proporción de verdaderos positivos identificados correctamente en relación con el total de casos reales positivos. En otras palabras, indica qué porcentaje de los positivos reales fue detectado por el modelo. Se define matemáticamente como:

$$\text{Recall} = \frac{VP}{VP+FN} \quad (2)$$

Donde:

VP: Verdaderos Positivos, corresponden a los casos que el modelo clasifica en una categoría y que realmente pertenecen a ella.
FN: Falsos Negativos, corresponden a los casos de una categoría que el modelo clasifica en otra categoría.

3.5.3. Puntaje F1

También conocido F1-score es una medida que combina precisión y recall en un único número. Es especialmente útil cuando se quiere balancear estos dos aspectos en problemas donde alguno de ellos puede ser más relevante que el otro. El F1-score se calcula como el promedio armónico entre Precisión y Recall y proporciona un balance entre la precisión y el recall. Matemáticamente se define como:

$$\text{F1-score} = 2 \text{ x } \frac{\text{PrecisiónxRecall}}{\text{Precisión+Recall}} \quad (3)$$

4. METODOLOGÍA

En el marco de esta investigación, se efectuaron experimentos usando diversas técnicas de preprocesamiento para analizar su influencia en las métricas de evaluación: Precisión, Recall y F1-score. Para la clasificación, se emplearon tres algoritmos: Random Forest, Naïve Bayes y K Vecinos más Cercanos, utilizando como base el Corpus Reuters-21578. La metodología adoptada se estructura en cuatro fases esenciales, las cuales están ilustradas en la Fig. 1. y se describen en detalle a continuación.

Fig. 1. *Esquema de Metodología Implementada*
Fuente: *elaboración propia.*

4.1. Selección de la Data

En el marco de este estudio, se generó un conjunto de datos balanceado a partir de artículos pertenecientes a determinadas categorías de la colección Reuters 21578. Para lograrlo, se llevó a cabo una filtración detallada del conjunto Reuters 21578. Inicialmente, se distinguió entre documentos etiquetados y no etiquetados, resultando en 11,367 documentos etiquetados distribuidos en 120 categorías. De estas, solo 10 cuentan con más de 100 documentos, como se muestra en la Tabla 1.

A partir de estas 10, se seleccionaron de manera aleatoria cinco categorías: Coffee, Crude, Earn, Gold y Sugar. Posteriormente, de cada categoría se extrajeron 80 documentos al azar, dando lugar a un conjunto balanceado.

Tabla 1: *categorías con más de 100 documentos*

Categoría	# de documentos por categoría
earn	3735
acq	2125
crude	355
trade	333
money	259
interest	211
ship	156
sugar	135
coffee	114
gold	100

Fuente: *elaboración propia.*

La Tabla 2 presenta el conjunto conformado por 5 categorías y un total de 400 documentos.

Tabla 2: *Conjunto de datos balanceado*

Categoría	# de documentos por categoría
Coffee	80
Crude	80
Earn	80
Gold	80
Sugar	80
∑	**400**

Fuente: *elaboración propia.*

4.2. Preprocesamiento

El preprocesamiento consiste en una serie de técnicas destinadas a limpiar, preparar y transformar el texto crudo, convirtiéndolo en un formato más apto para el proceso subsiguiente de clasificación. Las técnicas de preprocesamiento empleadas en este estudio se muestran en la tabla.3.

Tabla 3: *Técnicas de preprocesamiento*

#	Técnica
I	Tokenización.
II	Conversión a minúsculas.
III	Eliminación de palabras vacías, signos de puntuación y números.
IV	Eliminación de palabras con una frecuencia igual a uno.

Fuente: *elaboración propia.*

Las técnicas detalladas en la Tabla 3 sirven de base para establecer los escenarios experimentales de este estudio. Estos escenarios se configuran a través de combinaciones de dichas técnicas, tal como se ilustra en la Fig. 2.

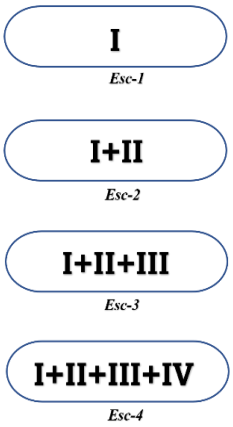

Fig. 2. *Escenarios experimentales*
Fuente: *elaboración propia.*

A continuación, se detallan los escenarios experimentales de la Fig. 2.:

Escenario 1 (Esc.1): Tokenización de la data.

Escenario 2 (Esc.2): Tokenización de la data y conversión de las palabras a minúsculas.

Escenario 3 (Esc.3): Tokenización de la data, conversión de palabras a minúsculas y eliminación de palabras vacías (stop words), signos de puntuación y números.

Escenario 4 (Esc.4): Tokenización, conversión a minúsculas, eliminación de palabras vacías (stop words), signos de puntuación, números y palabras con frecuencia de aparición única.

4.3. Procesamiento

Durante el procesamiento, se entrena cada algoritmo según los distintos escenarios, siguiendo la estructura delineada en el esquema de la Figura 3.

Fig. 3. *Esquema de Entrenamiento*
Fuente: *elaboración propia.*

Los datos son sometidos a un proceso de preprocesamiento siguiendo la configuración específica de técnicas propias de cada escenario. Una vez preprocesada la información, se procesa mediante algoritmos de aprendizaje supervisado: k-Vecinos Más Cercanos (k-NN), Random Forest (RF) y Naïve Bayes (NB). Finalmente, se muestran los resultados obtenidos en términos de métricas: Precisión, recall y puntaje F1, correspondientes a cada caso.

5. RESULTADOS

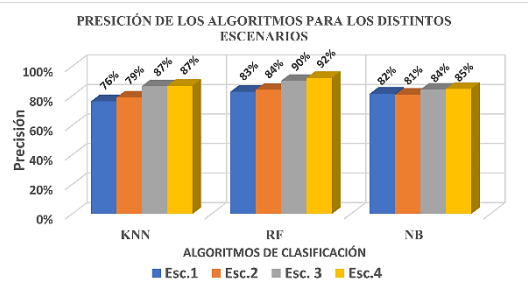

Fig. 4. *Métrica de Precisión de los algoritmos en los diferentes escenarios*
Fuente: *elaboración propia.*

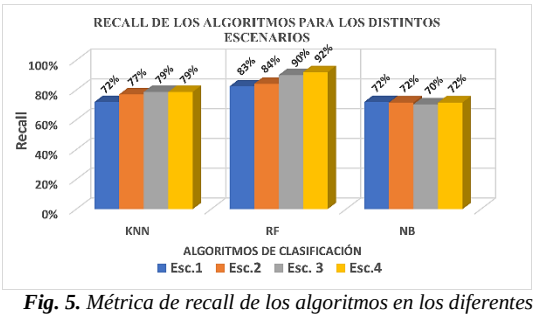

Fig. 5. *Métrica de recall de los algoritmos en los diferentes escenarios*
Fuente: *elaboración propia.*

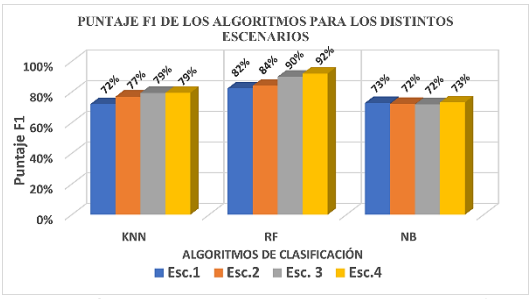

Fig. 6. *Métrica de F1-score de los algoritmos en los diferentes escenarios*
Fuente: *elaboración propia.*

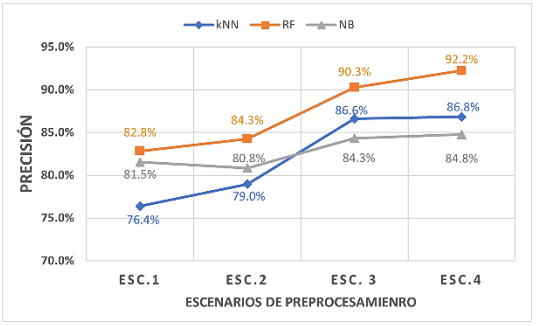

Fig. 7. *Impacto de las técnicas de preprocesamiento sobre los algoritmos en cada escenario*
Fuente: *elaboración propia.*

6. DISCUSIÓN

La Figura 4 muestra la precisión obtenida en cada escenario para los distintos algoritmos de clasificación.

Para el algoritmo k-Vecinos Más Cercanos, en el escenario 1 se alcanza una precisión del 76%, mientras que en el escenario 4 se registra un 87%. Esto indica que las técnicas de procesamiento incrementan la métrica de precisión en un 11%.

En el caso del algoritmo Random Forest, la precisión es del 83% en el escenario 1 y asciende al 92% en el escenario 4, lo que refleja una mejora del 11% tras aplicar las técnicas de procesamiento.

Por último, con el algoritmo Naïve Bayes, se observa una precisión del 82% en el escenario 1 y un 85% en el escenario 4, lo que representa una mejora del 3% debido a las técnicas de procesamiento.

Las Figuras 5 y 6 presentan los resultados de las métricas de evaluación, Recall y puntaje F1, respectivamente, para cada escenario y algoritmo de clasificación.

Se destaca que el algoritmo Naïve Bayes muestra los valores más bajos en ambas métricas, y no parece verse afectado significativamente por las técnicas de procesamiento. Por otro lado, en los algoritmos Random Forest y k-Vecinos Más Cercanos, las técnicas de procesamiento sí parecen tener un impacto positivo en las métricas. Específicamente, en Random Forest se registra un aumento del 9%, mientras que en k-Vecinos Más Cercanos, el incremento es del 7%.

En la Figura 7 se ilustra la precisión de cada algoritmo por escenario. De entrada, el algoritmo "k-Vecinos Más Cercanos" muestra la menor precisión, mientras que "Random Forest" destaca con el mejor desempeño. Con la aplicación progresiva de técnicas de preprocesamiento, se evidencia un efecto positivo en la precisión de los algoritmos. Tras completar el preprocesamiento, "Random Forest" lidera con una precisión del 92.2%. En contraste, el algoritmo "Naïve Bayes" reporta el menor rendimiento, aunque aún con un respetable 84.8% de precisión. Cabe mencionar que estos porcentajes son bastante satisfactorios para la clasificación automática de textos.

En todos los algoritmos empleados, se registra una mejora en la precisión, mostrando una tendencia ascendente en esta métrica.

7. CONCLUSIONES

Las técnicas de preprocesamiento demostraron ser fundamentales para mejorar la precisión en la clasificación automática de textos. Su aplicación sistemática llevó a mejoras significativas en las métricas de evaluación, en particular la precisión.

Entre los algoritmos evaluados, el "Random Forest" sobresalió con la precisión más alta, alcanzando un 92.2%. Aunque "Naïve Bayes" tuvo el rendimiento más bajo, aún logró un respetable 84.8%, demostrando que, en este contexto, todos los algoritmos seleccionados son adecuados para la tarea.

Como trabajo futuro sería provechoso explorar otras técnicas avanzadas de preprocesamiento o considerar la inclusión de algoritmos de clasificación no supervisados o más recientes para evaluar si pueden superar el desempeño actual.

RECONOCIMIENTO

Agradezco al Consejo de Ciencia y Tecnología (CONACYT) por el apoyo brindado con la beca 1192218 para la realización del posgrado Maestría en Administración de Tecnologías, en la Universidad de Guanajuato, México.

REFERENCIAS

[1] C. Guardiola González, “Clasificador de textos mediante técnicas de aprendizaje automático,” 2020. Accessed: Sep. 27, 2023. [Online]. Available: https://riunet.upv.es:443/handle/10251/133840

[2] Y. Li, “Automatic Classification of Chinese Long Texts Based on Deep Transfer Learning Algorithm,” in *2021 2nd International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*, IEEE, Nov. 2021, pp. 17–20. doi: 10.1109/ICAICE54393.2021.00011.

[3] D. Onita, “Active Learning Based on Transfer Learning Techniques for Text Classification,” *IEEE Access*, vol. 11, pp. 28751–28761, 2023, doi: 10.1109/ACCESS.2023.3260771.

[4] M. A. Tayal, V. Bajaj, A. Gore, P. Yadav, and V. Chouhan, “Automatic Domain Classification of Text using Machine Learning,” in *2023 International Conference on Communication, Circuits, and Systems (IC3S)*, IEEE, May 2023, pp. 1–5. doi: 10.1109/IC3S57698.2023.10169470.

[5] L. Zhang, B. Sun, F. Shu, and Y. Huang, “Comparing paper level classifications across different methods and systems: an investigation of Nature publications,” *Scientometrics*, 2022, doi: 10.1007/s11192-022-04352-3.

[6] C. Liu, Y. Sheng, Z. Wei, and Y.-Q. Yang, “Research of Text Classification Based on Improved TF-IDF Algorithm,” in *2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE)*, IEEE, Aug. 2018, pp. 218–222. doi: 10.1109/IRCE.2018.8492945.

[7] A. Rusli, A. Suryadibrata, S. B. Nusantara, and J. C. Young, "A Comparison of Traditional Machine Learning Approaches for Supervised Feedback Classification in Bahasa Indonesia," vol. VII, no. 1, 2020.

[8] D. Ji-Zhaxi, C. Zhi-Jie, C. Rang-Zhuoma, S. Maocuo, and B. Mabao, "A Corpus Preprocessing Method for Syllable-Level Tibetan Text Classification," in *2021 3rd International Conference on Natural Language Processing (ICNLP)*, IEEE, Mar. 2021, pp. 33–36. doi: 10.1109/ICNLP52887.2021.00011.

[9] A. Zdrojewska, J. Dutkiewicz, C. Jędrzejek, and M. Olejnik, "Comparison of the novel classification methods on the reuters-21578 corpus," in *Advances in Intelligent Systems and Computing*, Springer Verlag, 2019, pp. 290–299. doi: 10.1007/978-3-319-98678-4_30.

[10] Z. Chen, L. J. Zhou, X. Da Li, J. N. Zhang, and W. J. Huo, "The Lao text classification method based on KNN," in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 523–528. doi: 10.1016/j.procs.2020.02.053.

[11] M. Nasr, A. karam, M. Atef, K. Boles, K. Samir, and M. Raouf, "Natural Language Processing: Text Categorization and Classifications," *Advanced Networking and Applications*, vol. 12, no. 02, pp. 4542–4548, 2020.

[12] A. I. Kadhim, "Survey on supervised machine learning techniques for automatic text classification," *Artif Intell Rev*, vol. 52, no. 1, pp. 273–292, Jun. 2019, doi: 10.1007/s10462-018-09677-1.

[13] D. D. Lewis, "Machine Learning Repository," Documents came from Reuters newswire in 1987. Accessed: Oct. 18, 2022. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/reuters-21578+text+categorization+collection

[14] C. L. Hernández and J. E. Rodríguez, "Preprocesamiento de datos estructurados Structured Data Preprocessing," *Investigacion y desarrollo*, vol. 4, no. 2, pp. 27–48, 2013, doi: 10.14483/2322939X.4123.

[15] J. J. Paniagua Medina, E. Vargas Rodriguez, and R. Guzman Cabrera, "Machine Learning And The Reuters Collection-21578 In Document Classification," *Revista Colombiana De Tecnologias De Avanzada (RCTA)*, vol. 2, no. 40, Jul. 2023, doi: 10.24054/rcta.v2i40.2344.

[16] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information (Switzerland)*, vol. 10, no. 4. MDPI AG, 2019. doi: 10.3390/info10040150.

[17] L. A. Calvo-Valverde and J. A. Mena-Arias, "Evaluación de distintas técnicas de representación de texto y medidas de distancia de texto usando KNN para clasificación de documentos," *Revista Tecnología en Marcha*, Feb. 2020, doi: 10.18845/tm.v33i1.5022.

[18] T. Salles, M. Gonçalves, V. Rodrigues, and L. Rocha, "Improving random forests by neighborhood projection for effective text classification," *Inf Syst*, vol. 77, pp. 1–21, Sep. 2018, doi: 10.1016/j.is.2018.05.006.

[19] J. J. Espinosa Zúñiga, "Aplicación de algoritmos Random Forest y XGBoost en una base de solicitudes de tarjetas de crédito," *Ingeniería Investigación y Tecnología*, vol. 21, no. 3, pp. 1–16, Jul. 2020, doi: 10.22201/fi.25940732e.2020.21.3.022.

[20] M. Thangaraj and M. Sivakami, "Text classification techniques: A literature review," *Interdisciplinary Journal of Information, Knowledge, and Management*, vol. 13, pp. 117–135, 2018, doi: 10.28945/4066.

[21] A. Bhavani and B. Santhosh Kumar, "A Review of State Art of Text Classification Algorithms," in *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*, Institute of Electrical and Electronics Engineers Inc., Apr. 2021, pp. 1484–1490. doi: 10.1109/ICCMC51019.2021.9418262.