




Sistemas recomendadores como herramienta en la labor docente: una revisión sistemática

Recommender systems as a tool in teaching work: a systematic review

M. Torres-Herrera ^{a,*}, G. Cuaya-Simbro ^a, C. Canales-Castillo ^a

^a Maestría en Sistemas Computacionales, Tecnológico Nacional de México / ITS del Oriente del Estado de Hidalgo, Carretera Apan-Tepeapulco, Las Peñitas, 43900 Apan, Hidalgo, México.

Resumen

Esta revisión sistemática examina el estado actual de los sistemas recomendadores aplicados en la labor educativa. La investigación se centra en las tendencias de desarrollo de sistemas recomendadores, la relevancia y áreas de oportunidad, así como la importancia de profundizar en estos temas en futuras investigaciones. Los resultados muestran un creciente interés en el campo, específicamente en países como Estados Unidos, China y España, en donde su sistema educativo es de calidad, pero no así en América Latina. Se identifica que existe un gran interés, en general, por el desarrollo y uso de herramientas novedosas, que permitan mejorar los sistemas de educación, pero la mayoría de ellas no a través del uso de las nuevas tecnologías, como los sistemas de recomendación, además muchas de las herramientas reportadas en los manuscritos revisados se centran en el apoyo a los estudiantes y solo algunas de ellas en el apoyo a la labor docente, lo cual da evidencia que existe un área de oportunidad para el desarrollo de herramientas que apoyen a los docentes a través del uso de nuevas tecnologías. La metodología que se utilizó fue, PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses), con la cual se obtuvieron 121 manuscritos relacionados con el tema y los cuales fueron analizados de manera semiautomática a través del uso de Python.

Palabras Clave: Mejora educativa, Herramientas tecnológicas, Aprendizaje automático, Meta-análisis semiautomático.

Abstract

This systematic review examines the current state of recommender systems applied in educational work. The research focuses on trends in recommender system development, relevance and areas of opportunity, as well as the importance of delving deeper into these issues in future research. The results show a growing interest in the field, specifically in countries such as the United States, China and Spain, where their educational systems are of high quality, but not in Latin America. It is identified that there is a great interest, in general, in the development and use of novel tools that allow improving education systems, but most of them are not through the use of new technologies, such as recommender systems. In addition, many of the tools reported in the manuscripts reviewed are focused on supporting students and only some of them on supporting teaching, which shows that there is an area of opportunity for the development of tools that support teachers through the use of new technologies. The methodology used was PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses), with which 121 manuscripts related to the topic were obtained and which were analyzed semi-automatically using Python.

Keywords: Educational improvement, Technological tools, Machine Learning, Semi-automatic Meta-analysis.

1. Introducción

La educación es un campo crucial en el desarrollo de las personas y la sociedad en general. Para garantizar una educación efectiva, es necesario contar con estrategias de enseñanza adecuadas que permitan preparar a los estudiantes de manera correcta. Además, es esencial que los docentes tengan acceso a herramientas que les ayuden a mejorar su

propia competencia en los temas que enseñan y que también faciliten el aprendizaje de sus estudiantes.

En la actualidad, gracias a los avances tecnológicos, se han desarrollado sistemas recomendadores en el ámbito educativo que utilizan la inteligencia artificial (IA) para crear herramientas personalizadas y efectivas para los docentes, pero en su mayoría para los estudiantes. Estos sistemas tienen como objetivo sugerir a los docentes el uso de herramientas

*Autor para la correspondencia: 22030625m@itesa.edu.mx

Correo electrónico: 22030625m@itesa.edu.mx (Mariana Torres-Herrera), gcuaya@itesa.edu.mx (German Cuaya-Simbro), 22030714m@itesa.edu.mx (Ciro Canales-Castillo).

Historial del manuscrito: recibido el 26/04/2023, última versión-revisada recibida el 08/07/2023, aceptado el 14/07/2023, en línea (postprint) desde el 01/08/2023, publicado el 05/01/2024. **DOI:** <https://doi.org/10.29057/icbi.v11i22.11032>



didácticas que ayuden a la enseñanza, en general y en los estudiantes promover la autoevaluación, la autonomía y la reflexión continua en el proceso de aprendizaje.

Hasta donde se sabe, los sistemas recomendadores no han sido aplicados para sugerir estrategias de enseñanza-aprendizaje, es decir, que proporcionen al docente la descripción de procedimientos o recursos para hacer posible el aprendizaje del estudiante, como pueden ser: que el estudiante cree un mapa conceptual, o que realice alguna analogía o que haga un ensayo, sobre algún tema específico lo cual pueda garantizar o mejorar la probabilidad de que el alumno alcance un aprendizaje efectivo.

El desarrollo de un sistema recomendador de estrategias de enseñanza-aprendizaje, permitirá hacer frente a diferentes desafíos a los cuales se enfrenta el docente en la actualidad, entre los que se pueden mencionar: la visualización, supervisión, evaluación y calificación de estudiantes en un curso, debido al número creciente de alumnos, diseñar un curso que se adapte a las diferentes habilidades y experiencias de los estudiantes dado que requiere encontrar una estrategia adecuada para la mayoría de ellos, por mencionar algunos.

Pero, encontrar estrategias pedagógicas adecuadas para el proceso de enseñanza y aprendizaje, es importante debido a que contribuyen a mejorar la motivación, participación, colaboración e interacción entre estudiantes y profesores (Burch & Melby, 2019).

Esta revisión sistemática tiene como objetivo identificar las áreas de oportunidad para realizar investigación en el desarrollo de sistemas de recomendación específicamente para docentes y centrándose en la sugerencia de estrategias de enseñanza-aprendizaje para mejorar su labor.

Los resultados muestran que existe un interés creciente en el desarrollo de sistemas de recomendación en la educación, lo que indica que es un área de interés para el desarrollo de investigaciones al respecto, aunque la mayoría de los sistemas recomendadores reportados tiene como objetivo el apoyo a los estudiantes. Por otro lado, los resultados nos indican que son pocos los trabajos que reportan la implementación de sistemas de recomendación basados en IA, lo cual da evidencia de la oportunidad de aplicación de esta tecnología para el desarrollo de herramientas de apoyo a los docentes.

2. Materiales y métodos

2.1 Sistemas recomendadores

Los sistemas recomendadores son herramientas computacionales que ofrecen sugerencias personalizadas a los usuarios basándose en su historial de uso, preferencias y comportamiento (Ricci, Rokach & Shapira, 2011). Estos sistemas se basan en una combinación de técnicas de filtrado colaborativo y filtrado basado en contenido para poder ofrecer una experiencia de usuario personalizada y mejorar la relevancia de las recomendaciones (Felfernig, Friedrich, Jannach & Zanker, 2018). De manera general, existen 2 tipos principales de sistemas de recomendación:

El filtrado colaborativo: el cual trabaja sobre la idea de que, si a un usuario le gustó un producto en particular, también podría gustarle otro producto que le haya agradado a otro usuario con preferencias similares (Su & Khoshgoftaar, 2009).

Por otro lado, recomienda cosas similares (Linden, Smith & York, 2003).

El filtrado basado en contenido: el cual sugiere productos similares a los que el usuario ha preferido en el pasado, basándose en las características de los productos (Lops, de Gemmis & Semeraro, 2011). Este enfoque tiene la ventaja de que puede hacer recomendaciones a un usuario incluso si no hay otros usuarios con gustos similares.

Por otro lado, existen varios algoritmos que son el motor de inferencia de un sistemas de recomendación, entre los que se pueden mencionar, la factorización de matrices, algoritmos clustering y aprendizaje profundo (Zhang, Yao, Sun & Tay, 2019). Cabe señalar que, es una área abierta de investigación, la prueba de diferentes algoritmos para inferencia, dado que puede experimentarse con otros algoritmos, como son, redes bayesianas, máquinas de soporte vectorial, meta clasificadores, entre otros y validar la eficiencia de los mismos como parte de un sistema de recomendación.

2.2 Ambiente de desarrollo

Para la realización de la revisión sistemática y meta-análisis se utilizó Python, el cual es un lenguaje de alto nivel de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código y es utilizado ampliamente para desarrollar aplicaciones de todo tipo. (¿Qué Es Python?, 2023) Adicionalmente, para realizar esta revisión sistemática, se utilizaron librerías para automatizar varios procesos de la revisión sistemática, las cuales están desarrolladas en Python.

Las librerías que se usaron fueron Pandas, que es un paquete de Python que proporciona estructuras de datos rápidas, flexibles y expresivas diseñadas para que al trabajar con datos "relacionales" o "etiquetados" sea fácil e intuitivo. Además, esta librería, tiene el objetivo principal de convertirse esta librería de manipulación/análisis de datos de código abierto más poderosa y flexible disponible en cualquier idioma (The Pandas Development Team, 2023).

Finalmente, se utilizó Google-search-results, este paquete de Python está destinado a raspar y analizar los resultados de búsqueda de Google, Bing, Baidu, Yandex, Yahoo!, Home Depot, eBay y más, utilizando SerpApi el cual proporciona un generador de secuencias de comandos (Vikoki, 2023) y citeproc-py el cual es un procesador CSL (Citation Style Language) para Python, actualmente admite texto sin formato, reStructuredText y HTML; citeproc-py utiliza versiones semánticas (Machiels et al., 2021).

2.3 Metodología de revisión

En este trabajo se aplicó la metodología de PRISMA (Moher et al., 2009) la cual es, esencialmente, una lista de comprobación de requisitos que debe cumplir una revisión sistemática y un meta-análisis, para presentar la información. La Figura 1, muestra un diagrama de flujo del proceso que se siguió esta investigación.

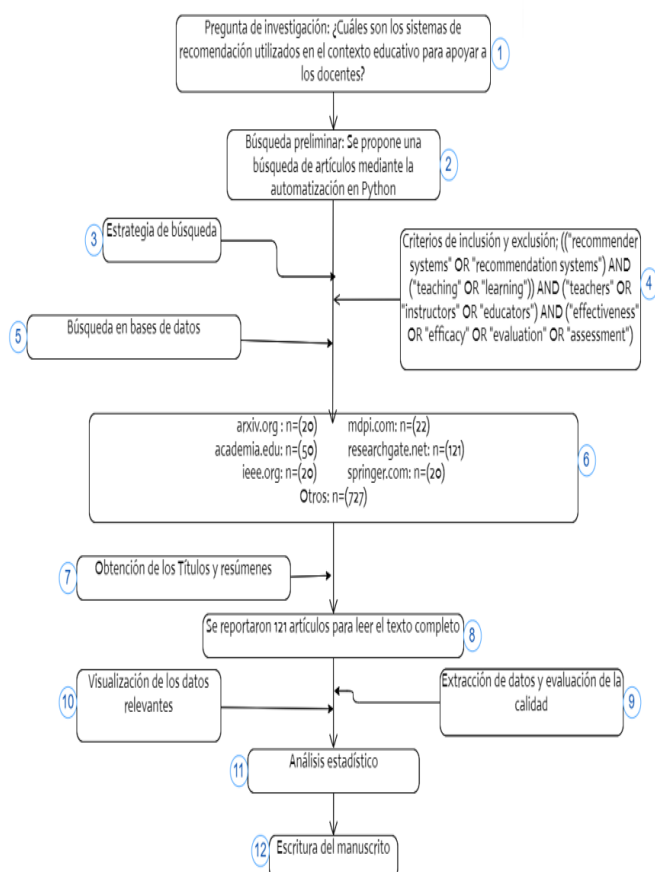


Figura 1: Diagrama de flujo de los pasos de la revisión sistemática y meta-análisis.

A continuación, se describe cada uno de los pasos de la metodología aplicada para la revisión sistemática:

1. Determinar la pregunta de investigación que dio pie al estudio del tema.

Se plantearon diferentes preguntas centradas en identificar el estado actual de los sistemas recomendadores en la educación, entre las que se pueden mencionar: ¿Cuál es el panorama actual de las tecnologías, estrategias y herramientas centradas en el apoyo a la enseñanza en un nivel medio superior o superior?, ¿Qué tipos de sistema de recomendación existen en el ámbito educativo y cuál es su aporte para mejorar la labor docente?, ¿Qué papel desempeña la inteligencia artificial en el diseño de dichas estrategias?

De lo anterior y después de un análisis sobre cómo hacer una generalización, se acordó guiar la búsqueda de literatura relevante con la siguiente pregunta de investigación: "¿Cuáles son los sistemas de recomendación utilizados en el contexto educativo para apoyar a los docentes?". Por otro lado, y dado que la mayoría de los trabajos sobre sistemas recomendadores son publicados en inglés, la pregunta anterior fue formulada en dicho idioma, como: Which are recommender system used in the education field focused in the teachers support?

2. Realizar la búsqueda automática mediante la programación de un algoritmo en Python usando la API de Google para conectar a Scholar.

3. Definición de la estrategia de búsqueda: se formuló la siguiente búsqueda avanzada específica que sirvió como guía para la ubicación de literatura relevante. ("recommender systems" OR "recommendation systems") AND ("teaching" OR "learning") AND ("teachers" OR "instructors" OR "educators") AND ("effectiveness" OR "efficacy" OR "evaluation" OR "assessment").

De lo anterior, se obtuvieron diversos artículos, con información sobre: autores, año de publicación, citas, resumen, entre otros.

4. Aplicar diferentes criterios de inclusión y exclusión iniciales para obtener los manuscritos más relevantes, dentro de los criterios de inclusión se pueden mencionar: año de publicación, que sólo fueran en un periodo de 2018 a 2023, relevancia de la investigación de acuerdo al tema y los criterios de exclusión que se aplicaron, eliminación de: manuscritos duplicados, no estuvieran en el idioma inglés.
5. Ejecutar el algoritmo de búsqueda automática en las diversas bases de datos que tiene Google, como por ejemplo: ResearchGate, IEEE, Springer, entre otros.
6. Agrupar los manuscritos de acuerdo a la base de datos de donde fueron obtenidos.
7. Después de identificar los temas latentes en los artículos mediante Latent Dirichlet Allocation (LDA), se aplicó una puntuación a los manuscritos para determinar su importancia al contestar la pregunta de investigación. Esta puntuación se basó en los pesos de los temas latentes asignados a cada artículo. Los artículos con una alta probabilidad de estar en los temas de interés fueron seleccionados y ordenados según su relevancia en relación con la pregunta de investigación.
8. Filtrar los artículos más relevantes de acuerdo a la calidad de la base de datos de donde fueron obtenidos. Este proceso se explica con más detalle en la Figura 2.
9. Realizar una revisión manual a detalle de los manuscritos para determinar su relevancia y enfoque hacia nuestra pregunta de investigación.
10. Generar gráficos para visualizar los datos de manuscritos.
11. Presentación de los resultados: se presentaron gráficamente los principales hallazgos de la revisión que se considera más relevantes.
12. Interpretación y discusión de los resultados: se realizó un análisis de los resultados de la revisión en relación con la literatura existente, se identificaron las limitaciones de los estudios incluidos y se redactó el reporte final.

Como se observa en la Figura 1, se obtuvieron un total de 980 trabajos arrojados de la búsqueda guiada con la estrategia descrita, un resumen de dichos trabajos es presentado en la Tabla 1.

Tabla 1: Resumen de los trabajos colectados de la búsqueda inicial.

Título	Año	Autores	Editorial
A Review of Educational Recommender Systems fo...	2021	M Dhahri, MK Khribi	ed.gov
Applying recommender systems and adaptive hype...	2013	B Vesin, A Klajnja-Miliević, M Ivanović	cai.sk
Personalized Student Assessment based on Learn...	2018	AL Gonçalves, LM Carlos, JB da Silva	ipp.pt
Using explanations for recommender systems in ...	2019	S Karga, M Satratzemi	uom.gr
Recommender systems for learning	2012	N Manouselis, H Drachsler, K Verbert, E Duval	core.ac.uk
...
Classifications of the Summative Assessment fo...	2021	MD Laddha, VT Lokare, AW Kiwelekar	arxiv.org
Intelligent System for Customizing Evaluation ...	2022	Mondez-Gurrola, RB Silva-Lopez	uacj.mx
Hybrid content and collaborative ...	2022	Tolety, V. B. P., & Prasad, E. V.	ieee.com
Hybrid Filtering Recommendation System in an E...	2022	M Baidada, K Mansouri, F Poirier	igi-global.com
CADA: a teacher-facing learning analytics dash...	2022	R Kaliisa, JA Dolonen	springer.com

La información colectada de cada manuscrito incluido es: el título de los manuscrito, año de publicación del documento, nombre de los autores, URL de localización del manuscrito, la editorial a la que pertenece el documento, el número de citas, por último, se incluye un fragmento breve del resumen, con el objetivo de examinar rápidamente el aporte del artículo.

3. Resultados y discusión

3.1 Filtración final de manuscritos

La Figura 2, muestra el diagrama de flujo PRISMA que se utiliza en los estudios SR/MA (Systematic Review / Meta-Analysis, por sus siglas en inglés), donde se detalla el resultado del filtrado realizado.

A continuación, se describe la manera en que se realizó el proceso de exclusión:

1. Presentación de las bases de datos: Se colocaron las bases de datos con la cantidad inicial de manuscritos encontrados en cada una de ellas
2. Obtención el total de manuscritos: Se realiza una suma de todo los trabajos encontrados en la búsqueda, obteniendo un total de 980 trabajos.
3. Primer exclusión: A partir de esto, se llevó a cabo la exclusión de 727 artículos, clasificados como "otros", ya que se encuentran en bases de datos que no cuentan con un gran número de trabajos relacionados con el tema en revisión.
4. Proyección de los títulos y resúmenes: Se obtienen los manuscritos filtrados, resultando sólo 253
5. Segunda exclusión: Posteriormente, se excluyeron trabajos en donde las bases de datos en las que aparecían no eran fuentes de información con una indexación de alto impacto, o no cumplían con los criterios de inclusión establecidos o no estaban

directamente relacionados con el tema de investigación. Y finalmente, se aplicó un criterio de exclusión adicional donde se elimina las bases de datos que tuvieran un total de publicaciones menores al 3% respecto del tema.

6. Presentación de los textos completos: Se presentan los artículos resultantes de la exclusión anterior, con los que se obtienen 245 trabajos.
7. Tercera exclusión: Se genera un filtrado, eliminando los manuscritos duplicados en las diferentes bases de datos.
8. Síntesis cualitativa: Se obtuvieron finalmente un total de 121 manuscritos, de los cuales se muestran en un resumen en la Tabla 2.

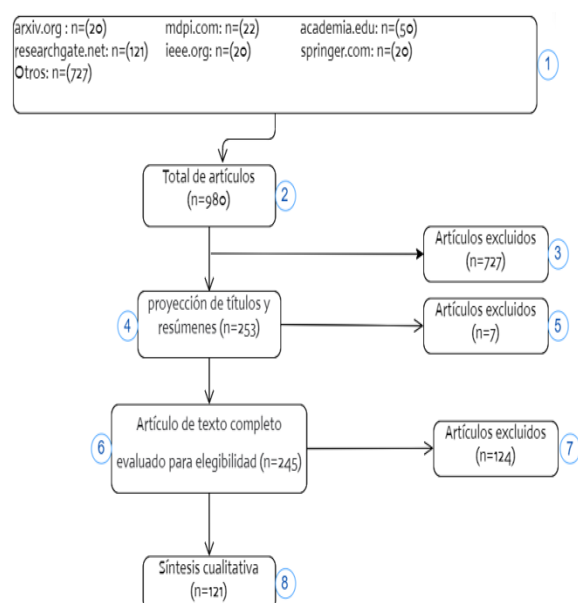


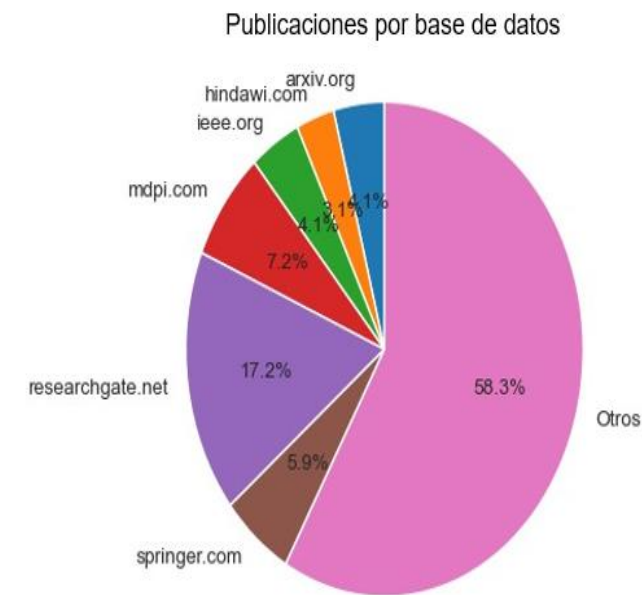
Figura 2: Diagrama de flujo de selección final de manuscritos para el meta-análisis.

Tabla 2: Resumen de los trabajos incluidos para el meta-análisis.

Título	Año	Autores	Editorial
A systematic review on educational ...	2023	Altomari, N., & Straniero, A. M.	springer.com
Complementing educational recommender system ...	2020	S abdi, H Khosravi, S Sadiq, D Gasevic	researchgate.net
Towards a folksonomy graph-based ...	2021	Qassimi, S., Abdelwahed ...	springer.com
Multi-stakeholder personalized learning with ...	2019	Y Zheng	researchgate.net
Learning to trust: Understading editorial ...	2021	T Hassan, B Edminson ...	arxiv.org
...
Application of C4.5 decision tree ...	2022	J Wang	hindawi.com
Integrating courses' relationship into predict...	2020	TN Huynh-Ly, Ht Le ...	researchgate.net
A recommender system for programming online ...	2018	R Yera Toledo, Y Caballero Mota ...	mdpi.com
Classifications of the summative assessment ...	2021	Md Laddha, VT Lokare ...	arxiv.org
CADA: a teacher-facing learning analytics ...	2022	R Kaliisa, JA Dononen	springer.com

Como se puede observar la mayoría de los títulos habla sobre sistemas recomendadores para distintos ámbitos en el área educativa. La Gráfica 1, muestra los porcentajes de las publicaciones por base de datos y deja ver las bases de datos que tienen más del 3% de trabajos publicados, éstas refieren a journals de mayor impacto científico.

procedencia de la publicación. Esto fue codificado como se muestra en la Figura 3.



Gráfica 1: Publicaciones por base de datos.

3.2 Principales hallazgos

Una vez con los manuscritos identificados como más relevantes, se realizó el meta-análisis de manera automática con la ayuda de un programa en Python, el cual, realiza una puntuación de los resúmenes y títulos de los manuscritos con base a palabras clave que son de nuestro interés de acuerdo a la investigación. Igualmente genera una discriminación de los manuscritos que no sean de interés con base a los criterios de exclusión, como el año de publicación, títulos duplicados,

```

import re
import pandas as pd
dataToool = []
for result in results:
    title = result.get("title")
    authors = result.get("publication_info", {}).get("summary")
    link = result.get("link")

    pattern = r"title:\s*{[^]*}"
    coincidence = re.search(pattern, str(result.get("resources", "")))
    value = coincidence.group(1) if coincidence else result.get("resources", "")
    journal = value
    citations = result.get("inline_links", {}).get("cited_by", {}).get("total")
    referenceAPA = result.get("inline_links", {}).get("serpapi_cite_link", {})
    abstract = result.get("snippet")

    try:
        year = re.search(r'\b\d{4}\b', authors).group(0)
    except AttributeError:
        year = "0"
    partes = authors.split(" - ")
    authors = partes[0]

    row = {"title": title, "year": year, "authors": authors, "link": link, "journal": journal, "citations":
    citations, "abstract": abstract, "reference": referenceAPA}
    dataToool.append(row)

df = pd.DataFrame(dataToool)
    
```

Figura 3: Obtención automática de manuscritos.

Como se observa en el código, con base a los resultados de la búsqueda de investigación, para cada elemento *result*, se extraen valores específicos del diccionario y se asignan a variables correspondientes. Se utiliza una expresión regular para buscar un título dentro de un valor específico y se asigna a la variable *value*. Posteriormente, se extraen otras propiedades relevantes y se asignan a variables correspondientes. Se intenta encontrar un año de publicación y se divide la cadena de autores. Se crea un diccionario con las propiedades extraídas y se agrega a la lista *dataToool*. Finalmente, se crea un DataFrame de pandas llamado *df* utilizando la lista *dataToool*.

La Figura 4 muestra el código con el cual, se importa el módulo *matplotlib.pyplot* como *plt*. Luego, se realiza una selección de datos en el DataFrame *datos*, manteniendo solo las filas donde el valor de la columna "anio" es mayor a 2017.

A continuación, se agrupan los datos por el campo "journal" y se cuenta la cantidad de publicaciones para cada buscador, almacenando el resultado en el DataFrame *publicaciones_por_buscador*. Se calcula el porcentaje de publicaciones para cada buscador en relación al total, redondeándolo a un decimal. A continuación, se realiza una serie de cálculos para sumar y restar porcentajes y cantidades de publicaciones, obteniendo un valor para la categoría "Otros". Se agrega esta nueva fila al DataFrame *publicaciones_por_buscador*. Finalmente, se crea un gráfico de pastel utilizando Matplotlib, mostrando las cantidades de publicaciones para cada buscador, con etiquetas y porcentajes correspondientes. El título de la gráfica es "Publicaciones por buscador" y se muestra la gráfica en pantalla utilizando *plt.show()*.

```
import matplotlib.pyplot as plt
datos = datos.loc[datos['anio'] > 2017]

# Agrupa los datos por buscador y cuenta la cantidad de publicaciones para cada buscador
publicaciones_por_buscador = df.groupby("journal").size().reset_index(name="cantidad_publicaciones")

# Calcula el porcentaje de publicaciones para cada buscador
total_publicaciones = publicaciones_por_buscador["cantidad_publicaciones"].sum()
publicaciones_por_buscador["porcentaje_publicaciones"] = publicaciones_por_buscador["cantidad_publicaciones"] / total_publicaciones * 100

# Redondea los porcentajes a un decimal
publicaciones_por_buscador["porcentaje_publicaciones"] = publicaciones_por_buscador["porcentaje_publicaciones"].apply(lambda x: round(x, 1))

sum_original = publicaciones_por_buscador["cantidad_publicaciones"].sum()

publicaciones_por_buscador = publicaciones_por_buscador[publicaciones_por_buscador["porcentaje_publicaciones"] > 3]

# Sumar la columna "porcentaje"
suma_porcentaje = publicaciones_por_buscador["porcentaje_publicaciones"].sum()

# Restar la suma de la columna "porcentaje" de un total de 100%
resta_porcentaje = 100 - suma_porcentaje

# Sumar la columna "porcentaje"
suma_porcentaje_cantidad_publicaciones = publicaciones_por_buscador["cantidad_publicaciones"].sum()

# Restar la suma de la columna "porcentaje" de un total de 100%
resta_porcentaje_cantidad_publicaciones = suma_porcentaje_cantidad_publicaciones - suma_porcentaje

nueva_fila = {'journal': 'Otros', 'cantidad_publicaciones': resta_porcentaje_cantidad_publicaciones, 'porcentaje_publicaciones': resta_porcentaje}

# Agregar la nueva fila al DataFrame
publicaciones_por_buscador = publicaciones_por_buscador.append(nueva_fila, ignore_index=True)

# Crea la gráfica de pastel utilizando Matplotlib
plt.pie(publicaciones_por_buscador["cantidad_publicaciones"],
        labels=publicaciones_por_buscador["journal"], autopct=lambda p: '{:1f}%'.format(p) if p > 3 else "",
        startangle=90)
plt.title("Publicaciones por buscador")
plt.show()
```

Figura 4: Agrupación de publicaciones por editorial, aplicando filtro de año y porcentaje de publicaciones.

La Figura 5, muestra cómo se importan los módulos pandas, numpy, CountVectorizer y LatentDirichletAllocation de Scikit-Learn. A continuación, se carga una tabla de artículos relevantes desde un archivo CSV y se realiza el preprocesamiento de los datos. Posteriormente, se crea una matriz término-documento utilizando CountVectorizer y se ejecuta el modelo de Latent Dirichlet Allocation (LDA) para identificar los temas latentes en los artículos. Después de lo anterior, se extraen los pesos de los temas latentes para cada artículo y se agregan a la tabla, se identifican los términos más importantes para cada tema y se imprimen. Para finalizar se seleccionan los artículos que tienen una alta probabilidad de estar en los temas de interés y se ordenan los artículos relevantes según los temas de interés.

```
import pandas as pd
import numpy as np
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.decomposition import LatentDirichletAllocation

# Cargar la tabla de artículos relevantes desde el archivo CSV
df = pd.read_csv('resultados.csv')
df.fillna(' ', inplace=True)

# Crear una matriz de términos-documentos utilizando CountVectorizer de Scikit-Learn
vectorizer = CountVectorizer(stop_words='english')
term_doc_matrix = vectorizer.fit_transform(df['abstract'])

# Ejecutar el modelo de LDA para identificar los temas latentes en los artículos
lda_model = LatentDirichletAllocation(n_components=20, random_state=123)
lda_model.fit(term_doc_matrix)

# Extraer los pesos de los temas latentes para cada artículo y agregarlos a la tabla
topic_weights = lda_model.transform(term_doc_matrix)
df = pd.concat([df, pd.DataFrame(topic_weights)], axis=1)

# Identificar los términos más importantes para cada tema
feature_names = vectorizer.get_feature_names()
topic_names = ['Topic {}'.format(i) for i in range(lda_model.n_components)]
top_terms = []
for weights in lda_model.components_:
    term_indices = np.argsort(weights)[-10:][::-1]
    top_terms.append([feature_names[i] for i in term_indices])

# Imprimir los términos más importantes para cada tema
for i, topic in enumerate(topic_names):
    print('Top terms for {}:'.format(topic))
    print(' '.join(top_terms[i]))

# Seleccionar los artículos que tienen una alta probabilidad de estar en los temas de interés
relevant_topics = [2, 6, 18] # Por ejemplo, los temas 1, 4 y 7 son de interés
relevant_articles = df[df[relevant_topics].sum(axis=1) > 0]

top_articles = relevant_articles.sort_values(by=relevant_topics, ascending=False)
```

Figura 5: Selección de manuscritos más relevantes.

A continuación, se reportan los principales hallazgos del proceso de meta-análisis realizado.

3.2.1 Tendencias en el desarrollo de sistemas recomendadores en la educación

La Gráfica 2, muestra cómo ha evolucionado el número de trabajos sobre sistemas recomendadores en la educación a lo largo del tiempo. En los años 2018 y 2019 se observa una cantidad similar de trabajos que reportan el desarrollo o aplicación de un sistema de recomendación en la educación, la mayoría de ellos para estudiantes, pero en el año 2020 se puede observar una disminución respecto a los dos años anteriores, esto probablemente debido a la pandemia que se vivió por COVID-19 donde muchas de las actividades tuvieron un decremento incluyendo las publicaciones relacionadas al desarrollo de sistemas de recomendación, pero en los siguientes años, 2021 y 2022, hubo un incremento en el número de publicaciones relacionadas al tema, debido a que la pandemia dejó de manifiesto la necesidad de tener herramientas para los docentes que apoyen al proceso de enseñanza-aprendizaje bajo diferentes circunstancias, puesto que durante el aislamiento, se tuvo la necesidad de contar con distintos métodos de enseñanza que fueran efectivas en las diferentes modalidades de educación: remota, semipresencial o híbrida.

Lo anterior da evidencia de que el desarrollo de herramientas para el apoyo a la docencia, en este caso sistemas recomendadores, es un área de investigación que muestra un interés creciente, esto sirve de apoyo a los investigadores para identificar una área donde ellos pueden aportar al campo del conocimiento a través del desarrollo este tipo de herramientas tecnológicas.

En el eje x de la Gráfica 2 se representa el año de publicación y en el eje y se muestra el número de manuscritos publicados, cabe mencionar que la gráfica fue construida considerando, los 980 trabajos obtenidos inicialmente.

- Comunidad de aprendizaje: Facilita la conexión con colegas afines para colaborar y compartir experiencias educativas.

Además, los sistemas recomendadores pueden ayudar a los docentes a optimizar el uso de los recursos educativos, lo que es especialmente valioso en el contexto actual de presupuestos limitados y de la creciente necesidad de la educación a distancia. De acuerdo con los resultados, sobre los países que reportan la aplicación de sistemas de recomendación en la educación, se debe promover la investigación en el desarrollo de sistemas recomendadores de técnicas de enseñanza como herramientas de apoyo a la labor docente en América Latina, es decir, se necesitan más esfuerzos para fomentar la investigación y la innovación en esta área para que la educación en esta región pueda beneficiarse de esta tecnología.

Es importante destacar que, aunque el uso de los sistemas recomendadores utilizados en el contexto educativo para apoyar a los docentes ha aumentado en los últimos años, todavía existen desafíos y limitaciones que deben abordarse en futuras investigaciones. En particular, es necesario desarrollar investigación sobre cómo integrar los sistemas recomendadores en los sistemas educativos existentes y cómo se pueden abordar los problemas de privacidad y ética relacionados con la recolección y el uso de los datos de los estudiantes.

Finalmente, los resultados muestran, que un área de investigación relevante y de interés sería crear sistemas de recomendación que puedan sugerir no sólo un tipo de estrategia de enseñanza, sino varios métodos de enseñanza y aún más que dichos sistemas tuvieran la capacidad de identificar el tipo de estrategia idóneo para un tema, grupo o situación específica lo cual implicaría profundizar más sobre el modo de construcción de dichos sistemas.

5. Agradecimientos

Este trabajo ha sido realizado gracias al financiamiento otorgado por la Convocatoria Proyectos de Investigación Científica, Desarrollo Tecnológico e Innovación 2023 para el proyecto “Plataforma Inteligente En La Nube Para Recomendación Automática De Estrategias De Enseñanza-Aprendizaje A Docentes De Pregrado, Colaboración JDC Colombia – ITESA México - Fase II” con clave: 16943.23-PD.

De igual modo se reconoce el esfuerzo conjunto de todos los autores del manuscrito quienes pertenecen a la Maestría en Sistemas Computacionales, de Tecnológico Nacional de México / ITS del Oriente del Estado de Hidalgo.

Referencias

¿Qué es Python? - Explicación del lenguaje Python - AWS. (2023). Amazon AWS. <https://aws.amazon.com/es/what-is/python/>

Abdi, S., Khosravi, H., Sadiq, S., & Gasevic, D. (2020, March). Complementing educational recommender systems with open learner models. In *Proceedings of the tenth international conference on learning analytics & knowledge* (pp. 360-365).

Altomari, N., & Straniero, A. M. (2023). A systematic review on educational robotics in Special Education. *Q-TIMES WEBMAGAZINE*, 2(1), 398-413.

Baidada, M., Mansouri, K., & Poirier, F. (2022). Hybrid Filtering Recommendation System in an Educational Context: Experiment in Higher Education in Morocco. *International Journal of Web-*

Based Learning and Teaching Technologies (IJWLTT), 17(1), 1-17.

da Silva, F. L., Slodkowski, B. K., da Silva, K. K. A., & Cazella, S. C. (2023). A systematic literature review on educational recommender systems for teaching and learning: research trends, limitations and opportunities. *Education and Information Technologies*, 28(3), 3289-3328.

Dhahri, M., & Khribi, M. K. (2021). A Review of Educational Recommender Systems for Teachers. *International Association for Development of the Information Society*.

Felfernig, A., Friedrich, G., Jannach, D., & Zanker, M. (2018). An introduction to recommender systems. In *Handbook of digital humanities* (pp. 179-203). Springer.

Gonçalves, A. L., Carlos, L. M., da Silva, J. B., & Alves, G. R. (2018, June). Personalized Student Assessment based on Learning Analytics and Recommender Systems. In *2018 3rd International Conference of the Portuguese Society for Engineering Education (CISPEE)* (pp. 1-7). IEEE.

Hassan, T., Edmison, B., Stelter, T., & McCrickard, D. S. (2021, June). Learning to trust: Understanding editorial authority and trust in recommender systems for education. In *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization* (pp. 24-32).

Huynh-Ly, T. N., Le, H. T., & Nguyen, T. N. (2020). Integrating courses' relationship into predicting student performance. *International Journal*, 9(4).

Kaliisa, R., & Dolonen, J. A. (2022). CADA: a teacher-facing learning analytics dashboard to foster teachers' awareness of students' participation and discourse patterns in online discussions. *Technology, Knowledge and Learning*, 1-22.

Karga, S., & Satratzemi, M. (2019). Using explanations for recommender systems in learning design settings to enhance teachers' acceptance and perceived experience. *Education and Information Technologies*, 24, 2953-2974.

Laddha, M. D., Lokare, V. T., Kiwelekar, A. W., & Netak, L. D. (2021). Classifications of the Summative Assessment for Revised Blooms Taxonomy by using Deep Learning. *arXiv preprint arXiv:2104.08819*.

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1), 76-80.

Laddha, M. D., Lokare, V. T., Kiwelekar, A. W., & Netak, L. D. (2021). Classifications of the Summative Assessment for Revised Blooms Taxonomy by using Deep Learning. *arXiv preprint arXiv:2104.08819*.

Lops, P., de Gemmis, M., & Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In *Recommender systems handbook* (pp. 73-105). Springer.

Machiels, B., Vanderberg, J. M., & Yarikoptic. (2021, Mayo 27). *citeproc-py* · PyPI. PyPI. Retrieved Abril, 2023, from <https://pypi.org/project/citeproc-py/#description>

Manouselis, N., Drachler, H., Verbert, K., & Duval, E. (2012). *Recommender systems for learning*. Springer Science & Business Media.

Moher, D., Liberati, A., Tetzlaff, J., & Altman, D. G. (2009, Julio 21). Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement. *PLoS Med*, 6(7). <https://doi.org/10.1371/journal.pmed.1000097>

Moubayed, A., Injadat, M., Nassif, A. B., Lutfiyya, H., & Shami, A. (2018). E-learning: Challenges and research opportunities using machine learning & data analytics. *IEEE Access*, 6, 39117-39138.

Raj, N. S., & Renumol, V. G. (2022). A systematic literature review on adaptive content recommenders in personalized learning environments from 2015 to 2020. *Journal of Computers in Education*, 9(1), 113-148.

Ricci, F., Rokach, L., & Shapira, B. (2011). Introduction to recommender systems handbook. In *Recommender systems handbook* (pp. 1-35). Springer.

- Silva López, R. B., & Méndez Gurrola, I. I. (2022). Intelligent System for Customizing Evaluation Activities Implemented in Virtual Learning Environments: Experiments and Results. *Computacion y sistemas*, 26(1), 473-484.
- Su, X., & Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009.
- The Pandas Development Team. (2023, April 3). pandas · PyPI. PyPI. Retrieved April, 2023, from <https://pypi.org/project/pandas/#description>
- Tolety, V. B. P., & Prasad, E. V. (2022). Hybrid content and collaborative filtering based recommendation system for e-learning platforms. *Bulletin of Electrical Engineering and Informatics*, 11(3), 1543-1549.
- Vesin, B., Klačnja-Milićević, A., Ivanović, M., & Budimac, Z. (2013). Applying recommender systems and adaptive hypermedia for e-learning personalization. *Computing and informatics*, 32(3), 629-659.
- Wang, X., Zhou, C., & Xu, X. (2019). Application of C4. 5 decision tree for scholarship evaluations. *Procedia Computer Science*, 151, 179-184.
- Yera Toledo, R., Caballero Mota, Y., & Martínez, L. (2018, April). A recommender system for programming online judges using fuzzy information modeling. In *Informatics* (Vol. 5, No. 2, p. 17). MDPI.
- Zheng, Y. (2019, July). Multi-stakeholder personalized learning with preference corrections. In *2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT)* (Vol. 2161, pp. 66-70). IEEE.
- Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1), 1-38.