ULPGC Universidad de Las Palmas de Gran Canaria | SIANI Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería

Trabajo de Curso

Recomendador de Revistas Científicas mediante Técnicas de Ciencia de Datos

ASIGNATURA: Ciencia de Datos en Ingeniería

MÁSTER: Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (SIANI)

AUTOR: Eduardo Ortega Zerpa

Curso académico: 2025–2026

Índice

1. Introducción

El aumento de revistas científicas en los últimos años ha incrementado la dificultad para los investigadores a la hora de seleccionar la revista más adecuada para publicar un artículo académico.

En este contexto, el uso de técnicas de Ciencia de Datos y Procesamiento del Lenguaje Natural permite desarrollar sistemas inteligentes capaces de analizar el contenido de un artículo y recomendar automáticamente la revista más apropiada.

El objetivo de este trabajo es el desarrollo de un sistema de recomendación de revistas científicas basado en el contenido textual de los artículos.

2. Descripción del problema

El problema se plantea como una tarea de clasificación multiclase de documentos. Cada documento corresponde a un artículo científico y cada clase representa una revista científica concreta.

Dado un documento d, compuesto por texto libre, se desea aprender una función:

$$f(d) \rightarrow r \in \mathcal{R}$$

donde $\mathcal{R}$ es el conjunto de revistas consideradas en el estudio.

3. Conjunto de datos

El conjunto de datos utilizado en este trabajo está compuesto por artículos científicos publicados entre los años 2020 y 2024 en distintas revistas de la editorial Elsevier, obtenidos de forma automática a través de la API oficial de Elsevier (Scopus y ScienceDirect) y con información adicional procedente de OpenAlex.

El objetivo del conjunto de datos es servir como base para un problema de **clasificación multiclase**, donde cada artículo debe ser asignado a la revista científica en la que fue publicado, utilizando exclusivamente información textual.

3.1. Revistas consideradas

En esta primera versión del sistema se han considerado las siguientes revistas:

- *Applied Ergonomics*
- *Expert Systems with Applications*
- *Journal of Visual Communication and Image Representation*
- *Neural Networks*

3.2. Campos disponibles por artículo

Para cada artículo recopilado se dispone de los siguientes campos principales:

- Título del artículo

- Resumen (abstract)
- Palabras clave (keywords)
- Año de publicación
- DOI
- Revista de publicación (etiqueta)

Los campos de título, resumen y palabras clave se concatenan para formar el texto de entrada del sistema de clasificación, mientras que la revista actúa como etiqueta de clase.

3.3. Obtención automática de los datos

La recopilación de los artículos se ha realizado mediante un script desarrollado específicamente para este trabajo, denominado `ElsevierExtractorAPI.py`. Este script implementa un pipeline completo de extracción, filtrado, normalización y almacenamiento de artículos científicos.

A continuación se describen los principales bloques funcionales del código.

3.3.1. Configuración y parámetros globales

En el bloque inicial del script se definen los parámetros generales del proceso:

- Clave de acceso a la API de Elsevier.
- Rango temporal de publicación (2020–2024).
- Lista de revistas objetivo.

3.3.2. Búsqueda de artículos en Scopus

El bloque de búsqueda implementa consultas a la API de Scopus mediante expresiones del tipo:

```
SRCTITLE("Nombre de la revista") AND PUBYEAR = año
```

Este paso permite obtener listados paginados de artículos publicados en una revista concreta durante un año específico. Para cada entrada se recupera, entre otros metadatos, el DOI del artículo.

3.3.3. Filtrado y recuperación por DOI

Una vez obtenidos los resultados de Scopus, se aplica un filtrado por DOI para descartar artículos que no pertenezcan a la editorial Elsevier. Únicamente se aceptan DOIs con el prefijo `10.1016/`.

Para cada DOI aceptado, se realiza una segunda petición a la API de ScienceDirect con el fin de recuperar información detallada del artículo, incluyendo título, resumen y metadatos completos.

3.3.4. Enriquecimiento con OpenAlex

Con el objetivo de enriquecer semánticamente el conjunto de datos, se realiza una consulta adicional a la API de OpenAlex utilizando el DOI del artículo. A partir de esta fuente se extraen:

- Palabras clave explícitas
- Conceptos asociados al artículo

Estas palabras clave se incorporan al artículo como un campo adicional.

3.3.5. Estructuración y almacenamiento

Cada artículo se normaliza finalmente en una estructura JSON que contiene todos los campos relevantes. Los artículos se almacenan siguiendo una jerarquía de directorios organizada por revista y año de publicación.

4. Metodología

Para el problema, se han implementado dos enfoques complementarios. El primero es una aproximación clásica basada en técnicas de representación vectorial y, el segundo, un enfoque conexionista basado en modelos de lenguaje profundo.

Ambos enfoques comparten una fase inicial de preparación de los datos, pero difieren en la forma en que representan el texto y en los modelos empleados para realizar la clasificación.

4.1. Carga y partición del conjunto de datos

La carga del conjunto de datos se realiza de forma automática a partir de la estructura de carpetas generada durante la fase de extracción de artículos científicos. Esta funcionalidad está implementada en el fichero `data_loader.py`, donde la función `load_dataset_from_folders` recorre los directorios correspondientes a cada revista, asignando una etiqueta de clase a cada documento en función de la carpeta de origen.

Una vez cargado el conjunto completo de datos, la estrategia de partición empleada depende del tipo de modelo considerado. En el caso de los modelos clásicos basados en técnicas de aprendizaje automático tradicional, el rendimiento se evalúa mediante validación cruzada estratificada de cinco pliegues, garantizando que la proporción de clases se mantiene constante en cada uno de los subconjuntos de entrenamiento y validación.

Por otro lado, debido al elevado coste computacional asociado al entrenamiento de modelos basados en arquitecturas Transformer, el modelo BERT se entrena utilizando una única partición estratificada fija del conjunto de datos, con una proporción del 80 % para entrenamiento y 20 % para evaluación.

4.2. Preprocesamiento del texto

En ambos enfoques se utiliza como entrada un texto unificado obtenido a partir de la concatenación del título, el resumen y las palabras clave de cada artículo. Esta concatenación se realiza durante la fase de carga de datos.

En la aproximación clásica, el preprocesamiento incluye además la normalización del texto a minúsculas y la eliminación de *stopwords* en inglés, operaciones que son gestionadas internamente por el vectorizador TF-IDF de `scikit-learn`. En el enfoque basado en BERT, este preprocesamiento se delega al tokenizador del modelo, que se encarga de la segmentación en subpalabras, el truncado y el relleno de las secuencias.

4.3. Aproximación clásica: TF-IDF + Linear SVM

La primera aproximación empleada se basa en técnicas tradicionales de recuperación de información y aprendizaje automático. Cada documento se representa mediante un vector TF-IDF que captura la importancia relativa de los términos dentro del corpus.

Esta representación se implementa mediante la clase `TfidfVectorizer`. El vectorizador se integra dentro de un `Pipeline` junto con un clasificador `LinearSVC`, capaz de encapsular todo el proceso de transformación y clasificación en una única estructura.

El entrenamiento del modelo se realiza en el fichero `train_classical.py`, donde el pipeline se ajusta utilizando el conjunto de entrenamiento. Tras el entrenamiento, el modelo se evalúa sobre el conjunto de prueba, obteniéndose las predicciones necesarias para calcular las métricas y generar los gráficos de los resultados.

4.4. Aproximación conexionista: BERT

La segunda aproximación se basa en el uso de modelos de lenguaje profundo preentrenados. Concretamente, se utiliza el modelo *bert-base-uncased*, que ha sido entrenado previamente sobre grandes corpus de texto en inglés y es capaz de capturar relaciones semánticas complejas entre palabras y frases.

El proceso de entrenamiento se implementa en el fichero `train_bert.py`. En primer lugar, las etiquetas textuales de las revistas se transforman en valores numéricos mediante un `LabelEncoder`. El texto se tokeniza utilizando `BertTokenizerFast`, aplicando truncado y relleno hasta una longitud máxima de 256 tokens.

El modelo se ajusta mediante *fine-tuning*, utilizando la clase `Trainer` de la librería `Transformers`. Se emplea una función de pérdida de entropía cruzada y se configuran hiperparámetros como el número de épocas, el tamaño de lote y la tasa de aprendizaje. Durante el entrenamiento, el modelo se evalúa al final de cada época y se conserva automáticamente la versión con mejor rendimiento sobre el conjunto de validación.

5. Evaluación

Una vez entrenados ambos modelos, se realiza una evaluación sobre el conjunto de prueba. A partir de las predicciones obtenidas, se calculan matrices de confusión normalizadas, métricas de precisión por clase y valores de F1-score por revista.

La generación de las gráficas se ha centralizado en el módulo `plots.py`, que contiene funciones reutilizables para la visualización.

6. Resultados y Discusión

En esta sección se analizan los resultados obtenidos por los dos enfoques propuestos para el sistema de recomendación de revistas científicas: un modelo clásico basado en

representaciones TF-IDF combinado con un clasificador Linear SVM, y un modelo basado en arquitecturas Transformer (BERT). El análisis se apoya en matrices de confusión normalizadas, métricas por clase y una comparación cualitativa del comportamiento de ambos enfoques.

6.1. Resultados del modelo clásico (TF-IDF + Linear SVM)

Los resultados del modelo clásico se obtienen mediante validación cruzada estratificada de cinco pliegues, garantizando que la proporción de clases se mantiene constante en cada partición y proporcionando una estimación robusta del rendimiento medio del clasificador.

6.1.1. Matriz de confusión normalizada

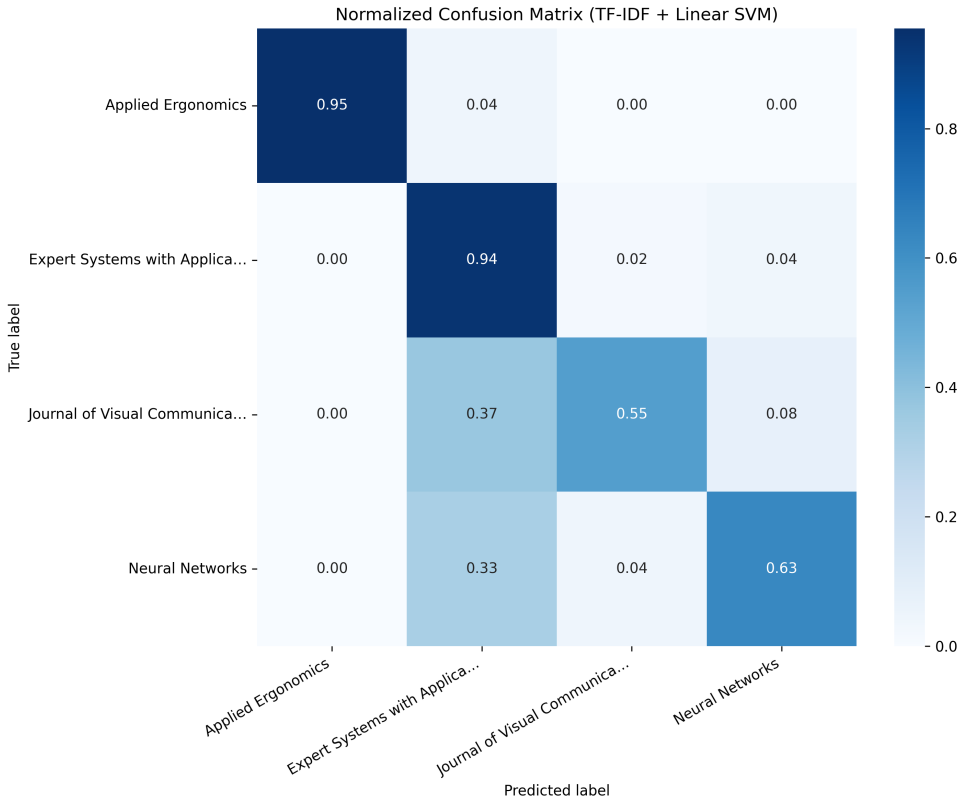

Figura 1: Matriz de confusión normalizada del modelo TF-IDF + Linear SVM

La Figura 1 muestra la matriz de confusión normalizada correspondiente al modelo TF-IDF + Linear SVM. Se observa un comportamiento claramente desigual entre las distintas revistas consideradas.

Las clases *Applied Ergonomics* y *Expert Systems with Applications* presentan tasas de acierto muy elevadas, del 95 % y 94 % respectivamente, lo que indica que el modelo identifica con gran fiabilidad los artículos pertenecientes a estas revistas. Este buen rendimiento puede atribuirse a la mayor homogeneidad temática de ambas publicaciones y a la presencia de vocabulario altamente distintivo que es capturado eficazmente mediante representaciones TF-IDF.

Por el contrario, la revista *Journal of Visual Communication and Image Representation* presenta una tasa de acierto sensiblemente inferior (55 %), mostrando una confusión

significativa con *Expert Systems with Applications*. De forma similar, la clase *Neural Networks* alcanza una accuracy del 63 %, evidenciando también confusión hacia dicha revista. Este patrón es coherente con el solapamiento conceptual existente entre áreas como visión por computador, aprendizaje automático y sistemas inteligentes, que limita la capacidad discriminativa de modelos basados únicamente en frecuencia de términos.

6.1.2. F1-score por revista

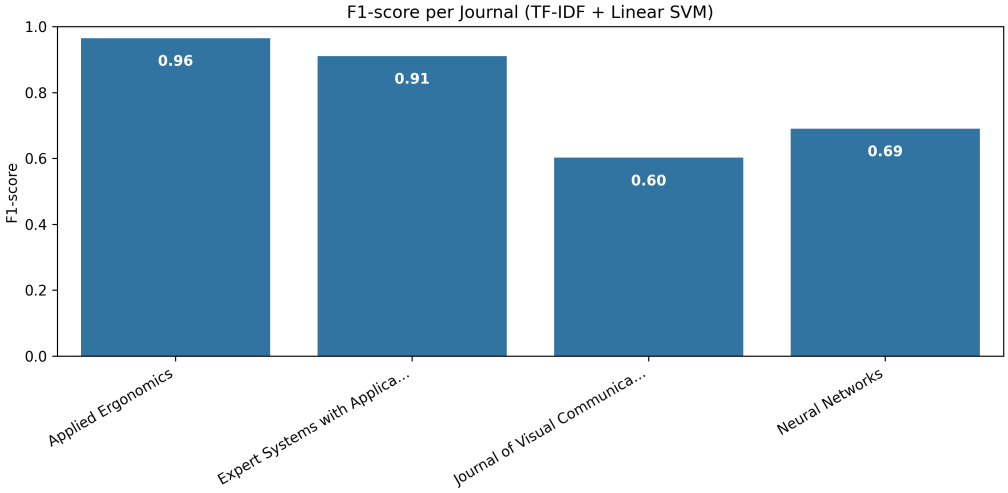

Figura 2: F1-score por revista para el modelo TF-IDF + Linear SVM

La Figura 2 muestra el F1-score por revista del modelo clásico. Las revistas *Applied Ergonomics* y *Expert Systems with Applications* alcanzan valores elevados de F1-score, 0.96 y 0.91 respectivamente, reflejando un equilibrio adecuado entre precisión y exhaustividad.

En contraste, *Journal of Visual Communication and Image Representation* obtiene un F1-score aproximado de 0.60, lo que confirma las dificultades observadas en la matriz de confusión para discriminar esta clase frente a otras revistas temáticamente próximas. La clase *Neural Networks* presenta un F1-score intermedio (0.69), indicando un rendimiento aceptable pero claramente inferior al de las clases con vocabulario más especializado.

6.1.3. Accuracy por revista

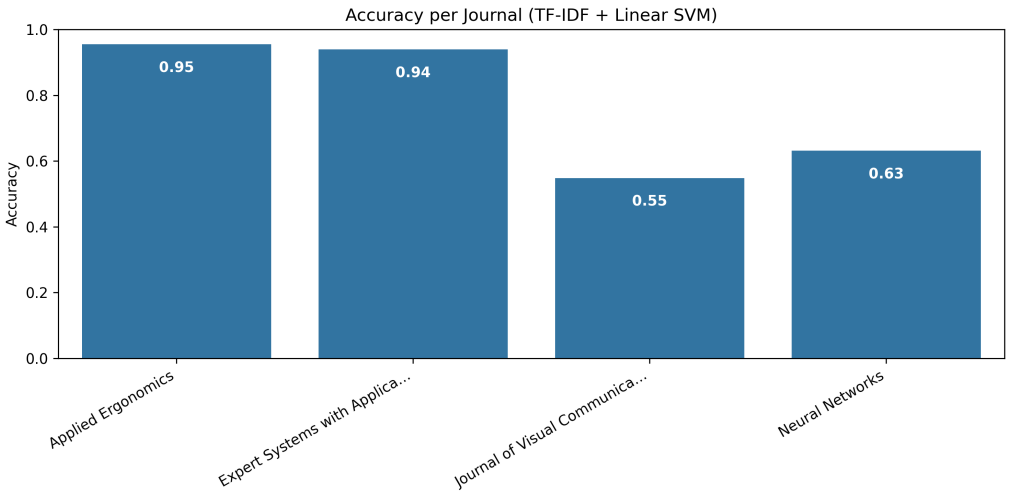

Figura 3: Accuracy por revista para el modelo TF-IDF + Linear SVM

La accuracy por clase, mostrada en la Figura 3, refuerza las conclusiones anteriores. Mientras que *Applied Ergonomics* y *Expert Systems with Applications* superan el 94 % de acierto, *Journal of Visual Communication and Image Representation* y *Neural Networks* presentan valores notablemente inferiores, del 55 % y 63 % respectivamente.

Estos resultados ponen de manifiesto que el enfoque TF-IDF + Linear SVM resulta altamente eficaz en dominios con terminología bien definida, pero muestra limitaciones evidentes cuando las clases comparten vocabulario y estructuras semánticas similares.

6.2. Resultados del modelo basado en BERT

A diferencia del modelo clásico, el modelo basado en BERT se entrena y evalúa utilizando una partición estratificada fija del conjunto de datos (80 % entrenamiento y 20 % evaluación), manteniendo la proporción de clases y reduciendo el coste computacional asociado al entrenamiento de arquitecturas Transformer.

6.2.1. Matriz de confusión normalizada

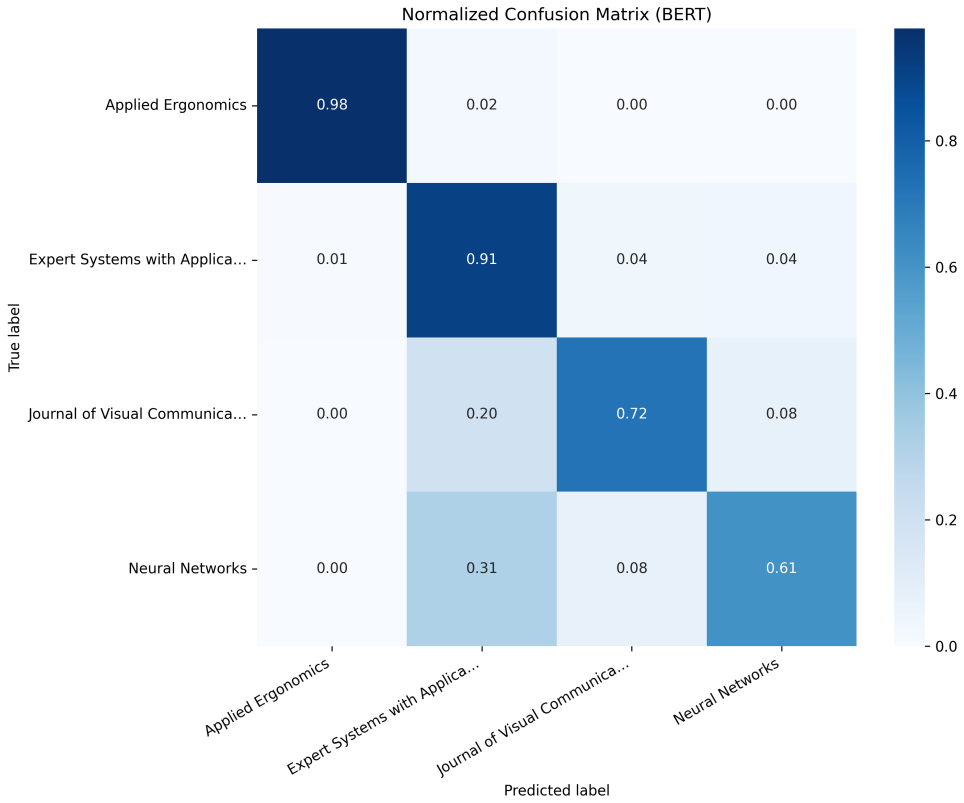

Figura 4: Matriz de confusión normalizada del modelo BERT

La matriz de confusión normalizada del modelo BERT, mostrada en la Figura 4, evidencia una mejora clara y consistente respecto al enfoque clásico. La diagonal principal presenta valores más elevados en todas las clases, destacando especialmente la mejora en *Journal of Visual Communication and Image Representation*, cuya tasa de acierto aumenta de forma significativa.

Esta mejora se explica por la capacidad de BERT para modelar el contexto semántico global del texto, capturando relaciones de largo alcance entre términos y reduciendo la ambigüedad presente en enfoques basados únicamente en frecuencia de palabras.

No obstante, persiste cierta confusión entre *Neural Networks* y *Expert Systems with Applications*, lo que sugiere que incluso modelos avanzados encuentran dificultades cuando los límites conceptuales entre revistas no están claramente definidos.

6.2.2. F1-score por revista

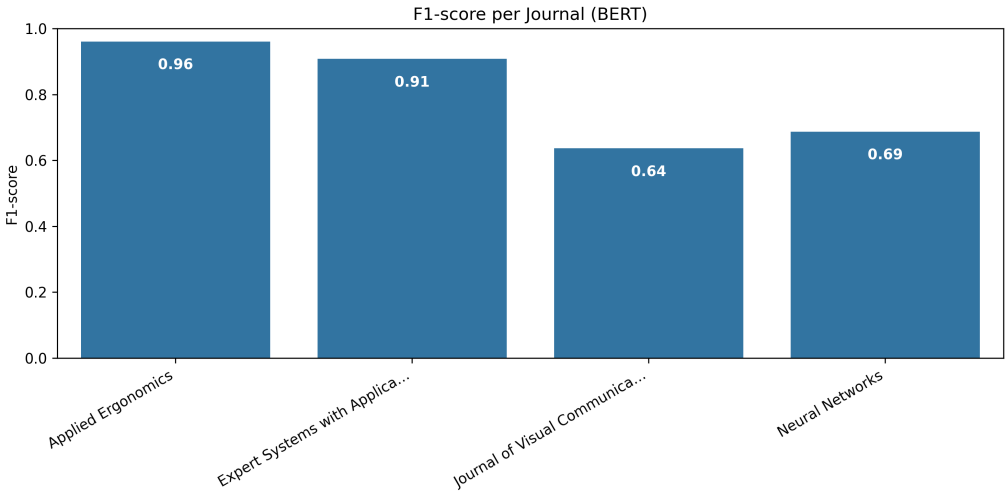

Figura 5: F1-score por revista para el modelo BERT

La Figura 5 muestra el F1-score por revista para el modelo BERT. Se observa una mejora consistente en todas las clases respecto al modelo clásico. *Applied Ergonomics* y *Expert Systems with Applications* alcanzan valores elevados, mientras que *Journal of Visual Communication and Image Representation* experimenta el incremento más notable, confirmando la idoneidad de BERT en escenarios donde el contexto semántico es determinante.

6.2.3. Accuracy por revista

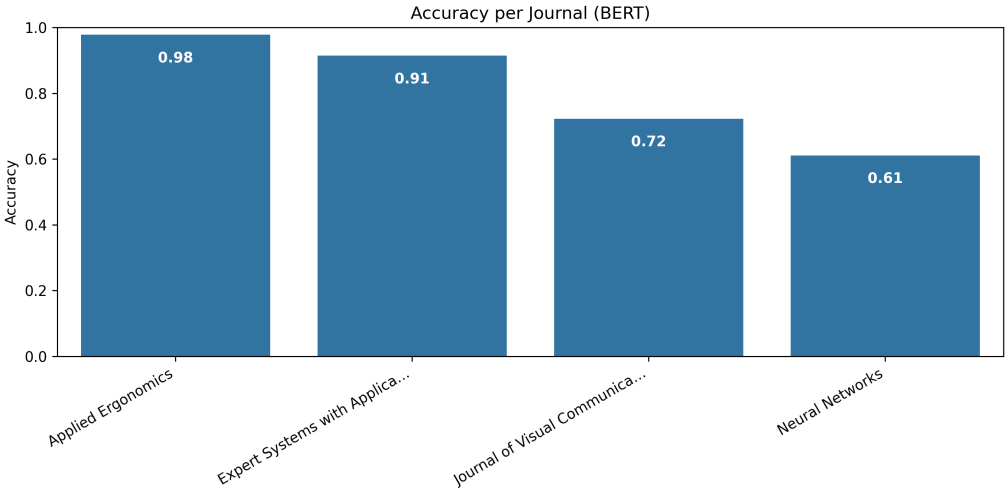

Figura 6: Accuracy por revista para el modelo BERT

La accuracy por clase del modelo BERT, representada en la Figura 6, consolida las conclusiones anteriores. Las revistas con perfiles temáticos bien definidos alcanzan valores muy elevados, mientras que las clases más complejas muestran mejoras sustanciales respecto al modelo clásico, aunque sin llegar a valores perfectos.

En conjunto, los resultados obtenidos demuestran que los modelos basados en Transformers superan de forma consistente a los enfoques clásicos en tareas de clasificación de textos científicos, especialmente en escenarios caracterizados por un alto solapamiento semántico entre clases.

7. Discusión

Comparando ambos modelos, se observa que el enfoque TF-IDF + Linear SVM resulta eficiente desde el punto de vista computacional y ofrece un rendimiento excelente en revistas con vocabulario específico y poco solapamiento temático. No obstante, su capacidad de generalización es limitada en contextos que mezclan disciplinas.

Por su parte, el modelo basado en BERT proporciona mejoras claras en revistas con contenido transversal, gracias a su capacidad para modelar contexto y relaciones semánticas complejas. Desde un punto de vista práctico, BERT sería la opción preferente para un sistema de recomendación robusto, mientras que el modelo clásico puede utilizarse como línea base o en entornos con recursos computacionales limitados.

8. Conclusiones

En este trabajo se ha desarrollado un sistema de recomendación de revistas científicas basado en el análisis del contenido textual de los artículos. Para abordar el problema se han implementado dos enfoques complementarios, uno clásico y otro basado en modelos de lenguaje profundo, permitiendo comparar su comportamiento de forma objetiva.

Los resultados obtenidos muestran que el enfoque clásico ofrece un rendimiento sólido en revistas con vocabulario específico y bien definido, mientras que el modelo basado en BERT presenta una mayor capacidad de generalización en revistas con contenidos más transversales. En conjunto, ambos modelos cumplen los objetivos planteados, siendo BERT el que alcanza un mejor rendimiento global.

Por otro lado, el análisis por revista ha permitido identificar las principales fuentes de confusión entre clases, lo que muestra la dificultad inherente del problema cuando existe solapamiento de temas entre distintas publicaciones.