ULPGC
Universidad de
Las Palmas de
Gran Canaria

SIANI
Instituto Universitario de Sistemas Inteligentes
y Aplicaciones Numéricas en Ingeniería

Ciencia de Datos en Ingeniería

Trabajo de curso: Recomendador de Revistas

Autor: Santiago Santana Martínez

[31/12/2025]

Índice

Resumen

En este trabajo se desarrolla un sistema inteligente de recomendación de revistas científicas basado en técnicas de Ciencia de Datos y Procesamiento de Lenguaje Natural. El sistema es capaz de sugerir la revista más adecuada para un artículo científico a partir de su contenido textual, aprendiendo dicho conocimiento a partir de artículos publicados previamente. Se exploran dos enfoques diferenciados: una aproximación clásica basada en modelos de clasificación tradicionales y una aproximación conexionista basada en redes neuronales recurrentes. Se analizan y comparan los resultados obtenidos, justificando las decisiones de diseño adoptadas.

1. Introducción

En la actualidad existe un elevado número de revistas científicas, lo que dificulta a los investigadores la elección de la revista más adecuada para enviar un artículo. Además de factores de impacto o prestigio, resulta fundamental que la temática del trabajo sea coherente con los artículos previamente publicados en la revista, ya que esto incrementa su visibilidad e interés dentro de la comunidad científica correspondiente.

1.1. Objetivo Principal

El objetivo principal de este trabajo es desarrollar un sistema inteligente capaz de recomendar una revista científica a partir del contenido textual de un artículo, concretamente su título, resumen y palabras clave. Este sistema aprende dicho conocimiento a partir de ejemplos de artículos previamente publicados en distintas revistas.

Para alcanzar este objetivo se hace uso de técnicas de Ciencia de Datos y Procesamiento de Lenguaje Natural, explorando dos paradigmas diferentes:

- Una aproximación clásica basada en modelos de clasificación tradicionales.
- Una aproximación conexionista basada en redes neuronales.

2. Descripción del Conjunto de Datos

El conjunto de datos utilizado en este trabajo está compuesto por artículos científicos publicados en distintas revistas del ámbito de la inteligencia artificial y áreas afines. Cada instancia del conjunto contiene información textual del artículo (título, resumen y palabras clave) junto con la revista en la que fue finalmente publicado, que se utiliza como etiqueta de clasificación.

Una característica relevante del conjunto de datos es la fuerte desigualdad en el número de artículos por revista. Mientras que algunas revistas cuentan con varios miles de artículos, otras apenas disponen de unas pocas decenas, lo que introduce un problema de desbalance de clases que puede afectar negativamente al entrenamiento y evaluación de los modelos.

La Figura 1 muestra la distribución original de artículos por revista, donde se observa claramente la presencia de clases muy minoritarias.

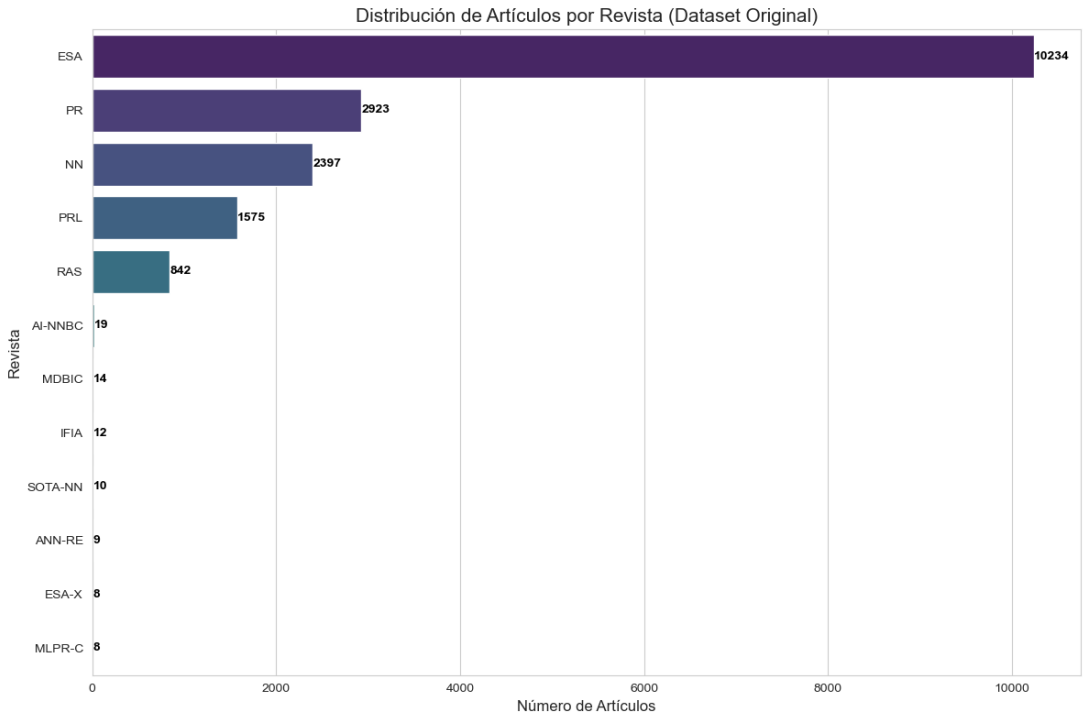

Figura 1: Distribución de artículos por revista en el conjunto de datos original.

Con el objetivo de garantizar significancia estadística y estabilidad en los resultados, se decidió eliminar aquellas revistas con un número de artículos inferior a un umbral mínimo. Tras un análisis exploratorio, se estableció un umbral de **20 artículos por revista**.

Esta decisión permite:

- Reducir el impacto del ruido introducido por clases extremadamente pequeñas.
- Evitar evaluaciones poco fiables debidas a la escasez de ejemplos.
- Prescindir de técnicas artificiales de rebalanceo, manteniendo la distribución real de los datos.

La Figura 2 muestra la distribución de artículos tras aplicar el filtrado, dando lugar a un conjunto de datos más equilibrado y representativo.

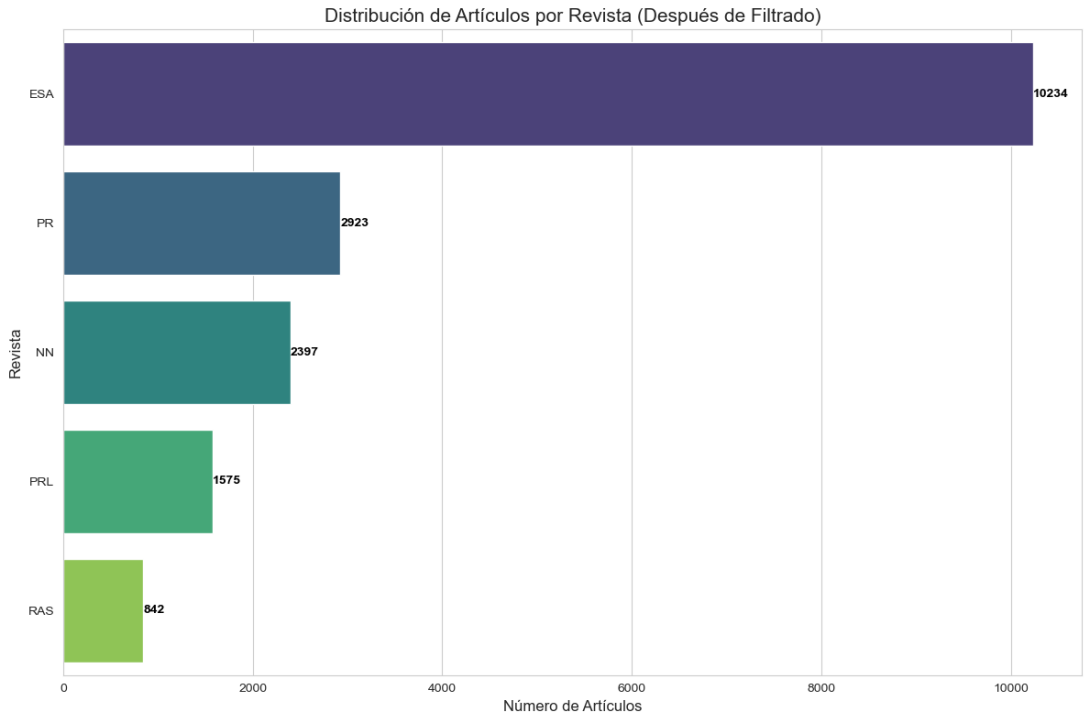

Figura 2: Distribución de artículos por revista tras aplicar el filtrado por número mínimo de artículos.

3. Preprocesamiento del Texto

Antes de proceder al entrenamiento de los modelos, el texto de los artículos fue sometido a una fase de preprocesamiento con el objetivo de reducir ruido y homogeneizar la representación textual.

Las operaciones básicas aplicadas incluyen:

- Conversión de todo el texto a minúsculas para evitar duplicidad de términos.
- Eliminación de caracteres especiales, símbolos y signos de puntuación.
- Tokenización del texto en palabras individuales.

En la aproximación clásica, gran parte de este preprocesamiento se delega al propio `TfidfVectorizer` de la librería `scikit-learn`, que incorpora mecanismos internos de normalización y tokenización.

Adicionalmente, se evaluó el impacto de aplicar **lematización** mediante la librería `NLTK`. La lematización reduce las palabras a su forma canónica, lo que teóricamente puede mejorar la generalización del modelo al agrupar variantes morfológicas de un mismo término.

Sin embargo, como se analizará en secciones posteriores, los experimentos muestran que la lematización no aporta mejoras significativas en este problema concreto, probablemente debido a que los modelos TF-IDF con n-gramas ya capturan suficiente información semántica.

4. Aproximación Clásica

En esta sección se describe la aproximación clásica empleada para el desarrollo del sistema de recomendación de revistas científicas. Este enfoque se basa en técnicas tradicionales de Procesamiento de Lenguaje Natural y modelos de clasificación supervisada, siguiendo las metodologías vistas a lo largo del curso.

4.1. Representación del Texto

Para la aproximación clásica se empleó la representación **TF-IDF**, que permite ponderar la importancia de los términos en función de su frecuencia dentro de un documento y de su capacidad discriminativa entre el conjunto de documentos.

Esta representación resulta especialmente adecuada para tareas de clasificación de texto, ya que reduce la influencia de términos muy frecuentes y destaca aquellos más relevantes para distinguir entre clases.

Con el objetivo de capturar tanto información léxica básica como expresiones compuestas frecuentes en el dominio científico, se consideraron n-gramas de tamaño uno y dos (unigramas y bigramas).

Adicionalmente, se evaluaron dos variantes de preprocesamiento:

- Una variante estándar, utilizando directamente TF-IDF con eliminación de *stopwords* en inglés.
- Una variante que incorpora un paso adicional de **lematización** mediante la librería `NLTK`, con el objetivo de reducir la variabilidad morfológica del texto.

4.2. Modelos de Clasificación

Sobre la representación TF-IDF se entrenaron y evaluaron los siguientes modelos de clasificación supervisada:

- Regresión Logística.
- Máquina de Vectores de Soporte Lineal (Linear SVM).
- Random Forest.

Estos modelos fueron seleccionados por su uso habitual en tareas de clasificación de texto y por presentar diferentes capacidades de modelado.

Cada modelo se entrenó y evaluó bajo las dos variantes de preprocesamiento descritas (con y sin lematización), permitiendo analizar de forma explícita el impacto de dicha técnica sobre el rendimiento final.

4.3. Evaluación

La evaluación de los modelos clásicos se realizó en dos fases complementarias:

- Validación cruzada estratificada, para estimar el rendimiento medio y la estabilidad de los modelos.

- Evaluación final sobre un conjunto de test independiente, reservado desde el inicio del proceso.

Como métricas de evaluación se utilizaron la precisión global, así como métricas por clase (precisión, *recall* y F1-score) y matrices de confusión, con el fin de analizar en detalle los patrones de error cometidos por los modelos.

4.4. Análisis de Resultados

La Figura 3 muestra los resultados obtenidos mediante validación cruzada para los modelos clásicos sin aplicar lematización, mientras que la Figura 4 presenta los resultados correspondientes tras incorporar dicho preprocesamiento.

```
===== Logistic Regression + TF-IDF (Cross-Validation) =====
accuracy: 0.6789 ± 0.0096
precision: 0.5808 ± 0.0070
recall: 0.6384 ± 0.0107
f1: 0.6035 ± 0.0088

===== Linear SVM + TF-IDF (Cross-Validation) =====
accuracy: 0.7124 ± 0.0118
precision: 0.6097 ± 0.0125
recall: 0.6080 ± 0.0098
f1: 0.6063 ± 0.0113

===== Random Forest + TF-IDF (Cross-Validation) =====
accuracy: 0.6857 ± 0.0057
precision: 0.6842 ± 0.0189
recall: 0.4711 ± 0.0085
f1: 0.5069 ± 0.0097
```

Figura 3: Resultados de validación cruzada para los modelos clásicos con TF-IDF sin lematización.

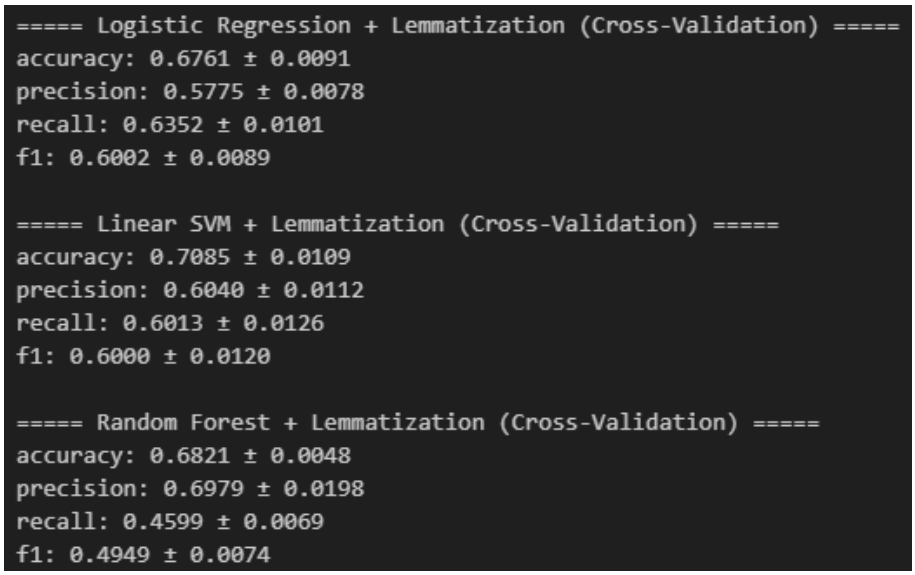

Figura 4: Resultados de validación cruzada para los modelos clásicos con TF-IDF y lematización.

Los resultados muestran que el modelo **Linear SVM con TF-IDF** obtiene el mejor rendimiento global, alcanzando los valores más altos de precisión media y mostrando una alta estabilidad entre validación cruzada y evaluación final.

La incorporación de lematización no produce mejoras significativas en ninguno de los modelos evaluados. En algunos casos, las diferencias observadas son marginales o incluso ligeramente negativas. Este comportamiento sugiere que la representación TF-IDF con n-gramas ya resulta suficientemente robusta para capturar la información discriminativa del texto, especialmente en un dominio técnico donde las variaciones morfológicas pueden tener significado propio.

Para profundizar en el análisis de errores, se estudiaron las matrices de confusión de los modelos con mejor rendimiento. La Figura 5 muestra la matriz de confusión del modelo Linear SVM con TF-IDF sin lematización.

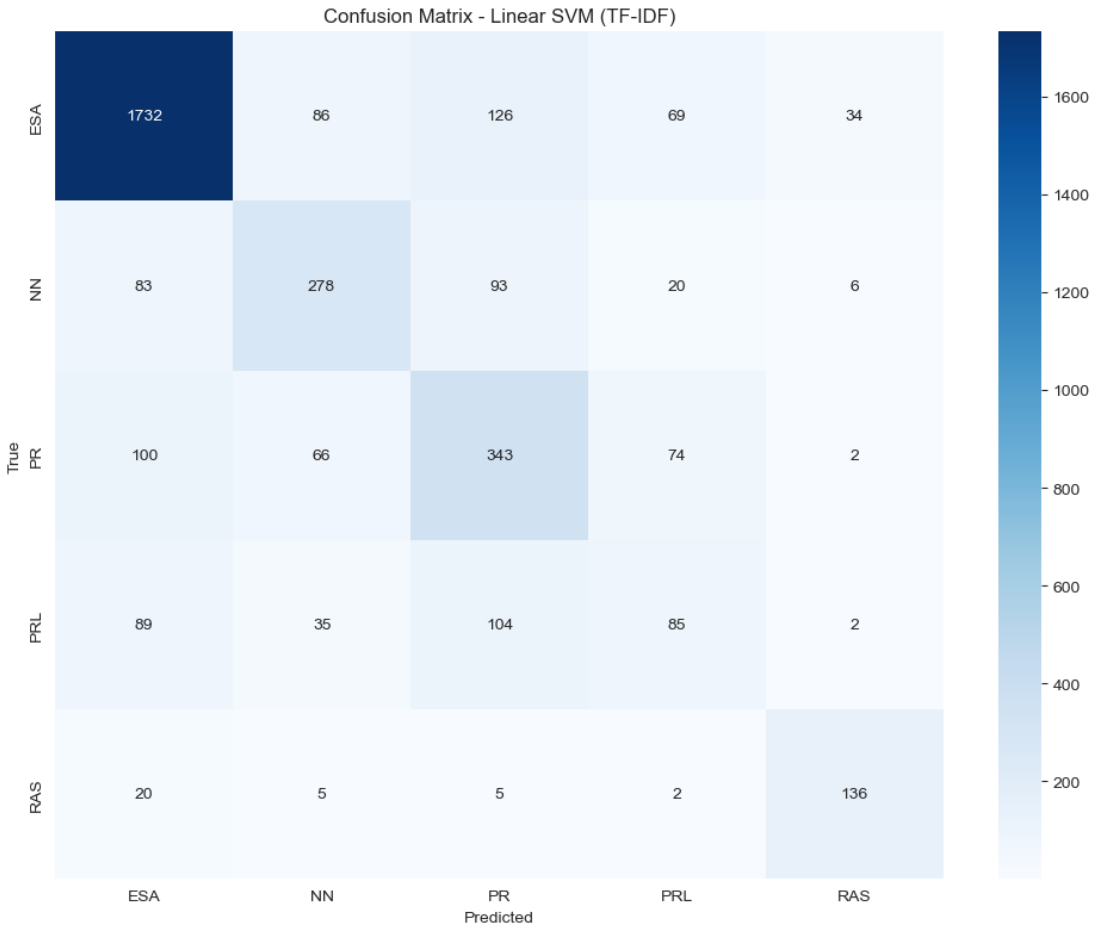

Figura 5: Matriz de confusión del modelo Linear SVM con TF-IDF sin lematización.

Por su parte, la Figura 6 presenta la matriz de confusión del mismo modelo tras aplicar lematización al texto de entrada.

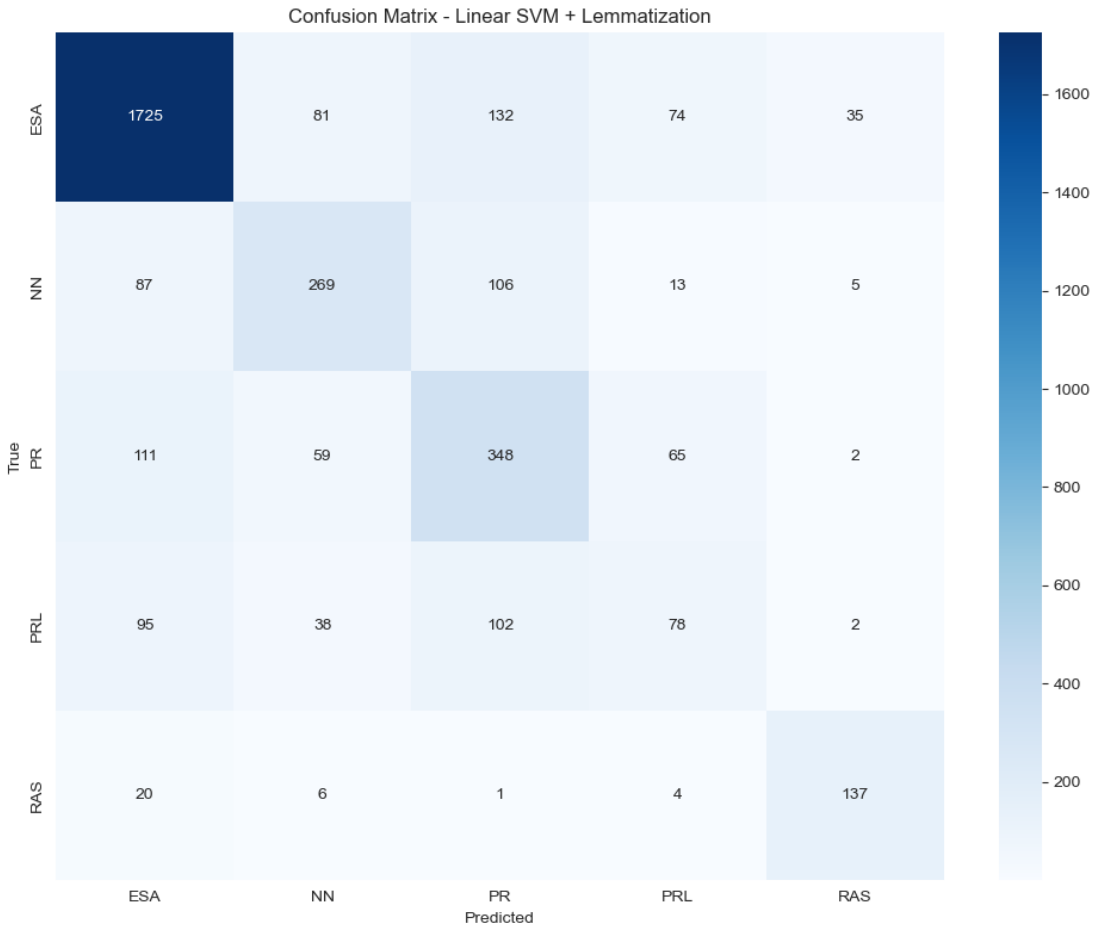

Figura 6: Matriz de confusión del modelo Linear SVM con TF-IDF y lematización.

El análisis comparativo de ambas matrices confirma que los patrones de error son muy similares en ambas variantes. En particular, los principales errores se producen entre revistas con dominios temáticos cercanos, como *Neural Networks* y *Pattern Recognition*, lo cual resulta coherente desde un punto de vista semántico.

No se observa una reducción clara de la confusión entre clases tras aplicar lematización, lo que refuerza la conclusión de que este preprocesamiento no aporta beneficios relevantes en el contexto concreto de este problema y conjunto de datos.

4.5. Discusión

A partir de los resultados obtenidos, se puede concluir que los modelos clásicos bien ajustados, en particular Linear SVM con representación TF-IDF, ofrecen un excelente compromiso entre rendimiento, estabilidad y coste computacional.

Este enfoque resulta especialmente adecuado en escenarios donde el tamaño del conjunto de datos es moderado y se busca una solución eficiente, interpretable y fácilmente reproducible. La comparación sistemática entre variantes con y sin lematización permite además justificar de forma experimental la elección de un preprocesamiento más simple, evitando complejidad innecesaria sin pérdida de rendimiento.

5. Aproximación Conexionista

En esta sección se describe la aproximación conexionista empleada para la construcción del sistema de recomendación, basada en modelos de redes neuronales profundas y representaciones de distribución del texto.

A diferencia de la aproximación clásica, este enfoque permite modelar explícitamente el orden de las palabras y capturar dependencias secuenciales dentro del texto, lo cual resulta especialmente relevante en documentos científicos con estructuras lingüísticas complejas.

5.1. Representación Distribucional del Texto

Para la representación del texto en la aproximación conexionista se empleó un modelo **Word2Vec**, entrenado desde cero utilizando exclusivamente el conjunto de datos del problema.

Previo al entrenamiento de Word2Vec, el texto fue sometido a un proceso de **tokenización simple**, consistente en:

- Conversión del texto a minúsculas.
- Eliminación de caracteres especiales y signos de puntuación.
- Separación del texto en tokens mediante espacios.

Esta tokenización sencilla permite preservar el vocabulario técnico del dominio sin aplicar normalizaciones adicionales que podrían eliminar información semántica relevante.

El modelo Word2Vec se entrenó utilizando el esquema *skip-gram*, obteniendo representaciones vectoriales densas de dimensión fija para cada palabra del vocabulario. Estas representaciones capturan relaciones semánticas y contextuales entre términos, sirviendo como entrada al modelo neuronal.

A partir del modelo Word2Vec entrenado se construyó una matriz de embeddings, donde cada fila corresponde al vector asociado a una palabra del vocabulario. Esta matriz se utilizó para inicializar la capa de embeddings del modelo BiLSTM (Bidirectional Long Short-Term Memory).

5.2. Arquitectura del Modelo

Se implementó un clasificador basado en una red neuronal recurrente **BiLSTM**. Esta arquitectura permite procesar la secuencia de palabras tanto en sentido directo como inverso, capturando dependencias contextuales en ambas direcciones.

La arquitectura del modelo consta de las siguientes capas:

- Capa de **embeddings**, inicializada con la matriz de embeddings obtenida a partir de Word2Vec.
- Capa **BiLSTM**, encargada de modelar las dependencias temporales de la secuencia.
- Capa **densa** final para la clasificación multiclase.

La salida final del modelo se obtiene concatenando los estados ocultos finales de la LSTM en ambas direcciones y proyectándolos sobre el espacio de clases mediante una capa totalmente conectada.

5.3. Entrenamiento y Regularización

El entrenamiento del modelo BiLSTM se realizó de forma supervisada, utilizando la función de pérdida de entropía cruzada categórica y el optimizador Adam. Destacar que fue entrenado haciendo uso de aceleración por GPU mediante CUDA, empleando la versión **CUDA 13.1**.

Durante el entrenamiento se monitorizaron tanto la pérdida como la precisión en los conjuntos de entrenamiento y validación. La Figura 7 muestra la evolución de estas métricas a lo largo de las distintas épocas.

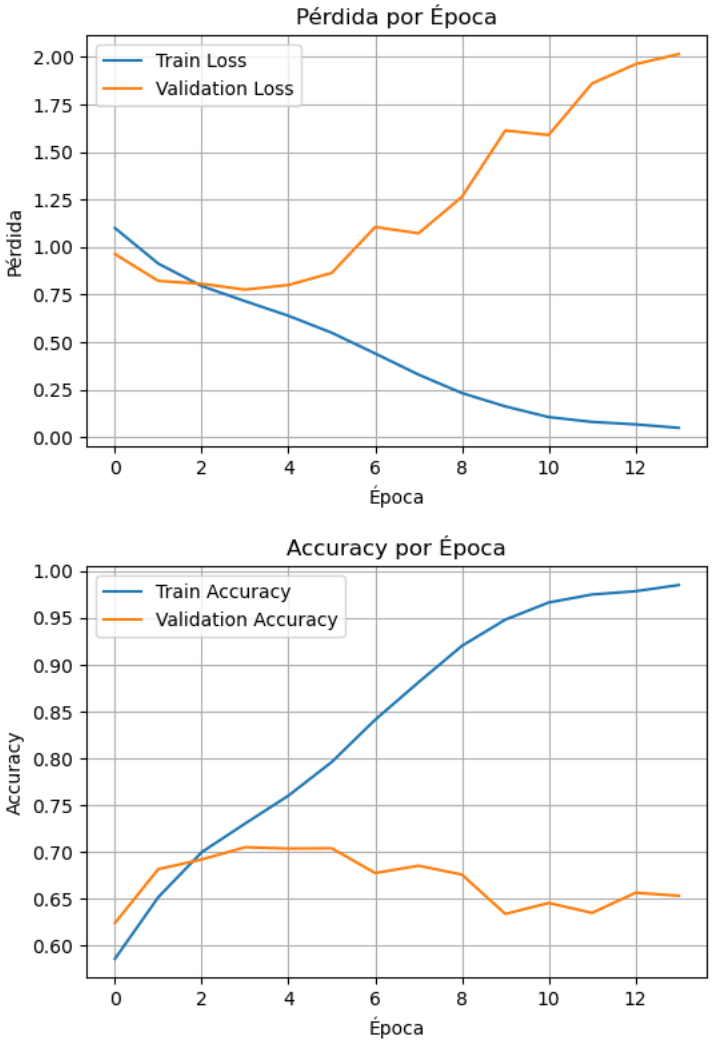

Figura 7: Evolución de la pérdida y de la precisión durante el entrenamiento del modelo BiLSTM.

Se observa un claro fenómeno de sobreajuste, caracterizado por una disminución progresiva de la pérdida de entrenamiento mientras que la pérdida de validación comienza a aumentar

tras un número reducido de épocas.

Para mitigar este comportamiento se incorporaron distintas técnicas de regularización:

- **Dropout**, aplicado a la salida de la capa recurrente.
- Penalización L2 mediante **weight decay**.
- **Early stopping**, deteniendo el entrenamiento cuando el rendimiento en validación deja de mejorar.

5.4. Resultados y Discusión

El modelo BiLSTM alcanza una precisión en el conjunto de test cercana al 70 %, situándose en valores comparables a los mejores modelos clásicos evaluados en este trabajo.

Para analizar con mayor detalle el comportamiento del modelo, se estudió su matriz de confusión, mostrada en la Figura 8.

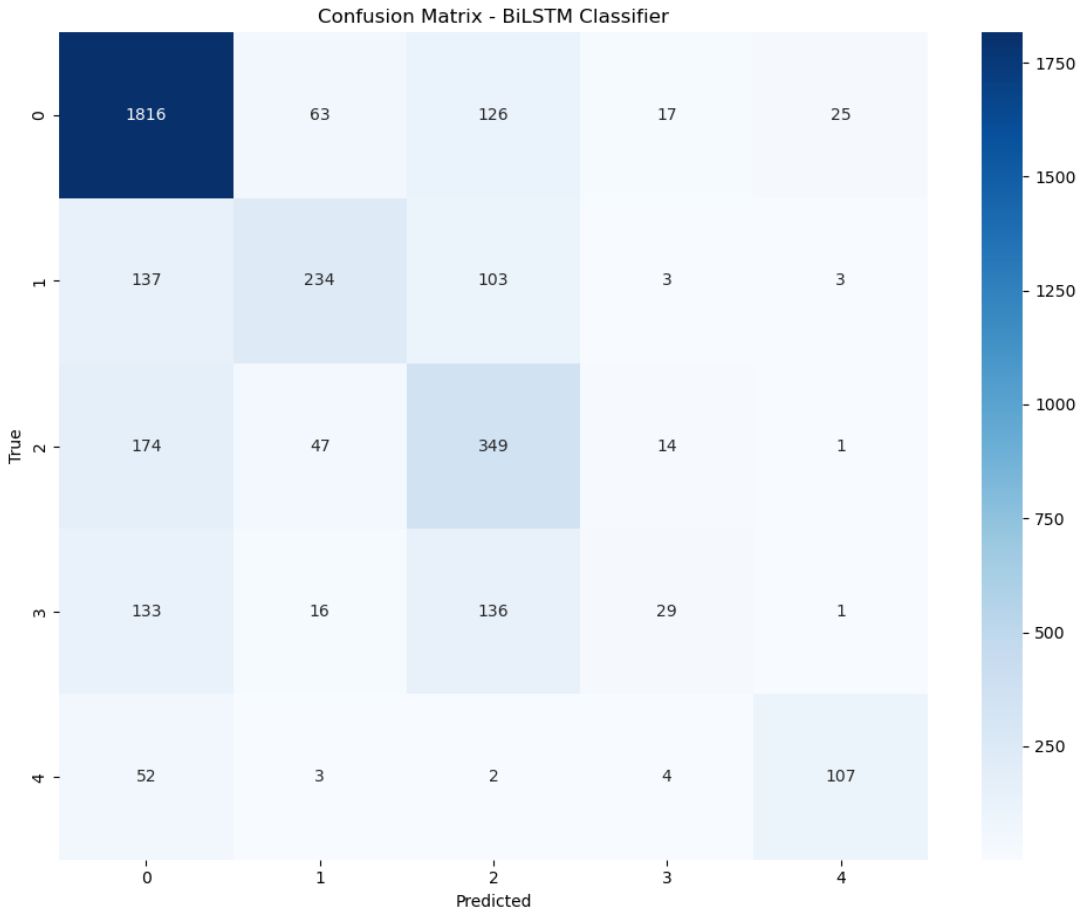

Figura 8: Matriz de confusión del modelo BiLSTM.

Al igual que en la aproximación clásica, los principales errores se producen entre revistas con temáticas cercanas, como *Neural Networks* y *Pattern Recognition*. Este comportamiento indica que el modelo captura adecuadamente la estructura semántica del problema, aunque sigue presentando dificultades para discriminar entre dominios muy próximos.

Si bien el modelo BiLSTM es capaz de modelar el orden y el contexto de las palabras, se observa que su mayor complejidad lo hace más propenso al sobreajuste en escenarios con un número limitado de datos, lo que limita su ventaja frente a los enfoques clásicos en este caso concreto.

6. Comparación Global de Modelos

La Tabla 1 recoge un resumen comparativo de los principales resultados obtenidos por los distintos modelos de clasificación evaluados en este trabajo, considerando tanto la precisión media obtenida mediante validación cruzada como el rendimiento final sobre el conjunto de test independiente.

Modelo	**Representación**	**Lematización**	$\hat{p}_{cv}$	$\hat{p}_{test}$
Regresión Logística	TF-IDF	No	$\approx 0{,}68$	$\approx 0{,}68$
Regresión Logística	TF-IDF	Sí	$\approx 0{,}68$	$\approx 0{,}68$
Linear SVM	TF-IDF	No	$\approx \mathbf{0{,}71}$	$\approx \mathbf{0{,}72}$
Linear SVM	TF-IDF	Sí	$\approx 0{,}71$	$\approx 0{,}71$
Random Forest	TF-IDF	No	$\approx 0{,}69$	$\approx 0{,}68$
Random Forest	TF-IDF	Sí	$\approx 0{,}68$	$\approx 0{,}68$
BiLSTM	Word2Vec	Implícita	$\approx 0{,}70$	$\approx 0{,}70$

Cuadro 1: Comparación de los modelos evaluados según representación del texto, uso de lematización y precisión obtenida en validación cruzada y conjunto de test.

En primer lugar, se observa que los modelos basados en representaciones clásicas de texto mediante TF-IDF ofrecen un rendimiento competitivo y estable. En particular, el modelo de Linear SVM destaca como el mejor clasificador global, alcanzando los valores más altos tanto en validación cruzada como en test.

La incorporación de lematización no produce mejoras significativas en ninguno de los modelos clásicos evaluados. En todos los casos, las diferencias observadas entre la versión con y sin lematización son marginales, lo que sugiere que la normalización léxica adicional no aporta información relevante en este problema concreto, probablemente debido a la naturaleza técnica y específica del vocabulario empleado en los artículos científicos.

Por su parte, el modelo Random Forest presenta un rendimiento inferior y una mayor discrepancia entre precisión y recall, especialmente en clases minoritarias. Este comportamiento es coherente con las dificultades de los modelos basados en árboles para manejar espacios de características muy dispersos y de alta dimensión, como los generados por TF-IDF.

Finalmente, el modelo conexionista basado en una arquitectura BiLSTM con embeddings Word2Vec alcanza un rendimiento comparable al de los mejores modelos clásicos, aunque sin superarlos de forma clara. Si bien este enfoque es capaz de capturar información secuencial y contextual del texto, los resultados obtenidos indican una mayor tendencia al sobreajuste y un coste computacional superior, lo que limita su ventaja práctica en el escenario considerado.

7. Conclusiones

En este trabajo se ha abordado el problema de la recomendación automática de revistas científicas a partir del título, resumen y palabras clave de un artículo, utilizando técnicas de Procesamiento del Lenguaje Natural y modelos de clasificación. Para ello, se han explorado dos aproximaciones diferenciadas: una aproximación clásica basada en representaciones TF-IDF y clasificadores tradicionales, y una aproximación conexionista basada en redes neuronales recurrentes.

En la primera parte del trabajo, se han aplicado técnicas de preprocesamiento estándar y se ha representado el texto mediante TF-IDF, entrenando distintos modelos de clasificación como Regresión Logística, Linear SVM y Random Forest. Los resultados obtenidos muestran que el modelo Linear SVM alcanza el mejor rendimiento global, con una precisión cercana al 72 %, superando al resto de modelos tanto en validación cruzada como en el conjunto de test. El análisis mediante matrices de confusión ha permitido identificar confusiones recurrentes entre revistas temáticamente cercanas, como Neural Networks, Pattern Recognition y Pattern Recognition Letters, lo cual resulta coherente desde un punto de vista semántico.

En la segunda parte del trabajo se ha implementado una aproximación conexionista basada en una arquitectura BiLSTM, utilizando embeddings Word2Vec entrenados específicamente sobre el corpus de artículos. Este modelo presenta una elevada capacidad de representación, alcanzando rápidamente altos valores de precisión en el conjunto de entrenamiento. No obstante, también se ha observado un claro fenómeno de sobreajuste, mitigado mediante la incorporación de técnicas de regularización como Dropout, regularización L2 (weight decay) y early stopping. A pesar de estas medidas, el rendimiento final del modelo BiLSTM resulta comparable, pero no claramente superior, al obtenido con los modelos clásicos.

La comparación entre ambas aproximaciones da a entender que, para el tamaño y la naturaleza del conjunto de datos utilizado, los métodos clásicos basados en TF-IDF y SVM continúan siendo altamente competitivos, ofreciendo un equilibrio favorable entre rendimiento, estabilidad y coste computacional. Por su parte, los modelos conexionistas aportan una mayor expresividad y capacidad de modelado secuencial, pero requieren una mayor cantidad de datos y un ajuste más cuidadoso de los hiperparámetros para traducirse en mejoras sustanciales de rendimiento.

En conjunto, los resultados obtenidos permiten concluir que la elección del modelo más adecuado depende en gran medida del contexto y de los recursos disponibles. En escenarios con conjuntos de datos de tamaño moderado y textos relativamente homogéneos, las técnicas clásicas siguen siendo una opción sólida y eficiente. No obstante, las aproximaciones basadas en redes neuronales constituyen una línea prometedora para trabajos futuros, especialmente en contextos con mayor volumen de datos o con textos más variados y complejos.