Modelo para la recomendación personalizada de noticias basado en técnicas de aprendizaje automático

Jamith Bolaños Vidal
José Miguel Buesaco Vela
Nydia Natalia Lozano Hernández

Proyecto Aplicado para optar al título de Magister en Ciencia de Datos

Directora
PhD Gloria Inés Álvarez Vargas

Codirector
PhD Diego Luis Linares Ospina

FACULTAD DE INGENIERIA Y CIENCIAS
MAESTRÍA EN CIENCIA DE DATOS
SANTIAGO DE CALI, DICIEMBRE DE 2024

Maestría en Ciencia de Datos
Facultad de Ingeniería y Ciencias

FICHA RESUMEN
TRABAJO DE GRADO DE MAESTRÍA

TITULO: "Modelo para la recomendación personalizada de noticias basado en técnicas de aprendizaje automático"

1. ÉNFASIS: N/A
2. TIPO DE PROYECTO: Aplicado
3. ÁREA DE TRABAJO: Sistemas personalizados de recomendación
4. ESTUDIANTE (S): Jamith Bolaños Vidal, José Miguel Buesaco Vela, Nydia Natalia Lozano Hernández
5. CORREO ELECTRÓNICO: jbvidal2006@javerianacali.edu.co, josemiguelbuesaco@javerianacali.edu.co, natalialozano06@javerianacali.edu.co
6. DIRECCIÓN Y TELÉFONO: Cra. 2ª # 19 CN 28 Popayán,3167741773; A304 T10 Balcones de la Pradera –Pasto-Nariño, 3163342577; Carrera 24 # 30-56 Edificio Malibú, Apto 301-Bucaramanga-Santander, 3013585676.
7. DIRECTOR: Phd. Gloria Inés Álvarez Vargas
8. VINCULACIÓN DEL DIRECTOR (en la universidad): Planta
9. CORREO ELECTRÓNICO DEL DIRECTOR: galvarez@javerianacali.edu.co
10. CO-DIRECTOR(ES) (Si aplica): Diego Luis Linares Ospina
11. GRUPO O EMPRESA QUE LO AVALA (Si aplica): N/A
12. OTROS GRUPOS O EMPRESAS: N/A
13. PALABRAS CLAVE (al menos 5): aprendizaje automático, sobresaturación de noticias, usuarios, modelos de recomendación personalizada, filtros y digitalización.
14. ODS QUE APLICA EL PROYECTO (Agenda 2030): Industria, innovación e infraestructura
15. FECHA DE INICIO (Desarrollo del proyecto): 29/01/2024
16. RESUMEN (máximo 400 palabras):

La digitalización ha generado que los usuarios se encuentren ante una sobreexposición de información, lo cual hace que tanto los usuarios como los medios de comunicación tradicionales y digitales se vean afectados. Para abarcar esta problemática, la ciencia de datos propone modelos de recomendación de noticias, los cuales tienen como objetivo analizar los gustos de los usuarios y, en función de estos generar filtros para proporcionarle al usuario una experiencia que ofrezca noticias de su interés. En este proyecto aplicado de ciencia de datos construimos un modelo basado técnicas de aprendizaje automático para la recomendación personalizada de noticias. Para lograr el objetivo de se realizaron distintas fases como la preparación de los datos, el modelado, el entrenamiento, la validación y finalmente se desarrolló un prototipo para la recomendación personalizada de noticias. Se aplicaron dos enfoques para las recomendaciones: el filtrado basado en contenido y el filtrado colaborativo, por la estructura de los datos utilizados, este último enfoque generó mejores recomendaciones. Los resultados mostraron que el modelo de Descomposición en Valores Singulares (SVD) presentó el mejor desempeño, esto determinado por la raíz del error cuadrático medio (RMSE) de 0,2461 en las predicciones y un F1-Score de 0,8118 en las listas personalizadas de recomendación de noticias.

TABLA DE CONTENIDO

INTRODUCCION

La ciencia de datos es un campo interdisciplinario que abarca diferentes procesos, métodos y sistemas que permiten utilizar los datos para la extracción de conocimiento, proporcionando herramientas útiles para representar realidades, facilitar el entendimiento y, finalmente, extraer conclusiones o facilitar la toma de decisiones. Entre los campos de análisis utilizados se encuentran la estadística, la minería de datos y el aprendizaje automático, solo por nombrar los más relevantes.

Al utilizar la ciencia de datos como una herramienta para resolver problemáticas de la vida cotidiana, nos encontramos con un desafío que afecta a muchos sectores. La gran disponibilidad de información presente en Internet genera una cantidad abrumadora de noticias y contenido. Esta sobrecarga informativa plantea un desafío que obliga a los usuarios a navegar a través de una gran cantidad de noticias para encontrar contenido relevante y significativo que sea de su interés.

El reto de la recomendación personalizada es una manifestación del problema de filtrado de información y existen dos enfoques básicos para abordarlo. Primero, el filtrado por contenido, que busca identificar características diferenciadoras en las noticias para vincularlas a los gustos de los usuarios, de tal forma que se les puedan recomendar nuevas noticias que compartan dichas características. Segundo, el filtrado colaborativo, que se basa en la idea de que, si dos o más usuarios comparten noticias en su historial de consumo, es muy probable que valoren de manera similar las nuevas noticias que lean.

Para abordar este desafío se utiliza el conjunto de datos de dominio público Microsoft News Dataset (MIND), el cual es procesado y transformado para ser usado posteriormente en las fases de modelado, entrenamiento y validación. Se implementan diferentes modelos, entre los cuales se destaca el filtrado colaborativo, al ser el enfoque más efectivo para lograr recomendaciones personalizadas, asegurando la calidad de los resultados. Se entrenan y evalúan varios modelos de aprendizaje automático, entre ellos k-Vecinos más Cercanos (KNN), Normal Predictor (NP), Descomposición en Valores Singulares (SVD), Factorización No Negativa de Matrices (NMF) y Clustering por k-medias (KM). De las técnicas trabajadas la Descomposición en Valores Singulares (SVD) es la de mejor desempeño.

Como resultado del proyecto, se obtiene un modelo con buen desempeño para desarrollar un prototipo y se elabora un documento final con los resultados obtenidos. De igual forma, se evidencia cómo la ciencia de datos cobra cada vez más importancia, siendo útil para la predicción de comportamientos de los usuarios, la optimización de productos, el ahorro de tiempo y la mejora en la calidad de vida de las personas.

1. DEFINICION DEL PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA

Las ventajas y beneficios de la digitalización y virtualización de la mayoría de las actividades diarias en todos los ámbitos de la vida de las personas son absolutamente incuestionables. Sin embargo, también existen problemas que ahora debe enfrentar cualquier usuario digital, como la cada vez mayor sobreexposición a información de diversos temas proveniente de todo tipo de fuentes. De esta forma, “uno de esos problemas es la dificultad que supone ordenar y filtrar los distintos artículos, dado que la cantidad de noticias disponibles aumenta día a día, así como las interacciones que reciben [1]”.

Desde hace años existen plataformas digitales de noticias que han complementado y que, en varios casos, sustituyeron a medios de comunicación tradicionales como los impresos, la radio y la televisión; para sobrevivir y mantenerse competitivos, los medios tradicionales han lanzado sus sitios web donde actualizan continuamente noticias en texto y en contenido multimedia, ya que la información disponible es abundante y cambiante.

El problema de la sobresaturación de información en los medios digitales debe importarles a dos actores principales: a los medios de comunicación, que están enfocados en atraer y retener la atención del público exponiendo contenidos de interés y de calidad, y, por otro lado, a los usuarios, que necesitan estar bien informados invirtiendo el menor tiempo posible y con la menor cantidad de clics en sus interacciones. Para atender este problema, la ciencia de datos ha desarrollado métodos que se enmarcan en los sistemas de recomendación, específicamente, sistemas personalizados de recomendación de noticias.

En general, en ciencia de datos, un sistema de recomendación es un “sistema inteligente que proporciona a los usuarios una serie de sugerencias personalizadas sobre un determinado tipo de elemento de un dominio particular”. Es decir, un buen sistema de recomendación de noticias es un apoyo a la toma de decisiones tanto para los medios de comunicación, en el sentido de qué presentar, como para los propios usuarios, en el sentido de qué noticias leer mejorando su experiencia de consulta.

Ahora bien, el corpus de la ciencia de datos está en constante evaluación, evolución y desarrollo; es así como se encuentran métodos para sistemas de recomendación de noticias clásicos basados en algoritmos convencionales y, en estos tiempos, con la amplia difusión de técnicas de inteligencia artificial y aprendizaje automático con algoritmos más complejos que generan mejores resultados, pero que, en la misma medida, exigen más recursos de diseño y de cómputo. De todas formas, ambos enfoques buscan la mejor forma de modelar el contenido de las noticias y el comportamiento de los usuarios.

1.2. FORMULACION DEL PROBLEMA

¿Cómo desarrollar un modelo para la recomendación personalizada de noticias basado en técnicas de aprendizaje automático?

Para esto, es necesario conocer:

- ¿Cómo preparar los datos para desarrollar modelos de recomendación personalizada de noticias?
- ¿Cómo modelar la solución que se ajuste a la recomendación personalizada de noticias?
- ¿Cómo entrenar los modelos de aprendizaje automático para la recomendación personalizada de noticias?
- ¿Cómo validar los modelos de aprendizaje automático para la recomendación personalizada de noticias?
- ¿Cómo desarrollar un prototipo para la recomendación personalizada de noticias basado en el modelo de aprendizaje automático de mejor desempeño?

2. OBJETIVOS DEL PROYECTO

2.1. OBJETIVO GENERAL

Desarrollar un modelo para la recomendación personalizada de noticias basado en técnicas de aprendizaje automático.

2.2. OBJETIVOS ESPECÍFICOS

- Preparar los datos para los modelos de recomendación de noticias basados en técnicas de aprendizaje automático.
- Modelar la solución que se ajuste a la recomendación personalizada de noticias.
- Entrenar los modelos para la recomendación personalizada de noticias.
- Validar los modelos de aprendizaje automático para la recomendación personalizada de noticias.
- Desarrollar un prototipo, a partir del modelo de mejor desempeño, para la recomendación personalizada de noticias.

3. JUSTIFICACION

Un modelo para la recomendación personalizada de noticias basado en técnicas de aprendizaje automático es de gran utilidad, ya que permite adaptar las recomendaciones de noticias basadas en el historial y las preferencias de cada usuario. Es decir, en este proyecto aplicado se desarrolla un sistema de recomendación de noticias por filtrado colaborativo, proporcionando así unas recomendaciones más personalizadas, logrando la satisfacción del usuario y mejorando su experiencia.

El incremento en la satisfacción del usuario y la mejora constante de la experiencia aumentan la probabilidad de retener al usuario en la plataforma o página web, lo cual es beneficioso para mejorar la orientación de los anuncios, permitiendo la monetización y publicidad eficiente, y aumentando la efectividad y el valor de la publicidad para anunciantes. Asimismo, los modelos de recomendación ayudan a los usuarios en la optimización del tiempo, permitiéndoles descubrir contenido de manera más eficiente, considerando que el mundo está inundado de información, por lo que se reduce el ruido al generar filtros.

Este proyecto es posible debido a la disponibilidad de conjuntos de datos de dominio público construidos para investigar la recomendación de noticias. Existen varios conjuntos de datos en diferentes idiomas disponibles para desarrollar este proyecto aplicado: Plista (alemán), Adressa (noruego), Globo (portugués), Yahoo! (inglés) y MIND (inglés). Es importante destacar que el acceso y la descarga del conjunto de datos MIND es gratuita para fines de investigación. Los conjuntos de datos disponibles proporcionan una base sólida para entrenar modelos de aprendizaje automático, permitiendo la construcción de modelos precisos y efectivos. Hoy en día se cuenta con grandes avances en la construcción e implementación de estos modelos, lo cual hace que los objetivos planteados sean más alcanzables y efectivos.

La capacidad de adaptación que tienen los modelos de aprendizaje automático y su habilidad para mejorarse y perfeccionarse continuamente a medida que se recopila más información son unas grandes ventajas, ya que permite que el algoritmo se adapte a las necesidades cambiantes de los usuarios, teniendo en cuenta que el ser humano, por naturaleza, no es estático y se encuentra en constante cambio y evolución durante su vida.

Para finalizar, es importante resaltar que este tipo de modelos puede aplicarse en diversos sectores económicos, entre los cuales se pueden destacar medios y comunicaciones, tecnología de la información, servicios financieros, negocios y mercados internacionales, educación, salud y ciencias de la vida, turismo, comercio electrónico, entre otros.

4. MARCO DE REFERENCIA

4.1. Contexto de la dinámica de publicación de noticias en la era digital

Durante los últimos años, los sistemas de recomendación se han vuelto muy populares. Entre los campos más comunes se encuentran las recomendaciones de noticias en diferentes plataformas digitales y páginas web. Un recomendador es un sistema complejo y sofisticado que, mediante distintas técnicas, se encarga de analizar la información de los usuarios, filtrarla y generar conocimiento accionable al predecir qué noticia será atractiva para el usuario, utilizando filtros colaborativos y basados en contenido.

Los sistemas de recomendación son sistemas utilizados hoy día para realizar recomendaciones personalizadas a usuarios de grandes conjuntos de datos. Se basan en el aprendizaje automático y parte de una base de datos muy grande, donde tendremos, por norma general, usuarios que valoran ítems (como pueden ser noticias, información, artículos, productos, películas, canciones etc.).

El sistema de recomendación aprende a partir de esta base de datos y recomienda a los usuarios nuevos ítems que sean de su agrado o concuerden con sus interacciones y comportamiento. Actualmente son muy utilizados en plataformas de contenidos audiovisuales o comercio electrónico, como Netflix, Youtube, Amazon, Spotify o eBay. Google News, Yahoo News, y Apple News son ejemplos de plataformas que emplean los sistemas de recomendación para noticias.

Estas plataformas tienen en común el hecho de trabajar con millones de usuarios, aunque cada una de ellas maneja distintos ítems. Las recomendaciones generadas por el sistema de recomendación ayudan a los usuarios a la toma de decisiones sobre ítems a elegir, algo muy útil cuando, como ocurre actualmente, la información que se le ofrece al usuario es muy abundante, y éste no tiene tiempo de examinar o valorar todas las opciones. Dado que el eje principal es la construcción de un modelo basado en técnicas de aprendizaje automático, enfocado en el desarrollo de un prototipo para la recomendación efectiva de noticias, se explican los principios teóricos más importantes para la comprensión y el desarrollo de este proyecto.

4.2. Ciencia de datos y aprendizaje automático

La ciencia de datos es un campo interdisciplinario que, de acuerdo con lo que dijo Cleveland [2], se describe mediante un diagrama de Venn, donde se ubica en primer lugar el

pensamiento cuantitativo y disciplinado presente en las matemáticas y la estadística. De la estadística se obtiene una comprensión de la variabilidad y la experiencia en el uso de herramientas estadísticas para trabajar con datos. En segundo lugar, la experiencia sustantiva brinda al científico de datos una comprensión del contexto disciplinario de un conjunto de datos, sin la cual será difícil o imposible elegir una metodología de análisis válida. Por último, las habilidades informáticas y de datos, junto con habilidades creativas para resolver problemas, permiten visualizar la estructura de los datos.

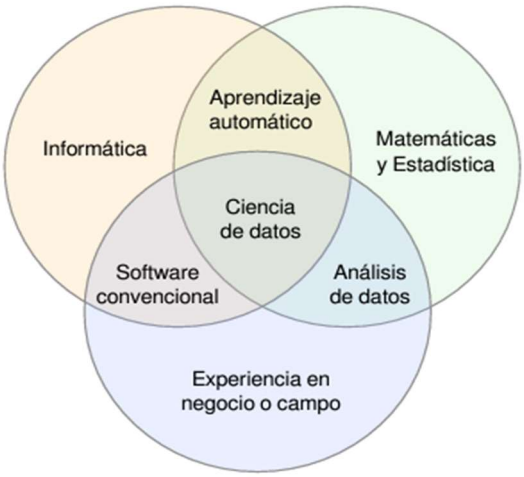

Figura 1. Diagrama de Venn: Ciencia de datos [2]

Se puede definir el aprendizaje automático como una disciplina dentro del campo de la informática, muy ligada a la inteligencia artificial (IA), mediante la cual se generan sistemas que aprenden de manera automática por sí mismos. En realidad, el sistema que aprende es un algoritmo capaz de valorar todos los datos de entrada y, a partir de ahí, realizar varias acciones, como predecir valores o recomendar valores futuros [3].

4.2.1. Tipos de aprendizaje automático

Hay dos áreas principales dentro del campo del aprendizaje automático:

- **Aprendizaje automático supervisado:** Es una técnica en la que se enseña o se entrena a la máquina utilizando datos bien etiquetados. Los modelos de aprendizaje automático supervisado se entrenan con conjuntos de datos etiquetados, lo que permite que los modelos aprendan y crezcan con el tiempo. Los datos de entrenamiento etiquetados indican que cada registro de entrenamiento también tiene la respuesta asociada adjunta. Por lo tanto, se

proporciona una guía clara al algoritmo de interpretación de datos. Luego, el algoritmo construirá y probará un modelo con las etiquetas dadas. Para cada iteración, una función de error validará si la predicción que el modelo generó se alinea con la etiqueta. Después, el modelo se ajustará ligeramente con cada iteración hasta que funcione bien con los datos de entrenamiento.

Hay dos tipos de aprendizaje supervisado. Lo fundamental que hay que recordar es que la clasificación separa los datos, mientras que la regresión se ajusta a ellos. La clasificación es el algoritmo que predice el valor de respuesta categórica para una observación o entrada determinada y predice un valor discreto [2].

- **Aprendizaje automático no supervisado**: Implica el entrenamiento a través de datos no etiquetados, lo que permite que el modelo actúe sobre esa información sin orientación. En el aprendizaje automático no supervisado, un programa busca patrones en datos no etiquetados. Cuando es difícil o demasiado costoso obtener suficientes datos de entrenamiento etiquetados, los métodos no supervisados se vuelven relevantes. Con el aprendizaje no supervisado, no proporcionamos al algoritmo datos de entrenamiento etiquetados. En cambio, buscamos que el algoritmo encuentre una forma de clasificar o separar los datos [2].

4.3. Sistemas de recomendación

Los sistemas de recomendación se utilizan ampliamente para ayudar a los lectores a filtrar una avalancha de información cada vez mayor. Estos sistemas implementan un método de filtrado de información para seleccionar productos de un flujo de información. Además, los sistemas de recomendación recopilan datos de los usuarios de forma explícita o implícita y, en función de la información recopilada, crean perfiles de usuario [4].

Los sistemas de recomendación se han considerado un remedio para superar el problema de la explosión de información, y gran parte del esfuerzo de investigación se ha centrado en desarrollar técnicas de recomendación altamente confiables. Los sistemas de recomendación tradicionales se clasifican según la información que utilizan y cómo la utilizan [5].

Los sistemas de recomendación se dividen en tres categorías, según cómo se generan las recomendaciones [6]:

- Sistemas de recomendación basados en contenido: estos sistemas recomiendan al usuario un elemento similar a los que el usuario prefería en el pasado.

- Sistemas de recomendación colaborativos: estos sistemas recomiendan un artículo al usuario en función de los gustos y preferencias de personas con intereses similares. Tienen la ventaja de poder recomendar elementos para los que hay poca o ninguna información semántica disponible (música, películas, libros).

- Sistemas de recomendación híbridos: estos sistemas combinan técnicas de recomendación colaborativas y basadas en contenido para mejorar la precisión de la recomendación.

En la Figura 2 se presenta un resumen de los diversos modelos de sistemas de recomendación.

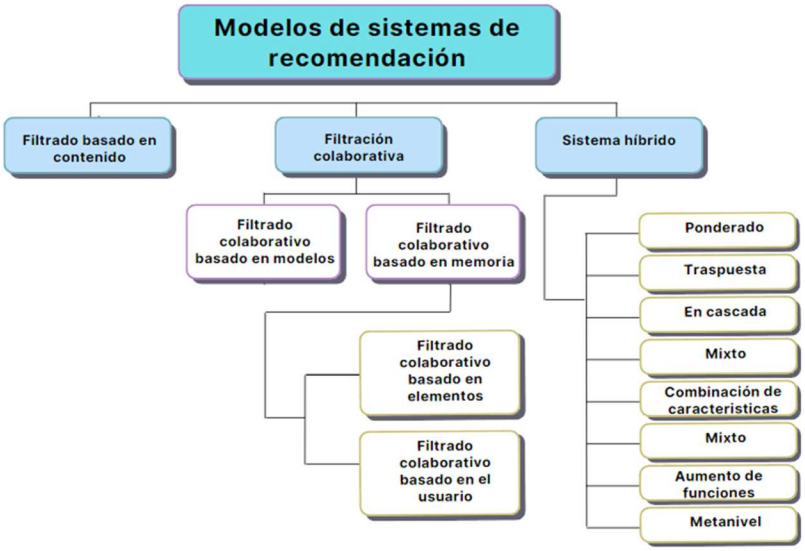

Figura 2. Descripción de los modelos de recomendación [7].

4.3.1. Recomendación basada en contenido

El filtrado basado en contenido recomienda *ítems* que estén dentro del perfil del usuario. Este perfil se puede construir de manera explícita, a partir de información solicitada al usuario, como por ejemplo mediante formularios donde el usuario expresa preferencias, o

de manera implícita, extrayendo información de los *ítems* en los que el usuario ha mostrado interés anteriormente.

Este tipo de sistema de recomendación depende mucho del contexto, ya que se requiere que los *ítems* tengan un conjunto de atributos (también llamados metadatos) que los describan. Estos atributos son especificados manualmente o se obtienen analizando información complementaria, como *tags*, comentarios, descripciones textuales o contenido multimedia, como imágenes, audio o video, por ejemplo. Por otro lado, se necesita que el formato del perfil del usuario se pueda relacionar con los atributos de los *ítems* de tal manera que permita obtener una estimación del interés que el usuario pueda tener en cada *ítem* [9].

4.3.2. Recomendación por filtrado colaborativo

El filtrado colaborativo es el conjunto de técnicas más popular para desarrollar sistemas de recomendación [8]. En este conjunto de técnicas, se intenta hacer la estimación y recomendación con base en el comportamiento o en las calificaciones que los usuarios dan a sus *ítems* [9].

Algunas de estas técnicas suponen que las opiniones de otros usuarios pueden ser utilizadas para estimar de manera adecuada las preferencias del usuario al cual se desean hacer recomendaciones. La idea intuitiva de esto es que, si un conjunto de usuarios está, en cierta medida, de acuerdo en el nivel de interés que tienen sobre un conjunto de *ítems*, entonces deberían coincidir en la misma medida en sus preferencias [10]. Es decir, existe una relación entre los gustos de los usuarios, y, al identificar dicha relación, es posible estimar el nivel de interés de un usuario en cada *ítem*.

Existe un *scikit* de Python denominado SURPRISE (*Simple Python Recommendation System Engine*), es una librería específica para crear y analizar sistemas de recomendación personalizado por filtrado colaborativo. A continuación, se describen cinco métodos de aprendizaje automático de SURPRISE, junto con sus algoritmos y los principales parámetros disponibles en la librería. Los métodos utilizados son los siguientes: k-Vecinos más Cercanos (KNN), Normal Predictor (NP), Descomposición en Valores Singulares (SVD), Factorización No Negativa de Matrices (NMF) y Clustering por k-medias [12].

4.3.2.1. Técnicas de filtrado colaborativo basadas en memoria

La técnica de filtrado colaborativo basado en memoria utiliza algoritmos que trabajan con el conjunto completo de tríadas para estimar el nivel de interés de un usuario en un *ítem*

dado. Para realizar la estimación, se utilizan funciones de agregación, de tal modo que, si tenemos una calificación desconocida del usuario sobre el *ítem*, podemos hacer una estimación de su valor, ya sea utilizando las calificaciones de otros usuarios similares al usuario (basado en usuarios) que sí han calificado. Otra opción es usar las calificaciones que el usuario ha hecho sobre *ítems* similares (basado en *ítems*) [9].

4.3.2.1.1. k Vecinos más Cercanos – KNN

K-Nearest Neighbor (*KNN*) es un algoritmo que clasifica a los k vecinos más cercanos de la tupla de prueba y la tupla de entrenamiento para clasificar un conjunto de datos. *KNN* clasifica los conjuntos de datos según la distancia más cercana, comparando la similitud entre cada elemento de datos mediante una ponderación basada en la distancia [11]. La distancia euclidiana, la similitud del coseno y la correlación de Pearson se utilizan principalmente como medidas para comparar similitudes. Cuando el algoritmo KNN se emplea en un sistema de recomendación, permite clasificar el patrón de búsqueda del usuario y predecir su preferencia futura. Después de analizar los patrones de datos de comportamiento del usuario, como los registros del servidor web y los datos de flujo de clics, el sistema puede clasificar elementos similares a los gustos del usuario y luego usar los resultados para recomendar elementos adecuados [7].

En la Tabla 1, se mencionan algunos parámetros disponibles para este algoritmo [12]:

Tabla 1. Parámetros para *KNN* [12]

Parámetro	Descripción	Valor predeterminado
k	Número (máx) de vecinos para la agregación	40
min_k	Número (mín) de vecinos para la agregación. Si no hay suficientes vecinos, la agregación de vecinos se establece en 0 y la predicción es equivalente a la media.	1
sim_options	Alternativas para la medida de similitud a emplear	msd

La estimación de la predicción se realiza con las siguientes ecuaciones [12]:

$$\hat{r}_{ui} = \frac{\sum_{v \in N_i^k(u)} \boldsymbol{sim}(\boldsymbol{u},\boldsymbol{v}) \cdot \boldsymbol{r_{vi}}}{\sum_{v \in N_i^k(u)} \boldsymbol{sim}(\boldsymbol{u},\boldsymbol{v})} \qquad \textit{Ecuación 1}$$

o

$$\hat{r}_{ui} = \frac{\sum_{j \in N_u^k(i)} \boldsymbol{sim}(\boldsymbol{i},\boldsymbol{j}) \cdot \boldsymbol{r_{uj}}}{\sum_{j \in N_u^k(i)} \boldsymbol{sim}(\boldsymbol{i},\boldsymbol{j})} \qquad \textit{Ecuación 2}$$

donde,

$\boldsymbol{\hat{r}_{ui}}$ es la calificación estimada del usuario ***u*** sobre el ítem ***i***

U es el conjunto de todos los usuarios, ***u*** y ***v*** denotan usuarios

$\boldsymbol{N_i^k(u)}$ representa al conjunto de vecinos más cercanos del usuario ***u*** que han calificado el ítem ***i***

$\boldsymbol{sim(u,v)}$ es la medida de similitud del usuario ***u*** con su vecino ***v***

$\boldsymbol{r_{vi}}$ es la calificación real observada del usuario ***v*** sobre el ítem ***i***

I es el conjunto de todos los ítems, ***i*** y ***j*** denotan ítems

$\boldsymbol{N_i^k(u)}$ representa al conjunto de ***k*** vecinos más cercanos del usuario ***u*** que han calificado al ítem ***i***

$\boldsymbol{N_u^k(i)}$ representa al conjunto de ***k*** vecinos más cercanos del ítem ***i*** que han sido calificados por el usuario ***u***

$\boldsymbol{r_{uj}}$ es la calificación real observada del usuario ***u*** sobre el ítem ***j***

4.3.2.2. Técnicas de filtrado colaborativo basadas en modelos

A diferencia de los sistemas de recomendación basados en memoria, esta técnica de sistemas de recomendación hace recomendaciones construyendo previamente un modelo con base en las calificaciones de los usuarios. Se aborda el problema de las estimaciones como un problema de datos perdidos o de clasificación y se utilizan algoritmos de *machine learning*, como redes neuronales, redes bayesianas, *clustering* o, incluso, se toma inspiración de algoritmos de otros tipos de problemas, como la factorización de matrices.

Una ventaja de estas técnicas es que son más rápidas al momento de estimar la relevancia de los *ítems*, aunque su desventaja es que, al agregar nuevos datos, ya sean usuarios, *ítems* o calificaciones, el modelo necesita ser actualizado [9].

4.3.2.2.1. Normal Predictor (NP)

Normal Predictor (NP) es un algoritmo que predice una calificación aleatoria basada en la distribución del conjunto de entrenamiento, que se supone normal. La predicción $\hat{\boldsymbol{r}}_{\boldsymbol{ui}}$ se genera a partir de una distribución normal $\boldsymbol{N}(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}}^{\boldsymbol{2}})$**,** estimada a partir de los datos de entrenamiento utilizando la estimación de máxima verosimilitud por medio de las siguientes ecuaciones [12]:

$$\hat{\mu} = \frac{1}{|R_{train}|} \sum_{r_{ui} \in R_{train}} r_{ui} \quad \textit{Ecuación 3}$$

$$\hat{\sigma} = \sqrt{\sum_{r_{ui} \in R_{train}} \frac{(r_{ui} - \hat{\mu})^2}{|R_{train}|}} \quad \textit{Ecuación 4}$$

donde,

$\hat{\boldsymbol{\mu}}$ es el promedio de las calificaciones obtenida a partir de los usuarios de entrenamiento

$|\boldsymbol{R_{train}}|$ denota el conjunto de datos de entrenamiento

$\boldsymbol{r_{ui}}$ es la calificación real observada del usuario ***u*** sobre el ítem ***i***

$\hat{\boldsymbol{\sigma}}$ es la desviación de las calificaciones de los usuarios de entrenamiento

4.3.2.2.2. Descomposición en Valores Singulares (SVD)

El famoso algoritmo SVD fue popularizado por Simon Funk durante el Premio Netflix. Cuando no se utilizan líneas base, es equivalente a la factorización de matrices probabilísticas [12].

En la Tabla 2, se mencionan algunos parámetros disponibles para este algoritmo [12]:

Tabla 2. Parámetros para *SVD* [12]

Parámetro	Descripción	Valor predeterminado
n_factors	La cantidad de factores.	100
n_epochs	Número de iteraciones del procedimiento SGD.	20
biased	Indica si se deben utilizar valores de referencia (o sesgos).	True
init_mean	Media de la distribución normal para la inicialización de vectores factoriales.	0
init_std_dev	Desviación estándar de la distribución normal para la inicialización de vectores factoriales.	0.1
lr_all	Tasa de aprendizaje para todos los parámetros.	0.005

La predicción se establece como:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T \cdot p_u \quad \textit{Ecuación 5}$$

donde,

$\hat{r}_{ui}$ es la calificación estimada del usuario ***u*** sobre el ítem ***i***

μ representa la media de todas las calificaciones de los usuarios del sistema

b_u es el sesgo en las calificaciones de los usuarios

b_i es el sesgo en las calificaciones de los ítems

$q_i^T \cdot p_u$ es la interacción entre el usuario ***u*** y el ítem ***i*** representada a través de los factores de los usuarios p_u y de los factores de los ítems q_i

4.3.2.2.3. Factorización No Negativa de Matrices (NMF)

Es un algoritmo de filtrado colaborativo basado en la factorización matricial no negativa, donde los factores de usuario y los factores de ítem se mantienen positivos. El procedimiento de optimización es un descenso de gradiente estocástico regularizado con

una elección específica del tamaño de paso que garantiza la no negatividad de los factores, siempre que sus valores iniciales también sean positivos [12].

En la Tabla 3 se mencionan algunos parámetros para este algoritmo [12]:

Tabla 3. Parámetros para ***NMF*** [12]

Parámetro	Descripción	Valor predeterminado
n_factors	La cantidad de factores.	15
n_epochs	Número de iteraciones del procedimiento SGD.	50
biased	Indica si se deben utilizar líneas base (o sesgos).	False
reg_pu	El término de regularización para los usuarios.	0.06
reg_qi	El término de regularización para los artículos.	0.06
reg_bu	El término de regularización para los usuarios (solo relevante para la versión con sesgos).	0.02

Este algoritmo es muy similar a SVD. La predicción se establece como [12]:

$$\hat{r}_{ui} = q_i^T \cdot p_u \quad \textit{Ecuación 6}$$

donde,

$\hat{\boldsymbol{r}}_{\boldsymbol{ui}}$ es la calificación estimada del usuario ***u*** sobre el ítem ***i***

$\boldsymbol{q}_{\boldsymbol{i}}^{\boldsymbol{T}} \cdot \boldsymbol{p}_{\boldsymbol{u}}$ es la interacción entre el usuario ***u*** y el ítem ***i*** representada a través de los factores de los usuarios $\boldsymbol{p}_{\boldsymbol{u}}$ y de los factores de los ítems $\boldsymbol{q}_{\boldsymbol{i}}$

4.3.2.2.4. Clustering por k-medias (KM)

Es un algoritmo de filtrado colaborativo basado en co-agrupamiento. Básicamente, a los usuarios y a los ítems se les asignan algunos clústeres y algunos co-grupos

En la Tabla 4 se mencionan algunos de los principales parámetros disponibles para este algoritmo [12]:

Tabla 4. Parámetros para *KM* [12]

Parámetro	Descripción	Valor predeterminado
k	El número (máximo) de vecinos a tener en cuenta para la agregación.	40
min_k	La cantidad mínima de vecinos a tener en cuenta para la agregación. Si no hay suficientes vecinos, la predicción se establece en la media global.	1
sim_options	Un diccionario de opciones para la medida de similitud.	N/A
verbose	Indica si se deben imprimir mensajes de seguimiento de estimación de sesgo, similitud, etc.	True

La predicción se establece como:

$$\hat{r}_{ui} = \overline{C_{ui}} + (\mu_u - \overline{C_u}) + (\mu_i - \overline{C_i}) \quad \textit{Ecuación 7}$$

donde,

$\hat{r}_{ui}$ es la calificación estimada del usuario ***u*** sobre el ítem ***i***

$\overline{C_{ui}}$ es la calificación media del agrupamiento C_{ui}

μ_u representa la media de todas las calificaciones dadas por el usuario ***u***

$\overline{C_u}$ es la media de la calificación de los usuarios del agrupamiento C_u

μ_i representa la media de todas las calificaciones dadas al ítem ***i***

$\overline{C_i}$ es la media de la calificación de los ítems del agrupamiento C_i

4.3.3. Métricas para la evaluación del desempeño de los sistemas de recomendación

La relevancia de los resultados está ligada a la satisfacción del usuario. La calidad de estos resultados se puede medir en términos de dos métricas: precisión y recuperación, las cuales se describirán a continuación:

- Precisión: La precisión se define como la proporción de elementos relevantes en el conjunto de resultados de búsqueda. La precisión se puede calcular como [13]:

$$Precisión = \frac{R}{S} \quad \textit{Ecuación 8}$$

donde,

R representa a los ítems que son recomendados y son relevantes para el usuario
S representa al total de ítems recomendados

- Recuperación (*Recall*): La recuperación se define como la proporción de elementos relevantes en el sistema que se encuentran en el conjunto de resultados de búsqueda. La recuperación se puede calcular como [13]:

$$Recuperación = \frac{R}{D} \quad \textit{Ecuación 9}$$

donde,

R representa a los ítems que son recomendados y son relevantes para el usuario
D representa al total de ítems relevantes para el usuario

La consideración de ambas métricas es crucial para evaluar la calidad de los resultados de búsqueda, ya que, en la práctica, la precisión y la recuperación pueden ser métricas contradictorias. Es probable que la inclusión de menos elementos en el conjunto de resultados de búsqueda para aumentar la precisión resulte en una menor recuperación.

Además de la precisión y la recuperación, también se puede considerar la *F1-score*, que es una combinación armonizada de precisión y recuperación. La *F1-score* se puede calcular como [13]:

$$F1\ Score = 2 * (Precisión * Recuperación)/(Precisión + Recuperación) \textit{ Ecuación 11}$$

La F1-score es una medida útil cuando se desea enfatizar el equilibrio entre precisión y recuperación.

4.3.4. Medidas dependientes de la escala

4.3.4.1. Raíz del error cuadrático medio (RMSE)

El error cuadrático medio (RMSE), también llamado desviación cuadrática media, es una medida de uso frecuente para evaluar la diferencia entre los valores pronosticados por un modelo y los valores realmente observados. Estas diferencias individuales, también llamadas residuos, se agregan en el RMSE para obtener una única medida de la capacidad de predicción [14].

Se define el RMSE (*Root Mean Square Error*) como la raíz cuadrada de la media de los errores al cuadrado:

$$RMSE = \sqrt[2]{MSE} = \sqrt[2]{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}$$

Ecuación 10

4.3.4.2. Error medio absoluto (MAE)

Se define MAE (Mean Absolute Error) como la magnitud promedio de los errores de un ejercicio de pronóstico sin tener en cuenta su signo, es decir, el promedio de los valores absolutos de los errores calculados [14]:

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t|$$

Ecuación 11

4.3.5. Métodos de validación y de optimización de hiperparámetros

- *K-fold Cross Validation:* La técnica de validación cruzada K-fold (KCV) es uno de los enfoques más utilizados por los profesionales para la selección de modelos y la estimación de errores de los clasificadores. La KCV consiste en dividir un conjunto de datos en k subconjuntos; luego, de forma iterativa, algunos de ellos se utilizan para aprender el modelo, mientras que los otros se explotan para evaluar su rendimiento. Sin embargo, a pesar del éxito de la KCV, solo existen métodos prácticos de regla empírica para elegir el número y la cardinalidad de los subconjuntos [5].
- Grid Search: El método tradicional de optimización de hiperparámetros es la búsqueda en malla, que simplemente realiza una búsqueda completa sobre un subconjunto dado del espacio de hiperparámetros del algoritmo de entrenamiento. Dado que el espacio de parámetros del algoritmo de aprendizaje automático puede incluir valores reales o ilimitados para algunos parámetros, es posible que necesitemos especificar un límite para aplicar la búsqueda en malla. La búsqueda en malla presenta dificultades en espacios de alta dimensionalidad, pero a menudo puede paralelizarse fácilmente, ya que los valores de los hiperparámetros con los que trabaja el algoritmo suelen ser independientes entre sí [4].

4.4. Antecedentes

4.4.1. Sistemas de recomendación basados en filtrado colaborativo: Aceleración mediante computación reconfigurable y aplicaciones predictivas sensoriales [15].

Debido a los avances digitales, numerosos campos como Big Data, Machine Learning, Blockchain y Cloud Computing han experimentado un crecimiento sustancial. En esta tesis, el punto focal son los sistemas de recomendación, una rama del aprendizaje automático utilizada principalmente en Amazon y Netflix. La investigación se divide en dos áreas de interés: acelerar algoritmos de recomendación mediante hardware reconfigurable basado en FPGA para entornos con datos y usuarios, abordar la complejidad computacional y explorar sistemas de recomendación como motores de predicción ambiental impulsados por mediciones sensoriales, enfatizando el pronóstico basado en el comportamiento.

Este trabajo contribuye al desarrollo del proyecto, ya que incluye un enfoque especifico en sistemas de recomendación, líneas de investigación complementarias, una aplicación práctica más allá del ámbito tradicional de recomendación ampliando el espectro de aplicaciones y evaluación de algoritmos de predicción utilizando la selección más adecuada de datos de prueba. Los cuatro capítulos que componen la tesis abordan cada aspecto, mostrando los aportes innovadores y la producción científica resultante.

4.4.2. A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields [7].

Este artículo examina las tendencias de investigación que vinculan los aspectos técnicos avanzados de los sistemas de recomendación utilizados en diferentes dominios de servicios con los aspectos comerciales de estos servicios, basándose en más de 135 artículos tomados de Google Scholar, desde el año 2010 hasta el 2021. Solo mediante servicios de aplicaciones se analizan de manera confiable el modelo de recomendación, la tecnología de extracción de datos y la investigación relacionada en el sistema de recomendación.

Para el proyecto, este artículo proporciona una revisión bibliográfica exhaustiva, información sobre la sistematización de estudios sobre modelos de recomendación, análisis de tendencias por año e información sobre la clasificación de campos de aplicación, con una visión más detallada del uso de estas tecnologías en diferentes contextos.

4.4.3. Introduction to Algorithmic Marketing [13]

Este libro es una guía completa de introducción al marketing algorítmico sobre la automatización avanzada del marketing, diseñada para científicos de datos, jefes de producto e ingenieros de software. El libro proporciona una visión general completa de la toma de decisiones automatizada y una justificación teórica, incluyendo fundamentos, aplicaciones, desafíos y oportunidades. De igual forma, proporciona información sobre cómo utilizar algoritmos para la recomendación de noticias de acuerdo con las preferencias de los usuarios, información útil para el desarrollo de un modelo eficaz.

4.4.4. Categorización de letras de canciones de un portal web usando agrupación [16].

En este trabajo se evidencian algoritmos de clasificación y agrupamiento para emplearlos en un sistema de recuperación de información musical, clasificando las canciones según el contenido de su letra por género, modo o tema. En esta investigación se usa un modelo no supervisado de minería de datos para agrupar letras de canciones recopiladas en un portal web, con el fin de ofrecer mejores resultados en las búsquedas de los usuarios. Inicialmente, el modelo identifica el lenguaje de la letra de la canción; luego, estas letras son representadas en un modelo de espacio vectorial *Bag of Words* (BOW) usando características de *Part of Speech* (POS). Posteriormente, se estima un número adecuado de agrupamientos, denominado (k), y se emplean algoritmos particionales y jerárquicos para obtener la clasificación de los grupos de canciones.

Para evaluar los resultados de cada agrupamiento, se utiliza el índice Davies-Bouldin (DBI). Esta investigación arroja resultados positivos, demostrando que es posible agrupar canciones únicamente con el contenido de la letra según el género, sentimiento y tema, gracias a técnicas no supervisadas que emplean solo la información de la letra.

4.4.5. Sistemas de recomendación de noticias basados en aprendizaje profundo [17]

En este trabajo se realizó una revisión sobre las técnicas utilizadas hoy en día en el campo de la recomendación de noticias, analizando el rendimiento y la eficiencia de cada una. Adicionalmente, se desarrolló una aplicación web, la cual tuvo como propósito analizar los mejores métodos para poner en producción modelos de aprendizaje automático, generando como resultado una interfaz gráfica que permite a los usuarios comparar el funcionamiento de los diferentes algoritmos sobre ejemplos reales.

En la elaboración de nuestro proyecto, el trabajo de grado nos aporta metodologías de evaluación para nuestro sistema de recomendación, permitiendo medir la efectividad, lo que proporcionaría una base sólida y estandarizada para los resultados. De igual forma, ofrece una perspectiva sobre los algoritmos a implementar y cómo se pueden combinar diferentes enfoques para mejorar la precisión y la relevancia de las recomendaciones en nuestro modelo. Finalmente, los casos de estudio allí reportados se convierten en referencias que nos pueden servir de guía e inspiración sobre cómo abordar los desafíos específicos con los cuales nos podemos topar en el desarrollo del trabajo.

4.4.6. Desarrollo de un sistema inteligente para la generación, identificación, clasificación, redacción y recomendaciones de noticias utilizando GPT-J [18]

El presente trabajo trata del desarrollo de un sistema inteligente mediante la utilización de lenguaje autorregresivo, que permite identificar, clasificar, generar y recomendar noticias a través de temas o parámetros ingresados por el usuario. El documento cuenta con seis fases. En la primera fase se estudian los diferentes métodos de recolección de datos, fundamentos de procesamiento de lenguaje natural, algoritmos de machine learning y deep learning. En la segunda fase se recolectan los datos; en la tercera fase se crea un repositorio; en la cuarta fase se configura el ambiente de desarrollo en base a los requisitos previos para implementar el modelo GPT-J. En la quinta fase, y quizás una de las más importantes para el desarrollo de nuestro proyecto, se realiza el diseño de los modelos de clasificación, generación y recomendación. Finalmente, en la sexta fase del documento nos topamos con el diseño y desarrollo de la aplicación web.

5. PREPARACION DE LOS DATOS

5.1. Obtención de los datos

En los últimos años, se han generado muy buenos *datasets* para la investigación en este tema. Entre estos se encuentran PLISTA, ADRESSA, GLOBO, YAHOO y MIND [21]. En el presente proyecto aplicado se utilizó MIND (*Microsoft News Dataset*) por ser el más completo y adecuado para los objetivos planteados.

MIND es un conjunto de datos de dominio público para uso académico y de investigación en sistemas personalizados de recomendación de noticias [19], y se puede descargar de forma gratuita conforme a los términos de licencia de *Microsoft Research* [20].

5.2. Entendimiento de los datos

La construcción de MIND se realizó a partir del comportamiento anonimizado de los usuarios activos de la plataforma *Microsoft News*. Contiene 1 millón de usuarios con el registro de su comportamiento en cuanto al consumo de noticias sobre más de 160 mil artículos digitales en inglés y más de 2 millones de interacciones. La muestra incluye a los usuarios que tienen un récord de al menos 5 noticias vistas durante las 6 semanas entre el 12 de octubre y el 22 de noviembre de 2019 [21]. MIND está conformado por 4 conjuntos de datos diferentes: *News, Behaviors, Entity y Relation*. En este proyecto aplicado se utilizan los 2 primeros.

El conjunto de datos *News* se utiliza únicamente para ilustrar el contenido de las noticias y para mejorar la presentación en las listas de recomendación generadas para los usuarios. Cada noticia tiene un identificador único que comienza con la letra "N" seguida de un número consecutivo; el código de identificación de la noticia sirve como vínculo con los otros conjuntos de datos de MIND. En la Tabla 5, se presentan los nombres de los campos utilizados del conjunto de datos *News* y un ejemplo de cada uno.

Tabla 5. Descripción del conjunto de datos News de MIND

Campo	Ejemplo
Id_News	N37378
Category	sports
SubCategory	golf
Title	PGA Tour winners
Abstract	A gallery of recent winners on the PGA Tour
URL	https://www.msn.com/en-us/sports/golf/pga-tour/

El conjunto de datos *Behaviors* registra el comportamiento de los usuarios respecto a su consumo de noticias y es la fuente de la cual se extrajeron los datos para modelar el sistema de recomendación por filtrado colaborativo. En la Tabla 6, se presentan los nombres de los campos utilizados del conjunto de datos *Behaviors* y un ejemplo de cada uno.

Tabla 6. Descripción del conjunto de datos *Behaviors* de MIND

Campo	Ejemplo
Id_Display	783
Id_User	U4558
Record	N23 N456 N239 N58 N371 N502 N495

El campo "*Record"* registra el comportamiento de cada usuario en cuanto al consumo de noticias, es decir, en qué noticias se interesó cada usuario. Estos datos son suficientes para modelar la interacción usuario-noticia y generar las recomendaciones por filtrado colaborativo. Es importante tener en cuenta que el mínimo de noticias vistas por un usuario para ser incluido en MIND es de 5.

MIND está disponible al público para fines académicos y de investigación en varias versiones, con los tamaños y características que se presentan en la Tabla 7 [22]:

Tabla 7. Características de las versiones de MIND

Versión MIND	News	Tamaño News	Behaviors		Tamaño Behaviors
			Interacciones	Usuarios	
Small-Train	51,281	39.29M	156,694	49,108	87.76M
Small-Validation	40,392	31.97M	70,937	48,593	40.85M
Large-Train	96,105	80.95M	2,186,682	698,365	1.28G
Large-Test	114,246	96.57M	2,341,618	698,012	1.36G
Large-Validation	68,394	56.32M	365,2	248,972	219.98M

5.3. Limpieza

Los dos conjuntos de datos, *News* y *Behaviors*, están muy bien construidos y estructurados. De todas maneras, se eliminan algunos registros con datos faltantes en los campos seleccionados. De esta forma, para la versión utilizada, *News* resulta en 48,615 registros correspondientes a noticias diferentes y *Behaviors* en 153,726 interacciones de 50,000 usuarios. Los modelos presentados a continuación se trabajaron con una muestra de datos

de MIND en su versión *Small-Train*, depurada, con un tamaño de 21,106 usuarios y 27,097 noticias. El muestreo es una de las principales técnicas en minería de datos para seleccionar un subconjunto de datos relevantes desde un conjunto de datos mucho más grande. El muestreo puede ser usado cuando procesar el conjunto de datos completo es demasiado costoso [23].

5.4. Transformación

El conjunto de datos de entrada de los modelos de recomendación implementados corresponde a una tripleta Usuario-Noticia-Clic. Entonces, es necesario transformar las interacciones registradas en *Behaviors* para modelar el comportamiento y la percepción de las noticias por parte de los usuarios. Para esto, se "explota" el contenido del campo "*Record*", lo cual significa que, por cada usuario, se registran en filas independientes cada una de las noticias que un usuario determinado leyó (ver Tabla 8).

Dado que la cantidad de artículos disponible es abundante y un usuario normalmente solo se va a interesar en unas cuantas noticias, es necesario inferir algunas noticias en las que el usuario no haya mostrado interés. Se asume que, con las restantes noticias, el usuario todavía no ha tenido contacto y son susceptibles de ser parte de una lista de recomendación. Es precisamente de dicha gran cantidad de noticias de donde, con las técnicas de aprendizaje automático, se deben seleccionar aquellas que se estime sean de mayor interés para cada uno de los usuarios del sistema.

En este caso, para la recomendación de noticias, el *Rating,* se establece de acuerdo a si un usuario determinado se interesó o no se interesó en una noticia específica. Se asume que, si al usuario se le presenta un subconjunto de noticias, puede hacer clic en las que le interesan y no hacer clic en las que no le interesan, de esta forma, con clic=1 se representan las noticias de interés para el usuario y con clic=0 las noticias que no son de interés para el usuario, un ejemplo de esta representación se muestra en la Tabla 8.

Así, se crea un campo binario denominado *Clic* para registrar el comportamiento de todos los usuarios del sistema. Los valores de 0, o sea, las noticias que no son de interés para el usuario, se generan aleatoriamente a partir del conjunto de noticias de interés para los usuarios del sistema. En la primera columna del conjunto de datos resultante, el usuario se repite tantas veces como la cantidad de noticias con las que haya tenido contacto, y en la columna *Clic* se registra si le interesó una noticia con un 1 o, si no le interesó con un 0. Utilizando el usuario U4558 presentado en la Tabla 6, un ejemplo de la transformación de los datos resultante se muestra en la Tabla 8.

Tabla 8. Tripleta de datos de entrada después de la transformación

Usuario	Noticia	Clic
U4558	N1	1
U4558	N2	1
U4558	N3	1
U4558	N4	1
U4558	N5	1
U4558	N6	1
U4558	N7	1
U4558	N8	1
U4558	N9	0
U4558	N10	0
U4558	N11	0
U4558	N12	0
U4558	N13	0
U4558	N14	0
U4558	N15	0

Finalmente, a partir de las tripletas de datos, se crea una matriz dispersa Usuario-Noticia. Esto se logra al pivotar la matriz de la Tabla 8, esto es estructurar los datos de tal forma que en las filas se ubiquen todos los usuarios únicos del sistema, en las columnas se dispongan todas las noticias únicas disponibles y, en la celda de la matriz, donde se cruce un Usuario con una Noticia determinada, se incluya un 1 si el usuario se interesó en la noticia, 0 si no se interesó y quede vacío si el usuario todavía no ha tenido contacto con la noticia. De esta manera, la mayoría de los valores de la matriz están vacíos, razón por la cual se trata de una matriz dispersa. Un esquema de la matriz Usuario-Noticia se presenta en la Tabla 9.

Para facilitar la comprensión de lo expuesto en el párrafo anterior, en la Tabla 9, las celdas con valores vacíos se reemplazan con una X. En el contexto de este proyecto aplicado de ciencia de datos, los métodos de aprendizaje automático implementados predicen el valor de las X a partir de la interacción histórica de los usuarios con las noticias.

En la matriz de la Tabla 9, las celdas con valor de 1 representan a las noticias que son de interés para los usuarios, las celdas con valor de 0 representan a las noticias que no son de interés para el usuario y las celdas con X representan a las noticias con las que el usuario todavía no ha tenido contacto o no conoce y, precisamente, los métodos de aprendizaje automático implementados predicen el interés que un usuario pueda llegar a tener en las noticias que todavía no conoce, esto con base en el comportamiento histórico de su consumo de noticias.

Tabla 9. Matriz dispersa Usuario-Noticia para los modelos de recomendación

Usuario/Noticia	N0	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	N13	N14
U0	0	X	0	1	0	0	X	0	X	0	0	X	0	0	1
U1	X	0	0	X	0	0	0	X	1	X	1	0	X	0	X
U2	1	0	X	0	0	1	X	1	0	X	0	1	X	1	1
U3	X	0	1	0	X	1	X	1	0	0	X	X	X	1	1
U4	X	X	X	1	1	0	X	1	X	1	1	0	X	X	0
U5	0	X	0	1	0	X	1	X	0	1	0	X	0	X	0
U6	X	0	1	1	0	X	1	0	0	X	0	X	0	X	X
U7	0	0	X	0	0	1	X	0	0	1	0	1	X	0	X
U8	X	0	0	X	0	0	X	0	0	X	0	X	1	0	1
U9	0	X	0	0	1	X	1	X	0	0	1	0	X	0	X

Las técnicas de aprendizaje automático para la recomendación personalizada de noticias implementadas en este proyecto aplicado toman como entrada la interacción de los usuarios con las noticias, dicha interacción está representada en la tripleta Usuario-Noticia-Clic con la estructura presentada en la Tabla 8, para predecir el interés de los usuarios hacia las noticias que todavía no conocen. Los valores a predecir, es decir, las salidas de los métodos de aprendizaje automático están representados por una X en la matriz Usuario-Noticia de la Tabla 9.

6. MODELADO

Se experimenta con dos enfoques para desarrollar el modelo de recomendación personalizada de noticias: el primer enfoque es la recomendación basada en contenido y, en el segundo, se modela la solución por filtrado colaborativo. De acuerdo con los objetivos planteados en este proyecto aplicado de ciencia de datos, las características de los datasets de dominio público disponibles y los resultados obtenidos, el desarrollo del proyecto se realizó con un enfoque en filtrado colaborativo basado en el usuario. A continuación, se explica el modelado por los dos enfoques mencionados y se explica por qué, en el contexto de este proyecto aplicado, el filtrado colaborativo presenta un mejor desempeño que el filtrado basado en contenido.

Para el enfoque basado en contenido, para definir las características representativas de cada noticia, se utilizó una codificación de la categoría y subcategoría, así como una representación vectorial del título de cada artículo mediante la Frecuencia del Término – Frecuencia Inversa de Documento (TF-IDF).

Para el enfoque por filtrado colaborativo, se configuraron y realizaron experimentos con cinco técnicas de aprendizaje automático para la recomendación personalizada de noticias: la primera, basada en memoria, denominada k-Vecinos más Cercanos (KNN), y otras cuatro basadas en modelos: Normal Predictor (NP), Descomposición en Valores Singulares (SVD), Factorización No Negativa de Matrices (NMF) y Clustering por k-medias (KM).

6.1. Modelado por un enfoque basado en contenido

El principio general para la recomendación personalizada con un enfoque basado en contenido es identificar elementos característicos en las noticias que han sido de interés para los usuarios. De esta forma, es posible recomendar a un usuario otras noticias que compartan los elementos característicos comunes identificados. En este caso, a excepción de la categoría y subcategoría, no se dispone de más elementos comunes y representativos que identifiquen a las noticias. Entonces, aprovechando que hay texto disponible, se aplicó una representación de la Frecuencia del Término – Frecuencia Inversa de Documento (TF-IDF) al texto del título para, junto con la codificación de la categoría y subcategoría de cada noticia, generar un vector de elementos característicos que represente la naturaleza de cada noticia.

Además del vector de elementos característicos que representa cada noticia, es necesario, de la misma forma, incluir en el modelo el perfil de los usuarios. Esto se realiza a partir de su consumo histórico de noticias, utilizando los mismos campos: Título, Categoría y

Subcategoría. Dado que un usuario tiene en su historial el consumo de una o más noticias, en el conjunto de datos de entrada del modelo, un mismo usuario va a aparecer varias veces. En la tabla 10, se presenta la configuración de los datos de entrada que se utilizan para modelar la recomendación personalizada de noticias con un enfoque basado en contenido.

Tabla 10. Datos de entrada para la recomendación basada en contenido

	Id_User	Id_News	Category	Subcategory	Title	Clicked
94430	U85438	N14881	movies	movies-celebrity	Jake Gyllenhaal heroically rescues Dalmatian i...	1
200662	U41516	N13840	finance	finance-companies	Airbus to sell 100 jets to US carrier Spirit A...	0
124659	U83913	N46568	news	newsworld	Netanyahu says Iran seeking means to attack Is...	1
169688	U22910	N24856	sports	football_nfl	Studs and duds from Jets' squeaker win vs. equ...	0
248432	U64111	N21434	news	newsus	Columbus statue beside Coit Tower vandalized w...	0
61804	U47315	N943	sports	football_nfl	'Bad on them, bad on the brand': How Dak Presc...	1
100699	U63521	N57558	foodanddrink	tipsandtricks	Having a Pie-Dough Meltdown? Make a Galette In...	1
97034	U4927	N54267	news	newspolitics	Mark Kelly raises astronomical sum in bid to s...	1
1816	U47685	N33213	news	newsopinion	Opinions | The math for Warren's health-care p...	1
70845	U2906	N37169	news	newscrime	A burglar hid in a Costco for hours before ste...	1

Como se observa en la Tabla 10, los datos se estructuran para que cada usuario tenga vinculadas las noticias en las que ha estado interesado y otras noticias en las que no ha estado interesado. Esto último se representa en una variable denominada *Clicked*, que toma los valores de: 1 para las noticias de interés del usuario y 0 para las noticias en las que no ha estado interesado. Con los datos estructurados de la forma presentada, es posible implementar un algoritmo de aprendizaje supervisado de clasificación que tome como variables de entrada el resultado de la representación vectorial de la categoría (elementos característicos del contenido), subcategoría y título de cada noticia, y como variable de salida la variable *Clicked*. El algoritmo de clasificación que se utiliza es k-vecinos más cercanos.

De esta forma, en el filtrado basado en contenido, se busca predecir si a un usuario le interesará (*Clic=1*) o no le interesará (*Clic=0*) una determinada noticia que todavía no ha visto, se trata de un modelo de aprendizaje automático supervisado de clasificación. Lo anterior, con base en un vector de elementos característicos del contenido de cada noticia, es decir, el modelo aprende, de acuerdo con la representación vectorial del contenido de los artículos, una forma para clasificar si nuevas noticias serán de interés o no para los usuarios.

Es importante validar los resultados de las predicciones realizadas por el modelo de clasificación por k-vecinos más cercanos. Para esto, se realiza una partición del conjunto de datos de entrada, asignando el 80% para entrenamiento y el 20% para prueba. La exactitud del modelo se mide en términos de Precisión, *Recall y F1-Score*. Los resultados de la validación del modelo de clasificación se presentan en la Tabla 11.

Tabla 11. Resultados de la validación del modelo de clasificación en *Clicked*=1 o *Clicked*=0

```
Accuracy: 0.8774077443341376
Confusion Matrix:
 [[21869  3405]
 [ 2794 22498]]
Classification Report:
              precision    recall  f1-score

           0       0.89      0.87      0.88
           1       0.87      0.89      0.88

    accuracy                           0.88
   macro avg       0.88      0.88      0.88
weighted avg       0.88      0.88      0.88
```

El modelo de clasificación descrito anteriormente, se utiliza para generar listas de recomendación de noticias para los usuarios. Para esto se crea una función que toma como entrada la base de datos de noticias disponibles que un determinado usuario todavía no ha visto y se han clasificado como de su interés. Finalmente, con base en las noticias que se estima son de interés para el usuario y en el algoritmo k-vecinos más cercanos, se definen las listas de recomendación con base en la similitud con respecto a las noticias en las que el usuario se ha interesado históricamente. En la Figura 3, se presenta un ejemplo de la lista de recomendación por filtrado basado en contenido.

	Id_News	Title	Prediction
14031	N22466	A 911 supervisor was streaming Netflix at work...	1.0
2974	N62853	A bodybuilder showed how fitness influencers c...	1.0
2980	N59691	California's legal weed profits going up in smoke	1.0
2979	N10770	Apparently Meghan Markle Isn't Allowed to Wear...	1.0
27649	N14947	How Meghan Markle, Prince Harry, and the rest ...	1.0
24967	N34245	No. 19 Michigan routs No. 8 Notre Dame 45-14 i...	1.0
9384	N5440	Feds take down the world's 'largest dark web c...	1.0
17150	N16363	Mom who touted daughter's 'bucket list' accuse...	1.0
22418	N34824	Birmingham couple charged with murder after ab...	1.0
6096	N986	Ken Fisher's sexist comments have cost his com...	1.0

Figura 3. Ejemplo de la lista de recomendación por filtrado basado en contenido

Aunque el modelo presentado para la recomendación personalizada de noticias basado en contenido es coherente en su implementación, la escasez de elementos característicos que

definan la naturaleza de las noticias (solo están disponibles la categoría y la subcategoría) hace que los vectores de representación de las noticias sean muy similares y, en consecuencia, las listas de recomendación generadas sean mínimamente variadas. En la mayoría de los casos, como se puede observar en la Figura 4, las listas de recomendación para los usuarios U59176, U25665 y U55207 son exactamente las mismas.

```
# Ejemplo de uso U59176
```

	Id_News	Title	Prediction
14031	N22466	A 911 supervisor was streaming Netflix at work...	1.0
2974	N62853	A bodybuilder showed how fitness influencers c...	1.0
2980	N59691	California's legal weed profits going up in smoke	1.0
2979	N10770	Apparently Meghan Markle Isn't Allowed to Wear...	1.0
27649	N14947	How Meghan Markle, Prince Harry, and the rest ...	1.0
24967	N34245	No. 19 Michigan routs No. 8 Notre Dame 45-14 i...	1.0
9384	N5440	Feds take down the world's 'largest dark web c...	1.0
17150	N16363	Mom who touted daughter's 'bucket list' accuse...	1.0
22418	N34824	Birmingham couple charged with murder after ab...	1.0
6096	N986	Ken Fisher's sexist comments have cost his com...	1.0

```
# Ejemplo de uso U25665
```

	Id_News	Title	Prediction
14031	N22466	A 911 supervisor was streaming Netflix at work...	1.0
2974	N62853	A bodybuilder showed how fitness influencers c...	1.0
2980	N59691	California's legal weed profits going up in smoke	1.0
2979	N10770	Apparently Meghan Markle Isn't Allowed to Wear...	1.0
27649	N14947	How Meghan Markle, Prince Harry, and the rest ...	1.0
24967	N34245	No. 19 Michigan routs No. 8 Notre Dame 45-14 i...	1.0
9384	N5440	Feds take down the world's 'largest dark web c...	1.0
17150	N16363	Mom who touted daughter's 'bucket list' accuse...	1.0
22418	N34824	Birmingham couple charged with murder after ab...	1.0
6096	N986	Ken Fisher's sexist comments have cost his com...	1.0

```
# Ejemplo de uso U55207
```

	Id_News	Title	Prediction
14031	N22466	A 911 supervisor was streaming Netflix at work...	1.0
2974	N62853	A bodybuilder showed how fitness influencers c...	1.0
2980	N59691	California's legal weed profits going up in smoke	1.0
2979	N10770	Apparently Meghan Markle Isn't Allowed to Wear...	1.0
27649	N14947	How Meghan Markle, Prince Harry, and the rest ...	1.0
24967	N34245	No. 19 Michigan routs No. 8 Notre Dame 45-14 i...	1.0
9384	N5440	Feds take down the world's 'largest dark web c...	1.0
17150	N16363	Mom who touted daughter's 'bucket list' accuse...	1.0
22418	N34824	Birmingham couple charged with murder after ab...	1.0
6096	N986	Ken Fisher's sexist comments have cost his com...	1.0

Figura 4. Similitud de las listas de recomendación por filtrado basado en contenido

Una de las principales limitantes del enfoque de recomendación basado en contenido es que si no es posible establecer unas características claramente diferenciables en los elementos a recomendar, como en este caso las noticias, en las que solo se dispone de la categoría, la subcategoría y la

posibilidad de representar vectorialmente el contenido del texto de los artículos, las recomendaciones generadas no son las mejores para obtener una buena experiencia del usuario, ya que todas las noticias son muy parecidas en los términos de la representación vectorial planteada.

Como trabajo futuro, para solucionar este problema en un entorno de producción real, se podría experimentar con:

- Otros métodos para la representación vectorial del contenido textual de las noticias
- Recolectando más datos distintivos de las noticias, que permitan caracterizarlas claramente, incluyendo: el autor, la región geográfica y creando más ramificaciones en la categorización de las noticias. El conjunto de datos utilizado en este proyecto aplicado no dispone de este tipo de datos, la amplitud (número de categorías y subcategorías) y la profundidad (número de subcategorías que forman cada categoría) no es suficiente para lograr una buena diferenciación en las noticias.

Por ejemplo, el conjunto de datos MIND (*Microsoft News Dataset*) utilizado en este proyecto aplicado, solo dispone de:

- Categoría: Deportes
- Subcategoría: Fútbol

Para generar mejores recomendaciones por el enfoque de filtrado basado en contenido sería necesario incluir más subcategorías, por ejemplo:

- Categoría: Deportes
- Subcategoría 1: Fútbol
- Subcategoría 2: Fútbol Masculino
- Subcategoría 3: Fútbol Masculino Europeo
- Subcategoría 4: Fútbol Masculino Europeo Liga Inglesa
- Subcategoría 5: Fútbol Masculino Europeo Liga Inglesa Jugadores Extranjeros

Seguramente, realizar este tipo de categorizaciones no es práctico en el caso de las noticias, como si lo es para otros elementos como: películas, música, viajes, libros, en fin, bienes más duraderos que una noticia. A los bienes más duraderos, los expertos les realizan minuciosas caracterizaciones que definen claramente los elementos haciéndolos más diferenciables y, de esta forma, generar mejores recomendaciones basadas en los gustos y preferencias de los usuarios.

Por las razones anteriormente expuestas, a continuación, se desarrolla el modelado del enfoque de recomendación por filtrado colaborativo basado en el usuario, enfoque que resuelve varios de los inconvenientes del filtrado basado en contenido generando mejores resultados en el contexto de las métricas de desempeño aplicadas.

6.2. Modelado por un enfoque basado en filtrado colaborativo

El filtrado colaborativo se basa en el historial de consumo de noticias de un usuario en relación con el comportamiento del consumo de noticias de otros usuarios del sistema. La idea central de la recomendación por filtrado colaborativo es que el interés de un usuario respecto a una noticia probablemente sea similar al de otro usuario, si los dos comparten noticias de interés.

Las recomendaciones del enfoque por filtrado colaborativo se realizan a través de la retroalimentación de otros usuarios. Esto permite que las listas de recomendación puedan contener noticias con la capacidad de sorprender porque no se basan solamente en los gustos históricos de los usuarios. El filtrado colaborativo puede recomendar noticias inesperadas, con alta probabilidad de que sean de interés para el usuario.

Los modelos de filtrado colaborativo pueden agruparse en dos clases generales: los métodos basados en vecinos y los métodos basados en modelos. En los métodos basados en vecinos (también llamados métodos basados en memoria), los registros del interés Usuario-Noticia almacenados en la memoria se utilizan directamente para predecir el interés del usuario en noticias que todavía no ha visto. En contraste, los métodos basados en modelos emplean los registros de interés que los usuarios han tenido en las noticias para entrenar un modelo predictivo. El objetivo de estos métodos es modelar las interacciones Usuario-Noticia mediante factores latentes “ocultos” que representan características tanto de los usuarios como de las noticias, como las preferencias de los usuarios y las categorías de las noticias. Este modelo se entrena utilizando los datos disponibles y luego se emplea para predecir el interés de los usuarios en noticias que todavía no han visto.

De esta forma, el problema a resolver mediante el enfoque de filtrado colaborativo para la recomendación personalizada de noticias consiste en estimar la posible respuesta que un usuario tenga ante una noticia con la que todavía no ha tenido contacto. Esto se realiza con base en la información histórica sobre el comportamiento de consumo de noticias de los usuarios, disponible en el campo *Record* del conjunto de datos *Behaviors.* En este proyecto aplicado, dicha estimación se denomina *Predicción del Interés*.

Para dar mayor claridad a lo mencionado en el párrafo anterior, los resultados de la Predicción del Interés se almacenan en las celdas con valor de X de la matriz Usuario-Noticia presentada en la Tabla 9. Como se observa a continuación, en la Tabla 12, los valores sombreados corresponden a la *Predicción del Interés*, que se puede obtener mediante alguno de los métodos de aprendizaje automático implementados en este proyecto. Por ejemplo, la predicción del interés del usuario U0 respecto a la noticia N1 es de 0,45.

Tabla 12. Matriz Usuario-Noticia con la *Predicción del Interés*

Usuario/Noticia	N0	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11	N12	N13	N14
U0	0	**0,45**	0	1	0	0	**0,20**	0	**0,59**	0	0	**0,09**	0	0	1
U1	**0,44**	0	0	**0,53**	0	0	0	**0,36**	1	**0,71**	1	0	**0,44**	0	**0,81**
U2	1	0	**0,63**	0	0	1	**0,60**	1	0	**0,08**	0	1	**0,53**	1	1
U3	**0,24**	0	1	0	**0,54**	1	**0,29**	1	0	0	**0,73**	**0,68**	**0,49**	1	1
U4	**0,05**	**0,38**	**0,77**	1	1	0	**0,27**	1	**0,54**	1	1	0	**0,27**	**0,35**	0
U5	0	**0,71**	0	1	0	**0,38**	1	**0,58**	0	1	0	**0,37**	0	**0,14**	0
U6	**0,80**	0	1	1	0	**0,82**	1	0	0	**0,62**	0	**0,84**	0	**0,58**	**0,72**
U7	0	0	**0,34**	0	0	1	**0,20**	0	0	1	0	1	**0,92**	0	**0,65**
U8	**0,63**	0	0	**0,70**	0	0	**0,62**	0	0	**0,81**	0	**0,38**	1	0	1
U9	0	**0,88**	0	0	1	**0,75**	1	**0,99**	0	0	1	0	**0,21**	0	**0,66**

Dado que el objetivo general de este proyecto aplicado es desarrollar un modelo para la recomendación personalizada de noticias basado en técnicas de aprendizaje automático, propósito que se puede materializar implementando modelos capaces de generar listas de recomendación de noticias, por tal razón es necesario establecer un tamaño adecuado para generar la lista personalizada de noticias a recomendar a cada usuario. Las condiciones para que una noticia sea candidata para formar parte de las listas de recomendación son: primero que las noticias sean nuevas para el usuario y segundo que el valor de la Predicción de Interés estimado sea mayor a un umbral definido.

Con base en el valor de la Predicción del Interés (un número entre 0 y 1) que cuantifica la posible respuesta que un usuario pueda tener acerca de una noticia que todavía no ha visto, se genera la lista de recomendación personalizada para cada usuario, al ordenar dichos valores de predicción de mayor a menor y hacer el corte en la cantidad de noticias para conformar la lista de recomendación personalizada de noticias.

Los métodos de aprendizaje automático implementados en este proyecto aplicado toman el mismo conjunto de datos de entrada, una tripleta Usuario-Noticia-Clic con la estructura presentada en la Tabla 8. Se realiza el entrenamiento de los modelos para predecir el interés en cada una de las noticias que cada uno de los usuarios del sistema todavía no conoce y la salida es una matriz Usuario-Noticia con la estructura presentada en la Tabla 12.

Entonces, en este proyecto aplicado se realiza la Predicción del Interés experimentando con cinco métodos de aprendizaje automático con un enfoque por filtrado colaborativo: el primero basado en memoria denominado k-Vecinos más Cercanos (KNN) y los otros cuatro basados en modelos: Normal Predictor (NP), Descomposición en Valores Singulares (SVD), Factorización No Negativa de Matrices (NMF) y Clustering por k-medias (KM). Lo anterior con el objetivo de entrenarlos y evaluarlos para escoger el método de mejor desempeño y, finalmente, desarrollar un prototipo para la generación automática de listas personalizadas de noticias de tamaño L para cada usuario.

Después de estimar el interés de cada usuario sobre cada una de las noticias que todavía no conoce, es necesario definir:

- El valor de L, un número entero que representa el tamaño de la lista personalizada, es decir, la cantidad de noticias para recomendar a cada usuario.
- El valor de U, un número decimal que representa el umbral sobre el cual se establece si una noticia será o no será de interés para un usuario.

Cada uno de los cinco métodos de aprendizaje automático implementados predice el interés de los usuarios de una forma diferente, pero parten del mismo conjunto de datos de entrada. Una vez estimada la predicción, es decir, llegando a la estructura de la matriz presentada en la Tabla 12, el procedimiento para la generación de las recomendaciones es igual y, se describe a continuación. De esta forma, es posible analizar los resultados para seleccionar el mejor método para desarrollar el prototipo del sistema de recomendación personalizada de noticias. En la Figura 5, se presenta la estructura general de la solución para la recomendación personalizada de noticias con un enfoque de filtrado colaborativo.

A continuación, se describe la estructura general de la solución, siguiendo el esquema presentado en la Figura 5. Se inicia con la explicación del procedimiento de los cinco métodos de aprendizaje automático por filtrado colaborativo basado en el usuario implementados para la predicción del interés: primero, los métodos basados en memoria con *k*-Vecinos más Cercanos (KNN); luego, los métodos basados en modelos: *Normal Predictor* (NP), Descomposición en Valores Singulares (SVD), Factorización No Negativa de Matrices (NMF) y Clustering por *k*-medias (KM).

Posteriormente, se describe la validación de los resultados empleando una técnica clásica de validación cruzada de tres partes y el método de optimización de hiperparámetros mediante “búsqueda por cuadrícula” (GridSearchCV).

Con los resultados de la Predicción del Interés validados y, después de optimizar los hiperparámetros para asegurar el mejor desempeño de los métodos de aprendizaje automático implementados, se pueden generar finalmente las listas de recomendación personalizadas de noticias de tamaño *L* para cada usuario.

La validación de los resultados de la Predicción del Interés se realiza con la raíz del error cuadrático medio (RMSE) y el error medio absoluto (MAE) y la validación de las listas de recomendación se realiza con métricas como: la Precisión, el Recall y el F1-Score. De esta forma, se tienen las bases para seleccionar el mejor método de mejor desempeño y utilizarlo para desarrollar un sistema de recomendación personalizada de noticias.

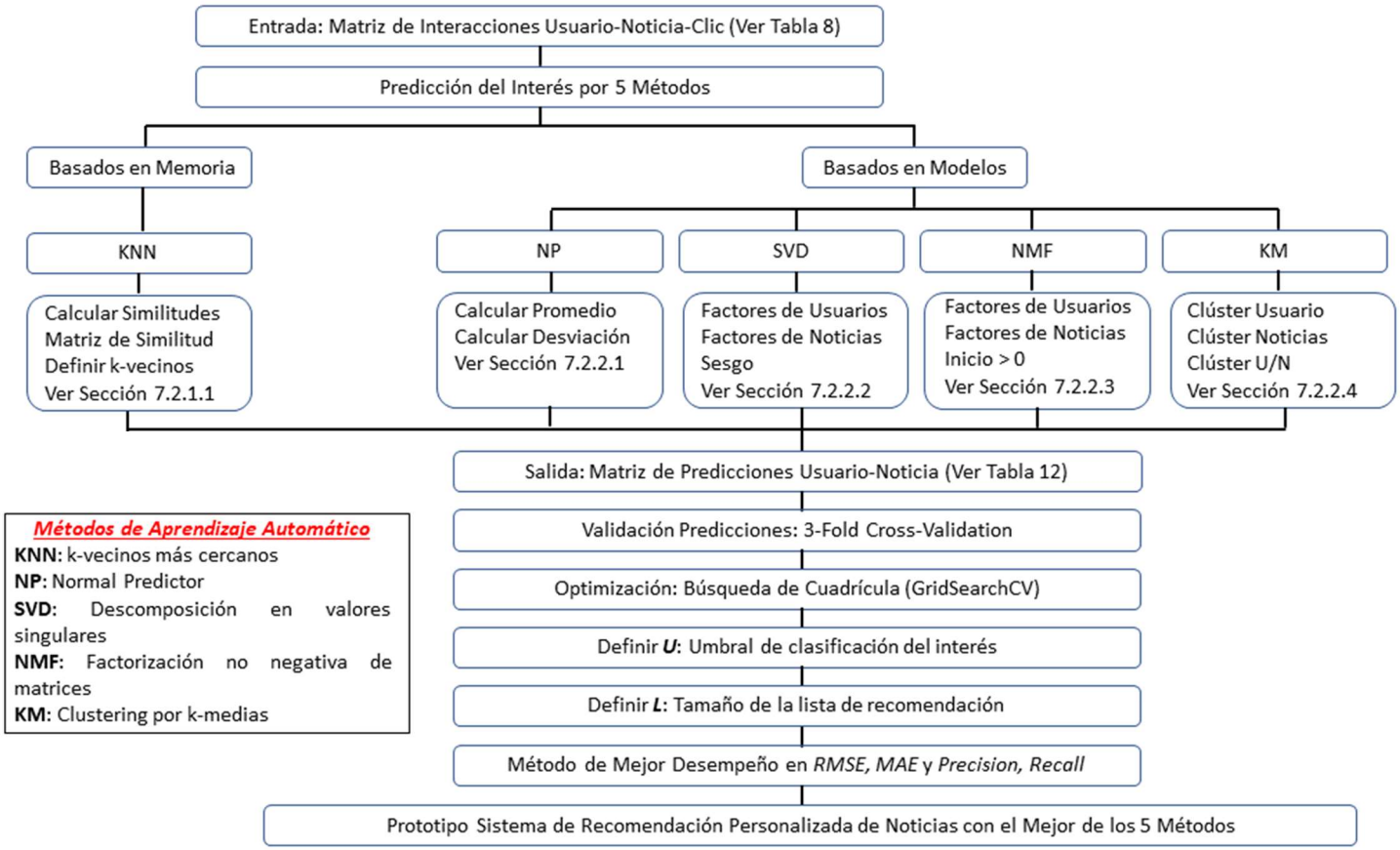

Figura 5. Estructura general de la solución para la recomendación personalizada de noticias por un enfoque de filtrado colaborativo

6.2.1. Métodos de filtrado colaborativo basados en vecinos (basados en memoria)

6.2.1.1. k Vecinos más Cercanos KNN

A partir de la matriz Usuario-Noticia, con una estructura igual a la presentada en la matriz de la Tabla 9, la implementación del método k Vecinos más Cercanos (KNN) para estimar la predicción del interés de cada usuario en cada una de las noticias que todavía no ha visto, consiste en los siguientes pasos:

- Calcular la similitud entre cada par de usuarios aplicando la Correlación de Pearson o la Desviación Media Cuadrática (MSD)
- Obtener la matriz de similitud entre todos los usuarios del sistema, se trata de una matriz simétrica Usuario-Usuario con una diagonal de 1's que representa la similitud de un usuario son sigo mismo.
- Ordenar de mayor a menor la similitud de cada usuario con respecto a los demás usuarios del sistema, se realiza iterativamente en cada columna de la matriz de similitud Usuario-Usuario.
- Definir el valor de k que representa la cantidad de vecinos más cercanos que se utilizará para estimar la predicción del interés de cada usuario con relación a cada noticia que todavía no ha visto.
- Establecer el subconjunto de noticias candidatas para cada usuario como la unión de todas las noticias de interés de sus k vecinos más cercanos.
- Realizar la predicción del interés, definido como $c_{u,n}$, de cada usuario sobre cada noticia contenida en el sistema que todavía no ha visto, esto se representa como el promedio de la intersección de las noticias de sus ***k*** vecinos más cercanos ponderado por el valor de su medida de similitud. Por ejemplo, para cualquier noticia ***n*** que el usuario objetivo ***u*** todavía no haya visto, si el número de ***k*** vecinos se define como 5, la predicción del interés del usuario ***u*** para la noticia ***n*** se puede determinar con la siguiente ecuación:

$$\hat{c}_{u,n} = \frac{\sum_{k \in V_{u,n}} sim(u,n) \cdot c_{k,n}}{\sum_{k \in V_{u,n}} sim(u,n)}$$

donde,

$\hat{c}_{u,n}$ es la predicción del interés del usuario ***u*** sobre la noticia ***n***
k representa a los vecinos cercanos definidos para el usuario ***u***

$V_{u,n}$ simboliza al conjunto de los k-vecinos más cercanos del usuario ***u*** que tuvieron contacto con la noticia ***n***
$sim(u,n)$ es la medida de similitud del usuario ***u*** con su vecino ***k***
$c_{k,n}$ representa el interés del vecino ***k*** sobre la noticia ***n,*** 1 se interesó, 0 no se interesó, n/a si no ha tenido contacto con la noticia ***n***

En la Tabla 13, se presenta un ejemplo con un número de vecinos igual a 5, donde solo 3 de ellos tuvieron contacto con la noticia ***n***. Al realizar los cálculos el resultado de la predicción de interés del usuario ***u*** sobre la noticia ***n*** es de 0,7430.

Tabla 13. Ejemplo para el cálculo de la predicción del interés por *KNN*

Vecino k	**k1**	**k2**	**k3**	**k4**	**k5**
Medida de Similitud sim(u,k)	0.8925	0.5678	0.4345	0.3941	0.3638
Ck,n	1	n/a	0	n/a	1
sim(u,k)*ck,n de los k ∈ Vu,n	0.8925	n/a	0	n/a	0.3638
∑ sim(u,k)*ck,n de los k ∈ Vu,n	12.563				
∑ sim(u,k) de los k ∈ Vu,n	16.908				
Cun	0.743				

6.2.2. Métodos de filtrado colaborativo basados en modelos

6.2.2.1. Normal Predictor NP

A partir de la matriz Usuario-Noticia, con una estructura igual a la presentada en la matriz de la Tabla 9, la implementación del método Normal Predictor *NP* para estimar la *Predicción del Interés* de cada usuario en cada una de las noticias que todavía no ha visto, consiste en los siguientes pasos:

- Calcular la media del interés de todos los usuarios del conjunto de datos de entrenamiento sobre las noticias que consultaron, esto se define como $\hat{\mu}$**.**
- Calcular la desviación con respecto a la media estimada $\hat{\mu}$ en el conjunto de datos de entrenamiento, esto se define como $\hat{\sigma}$**.**
- La predicción del interés del usuario ***u*** para la noticia ***n***, definida como $\hat{c}_{u,n}$ se obtiene a partir de una distribución normal con media $\hat{\mu}$ y varianza $\hat{\sigma}^2$.

6.2.2.2. Descomposición en Valores Singulares SVD

A partir de la matriz Usuario-Noticia, con una estructura igual a la presentada en la matriz de la Tabla 9, la implementación del método de Descomposición en Valores Singulares *SVD* para estimar la *Predicción del Interés* de cada usuario en cada una de las noticias que todavía no ha visto, consiste en:

- La Descomposición en Valores Singulares es una derivación del método de Factorización de Matrices al que se le agregan sesgos del comportamiento de los usuarios y de la percepción de las noticias.
- El filtrado colaborativo por este método SVD se basa en que el interés en los artículos está basado en un conjunto de factores latentes intrínsecos a los usuarios y a las noticias.
- Los factores latentes están ocultos y el método SVD no puede categorizarlos para conocer su naturaleza como por ejemplo las categorías de las noticias o las características demográficas de los usuarios.
- La predicción del interés del usuario ***u*** para la noticia ***n***, definida como $\hat{c}_{u,n}$ se obtiene a partir de la siguiente ecuación:

$$\hat{c}_{u,n} = \mu + b_u + b_n + q_n^T \cdot p_u$$

donde:

$\hat{c}_{u,n}$ es la predicción del interés del usuario ***u*** sobre la noticia ***n***
μ es la media del interés teniendo en cuenta a todos los usuarios y a todas las noticias del sistema
b_u es el sesgo (bias) del usuario ***u***
b_n es el sesgo (bias) de la noticia ***n***
$q_n^T \cdot p_u$ representa la interacción entre el usuario ***u*** y la noticia ***n***

- De esta forma, la predicción del interés del usuario ***u*** para la noticia ***n***, definida como $\hat{c}_{u,n}$ está determinada como la media del comportamiento de todo el conjunto de datos, más/menos un ajuste de acuerdo con el comportamiento del usuario, más/menos un ajuste de acuerdo a cómo se percibe la noticia y más/menos a la interacción entre el usuario y el ítem.
- Es importante recalcar que cuando no se aplican sesgos el método SVD es equivalente a la Factorización Probabilística de Matrices.

6.2.2.3. Factorización No Negativa de Matrices NMF

A partir de la matriz Usuario-Noticia, con una estructura igual a la presentada en la matriz de la Tabla 9, la implementación del método de Factorización No Negativa de Matrices NMF para estimar la *Predicción del Interés* de cada usuario en cada una de las noticias que todavía no ha visto, consiste en:

- Una Factorización Probabilística de Matrices en la que los factores de usuario y noticia se mantienen positivos.
- La predicción del interés del usuario $\boldsymbol{u}$ para la noticia $\boldsymbol{n}$, definida como $\hat{\boldsymbol{c}}_{\boldsymbol{u,n}}$ se obtiene a partir de la siguiente ecuación:

$$\hat{c}_{u,n} = q_n^T \cdot p_u$$

6.2.2.4. Clustering por k-medias

A partir de la matriz Usuario-Noticia, con una estructura igual a la presentada en la matriz de la Tabla 9, la implementación del método de Clustering por K-medias para estimar la *Predicción del Interés* de cada usuario en cada una de las noticias que todavía no ha visto, consiste en:

- Un método de filtrado colaborativo basado en co-agrupamiento.
- Se generen algunos agrupamientos de usuarios $\boldsymbol{G_u}$ y de noticias $\boldsymbol{G_n}$
- Se generan algunos agrupamientos $\boldsymbol{G_{un}}$
- La predicción del interés del usuario $\boldsymbol{u}$ para la noticia $\boldsymbol{n}$, definida como $\hat{\boldsymbol{c}}_{\boldsymbol{u,n}}$ se obtiene a partir de la siguiente ecuación:

$$\hat{c}_{u,n} = \overline{G_{un}} + (\mu_u - \overline{G_u}) + (\mu_n - \overline{G_n})$$

donde:

$\hat{\boldsymbol{c}}_{\boldsymbol{u,n}}$ es la predicción del interés del usuario $\boldsymbol{u}$ sobre la noticia $\boldsymbol{n}$
$\overline{\boldsymbol{G_{un}}}$ es la calificación media del agrupamiento $\boldsymbol{G_{un}}$
$\boldsymbol{\mu_u}$ es la media del interés de los usuarios
$\overline{\boldsymbol{G_u}}$ es la media del interés de los usuarios del agrupamiento $\boldsymbol{G_u}$
$\boldsymbol{\mu_n}$ es la media del interés en las noticias
$\overline{\boldsymbol{G_n}}$ es la media del interés en las noticias del agrupamiento $\boldsymbol{G_n}$

Siguiendo la estructura general de la solución presentada en la Figura 5, una vez se han estimado las predicciones por cada uno de los cinco métodos de aprendizaje automático, se procede a validar los resultados de las predicciones por un método de validación cruzada de 3 partes y a la optimización de hiperparámetros por "búsqueda de cuadrícula" o GridSearchCV.

6.3. Validación cruzada del enfoque basado en filtrado colaborativo

En este proyecto aplicado, la validación de los cinco modelos de aprendizaje automático para la recomendación personalizada de noticias por filtrado colaborativo se realiza mediante validación cruzada de tres partes (k-fold cross-validation), lo que implica dividir el conjunto de datos de la tripleta Usuario-Noticia-Clic en tres subconjuntos. Los modelos se entrenan en dos de los subconjuntos y se prueban en el subconjunto restante. Este proceso se repite tres veces, usando cada subconjunto una vez como datos de prueba. De cada repetición se obtiene una medida de desempeño, y el desempeño del modelo se calcula como el promedio de las medidas obtenidas en las tres repeticiones.

En este proyecto aplicado, específicamente para la evaluación y validación de las predicciones (es decir, para la evaluación de la diferencia entre los valores predichos del interés y el interés realmente observado), se utiliza un método de validación cruzada de tres partes, empleando como métricas de desempeño la raíz del error cuadrático medio (RMSE) y el error medio absoluto (MAE). Estas métricas se calculan para las predicciones del interés de cada usuario y, posteriormente, se promedian para obtener el desempeño de cada modelo.

6.4. Optimización de hiperparámetros de los métodos basados en filtrado colaborativo.

En el aprendizaje automático, la optimización de hiperparámetros consiste en encontrar los valores de los parámetros que generen los mejores resultados. Estos parámetros no son derivados del proceso de entrenamiento, es decir, deben especificarse manualmente en la configuración inicial. Dado que la cantidad de parámetros de cada modelo es grande, es necesario implementar algún procedimiento eficiente para probar las posibles combinaciones de parámetros y encontrar aquella que genere los mejores resultados en términos de las métricas de desempeño seleccionadas: RMSE y MAE.

En este proyecto aplicado, la optimización de parámetros se realiza a través de la "búsqueda por cuadrícula" o *GridSearch*, que consiste en seleccionar un conjunto de valores para cada uno de los hiperparámetros que se desea ajustar, para después probar todas las combinaciones posibles de valores. Todo esto se realiza a lo largo de un procedimiento de

validación cruzada de tres partes, para determinar la combinación óptima que genere los mejores resultados en los modelos, desde la perspectiva de las combinaciones definidas.

La validación de la estimación de la predicción del interés de cada usuario en cada noticia que todavía no ha visto está integrada a la optimización de hiperparámetros de los cinco modelos de aprendizaje automático implementados. En la Tabla 14, se describen las combinaciones de parámetros con las que se experimentó.

Tabla 14. Hiperparámetros en las técnicas de filtrado colaborativo

Técnica	Parámetros	Experimentos
KNN	k = (30, 40, 50); min_k = (1, 3); sim_options = (msd, pearson)	12
Normal Predictor	No tiene parámetros	1
SVD	n_factors = (80, 100, 120); n_epochs = (15, 20, 25); lr_all = (0.0025, 0.0050, 0.0075); reg_all = (0.01, 0.02, 0.03)	81
NMF	n_factors = (10, 15, 20); n_epochs = (30, 50, 70); init_low = 0, init_high = 1	9
KM	n_cltr_u = (3, 5, 7); n_cltr_i = (3, 5, 7); n_epochs = (15, 20, 25)	27

A continuación, se presentan los resultados del exhaustivo proceso de optimización de hiperparámetros con el objetivo de mejorar el desempeño de los modelos para la recomendación personalizada de noticias en cuanto a las desviaciones de las predicciones del interés y el interés realmente observado, para esto se utiliza la raíz del error cuadrático medio (*RMSE*) y el error medio absoluto (*MAE*).

La estructura del informe de resultados de optimización y validación contiene para las métricas (*RMSE* y *MAE*) el desempeño de cada uno de los 3 subconjuntos (3-*fold cross-validation*), su media y su desviación estándar, el ranking que ocupa cada combinación de hiperparámetros en el total de experimentos realizados por cada modelo, los tiempos de procesamiento de cada iteración para entrenamiento y prueba y la especificación de la combinación de parámetros de cada experimento.

En el modelo *KNN*, *k* es un número entero que representa el número (máximo) de vecinos a tener en cuenta para la agregación, *min_k* es el número mínimo de vecinos que se tienen en cuenta para realizar el proceso de agregación, si no hay suficientes vecinos, la agregación de vecinos se establece en cero y la predicción se define como la media del usuario y *sim_options* el método de similitud a emplear.

En general, si se definen valores de *k* más bajos se generan predicciones más precisas porque se calculan con usuarios muy similares al usuario objetivo, pero se tiene la desventaja de que se pueden quedar muchas noticias sin predicción, por el contrario, con valores altos de *k* se podrán generar predicciones para casi todas las noticias, pero serán menos personalizadas. En el *KNN* implementado en este proyecto aplicado se puede experimentar con *min_k* para gestionar de mejor manera esta situación. También se experimenta con dos opciones para el cálculo de la similitud entre cada par de usuarios: *msd* calcula la similitud con la diferencia cuadrática media y *pearson* calcula la similitud con el coeficiente de correlación de Pearson entre todos los pares de usuarios del sistema.

El modelo Normal Predictor *NP*, es un algoritmo que predice el interés de un usuario ante una noticia que todavía no ha visto en forma aleatoria en función de una distribución que se supone normal, con media y desviación estándar calculadas a partir del comportamiento de todos los usuarios disponibles en el sistema, por tal razón no tiene hiperparámetros a optimizar.

En el modelo de Descomposición en Valores Singulares *SVD*, *n_factors* es la cantidad de factores latentes definida, *n_epochs* representa el número de iteraciones del procedimiento de descenso de gradiente estocástico (SGD), *lr_all* es la tasa de aprendizaje para todos los parámetros y reg_all es el término de regularización para todos los parámetros.

En el modelo de Factorización No Negativa de Matrices *NMF*, al igual que en el modelo *SVD*, *n_factors* es la cantidad de factores latentes definida, *n_epochs* representa el número de iteraciones del procedimiento de descenso de gradiente estocástico (SGD), *init_low* es el límite inferior para la inicialización aleatoria de factores, debe ser mayor que cero para garantizar que los factores latentes no sean negativos, *init_high* representa al límite superior para la inicialización aleatoria de factores latentes.

En el modelo de Clustering por k-medias *KM*, *n_cltr_u* es un número entero que representa el número de clústeres de usuarios, *n_cltr_i* es un número entero que representa el número de clústeres de noticias y *n_epochs* es un número entero que representa el número de iteraciones del ciclo de optimización. El método de optimización es el Descenso de Gradiente Estocástico (SGD) con una elección específica del tamaño de paso que garantiza la no negatividad de los factores.

En la Tabla 15, se presenta el resumen de los resultados con la mejor combinación de hiperparámetros con los que se experimentó en cada una de las técnicas de recomendación por filtrado colaborativo, se incluye el promedio de las medidas de desempeño: raíz del error cuadrático medio (RMSE) y el error medio absoluto (MAE), así como los tiempos promedio empleados para entrenamiento y prueba. De esta forma, se puede asegurar los

mejores resultados posibles en la estimación de la predicción del interés y una mejor calidad en las listas de recomendación personalizadas de noticias que se puedan generar.

Tabla 15. Resumen de los mejores resultados de cada modelo

Técnica	RMSE	MAE	Tiempo (Seg.)		Parámetros
			Train	Test	
KNN	0,2519	0,1125	37	117	k=30, min_k=1, sim_options=msd
NP	0,6156	0,5100	27	5	No tiene parámetros
SVD	0,2461	0,1528	32	9	n_factors=120, n_epcohs=25, lr_all=0.0075, reg_all=0.01
NMF	0,2612	0,1667	65	8	n_factors=20, n_epcohs=70, init_low=0, init_high=1
KM por RMSE	0,2762	0,1459	34	7	n_cltr_u=7, n_cltr_i=3, n_epcohs=15
KM por MAE	0,2775	0,1438	47	7	n_cltr_u=7, n_cltr_i=5, n_epcohs=20

En el Anexo 1, se presentan los resultados de optimización de hiperparámetros (Tablas 22, 23, 24 y 25), a través de un procedimiento de validación cruzada de tres partes, para los métodos de aprendizaje automático implementados para estimar la predicción del interés, en cada una de ellas está resaltado en amarillo, la combinación de parámetros que mejores resultados genera para cada modelo, esto en términos de *RMSE* y de *MAE.*

Considerando el nivel de error en las estimaciones y el tiempo de entrenamiento empleado, la técnica de mejor desempeño es la Descomposición en Valores Singulares (*SVD*), mientras que la de peor desempeño, aunque tenga un tiempo de entrenamiento menor, es *Normal Predictor (NP)*. *SVD* presenta un mejor desempeño en los errores de las estimaciones por RMSE que *KNN, NMF* y *KM* y, además, con un tiempo de entrenamiento menor. Al considerar los errores por MAE, aunque KNN presenta mejor desempeño, su tiempo de procesamiento es notablemente más alto.

Siguiendo la estructura general de la solución presentada en la Figura 5, una vez se han estimado las predicciones por cada uno de los cinco métodos de aprendizaje automático, se han validado los resultados a través de un método de validación cruzada de tres partes y se han optimizado sus hiperparámetros para asegurar los mejores resultados de cada método de predicción, se procede a definir los valores para ***L*** y para ***U***, estas variables se definen de la misma forma a partir de la matriz de predicciones Usuario-Noticia.

- Definir el umbral ***U*** para clasificar las predicciones como: si le interesa (Clic = 1) o no le interesa (Clic = 0) la noticia al usuario, siguiendo con el ejemplo presentado en la Tabla 13, si se define el umbral en 0,7000, la noticia ***n*** se estima de interés para el usuario ***u*** porque la predicción $\hat{c}_{u,n}$ es mayor al umbral ***U***: $\hat{c}_{u,n}$ = 0,7430 >= 0,7000, entonces Clic=1.
- Definir el tamaño ***L*** de la lista de recomendación, es decir, la cantidad de noticias que se van a sugerir a cada usuario. El objetivo del sistema de recomendación de

noticias es apoyar a los usuarios en que noticias leer y presentar una selección con las noticias que, se estima, tengan alta probabilidad de ser de interés para el usuario. Entonces, el valor de ***L*** no puede ser ni muy grande ni muy pequeño para lograr generar una buena experiencia para el usuario.

- Generar la lista de recomendación personalizada para cada usuario. Para esto se ordena de mayor a menor los valores de las predicciones del interés de cada usuario y se seleccionan las ***L*** noticias con el valor de la predicción más alto.

6.5. Definición del tamaño de las listas de recomendación y del umbral para la clasificación del interés.

La salida o el resultado de cada modelo de recomendación implementado es una lista personalizada de tamaño ***L*** de noticias para cada usuario del sistema. Dicha lista de noticias se genera ordenando de mayor a menor el interés estimado (Predicción del Interés $\hat{\boldsymbol{c}}_{\boldsymbol{u,n}}$) en cada noticia con la que los usuarios todavía no han tenido contacto, por lo tanto, el valor de ***L*** afecta el desempeño y la calidad de las recomendaciones. Otro parámetro general para todos los modelos es el umbral ***U*** para predecir el interés (Clic = 1) o el no interés (Clic = 0) de un usuario hacia una determinada noticia. Se realizaron varios experimentos combinando diferentes valores para ***L*** (8, 10, 12 y 15 noticias) y para ***U*** (0.5, 0.6, 0.7 y 0.8) con el fin de observar su efecto en el desempeño de las recomendaciones.

La definición del valor de ***L*** es esencial para que el modelo genere automáticamente las listas de recomendación para cada usuario del sistema. Al igual que se validaron los resultados para la *Predicción del Interés* con las métricas RMSE y MAE***,*** es necesario validar la calidad de las listas personalizadas de recomendación de noticias para cada usuario.

6.5.1. Evaluación y validación de la calidad de las listas de recomendación

En este proyecto aplicado, al hablar de la calidad de las listas de recomendación de tamaño ***L*** se está haciendo referencia a verificar que el sistema esté generando sugerencias de noticias que sean relevantes para los usuarios.

Lo anterior, en el modelo de recomendación personalizada de noticias, se complementa en dos métricas de desempeño: la *Precisión* y el *Recall* (o recuperación). Con la *Precisión* se evalúa qué porcentaje de noticias son relevantes para un usuario con respecto al total de noticias que se le está recomendando y con *Recall* se verifica qué porcentaje de noticias el modelo está recomendando con respecto al número de noticias relevantes para un usuario

específico. Estas métricas se calculan para cada uno de los usuarios y al promediarlas se obtiene el desempeño del modelo.

En este contexto, el término *relevante* hace referencia a la clasificación automática del interés en una noticia a partir del valor de la *Predicción del Interés* $\hat{c}_{u,n}$ que se obtiene por cada uno de los cinco métodos de aprendizaje automático implementados. Para esto es necesario definir un umbral ***U*** de clasificación. Entonces, si la *Predicción del Interés* $\hat{c}_{u,n}$ es mayor o igual al umbral ***U,*** se clasifica como ***Clic=1***, o sea, noticia relevante o de interés para el usuario o, en caso contrario, ***Clic=0***, es decir, noticia no relevante o de no interés para el usuario.

La aplicación de estas dos métricas es muy importante para evaluar la calidad de las listas de recomendación, y dado que, en la práctica, *Precisión* y *Recall* pueden ser medidas contradictorias porque la inclusión de menos noticias en las listas de recomendación para aumentar la *Precisión* puede que genere como resultado una menor Recuperación (*Recall*). Entonces, para balancear o equilibrar las dos métricas, se calcula la F1-Score que es la media armónica de *Precisión* y *Recall*.

Teniendo en cuenta las anteriores consideraciones, en el Anexo 2 (Tablas 26, 27, 28, 29 y 30), se presentan los resultados de la experimentación para definir los mejores valores de ***L*** (tamaño de la lista personalizada de noticias a recomendar) y ***U*** (umbral de clasificación Clic=1 y Clic=0) que mejor se adecúen a los objetivos de este proyecto aplicado. Para evaluar el impacto de ***L*** y ***U*** en las métricas de calidad de las listas personalizadas de recomendación de noticias, se experimenta con veinte combinaciones de tamaños de lista y umbrales para cada método: ***L*** = 8, 9, 10 y 12 noticias y ***U*** = 0.5, 0.6, 0.7, 0.8 y 0.9.

De acuerdo con los resultados de los experimentos (Ver Anexo 2 Tablas 26, 27, 28, 29 y 30), en general se observa que para valores de ***L*** más grandes se mejora el *F1-Score* de las listas de recomendación generadas. Por el contrario, a medida que aumenta el valor de ***U***, la calidad de las listas de recomendación disminuye. Es importante analizar el efecto en la *Precisión* y el *Recall* ante los cambios en ***L*** y ***U***.

De acuerdo con los 20 experimentos realizados con cada uno de los cinco métodos implementados, se observa lo siguiente:

- La *Precisión* se mantiene casi constante para todos los valores de ***U*** explorados, al variar ***L*** entre 8, 10, 12 y 15 noticias.
- El *Recall* es sensible a los cambios de ***L***, se observa una mejora significativa en esta métrica al aumentar el tamaño ***L*** de la lista de noticias.
- Los incrementos en el valor de ***U*** afectan negativamente las métricas de calidad de las listas de recomendación, más notablemente a *Recall* que a la *Precisión*.

- El comportamiento descrito en los tres puntos anteriores se repite en los cinco métodos de aprendizaje automático implementados.

Para dar mayor claridad a lo expuesto acerca del impacto de ***L*** (tamaño de la lista de recomendación) y ***U*** (umbral para definir Clic = 1 o Clic = 0) en la calidad de las métricas de evaluación de las listas de recomendación de noticias en los cinco métodos de aprendizaje automático implementados en el presente proyecto aplicado, en las tablas de resultados (Ver Anexo 2) se sombrea en color verde la combinación de ***L*** y ***U*** con mejor desempeño y en color rojo la de peor desempeño, esto según el resultado de la F1-Score de cada método.

En todos los métodos de aprendizaje automático, la combinación de ***L*** y ***U*** con mejor desempeño corresponde al mayor valor ***L*** con el menor valor de ***U***, mientras que el peor desempeño se da con el menor valor ***L*** y con el mayor valor de ***U***. El método de mejor desempeño, con un *F1-Score* de 0.8537, es *SVD*, seguido muy de cerca por *KNN* con 0.8403, *NMF* con 0.8351 *y KM* con 0.8300. La técnica básica *Normal Predictor (NP)* se aleja considerablemente de estos buenos resultados, obteniendo un *F1-Score* de 0.4949.

Dado que, uno de los objetivos de los sistemas de recomendación es apoyar a los usuarios en la toma de decisiones con respecto a qué noticias leer y generar una buena experiencia, el tamaño ***L*** de la lista de recomendación no debe ser ni demasiado grande ni demasiado pequeño, ya que se busca que el usuario pueda evaluar rápidamente una oferta diversa de noticias. Por esta razón, se selecciona *L* = 12 noticias. Por otro lado, para evaluar la calidad de las recomendaciones, se utiliza un umbral ***U*** medio para definir si las noticias de la lista son de interés (Clic = 1) o no (Clic = 0) para los usuarios.

Entonces, la combinación de ***L*** y ***U*** que mejor se adapta a los objetivos del modelo de recomendación en este proyecto aplicado es ***L*** = 12 y ***U*** = 0.7. Con estos valores, la mejor calidad de las listas de recomendación, con un F1-Score de 0.8118, se observa en la técnica *SVD*, seguida de cerca por *KNN* con 0.8049, *NMF* con 0.7832 *y KM* con 0.7874. El resultado de *Normal Predictor (NP)* es considerablemente inferior a los otros métodos implementados, con tan solo 0.3540. En la Tabla 16, se presenta el resumen de los resultados de la evaluación de la calidad de las listas de recomendación de tamaño ***L*** = 12 noticias y umbral ***U*** = 0.7 generadas por cada uno de los métodos implementados.

Tabla 16. *Evaluación de las listas de recomendación para L* = 12 y *U* = 0.7

Técnica	Precision	Recall	F1-Score
KNN	0,9306	0,7091	0,8049
Normal Predictor	0,4494	0,2920	0,3540
SVD	0,9308	0,7197	0,8118
NMF	0,9194	0,6822	0,7832
KM	0,9076	0,6953	0,7874

Teniendo en cuenta que el objetivo general de este proyecto aplicado es generar listas de recomendación, en la medida de lo posible, completas y para todos los usuarios, después de evaluar: la calidad de las estimaciones con la raíz del error cuadrático medio (*RMSE*) y con el error medio absoluto (*MAE*) y la calidad de las listas de recomendación generadas con *Precisión, Recall* y *F1-Score*, es muy importante analizar el porcentaje de usuarios a los cuales los modelos les generan listas de recomendación completas, parciales o no es posible generar ninguna recomendación.

Una de las principales razones por las que los métodos de aprendizaje automático no puedan generar recomendaciones es una baja interacción con el conjunto de noticias, esto se presenta con los usuarios nuevos que tienen poco historial en el conjunto de datos.

En la Tabla 17, se resume la capacidad de cada método para generar las listas personalizadas de recomendación de noticias. De esta forma, en la columna ***L'*** se registra la cantidad de noticias que se generan para cada usuario. Por ejemplo, ***L'*** = 12 indica que el modelo está en la capacidad de generar las listas con la cantidad esperada de 12 noticias, ***L'*** = 9 indica la cantidad de usuarios a los que el modelo solo les genera listas de recomendación de 9 noticias de las 12 que son el objetivo de tamaño de lista a recomendar, ***L'*** = 0 indica la cantidad de usuarios a los que el modelo no le recomienda ninguna noticia.

Tabla 17. *Tamaños de las listas de recomendación generadas*

L'	KNN	%KNN	NP	%NP	SVD	%SVD	NMF	%NMF	KM	%KM
12	10.649	50,5%	10.622	50,3%	10.691	50,7%	10.641	50,4%	10.677	50,6%
11	599	2,8%	546	2,6%	522	2,5%	554	2,6%	593	2,8%
10	592	2,8%	616	2,9%	593	2,8%	615	2,9%	625	3,0%
9	654	3,1%	729	3,5%	683	3,2%	683	3,2%	657	3,1%
8	751	3,6%	774	3,7%	743	3,5%	740	3,5%	731	3,5%
7	827	3,9%	816	3,9%	877	4,2%	838	4,0%	799	3,8%
6	981	4,6%	958	4,5%	957	4,5%	980	4,6%	978	4,6%
5	1.127	5,3%	1.092	5,2%	1.063	5,0%	1.130	5,4%	1.132	5,4%
4	1.229	5,8%	1.267	6,0%	1.240	5,9%	1.157	5,5%	1.190	5,6%
3	1.229	5,8%	1.296	6,1%	1.257	6,0%	1.295	6,1%	1.214	5,8%
2	1.197	5,7%	1.134	5,4%	1.226	5,8%	1.197	5,7%	1.257	6,0%
1	902	4,3%	885	4,2%	866	4,1%	930	4,4%	893	4,2%
0	369	1,7%	371	1,8%	388	1,8%	346	1,6%	360	1,7%

De acuerdo con los resultados, se observa que los 5 métodos de aprendizaje automático pueden generar listas de recomendación completas, esto es ***L*** = 12 noticias, para aproximadamente la mitad de los 21.106 usuarios estudiados. La personalización para la generación de las listas de recomendación lograda es muy buena, ya que para más del 75% de los usuarios los métodos de aprendizaje automático generan recomendaciones, por lo menos 4 noticias, con una alta probabilidad de que sean de su interés. La cantidad de usuarios para los cuales los métodos no les pueden generar ninguna recomendación personalizada es muy baja, alrededor del 1,7%, equivalente a menos de 360 usuarios.

Finalmente, en las Figuras 6, 7, 8, 9 y 10, se presenta un ejemplo de la salida como resultado de las listas de recomendación de tamaño ***L*** = 12 noticias para ***U*** = 0.7 por cada uno de los métodos implementados, esto para el usuario de ejemplo *U62541*.

```
Recomendaciones para el usuario  U62541:
Noticia:  N56469, Indice Recomendación:  1
Noticia:  N29510, Indice Recomendación:  1
Noticia:  N22816, Indice Recomendación:  1
Noticia:  N306, Indice Recomendación:  1
Noticia:  N35710, Indice Recomendación:  1
Noticia:  N32928, Indice Recomendación:  1
Noticia:  N3491, Indice Recomendación:  1
Noticia:  N28313, Indice Recomendación:  1
Noticia:  N2142, Indice Recomendación:  1
Noticia:  N22816, Indice Recomendación:  1
Noticia:  N35710, Indice Recomendación:  1
Noticia:  N28313, Indice Recomendación:  1
```

Figura 6. *Lista de recomendación* ***L*** *= 12 noticias para el usuario U62541 por KNN*

```
Recomendaciones para el usuario  U62541:
Noticia:  N45794, Indice Recomendación:  1
Noticia:  N56117, Indice Recomendación:  1
Noticia:  N39122, Indice Recomendación:  1
Noticia:  N56469, Indice Recomendación:  1
Noticia:  N61210, Indice Recomendación:  1
Noticia:  N43654, Indice Recomendación:  1
Noticia:  N12353, Indice Recomendación:  0.994103746621823
Noticia:  N35710, Indice Recomendación:  0.70429488612568
Noticia:  N47020, Indice Recomendación:  0.6892559142505513
Noticia:  N19315, Indice Recomendación:  0.5590259903743884
Noticia:  N29732, Indice Recomendación:  0.5529304659999943
Noticia:  N40057, Indice Recomendación:  0.5469764420383523
```

Figura 7. *Lista de recomendación* ***L*** *= 12 noticias para el usuario U62541 por NP*

```
Recomendaciones para el usuario  U62541:
Noticia:  N39677, Indice Recomendación:  1
Noticia:  N8834, Indice Recomendación:  1
Noticia:  N4866, Indice Recomendación:  0.9637743637383992
Noticia:  N28313, Indice Recomendación:  0.9501511846848265
Noticia:  N28313, Indice Recomendación:  0.9501511846848265
Noticia:  N26682, Indice Recomendación:  0.9493778862206851
Noticia:  N35710, Indice Recomendación:  0.9312048825840716
Noticia:  N35710, Indice Recomendación:  0.9312048825840716
Noticia:  N47020, Indice Recomendación:  0.8966152326266191
Noticia:  N6225, Indice Recomendación:  0.7928710376971837
Noticia:  N28290, Indice Recomendación:  0.5690043169406583
Noticia:  N41322, Indice Recomendación:  0.35896652715552746
```

Figura 8. *Lista de recomendación* ***L*** *= 12 noticias para el usuario U62541 por SVD*

```
Recomendaciones para el usuario  U62541:
Noticia:  N22816, Indice Recomendación:  0.9159591127831531
Noticia:  N45794, Indice Recomendación:  0.9136471602538377
Noticia:  N45794, Indice Recomendación:  0.9136471602538377
Noticia:  N33096, Indice Recomendación:  0.9088919303811468
Noticia:  N306, Indice Recomendación:  0.9087279850765986
Noticia:  N35710, Indice Recomendación:  0.9052942040313824
Noticia:  N55556, Indice Recomendación:  0.9020732173246138
Noticia:  N56469, Indice Recomendación:  0.8989540746888132
Noticia:  N56469, Indice Recomendación:  0.8989540746888132
Noticia:  N18285, Indice Recomendación:  0.890923250055295
Noticia:  N18285, Indice Recomendación:  0.890923250055295
Noticia:  N39677, Indice Recomendación:  0.836642588550395
```

Figura 9. *Lista de recomendación* ***L*** *= 12 noticias para el usuario U62541 por NMF*

```
Recomendaciones para el usuario  U62541:
Noticia:  N47020, Indice Recomendación:  1
Noticia:  N28313, Indice Recomendación:  1
Noticia:  N35710, Indice Recomendación:  1
Noticia:  N47020, Indice Recomendación:  1
Noticia:  N9803, Indice Recomendación:  1
Noticia:  N3491, Indice Recomendación:  1
Noticia:  N26682, Indice Recomendación:  0.9868463294686164
Noticia:  N39677, Indice Recomendación:  0.9737019995717092
Noticia:  N39677, Indice Recomendación:  0.9737019995717092
Noticia:  N29510, Indice Recomendación:  0.972396778688175
Noticia:  N29510, Indice Recomendación:  0.972396778688175
Noticia:  N14001, Indice Recomendación:  0.5204001127792562
```

Figura 10. *Lista de recomendación* ***L*** *= 12 noticias para el usuario U62541 por KM*

6.6. Aspectos de la implementación

El entorno de implementación de los modelos para la recomendación personalizada de noticias fue Python en Google Colab. Para evaluar los tiempos de carga, preparación, procesamiento de los datos, entrenamiento de los modelos se probaron varias configuraciones de hardware disponibles tanto en la versión gratuita como en la versión Google Colab Pro. En total, se probaron siete configuraciones de implementación, las cuales se presentan en la Tabla 18, ordenadas de la de mayor a la de menor capacidad computacional:

Tabla 18. Configuraciones del entorno de implementación de Python en Google Colab y Google Colab Pro

N	Configuración
1	TPU-v2-8
2	T4 GPU alta disponibilidad de RAM
3	Procesador gráfico T4
4	GPU L4
5	GPU A100
6	CPU alta disponibilidad de RAM
7	CPU

Por la mayor eficiencia en los tiempos de procesamiento, entrenamiento y validación de los métodos de aprendizaje automático implementados, se trabajó con la configuración TPU – v2-8 de Google Colab Pro. Esta es la configuración que genera los resultados en menor tiempo.

Para el caso del modelo basado en contenido se utilizó la librería *SKLEARN*, específicamente la transformación de los datos referentes a la codificación de la Categoría y de la Subcategoría de las noticias se realizó con *OneHotEncoder* y la representación vectorial del contenido textual del título con *TfidfVectorizer*. El entrenamiento del modelo de recomendación basado en contenido se hizo con el algoritmo *KNeighborsClassifier*. La validación de resultados con una partición del conjunto de datos de entrada en 80% para entrenamiento y 20% para prueba, se realizó con *train_test_split*. Las métricas de desempeño se obtuvieron con *classification_report* y *accuracy_score*. Finalmente, la automatización de las recomendaciones se realizó con un *Pipeline* para integrar todo el procedimiento y generar las listas de recomendación personalizadas de noticias por un enfoque basado en contenido.

Para el caso del modelo de recomendación por filtrado colaborativo, la etapa de preparación y de tratamiento de datos se trabajó con las librerías *Pandas* y *Numpy*. De esta forma, se logró obtener la tripleta de datos Usuario-Noticia-Clic con la estructura apta para entrenar los cinco métodos de aprendizaje automático. Los métodos k Vecinos más Cercanos *KNN* y Descomposición en Valores Singulares *SVD* se implementaron, entrenaron y validaron de forma “manual”, es decir, sin utilizar ninguna librería específica de Python para la recomendación personalizada, esto es con las librerías generales Pandas, *Numpy* y *Math.*

Con el objetivo de mejorar la eficiencia, la gestión de los resultados y la generación automática de las listas de recomendación de noticias, se investigó algunas librerías de uso específico para propósitos de recomendación personalizada. La mejor opción fue la librería *SURPRISE* (*Simple Python Recommendation System Engine*) es una librería específica para crear y analizar sistemas de recomendación personalizado por filtrado colaborativo. Se decidió trabajar con esta herramienta porque proporciona un muy buen control sobre los experimentos planteados, cuenta con una excelente documentación de apoyo y ejemplos claros y variados para conocer todas sus funcionalidades. Tiene a disposición varios algoritmos de aprendizaje automático para la predicción que son fáciles de usar. Además, cuenta con varias herramientas para evaluar y comparar fácilmente el desempeño de los algoritmos.

En *SURPRISE*, los cinco métodos de aprendizaje automático implementados se corresponden de la siguiente forma:

- k Vecinos más Cercanos *KNNBasic*
- Predicción Basada en una Distribución Normal *NormalPredictor*
- Descomposición en Valores Singulares *SVD*
- Factorización No Negativa de Matrices *NMF*
- Clustering por k-medias *CoClustering*

También se utilizó *Dataset* para la configuración y el alistamiento de los datos en la estructura necesaria para el procesamiento, *Reader* para el cargue de los mismos, *accuracy* para obtener las métricas definidas para evaluar y validar el desempeño en los errores de las predicciones (RMSE y MAE), *KFold* para implementar la validación cruzada de tres partes y *GridSearchCV* para la optimización de los hiperparámetros disponibles en cada modelo con el objetivo de mejorar su desempeño y lograr mejores resultados. Además, se implementaron dos funciones, una para mapear las predicciones y otra para la generación automática de las listas de recomendación personalizada de noticias para cada usuario, esto validando la calidad de las recomendaciones utilizando las métricas de *Precisión*, *Recall* y *F1-Score*.

7. ANALISIS DE LOS RESULTADOS

En este proyecto aplicado de ciencia de datos, se utilizaron los dos enfoques básicos para la recomendación personalizada: el filtrado basado en contenido y el filtrado colaborativo. A su vez, los métodos del enfoque por filtrado colaborativo implementados se dividen en dos grupos: los métodos basados en vecinos (o también basados en memoria) y los métodos basados en modelos.

Para realizar un correcto análisis de los resultados obtenidos, es muy importante tener en mente el tamaño del conjunto de datos utilizado, que es exactamente el mismo en todos los modelos implementados. Los datasets utilizados son: *NEWS* con 50.537 noticias diferentes y *BEHAVIORS* que después de transformado contiene 21.106 usuarios únicos.

En el caso del enfoque basado en contenido se creó una matriz Usuario-Característica de 153.726 usuarios no únicos (los usuarios se repiten de acuerdo a su historial de consumo de noticias) por 1.281 características únicas, 281 por las Categorías y Subcategorías únicas y 1.000 características únicas por el vocabulario de palabras identificadas en los títulos de las noticias.

En el caso del enfoque de recomendación por filtrado colaborativo se creó una matriz Usuario-Noticia de 21.106 usuarios únicos (filas) por 50.537 noticias únicas (columnas), es decir, una matriz dispersa con más mil millones de valores de los cuales solamente cerca de dos millones contienen un valor (los demás están vacíos y corresponden al objetivo de predicción de los modelos), esto es equivalente a que solo el 0,19% de los datos tiene registrado el interés o no interés de un usuario en una noticia indicado por un cero o un uno (Clic=1 o Clic=0). De esta forma, se puede apreciar la escasez de los datos y, la capacidad que tienen los modelos de filtrado colaborativo para generar buenas recomendaciones a partir de una matriz tan dispersa. Para el filtrado colaborativo el dataset *NEWS* solamente se utiliza para la presentación de la lista de recomendación, vinculando con el *Id_News* su *Categoría, Subcategoría* y su *Título*.

7.1. Análisis de los resultados del modelo basado en contenido

El método implementando para generar la recomendación personalizada de noticias por el enfoque basado en contenido es coherente con lo que se busca en este contexto, esto es identificar los elementos característicos comunes de las noticias que han sido de interés para un usuario determinado, de tal forma que se le puedan recomendar otras noticias que compartan dichos elementos característicos. Del conjunto de datos de noticias utilizado, se

logró extraer los rasgos comunes de las noticias desde la *Categoría* y la *Subcategoría* y representando vectorialmente el contenido textual del *Título* de las noticias.

La idea principal del enfoque de recomendación basado en contenido es que las noticias contengan una cantidad significativa de rasgos característicos que permitan identificar su naturaleza a través de un vector de características estandarizado. Aunque, con los datos disponibles se logró la representación vectorial de las características de las noticias y se logró implementar un modelo que incluye un perfil de usuario que, de igual forma, describe los rasgos característicos de las noticias que le interesan, al momento de evaluar y validar los resultados de las listas de recomendación los resultados no fueron los esperados.

Como se observa en la Figura 4, las listas de recomendación generadas son iguales para muchos de los usuarios, esto indica que la personalización lograda por el enfoque de recomendación basado en contenido no es adecuada para los objetivos de este proyecto aplicado. La principal razón de estos resultados no satisfactorios es que la disponibilidad de rasgos característicos de las noticias es, desde los métodos con los que se experimentó, muy pobre, es decir, los rasgos característicos seleccionados no son suficientes para definir la naturaleza de las noticias y, por lo tanto, el modelo no está en la capacidad de diferenciar entre todas las noticias y todos los elementos son muy parecidos. Básicamente, se dispone de una sola variable (la *Subcategoría*) para representar el vector de características porque dicha variable está muy correlacionada con la *Categoría*. La diferenciación de las noticias en función de las palabras que contenga tampoco genera buenos resultados al momento de identificar su naturaleza y de vincularla al perfil del usuario.

Aunque para clasificar las noticias, entre de interés (*Clicked=1*) y de no interés (*Clicked=0*) para los usuarios, el modelo implementado de filtrado por contenido para la recomendación personalizada de noticias es muy eficiente y preciso, tiene otro gran problema que es el tiempo necesario para generar una lista de recomendación para un usuario, es inaceptablemente largo, superando los sesenta minutos para generar una sola lista de recomendación.

Por los aspectos negativos mencionados anteriormente: primero, la muy baja personalización en las listas de recomendación generadas y, segundo, la ineficiencia en los tiempos de procesamiento, se decidió no utilizar el enfoque basado en contenido para el desarrollo del prototipo del sistema de recomendación personalizada de noticias. Estos problemas se pueden solucionar y gestionar de mejor manera con el enfoque de recomendación basado en filtrado colaborativo.

7.2. Análisis de los resultados de los métodos por filtrado colaborativo

El enfoque para la recomendación personalizada de noticias por filtrado colaborativo utilizado en este proyecto aplicado supera algunos de los inconvenientes del enfoque basado en contenido. Después de implementar todos los métodos y analizar los resultados generados se observa que, en el contexto de la recomendación de noticias de este proyecto aplicado, el filtrado colaborativo genera mejores resultados y es computacionalmente más eficiente que el filtrado basado en contenido, lo anterior básicamente porque este último se basa en el historial de consumo de noticias de un usuario en relación con el comportamiento del consumo de noticias de los otros usuarios del sistema, entonces no es necesario encontrar características diferenciadoras en las noticias, lo cual como se mencionó, no es tan fácil desde los datos disponibles. La idea central de la recomendación por filtrado colaborativo es que el interés de un usuario con respecto a una noticia es probable que sea similar al de otro usuario, si los dos usuarios comparten noticias de interés.

En este proyecto aplicado se implementaron cinco métodos de aprendizaje automático para la recomendación personalizada de noticias bajo un enfoque de filtrado colaborativo, estos métodos se dividen de dos formas: basados en memoria con k Vecinos más Cercanos (*KNN*) y los otros, basados en modelos con Normal Predictor (*NP*), Descomposición en Valores Singulares (*SVD*), Factorización No Negativa de Matrices (*NMF*) y Clustering por k-medias (*KM*). A continuación, se analizan los resultados obtenidos después de realizar un adecuado entrenamiento y un minucioso proceso de validación.

En general, los métodos de recomendación implementados, a excepción del método Normal Predictor (NP), logran buenos resultados de personalización en las sugerencias, esto determinado a través de las métricas de evaluación de las desviaciones entre la *Predicción del interés* y el interés realmente observado en los usuarios de prueba (*RMSE* y *MAE*); y también a través de las métricas para la evaluación de la calidad de las listas de recomendación para verificar que las sugerencias sean relevantes para los usuarios (*Precisión, Recall* y *F1-Score*).

Como se presentó en el capítulo de modelado, específicamente en el esquema de la Figura 5 (Estructura general de la solución para la recomendación personalizada de noticias por un enfoque de filtrado colaborativo), los modelos se entrenaron tomando el mismo conjunto de datos de entrada con el objetivo de que los resultados y tiempos de procesamiento sean comparables. De esta forma, después de realizar la validación de los resultados de la *Predicción del Interés*, de optimizar los hiperparámetros disponibles en cada modelo y de evaluar y de analizar el tamaño *L* de la lista personalizada de noticias a recomendar y el umbral *U* de clasificación entre Clic=1 y Clic=0. Es momento de seleccionar el mejor modelo

integrando las métricas de calidad de las predicciones con las métricas de calidad de las listas de recomendación y también con los tiempos de procesamiento de cada modelo.

Para seleccionar el mejor modelo de los cinco implementados en la Tabla 19, se presenta el resumen de los resultados integrados, incluyendo las métricas de calidad de las predicciones y las métricas de calidad de las listas de recomendación, esto junto con los tiempos de procesamiento necesario para el entrenamiento y prueba de cada modelo. Los resultados presentados a continuación están basados en un tamaño de lista de recomendación ***L*** = 12 noticias y un umbral de clasificación ***U*** = 0.7.

Tabla 19. Comparación de los resultados para los 5 modelos con ***L***=12 y ***U***=0.7

Técnica	Calidad Predicciones		Calidad Recomendaciones			Tiempo (Seg.)	
	RMSE	MAE	Precisión	Recall	F1-Score	Train	Test
KNN	0,2519	0,1125	0,9306	0,7091	0,8049	37	117
NP	0,6156	0,5100	0,4494	0,2920	0,3540	27	5
SVD	0,2461	0,1528	0,9308	0,7197	0,8118	32	9
NMF	0,2612	0,1667	0,9194	0,6822	0,7832	65	8
KM	0,2762	0,1459	0,9076	0,6953	0,7874	34	7

Aunque otros valores ***L*** de tamaño de lista de recomendación y de umbral ***U*** de clasificación pueden generar mejores resultados de listas de recomendación, en este proyecto aplicado se considera que 12 noticias es un tamaño deseable por el usuario, listas más pequeñas pueden mejorar la *Precisión*, pero dejan menos opciones de noticias al usuario, es decir, se afecta negativamente la Recuperación (*Recall*).

En este proyecto aplicado se considera que la calidad de las listas de recomendación generadas por los modelos es el factor fundamental para decidir cuál es el de mejor desempeño, seguido de la calidad de las predicciones y de los tiempos de procesamiento de los datos. De esta forma, el mejor método, de los cinco analizados, para la recomendación personalizada de noticias es Descomposición en Valores Singulares (SVD) con la mejor calidad de las listas de recomendación por el F1-Score más alto de 0,8118, la mejor calidad de las predicciones por el RMSE más bajo de 0,2461 y el segundo menor tiempo de entrenamiento con 32 segundos. Con relación al tiempo de entrenamiento el mejor modelo es Normal Predictor (NP) pero no se lo considera para el desarrollo del prototipo del sistema de recomendación porque tiene unos indicadores de calidad de predicciones y de calidad de listas de recomendación inaceptables para los objetivos de este proyecto aplicado.

8. SISTEMA DE RECOMENDACIÓN

8.1. *Selección del modelo para el desarrollo del prototipo*

Después de realizar la adecuada preparación y limpieza de los datos, de tal forma que se adapten a los requerimientos de entrada de las 5 técnicas de aprendizaje automático implementadas para la recomendación personalizada de noticias por filtrado colaborativo. Pasando por las fases de entrenamiento, evaluación y validación de los resultados obtenidos con métricas enfocadas en la evaluación de las estimaciones (*RMSE* y *MAE*) y métricas enfocadas en la calidad de las listas de recomendación (*Precisión, Recall* y *F1-Score*). Además, se aplicó un procedimiento para la optimización de los parámetros de cada técnica para mejorar su desempeño. Se evidenció que la mejor técnica para alcanzar el objetivo general de este proyecto aplicado, que es generar recomendación personalizada de noticias, es la Descomposición en Valores Singulares (*SVD*), con esta técnica de aprendizaje automático se logran los menores errores en las estimaciones, la mejor calidad en las listas de noticias recomendadas y el menor consumo de recursos computacionales y de tiempos de procesamiento.

Por las razones mencionadas, se desarrolló el prototipo del sistema de recomendación personalizada de noticias basado en el algoritmo de Descomposición en Valores Singulares (*SVD*). A continuación, se describe el prototipo desarrollado y los resultados logrados.

8.2. *Desarrollo del prototipo del sistema de recomendación*

El desarrollo del prototipo se llevó a cabo con el objetivo de mostrar los resultados de recomendación de noticias que empleará el modelo de Descomposición en Valores Singulares (SVD) para generar recomendaciones personalizadas. Este enfoque permite a los usuarios explorar contenido relevante según sus interacciones previas con las noticias. A continuación, se detallan las tecnologías utilizadas, así como los casos de uso correspondientes a la funcionalidad principal del prototipo.

- *Python:* Se utilizó ***Python*** como lenguaje de programación principal para implementar el modelo de recomendación. La biblioteca *Scikit-SurPRISE* facilitó la creación y entrenamiento del modelo *SVD*, permitiendo manejar datos de usuario y noticias de manera eficiente.
- *FastAPI*: Esta herramienta se eligió para construir la API que sirve como intermediario entre el *frontend* y el modelo de recomendación. *FastAPI* permite

manejar solicitudes HTTP de manera rápida y sencilla, además de ofrecer documentación automática de la API.

- *PHP*: El *backend* se desarrolló a través del lenguaje de programación *PHP*, el cual recibe la petición del *frontend* con el identificador del usuario y hace el llamado a la API a través de *curl*. Una vez obtiene la información relacionada al usuario, retorna los datos en formato JSON al *frontend* para su visualización.
- *XAMPP*: Se emplea la herramienta *XAMPP* como servidor de *Apache* local para desplegar el *backend* de *PHP*.
- *JavaScript*: Se utilizó *JavaScript* para implementar la lógica en el *frontend*. Este lenguaje permite una interacción dinámica con la interfaz de usuario, facilitando la selección del ID del usuario y la recuperación de las recomendaciones de noticias.
- *HTML* y *CSS*: Se utilizaron *HTML* y *CSS* para crear la interfaz de usuario, asegurando que sea intuitiva y atractiva visualmente. Se implementaron estilos que imitan la apariencia de plataformas modernas, mejorando la experiencia de navegación.
- *Ngrok*: Para exponer el servidor de desarrollo de *FastAPI* a Internet, se utilizó *Ngrok*. Esto permite a los usuarios acceder al prototipo en línea de manera sencilla y segura durante las fases de prueba.

En la Figura 11, se describe la arquitectura del prototipo. En ella se puede observar de manera gráfica cómo se encuentran interconectados los componentes principales del sistema de recomendación personalizada de noticias. Se muestra una visión general acerca de la comunicación entre los usuarios, la API y el acceso al modelo entrenado de SVD, desarrollado en el lenguaje de programación Python.

El modelo inicia con el cliente, quien hace uso del *frontend* construido con *HTML* y *JavaScript*. El cliente accede al prototipo a través de un navegador web. Es necesario aclarar que el servidor que aloja el aplicativo, en este caso, es un servidor local. Para el recomendador de noticias, este servidor es también local.

En el *backend* encontramos la API, que actúa como una interfaz que comunica las solicitudes del cliente con el modelo. En esta API se gestionan las solicitudes realizadas por el cliente mediante el navegador web y se envían las respuestas de acuerdo con cada solicitud. Esta API implementa dos funcionalidades principales:

- Obtener identificadores válidos de usuarios en el conjunto de datos, los cuales se despliegan en la interfaz principal del usuario.
- Consultar las noticias recomendadas en el momento en que el usuario elige un identificador.

Esta consulta retorna un listado de noticias. Para obtener el listado, se utiliza el recomendador de noticias que implementa el modelo *SVD* desarrollado en *Python*. La

respuesta se envía en formato *JSON*, lo que posteriormente permite visualizar el listado de noticias recomendadas en un navegador web.

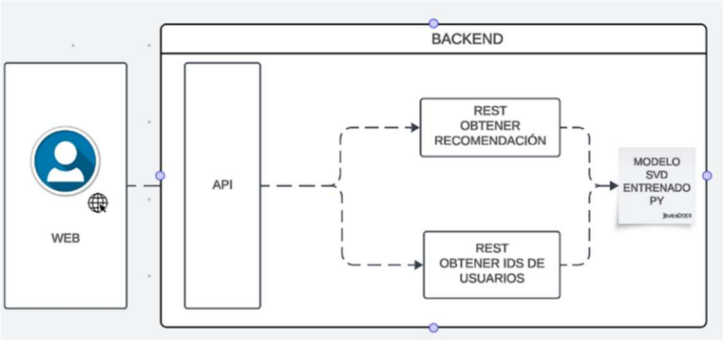

Figura 11. *Arquitectura del prototipo*

En la Figura 12 se puede observar, en orden cronológico, cómo suceden las acciones desde el momento en que el usuario ingresa a la plataforma web. El primer paso ocurre cuando el usuario accede al recomendador de noticias por medio de un navegador web, conectándose al servidor local.

El primer paso es la solicitud del *frontend* a la API para obtener el listado de identificadores de usuarios válidos en el *dataset*. Esta petición *HTTP* viaja por medio de la API al *backend* y, a su vez, se consulta en *Python* para retornar el listado alojado en el *dataset*. Una vez se presentan los identificadores al usuario, este selecciona uno de ellos desde una lista desplegable. Al seleccionar un usuario del listado, se envía nuevamente una petición *HTTP* que viaja hasta el modelo *SVD*, cuyos valores son retornados en formato *JSON*. En este momento, el *frontend* recibe los resultados y los organiza mediante *HTML* y *JavaScript* para que la información se presente de la mejor manera.

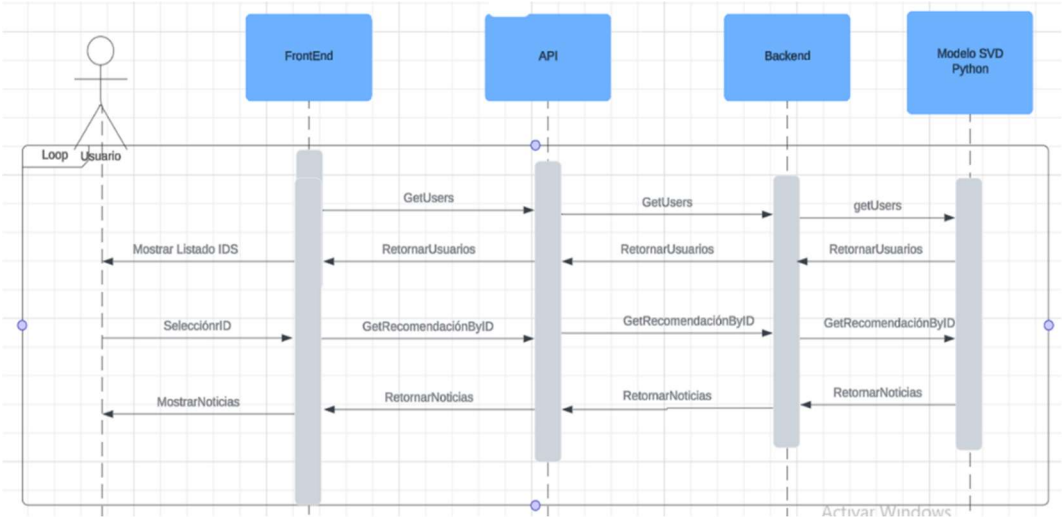

Figura 12. *Diagrama de secuencia del prototipo*

Casos de Uso

Tabla 20. Caso de uso para seleccionar usuario por identificador

Caso de Uso 1: Selección del ID de Usuario	
Descripción:	El usuario selecciona un ID de usuario de una lista desplegable
Actor Principal:	Usuario
Precondiciones	El usuario ha accedido a la interfaz del prototipo
Flujo Principal:	• El usuario abre la página del prototipo. • Se presenta una lista desplegable con IDs de usuario. • El usuario selecciona un ID de la lista. • El usuario hace clic en el botón "Obtener Recomendaciones".

Tabla 21. Caso de uso de para obtener recomendación de noticias

Caso de Uso 2: Obtención de Recomendaciones de Noticias	
Descripción:	El sistema muestra las noticias vistas y las recomendaciones generadas por el modelo SVD.
Actor Principal:	Sistema
Precondiciones	El usuario ha seleccionado un ID de usuario y ha hecho clic en "Obtener Recomendaciones".
Flujo Principal:	9. El sistema recibe la solicitud del ID de usuario a través de la API. • El modelo SVD procesa la solicitud y recupera las noticias vistas por el usuario y las recomendaciones correspondientes. • El sistema envía la información de vuelta al frontend en formato JSON. • El frontend procesa los datos y los presenta en la interfaz, mostrando dos secciones: "Viewed News" (Noticias Vistas) y "Recommended News" (Noticias Recomendadas). • El usuario puede visualizar la información de manera clara y comprensible.

8.3. Pruebas del prototipo

Las pruebas del sistema son fundamentales para garantizar que el prototipo de recomendación de noticias funcione de manera correcta y eficiente. A continuación, se describen las diferentes pruebas realizadas, que incluyen la validación de la lógica del modelo, el funcionamiento de la API y la correcta visualización de los resultados en el frontend.

Pruebas del Modelo SVD: Se realizaron pruebas para verificar que el modelo SVD genere recomendaciones precisas y coherentes para cada usuario. Las pruebas se enfocaron en evaluar la precisión y el desempeño del modelo, y se realizaron de la siguiente manera:

- Prueba de precisión del modelo: Se utilizó la métrica de precisión para evaluar la calidad de las recomendaciones generadas por el modelo *SVD*. El conjunto de datos de prueba se comparó con las predicciones del modelo para medir qué tan bien el modelo recomendó noticias relevantes a los usuarios.
- Prueba de recomendaciones para usuarios específicos: Se seleccionaron varios usuarios del conjunto de datos y se revisaron las recomendaciones generadas. Se buscó verificar que las recomendaciones fueran coherentes con el historial de noticias vistas por cada usuario.

Pruebas de la API con FastAPI: La API desarrollada con *FastAPI* fue sometida a pruebas para asegurar su correcto funcionamiento y respuesta adecuada a las solicitudes desde el frontend. Las pruebas incluyeron:

- Prueba de conexión: Se verificó que la API estuviera correctamente expuesta y accesible mediante Ngrok, permitiendo el acceso desde cualquier dispositivo conectado a Internet durante el desarrollo y la fase de pruebas.
- Prueba de endpoints: Se probaron los endpoints principales de la API, como el endpoint /get_recommendations, para asegurar que las solicitudes de los usuarios devolvieran las noticias vistas y recomendadas de forma correcta. Se enviaron solicitudes tanto válidas como inválidas (ID de usuarios inexistentes) para evaluar el manejo de errores, para estas pruebas se hizo uso de la herramienta Postman
- Prueba de rendimiento: Se realizaron pruebas de carga para evaluar cómo respondía la API a múltiples solicitudes simultáneas. Esto permitió verificar que la API mantenía un tiempo de respuesta adecuado incluso bajo carga, garantizando una experiencia fluida para el usuario final.

Pruebas del Frontend: Se llevaron a cabo pruebas en la interfaz de usuario para asegurar una experiencia de navegación intuitiva y la correcta visualización de las noticias. Estas pruebas incluyeron:

- Prueba de interacción con el selector de usuarios: Se probó que el selector de usuarios mostrara todos los IDs disponibles y que la selección de un ID activara correctamente la solicitud de recomendaciones. Se verificó que la lista se actualizara de acuerdo con los datos del backend.
- Prueba de visualización de resultados: Se comprobó que las noticias vistas y recomendadas se mostraran correctamente en la interfaz una vez obtenida la respuesta de la API. Esto incluyó verificar que cada noticia mostrara su título, categoría, subcategoría y descripción de manera adecuada, con un formato consistente y legible.
- Prueba de manejo de errores en el frontend: Se simularon situaciones en las que la API devolvía errores (como un ID de usuario inexistente), para asegurar que el frontend mostrara mensajes de error apropiados al usuario y que la experiencia de navegación no se viera afectada negativamente.

Pruebas de Integración: Se realizaron pruebas de integración para asegurar que todos los componentes del prototipo funcionaran en conjunto de manera correcta. Estas pruebas se enfocaron en verificar que la interacción entre el frontend, la API y el modelo de recomendación ***SVD*** fuera fluida y sin problemas.

- Prueba de flujo completo: Se simuló el flujo completo desde la selección de un ID de usuario hasta la visualización de las recomendaciones, verificando que cada paso se ejecutara correctamente. Esto incluyó la interacción con el selector de usuarios, la llamada a la API, la generación de recomendaciones y su visualización en la interfaz.
- Prueba de consistencia de datos: Se verificó que los datos de noticias vistas y recomendadas obtenidos desde la API coincidieran con las predicciones generadas por el modelo SVD, asegurando que no hubiera discrepancias entre lo procesado por el backend y lo mostrado al usuario.

8.4. Producción en un entorno local

La fase de producción en un entorno local tiene como objetivo implementar el prototipo de recomendación de noticias de manera que pueda ser ejecutado de forma independiente y controlada en una máquina local. Esta etapa asegura que el sistema funcione correctamente fuera del entorno de desarrollo y sea accesible para pruebas continuas y futuras iteraciones sin depender de conexiones externas, como Google Colab.

Preparación del Entorno Local: Antes de poner el sistema en producción en un entorno local, fue necesario realizar una serie de configuraciones para asegurar que todos los componentes estuvieran disponibles y operativos:

- Instalación de Python y dependencias: Se instaló Python en el entorno local, junto con las librerías necesarias para el desarrollo, tales como scikit-surprise para el modelo SVD, FastAPI para la creación de la API, y Uvicorn para el servidor de la API. Además, se configuró un entorno virtual para mantener las dependencias organizadas y evitar conflictos con otros proyectos.
- Carga de los datos locales: Los archivos de datos (news.tsv y behaviorstsv) fueron descargados desde Google Drive y almacenados localmente. Esto permitió que el sistema accediera a los datos sin necesidad de conexión a Internet, garantizando un funcionamiento más rápido y seguro.
- Entrenamiento del modelo: Se ejecutó el entrenamiento del modelo SVD de forma local utilizando los datos cargados, generando las predicciones necesarias para recomendar noticias a los usuarios. El entrenamiento local permitió ajustar el modelo y evaluar su rendimiento de manera más rápida, sin depender de recursos en la nube.

Configuración de la API con FastAPI: La API fue configurada para funcionar en el entorno local, permitiendo que el frontend pudiera realizar solicitudes de manera rápida y sin la latencia de una conexión remota. Los pasos realizados fueron:

- Despliegue de la API con Uvicorn: Se utilizó Uvicorn para iniciar el servidor de la API en un entorno local, ejecutando el comando uvicorn main:app --reload desde la línea de comandos. Esto permitió exponer los endpoints para obtener el listado de usuarios y las recomendaciones.
- Prueba de la API localmente: La API fue probada en el navegador y con herramientas como Postman para asegurar que los endpoints respondieran correctamente a las solicitudes de obtención de recomendaciones y la lista de usuarios. Se verificó que la API devolviera respuestas rápidas y precisas.

Implementación del Frontend en el Entorno Local: La interfaz de usuario, desarrollada en HTML, CSS y JavaScript, fue configurada para interactuar directamente con la API desplegada localmente. Los pasos realizados para esta implementación fueron:

- Configuración de las rutas locales: El archivo HTML fue modificado para que las solicitudes de la API apuntaran a http://localhost:8000, la dirección donde la API estaba desplegada localmente. Esto permitió que el frontend pudiera enviar y recibir datos de la API sin necesidad de conexión a servidores externos.
- Pruebas de la interfaz de usuario: La interfaz fue probada en varios navegadores (como Chrome y Firefox) para asegurar la compatibilidad y el correcto

funcionamiento. Se verificó que el selector de usuarios cargara los IDs disponibles y que las recomendaciones se mostraran adecuadamente al seleccionar un usuario.

- Ajustes de estilo y diseño: Se realizaron ajustes finales en los estilos CSS para asegurar una presentación visual óptima de los resultados, haciendo que la experiencia de usuario fuera lo más fluida posible. A continuación, en la Figura 13, se muestra la interfaz de usuario generada en el prototipo.

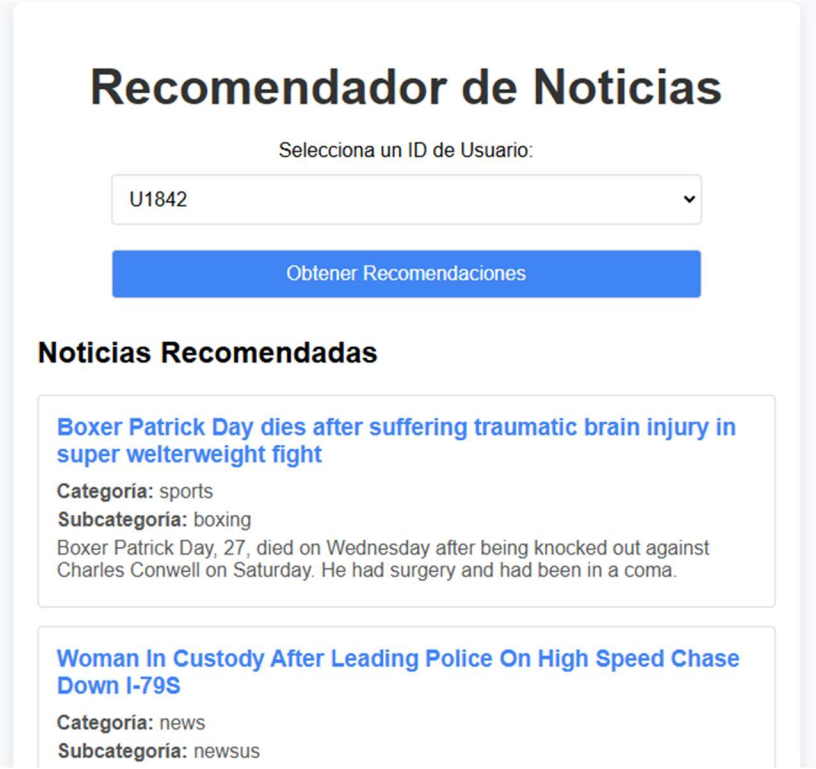

***Figura 13*.** Ejemplo del prototipo del sistema de recomendación

9. CONCLUSIONES

- En este proyecto aplicado se logró desarrollar un modelo de recomendación de noticias, con un muy buen nivel de personalización, esto determinado porque los modelos pueden generar recomendaciones personalizadas al 98% de los usuarios del sistema. Para lograrlo, se experimentó con los dos enfoques fundamentales de recomendación: uno basado en contenido y el otro basado en filtrado colaborativo. En el contexto de este trabajo, los experimentos mostraron que el enfoque de recomendación por filtrado colaborativo generó mejores resultados que el filtrado por contenido.

- Los datos de entrada para los métodos de aprendizaje automático por filtrado colaborativo implementados consisten en una matriz dispersa con la tripleta Usuario-Noticia-Clic. Fue muy interesante comprobar cómo, a pesar de la gran escasez de datos de valoración de los usuarios, lo cual es normal en los sistemas de recomendación, el modelo desarrollado es capaz de generar buenos resultados de recomendación personalizada, con una Precisión de 0.9308, lo que indica que las noticias recomendadas son relevantes para el usuario.

- Los resultados del modelo implementado por filtrado basado en contenido no fueron los esperados, básicamente porque de los datos disponibles no se logró establecer una forma para la identificación de características que permitieran diferenciar, de una forma estandarizada, unas noticias de otras. Solamente se tenían dos variables muy correlacionadas para establecer características distintivas: la categoría y la subcategoría. Por tal razón, también se experimentó representando vectorialmente el contenido textual del título. A pesar de esto, los resultados de personalización de las listas de recomendación no fueron satisfactorios. Para resolver este inconveniente, como trabajo futuro se plantea, experimentar con otros métodos para la representación vectorial del contenido textual de las noticias y, también buscar mayor amplitud (más categorías) y profundidad (más subcategorías en cada categoría) para caracterizar con más detalle las noticias y lograr una mejor diferenciación que permite generar recomendaciones basadas en contenido.

- En cuanto a la utilización de recursos computacionales, los métodos de aprendizaje automático para la recomendación personalizada de noticias por filtrado colaborativo con un enfoque basado en modelos, son más eficientes, empleando menos del 10% del tiempo de procesamiento, en comparación con los métodos con

un enfoque basado en memoria. Básicamente porque estos últimos, como k-vecinos más cercanos (KNN) utilizan directamente todos los registros históricos del interés Usuario-Noticia almacenados en el sistema para lograr el objetivo; en contraste, los métodos basados en modelos utilizan los registros históricos del interés Usuario-Noticia para entrenar un modelo predictivo.

- Las listas de recomendación generadas por los métodos de filtrado colaborativo, evaluadas por el F1-Score, logran un buen nivel de calidad cercano a 0.8000. Lo anterior, a excepción de la técnica Normal Predictor (NP) que logró solamente un 0.3500. Las otras cuatro técnicas implementadas generaron muy buenos resultados de personalización y eficiencia. Esto se debió al buen desempeño de los modelos en la estimación de las predicciones.

- Aunque los resultados en cuanto a RMSE, MAE, Precisión y Recall fueron similares para k-Vecinos más Cercanos (KNN), Descomposición en Valores Singulares (SVD), Factorización No Negativa de Matrices (NMF) y Clustering por k-medias (KM), en SVD se observó un desempeño levemente superior, con un RMSE de 0.2461 y un F1-Score de 0.8118. Entonces, por estos buenos resultados y por los menores tiempos en el procesamiento de los datos, se seleccionó la técnica SVD para desarrollar el prototipo del sistema de recomendación personalizada de noticias.

- Con el modelo de mejor desempeño, Descomposición en Valores Singulares (SVD), en el contexto de las métricas de evaluación y validación de los resultados utilizados en este proyecto aplicado, se logró desarrollar un prototipo del sistema de recomendación personalizada de noticias para presentar de manera visual e interactiva el funcionamiento del modelo en un entorno de producción local. Esta última fase del proyecto es muy importante porque se puede asegurar que el modelo funciona correctamente fuera del entorno de desarrollo y sea accesible para pruebas continuas sin depender de conexiones externas como Google Colab.

10. REFERENCIAS BIBLIOGRAFICAS

[1] P. Doblas Martín, "Clasificación de noticias mediante técnicas de procesamiento del lenguaje natural basadas en aprendizaje profundo," Trabajo de Grado, Universidad de Málaga, Departamento de Lenguajes y Ciencias de la Computación, Programa de Ingeniería Informática, 2021.

[2] W. S. Cleveland, "Data science: An action plan for expanding the technical areas of the field of statistics," International Statistical Review, vol. 69, no. 1, pp. 21–26, 2001.

[3] F. Pajuelo Holguera, "Sistemas de recomendación basados en filtrado colaborativo: Aceleración mediante computación reconfigurable y aplicaciones predictivas sensoriales," Tesis Doctoral, Universidad de Extremadura, Programa de Doctorado en Tecnología Aeroespacial: Ingenierías Electromagnética, Electrónica, Informática y Mecánica, 2021.

[4] N. Jonnalgedda, S. Gauch, K. Labille, and S. Alfarhood, "Incorporating popularity in a personalized news recommender system," PeerJ Computer Science, vol. 2, no. 63, pp. 1–20, Jun. 2016. [Online]. Available: https://peerj.com/articles/cs-63/. DOI: 10.7717/peerj-cs.63.

[5] A. Tuzhilin and G. Adomavicius, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734-749, Jun. 2005.

[6] M. Balabanovic and Y. Shoham, "Fab: content-based, collaborative recommendation," *Communications of the ACM*, vol. 40, no. 3, pp. 66–72, Mar. 1997.

[7] H. Ko, S. Lee, Y. Park, and A. Choi, "A Survey of Recommendation Systems: Recommendation Models, Techniques, and Application Fields," *Electronics*, vol. 11, no. 1, p. 141, Jan. 2022. DOI: 10.3390/electronics11010141.

[8] X. Amatriain, "Recommender Systems," Machine Learning Summer School 2014 @ CMU, Jul. 2014. [Online]. Available: http://www.slideshare.net/xamat/recommender-systems-machine-learning-summer-school-2014-cmu/. [Accessed: Sep. 29, 2024].

[9] O. Escamilla González and S. Marcellin Jacques, "Estado del arte en los sistemas de recomendación," *Research in Computing Science*, vol. 135, Universidad Nacional Autónoma de México, Posgrado en Ciencias e Ingeniería en Computación, Ciudad de México, pp. 25–40, 2017.

[10] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan, "Collaborative Filtering Recommender Systems," *NOW the Essence of Knowledge*, vol. 4, no. 2, pp. 81–173, 2011. [Online]. Available: http://files.grouplens.org/papers/FnTCFRecsysSurvey.pdf. [Accessed: Sep. 29, 2024].

[11] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, "Enfoque basado en modelos KNN en la clasificación," in *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA and ODBASE*, R. Meersman, Z. Tari, and D. C. Schmidt, Eds., Lecture Notes in Computer Science, vol. 2888, Springer, Berlin/Heidelberg, Germany, 2003, pp. 986–996.

[12] A. Hug, "Prediction algorithms package — Surprise 1.1.1 documentation," *Surprise.readthedocs.io*, 2024. [Online]. Available: https://surprise.readthedocs.io/en/stable/prediction_algorithms_package.html. [Accessed: Sep. 29, 2024].

[13] I. Katsov, *Introduction to Algorithmic Marketing*, GridDynamics, 2018. ISBN: 978-0-692-98904-3.

[14] J. Vélez Correa and P. Nieto Figueroa, "Validación de medidas de evaluación para el pronóstico de la tasa de cambio en Colombia," Colegio de Estudios Superiores de Administración (CESA), Maestría en Finanzas Corporativas, Bogotá, 2016.

[15] F. Pajuelo Holguera, "Sistemas de recomendación basados en filtrado colaborativo: Aceleración mediante computación reconfigurable y aplicaciones predictivas sensoriales," Tesis Doctoral, Universidad de Extremadura, Programa de Doctorado en Tecnología Aeroespacial: Ingenierías Electromagnética, Electrónica, Informática y Mecánica, 2021.

[16] F. L. Parra Anzola, "Categorización de letras de canciones de un portal web usando agrupación," Tesis de pregrado, Universidad Nacional de Colombia, Facultad de Ingeniería, Departamento de Ingeniería de Sistemas y Computación, Bogotá, Colombia, 2013.

[17] E. Morales Agostinho, "Sistemas de recomendación de noticias basados en aprendizaje profundo," Trabajo Fin de Grado, Escuela Politécnica Superior, Universidad Autónoma de Madrid, Madrid, España, 2021.

[18] D. D. Guzmán Chuva and D. F. Tixi Uyaguari, "Desarrollo de un sistema inteligente para la generación, identificación, clasificación, redacción y recomendaciones de noticias utilizando GPT-J," Trabajo de titulación, Carrera de Computación, Universidad Politécnica Salesiana, Cuenca, Ecuador, 2023.

[19] Microsoft, "Microsoft News Dataset," *Microsoft Learn*. [Online]. Available: https://learn.microsoft.com/en-us/azure/open-datasets/dataset-microsoft-news?tabs=azureml-opendatasets. [Accessed: Sep. 29, 2024].

[20] Microsoft, "MSR License Data," *GitHub*. [Online]. Available: https://github.com/msnews/MIND/blob/master/MSR%20License_Data.pdf. [Accessed: Sep. 29, 2024].

[21] F. Wu, Y. Qiao, J. H. Chen, C. Wu, y T. Qi, "A Large-scale Dataset for News Recommendation," Microsoft Research, Microsoft, Tsinghua University, en *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[22] Microsoft News, "Microsoft News GitHub Page," [En línea]. Disponible: https://msnews-github-io.translate.goog/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=sc. [Accedido: 29-Sep-2024].

[23] F. Ricci, L. Rokach, B. Shapira, P. Kantor, *Recommender Systems Handbook*, New York. Springer Science+Bussiness Media, 2011.

11. ANEXOS

Anexo 1. Resultados de la optimización por *GridSearchCV*

Tabla 22. Resultados de optimización de hiperparámetros modelo *KNN*

#	RMSE						MAE						time				parameters		
	split			mean	std	rank	split			mean	std	rank	fit		test				
	0	1	2				0	1	2				mean	std	mean	std	k	min_k	sim
0	0,2521	0,2518	0,2517	0,2519	0,0002	1	0,1127	0,1126	0,1124	0,1125	0,0001	1	36,5	0,3	116,5	1,1	30	1	msd
1	0,2619	0,2610	0,2610	0,2613	0,0004	7	0,1196	0,1193	0,1191	0,1193	0,0002	6	56,5	0,3	117,0	0,7	30	1	pearson
2	0,2536	0,2532	0,2531	0,2533	0,0002	2	0,1163	0,1162	0,1159	0,1161	0,0002	3	36,7	0,4	115,7	0,5	30	3	msd
3	0,2723	0,2720	0,2718	0,2720	0,0002	10	0,1396	0,1396	0,1393	0,1395	0,0001	10	56,5	0,2	116,8	0,5	30	3	pearson
4	0,2539	0,2536	0,2535	0,2537	0,0002	3	0,1157	0,1156	0,1154	0,1156	0,0001	2	36,7	0,3	119,8	0,7	40	1	msd
5	0,2624	0,2615	0,2616	0,2618	0,0004	8	0,1211	0,1208	0,1207	0,1209	0,0002	7	57,1	0,4	120,2	1,1	40	1	pearson
6	0,2553	0,2551	0,2549	0,2551	0,0002	5	0,1194	0,1192	0,1190	0,1192	0,0002	5	36,8	0,3	120,2	0,5	40	3	msd
7	0,2728	0,2725	0,2723	0,2725	0,0002	11	0,1411	0,1411	0,1409	0,1410	0,0001	11	56,5	0,4	121,1	0,7	40	3	pearson
8	0,2552	0,2549	0,2548	0,2550	0,0002	4	0,1180	0,1179	0,1177	0,1179	0,0001	4	37,1	0,2	122,7	0,3	50	1	msd
9	0,2626	0,2618	0,2618	0,2621	0,0004	9	0,1221	0,1218	0,1217	0,1218	0,0002	9	56,6	0,4	129,2	5,5	50	1	pearson
10	0,2566	0,2563	0,2561	0,2564	0,0002	6	0,1216	0,1215	0,1213	0,1215	0,0001	8	36,5	0,2	122,8	0,4	50	3	msd
11	0,2730	0,2728	0,2726	0,2728	0,0002	12	0,1420	0,1421	0,1419	0,1420	0,0001	12	56,4	0,2	122,6	0,5	50	3	pearson

Tabla 23. Resultados de optimización de hiperparámetros modelo *SVD* (1/2)

#	RMSE						MAE						time				parameters			
	split			mean	std	rank	split			mean	std	rank	fit		test					
	0	1	2				0	1	2				mean	std	mean	std	n_factors	n_epochs	lr_all	reg_all
0	0,2855	0,2852	0,2848	0,2852	0,0003	74	0,2020	0,2020	0,2015	0,2019	0,0002	73	14,6	0,2	8,0	0,6	80	15	0,0025	0,01
1	0,2863	0,2860	0,2855	0,2859	0,0003	76	0,2047	0,2046	0,2041	0,2044	0,0003	76	13,9	0,0	7,5	0,4	80	15	0,0025	0,02
2	0,2871	0,2868	0,2862	0,2867	0,0004	79	0,2071	0,2071	0,2065	0,2069	0,0003	79	13,8	0,1	7,6	0,6	80	15	0,0025	0,03
3	0,2686	0,2684	0,2682	0,2684	0,0002	48	0,1773	0,1773	0,1770	0,1772	0,0001	48	13,8	0,0	7,9	0,6	80	15	0,0050	0,01
4	0,2711	0,2708	0,2703	0,2708	0,0003	52	0,1813	0,1813	0,1808	0,1811	0,0002	50	13,8	0,1	7,3	0,0	80	15	0,0050	0,02
5	0,2731	0,2725	0,2720	0,2725	0,0005	57	0,1847	0,1845	0,1840	0,1844	0,0003	55	13,9	0,1	7,3	0,0	80	15	0,0050	0,03
6	0,2609	0,2607	0,2600	0,2605	0,0004	22	0,1674	0,1674	0,1668	0,1672	0,0003	21	13,9	0,1	7,2	0,1	80	15	0,0075	0,01
7	0,2650	0,2647	0,2641	0,2646	0,0004	36	0,1724	0,1723	0,1718	0,1722	0,0003	36	13,8	0,1	7,2	0,0	80	15	0,0075	0,02
8	0,2680	0,2675	0,2672	0,2675	0,0003	46	0,1763	0,1760	0,1758	0,1760	0,0002	42	13,8	0,1	7,2	0,0	80	15	0,0075	0,03
9	0,2775	0,2772	0,2767	0,2771	0,0003	66	0,1897	0,1896	0,1891	0,1895	0,0003	64	18,3	0,1	7,3	0,0	80	20	0,0025	0,01
10	0,2790	0,2784	0,2781	0,2785	0,0004	69	0,1929	0,1926	0,1922	0,1926	0,0003	67	18,4	0,1	7,2	0,1	80	20	0,0025	0,02
11	0,2799	0,2796	0,2790	0,2795	0,0004	72	0,1955	0,1955	0,1949	0,1953	0,0003	70	18,3	0,1	7,3	0,0	80	20	0,0025	0,03
12	0,2625	0,2626	0,2617	0,2623	0,0004	29	0,1688	0,1691	0,1683	0,1687	0,0003	27	20,1	1,4	8,6	1,0	80	20	0,0050	0,01
13	0,2662	0,2658	0,2655	0,2658	0,0003	40	0,1735	0,1734	0,1731	0,1734	0,0002	39	21,3	0,1	9,3	0,1	80	20	0,0050	0,02
14	0,2687	0,2684	0,2680	0,2683	0,0003	47	0,1773	0,1771	0,1767	0,1770	0,0003	43	21,3	0,2	9,3	0,1	80	20	0,0050	0,03
15	0,2547	0,2544	0,2542	0,2544	0,0002	8	0,1597	0,1595	0,1595	0,1596	0,0001	6	21,3	0,2	9,2	0,0	80	20	0,0075	0,01
16	0,2602	0,2601	0,2593	0,2598	0,0004	20	0,1657	0,1657	0,1651	0,1655	0,0003	17	21,5	0,1	9,3	0,0	80	20	0,0075	0,02
17	0,2641	0,2639	0,2634	0,2638	0,0003	34	0,1700	0,1700	0,1696	0,1699	0,0002	30	21,5	0,2	9,3	0,1	80	20	0,0075	0,03
18	0,2719	0,2718	0,2713	0,2717	0,0003	55	0,1813	0,1813	0,1808	0,1811	0,0002	49	26,6	0,1	9,3	0,1	80	25	0,0025	0,01
19	0,2740	0,2737	0,2732	0,2736	0,0003	60	0,1850	0,1849	0,1844	0,1848	0,0002	58	26,7	0,1	9,3	0,0	80	25	0,0025	0,02
20	0,2753	0,2750	0,2746	0,2750	0,0003	63	0,1879	0,1878	0,1873	0,1877	0,0003	61	26,5	0,1	9,2	0,0	80	25	0,0025	0,03
21	0,2581	0,2577	0,2572	0,2577	0,0004	13	0,1630	0,1627	0,1624	0,1627	0,0002	12	26,6	0,1	9,3	0,0	80	25	0,0050	0,01
22	0,2626	0,2622	0,2618	0,2622	0,0003	28	0,1682	0,1680	0,1676	0,1680	0,0002	24	26,5	0,2	9,3	0,0	80	25	0,0050	0,02
23	0,2659	0,2657	0,2650	0,2655	0,0004	38	0,1723	0,1722	0,1717	0,1721	0,0003	35	26,4	0,2	9,3	0,0	80	25	0,0050	0,03
24	0,2498	0,2494	0,2490	0,2494	0,0003	3	0,1540	0,1536	0,1536	0,1537	0,0002	3	26,5	0,2	9,2	0,0	80	25	0,0075	0,01
25	0,2561	0,2559	0,2556	0,2559	0,0002	10	0,1605	0,1605	0,1602	0,1604	0,0001	9	26,5	0,1	9,2	0,1	80	25	0,0075	0,02
26	0,2614	0,2610	0,2605	0,2610	0,0003	24	0,1657	0,1657	0,1653	0,1655	0,0002	18	26,4	0,2	9,2	0,0	80	25	0,0075	0,03
27	0,2854	0,2853	0,2849	0,2852	0,0002	73	0,2019	0,2022	0,2017	0,2019	0,0002	74	18,0	0,1	9,2	0,1	100	15	0,0025	0,01
28	0,2866	0,2861	0,2857	0,2861	0,0004	77	0,2051	0,2048	0,2043	0,2047	0,0003	77	18,1	0,1	9,7	0,7	100	15	0,0025	0,02
29	0,2872	0,2869	0,2865	0,2869	0,0003	80	0,2074	0,2074	0,2069	0,2072	0,0002	80	17,9	0,1	9,6	0,7	100	15	0,0025	0,03
30	0,2679	0,2674	0,2669	0,2674	0,0004	45	0,1774	0,1773	0,1769	0,1772	0,0002	47	18,1	0,1	9,2	0,1	100	15	0,0050	0,01
31	0,2704	0,2701	0,2696	0,2700	0,0004	50	0,1814	0,1813	0,1808	0,1812	0,0002	51	18,1	0,2	9,2	0,0	100	15	0,0050	0,02
32	0,2724	0,2720	0,2714	0,2719	0,0004	56	0,1848	0,1846	0,1841	0,1845	0,0003	56	17,9	0,1	10,1	0,8	100	15	0,0050	0,03
33	0,2594	0,2590	0,2587	0,2590	0,0003	17	0,1670	0,1669	0,1666	0,1668	0,0001	20	17,9	0,1	8,7	0,9	100	15	0,0075	0,01
34	0,2636	0,2632	0,2628	0,2632	0,0003	31	0,1721	0,1719	0,1716	0,1719	0,0002	32	17,9	0,1	10,0	0,7	100	15	0,0075	0,02
35	0,2668	0,2667	0,2662	0,2665	0,0003	41	0,1761	0,1762	0,1757	0,1760	0,0002	41	18,0	0,1	9,7	0,8	100	15	0,0075	0,03
36	0,2769	0,2767	0,2766	0,2767	0,0001	65	0,1895	0,1896	0,1893	0,1895	0,0001	65	23,8	0,1	9,2	1,2	100	20	0,0025	0,01
37	0,2785	0,2782	0,2776	0,2781	0,0004	68	0,1929	0,1928	0,1922	0,1927	0,0003	68	23,8	0,1	9,6	0,7	100	20	0,0025	0,02
38	0,2797	0,2793	0,2788	0,2793	0,0004	71	0,1958	0,1957	0,1951	0,1955	0,0003	71	24,0	0,1	9,8	0,7	100	20	0,0025	0,03
39	0,2616	0,2612	0,2605	0,2611	0,0004	25	0,1689	0,1687	0,1681	0,1686	0,0003	26	23,8	0,1	9,0	1,3	100	20	0,0050	0,01

Tabla 23. Resultados de optimización de hiperparámetros modelo *SVD* (2/2)

#	RMSE						MAE						time				parameters			
	split			mean	std	rank	split			mean	std	rank	fit		test					
	0	1	2				0	1	2				mean	std	mean	std	n_factors	n_epochs	lr_all	reg_all
40	0,2650	0,2646	0,2644	0,2647	0,0002	37	0,1734	0,1732	0,1731	0,1732	0,0002	38	23,7	0,1	10,2	0,6	100	20	0,0050	0,02
41	0,2678	0,2675	0,2669	0,2674	0,0004	44	0,1772	0,1772	0,1767	0,1770	0,0003	44	23,8	0,1	9,1	1,2	100	20	0,0050	0,03
42	0,2528	0,2526	0,2523	0,2526	0,0002	6	0,1591	0,1590	0,1587	0,1590	0,0002	5	23,9	0,1	9,1	1,3	100	20	0,0075	0,01
43	0,2586	0,2581	0,2577	0,2581	0,0004	15	0,1651	0,1651	0,1647	0,1650	0,0002	15	23,8	0,1	10,3	0,7	100	20	0,0075	0,02
44	0,2628	0,2624	0,2622	0,2625	0,0002	30	0,1698	0,1697	0,1695	0,1697	0,0002	29	23,8	0,1	9,6	1,5	100	20	0,0075	0,03
45	0,2715	0,2714	0,2708	0,2712	0,0003	53	0,1814	0,1815	0,1811	0,1813	0,0002	53	29,7	0,1	9,0	1,3	100	25	0,0025	0,01
46	0,2735	0,2731	0,2726	0,2731	0,0004	59	0,1852	0,1850	0,1845	0,1849	0,0003	59	29,6	0,1	9,1	1,3	100	25	0,0025	0,02
47	0,2750	0,2747	0,2741	0,2746	0,0004	62	0,1882	0,1881	0,1876	0,1880	0,0003	62	29,8	0,1	10,2	0,7	100	25	0,0025	0,03
48	0,2565	0,2563	0,2559	0,2562	0,0002	11	0,1627	0,1625	0,1622	0,1624	0,0002	11	29,8	0,1	9,2	1,2	100	25	0,0050	0,01
49	0,2612	0,2610	0,2605	0,2609	0,0003	23	0,1680	0,1679	0,1675	0,1678	0,0002	23	29,9	0,1	10,2	0,7	100	25	0,0050	0,02
50	0,2648	0,2644	0,2639	0,2644	0,0004	35	0,1723	0,1721	0,1717	0,1720	0,0003	34	29,8	0,0	9,2	0,1	100	25	0,0050	0,03
51	0,2478	0,2476	0,2473	0,2476	0,0002	2	0,1532	0,1534	0,1532	0,1532	0,0001	2	29,7	0,1	10,2	0,6	100	25	0,0075	0,01
52	0,2545	0,2543	0,2536	0,2542	0,0004	7	0,1602	0,1602	0,1596	0,1600	0,0003	8	29,8	0,1	9,2	0,1	100	25	0,0075	0,02
53	0,2598	0,2594	0,2590	0,2594	0,0003	18	0,1653	0,1653	0,1650	0,1652	0,0001	16	29,7	0,1	9,2	0,0	100	25	0,0075	0,03
54	0,2856	0,2855	0,2851	0,2854	0,0002	75	0,2022	0,2022	0,2019	0,2021	0,0002	75	19,1	0,1	9,7	0,7	120	15	0,0025	0,01
55	0,2867	0,2863	0,2859	0,2863	0,0003	78	0,2052	0,2051	0,2047	0,2050	0,0002	78	19,2	0,0	9,8	0,6	120	15	0,0025	0,02
56	0,2876	0,2870	0,2866	0,2871	0,0004	81	0,2078	0,2076	0,2071	0,2075	0,0003	81	19,2	0,1	10,3	0,7	120	15	0,0025	0,03
57	0,2670	0,2669	0,2663	0,2667	0,0003	42	0,1773	0,1774	0,1768	0,1772	0,0003	45	19,2	0,1	9,3	0,0	120	15	0,0050	0,01
58	0,2696	0,2694	0,2690	0,2693	0,0003	49	0,1814	0,1813	0,1809	0,1812	0,0002	52	19,1	0,1	9,3	0,0	120	15	0,0050	0,02
59	0,2718	0,2716	0,2710	0,2715	0,0003	54	0,1848	0,1849	0,1843	0,1847	0,0003	57	19,1	0,1	9,3	0,1	120	15	0,0050	0,03
60	0,2582	0,2579	0,2576	0,2579	0,0003	14	0,1667	0,1667	0,1665	0,1667	0,0001	19	19,2	0,1	9,2	0,1	120	15	0,0075	0,01
61	0,2626	0,2623	0,2616	0,2622	0,0004	27	0,1719	0,1720	0,1714	0,1718	0,0003	31	19,2	0,1	9,2	0,1	120	15	0,0075	0,02
62	0,2659	0,2656	0,2651	0,2656	0,0003	39	0,1761	0,1761	0,1757	0,1760	0,0002	40	19,2	0,1	9,2	0,1	120	15	0,0075	0,03
63	0,2767	0,2767	0,2761	0,2765	0,0003	64	0,1896	0,1899	0,1891	0,1895	0,0003	66	25,4	0,1	9,1	0,0	120	20	0,0025	0,01
64	0,2783	0,2779	0,2775	0,2779	0,0003	67	0,1931	0,1929	0,1926	0,1929	0,0002	69	25,5	0,1	9,1	0,1	120	20	0,0025	0,02
65	0,2796	0,2791	0,2790	0,2792	0,0003	70	0,1960	0,1959	0,1957	0,1958	0,0001	72	25,6	0,1	9,2	0,0	120	20	0,0025	0,03
66	0,2604	0,2598	0,2595	0,2599	0,0004	21	0,1685	0,1684	0,1681	0,1683	0,0002	25	25,6	0,1	9,2	0,0	120	20	0,0050	0,01
67	0,2640	0,2635	0,2632	0,2636	0,0003	33	0,1732	0,1730	0,1727	0,1730	0,0002	37	25,5	0,0	9,1	0,1	120	20	0,0050	0,02
68	0,2673	0,2667	0,2663	0,2668	0,0004	43	0,1775	0,1772	0,1769	0,1772	0,0002	46	25,4	0,1	9,3	0,1	120	20	0,0050	0,03
69	0,2516	0,2512	0,2509	0,2512	0,0003	4	0,1588	0,1586	0,1587	0,1587	0,0001	4	25,4	0,0	9,2	0,1	120	20	0,0075	0,01
70	0,2573	0,2569	0,2567	0,2570	0,0003	12	0,1650	0,1649	0,1646	0,1648	0,0001	13	25,4	0,1	9,2	0,0	120	20	0,0075	0,02
71	0,2618	0,2613	0,2611	0,2614	0,0003	26	0,1697	0,1697	0,1694	0,1696	0,0001	28	25,4	0,0	9,2	0,0	120	20	0,0075	0,03
72	0,2712	0,2708	0,2703	0,2708	0,0004	51	0,1817	0,1815	0,1811	0,1814	0,0003	54	31,8	0,1	9,2	0,0	120	25	0,0025	0,01
73	0,2730	0,2726	0,2722	0,2726	0,0003	58	0,1852	0,1850	0,1847	0,1849	0,0002	60	31,9	0,1	9,3	0,0	120	25	0,0025	0,02
74	0,2747	0,2743	0,2739	0,2743	0,0003	61	0,1884	0,1883	0,1879	0,1882	0,0002	63	31,9	0,1	9,3	0,0	120	25	0,0025	0,03
75	0,2549	0,2549	0,2544	0,2548	0,0003	9	0,1619	0,1622	0,1618	0,1620	0,0002	10	31,8	0,1	9,3	0,1	120	25	0,0050	0,01
76	0,2597	0,2597	0,2592	0,2595	0,0002	19	0,1677	0,1677	0,1672	0,1675	0,0002	22	31,8	0,1	9,2	0,0	120	25	0,0050	0,02
77	0,2637	0,2635	0,2629	0,2633	0,0003	32	0,1720	0,1721	0,1715	0,1719	0,0002	33	31,9	0,3	9,2	0,1	120	25	0,0050	0,03
78	0,2465	0,2461	0,2458	0,2461	0,0003	1	0,1530	0,1528	0,1525	0,1528	0,0002	1	31,8	0,2	9,3	0,1	120	25	0,0075	0,01
79	0,2530	0,2526	0,2522	0,2526	0,0003	5	0,1598	0,1595	0,1593	0,1596	0,0002	7	31,8	0,2	9,2	0,0	120	25	0,0075	0,02
80	0,2586	0,2581	0,2578	0,2582	0,0003	16	0,1651	0,1650	0,1647	0,1649	0,0002	14	31,9	0,1	9,3	0,0	120	25	0,0075	0,03

Tabla 24. Resultados de optimización de hiperparámetros modelo *NMF*

#	RMSE						MAE						time				parameters			
	split			mean	std	rank	split			mean	std	rank	fit		test					
	0	1	2				0	1	2				mean	std	mean	std	n_factors	n_epochs	init_low	init_high
0	0,2704	0,2700	0,2700	0,2701	0,0002	9	0,1720	0,1718	0,1716	0,1718	0,0001	9	25,9	0,6	7,5	0,6	10	30	0	1
1	0,2670	0,2671	0,2669	0,2670	0,0001	6	0,1699	0,1700	0,1699	0,1699	0,0001	6	41,1	1,0	7,7	1,1	10	50	0	1
2	0,2641	0,2644	0,2650	0,2645	0,0004	4	0,1680	0,1683	0,1687	0,1683	0,0003	3	55,9	0,3	6,9	0,5	10	70	0	1
3	0,2695	0,2691	0,2693	0,2693	0,0002	8	0,1716	0,1714	0,1715	0,1715	0,0001	8	26,4	0,2	7,8	1,0	15	30	0	1
4	0,2655	0,2655	0,2649	0,2653	0,0003	5	0,1693	0,1693	0,1689	0,1692	0,0002	5	44,1	0,4	7,0	0,6	15	50	0	1
5	0,2629	0,2625	0,2623	0,2626	0,0003	2	0,1676	0,1674	0,1672	0,1674	0,0001	2	60,9	0,3	7,0	0,6	15	70	0	1
6	0,2687	0,2686	0,2687	0,2687	0,0001	7	0,1712	0,1711	0,1713	0,1712	0,0001	7	28,2	0,4	7,7	0,9	20	30	0	1
7	0,2643	0,2643	0,2644	0,2644	0,0000	3	0,1687	0,1687	0,1688	0,1687	0,0000	4	46,6	0,1	6,9	0,5	20	50	0	1
8	0,2612	0,2612	0,2614	0,2612	0,0001	1	0,1667	0,1667	0,1668	0,1667	0,0000	1	64,7	0,4	7,6	0,1	20	70	0	1

Tabla 25. Resultados de optimización de hiperparámetros modelo *KM*

#	RMSE						MAE						time				parameters		
	split			mean	std	rank	split			mean	std	rank	fit		test				
	0	1	2				0	1	2				mean	std	mean	std	n_cltr_u	n_cltr_i	n_epochs
0	0,2769	0,2775	0,2777	0,2774	0,0003	10	0,1473	0,1470	0,1476	0,1473	0,0002	27	31,2	0,5	7,7	0,1	3	3	15
1	0,2765	0,2770	0,2781	0,2772	0,0006	9	0,1467	0,1464	0,1476	0,1469	0,0005	26	39,4	0,5	7,3	0,5	3	3	20
2	0,2765	0,2762	0,2780	0,2769	0,0008	8	0,1473	0,1452	0,1470	0,1465	0,0009	24	48,4	0,7	7,3	0,5	3	3	25
3	0,2786	0,2787	0,2790	0,2788	0,0002	24	0,1458	0,1457	0,1464	0,1460	0,0003	20	32,0	0,4	6,9	0,5	3	5	15
4	0,2776	0,2784	0,2792	0,2784	0,0006	16	0,1455	0,1451	0,1456	0,1454	0,0002	16	42,0	0,7	7,2	0,6	3	5	20
5	0,2784	0,2786	0,2792	0,2787	0,0003	21	0,1457	0,1458	0,1461	0,1459	0,0002	18	51,9	0,5	7,2	0,6	3	5	25
6	0,2781	0,2788	0,2787	0,2785	0,0003	17	0,1448	0,1448	0,1451	0,1449	0,0001	11	34,2	0,4	6,8	0,6	3	7	15
7	0,2784	0,2788	0,2800	0,2791	0,0007	27	0,1453	0,1454	0,1452	0,1453	0,0001	15	45,0	0,8	7,3	1,1	3	7	20
8	0,2782	0,2792	0,2794	0,2789	0,0005	26	0,1445	0,1448	0,1448	0,1447	0,0001	9	55,8	0,7	7,6	0,1	3	7	25
9	0,2761	0,2763	0,2768	0,2764	0,0003	3	0,1465	0,1455	0,1448	0,1456	0,0007	17	32,1	0,7	7,3	0,5	5	3	15
10	0,2767	0,2772	0,2764	0,2768	0,0003	7	0,1456	0,1467	0,1462	0,1461	0,0005	22	41,8	0,5	7,3	0,6	5	3	20
11	0,2761	0,2754	0,2773	0,2763	0,0008	2	0,1471	0,1474	0,1462	0,1469	0,0005	25	52,1	0,8	7,3	0,6	5	3	25
12	0,2778	0,2762	0,2784	0,2775	0,0009	11	0,1454	0,1446	0,1448	0,1449	0,0003	13	33,9	0,6	7,3	0,6	5	5	15
13	0,2761	0,2769	0,2770	0,2767	0,0004	6	0,1443	0,1447	0,1445	0,1445	0,0002	7	44,1	0,5	7,1	0,5	5	5	20
14	0,2775	0,2772	0,2791	0,2779	0,0008	15	0,1447	0,1451	0,1449	0,1449	0,0002	12	54,4	0,6	6,8	0,6	5	5	25
15	0,2778	0,2790	0,2798	0,2789	0,0008	25	0,1446	0,1445	0,1446	0,1446	0,0001	8	35,4	0,5	7,2	0,6	5	7	15
16	0,2790	0,2781	0,2789	0,2787	0,0004	20	0,1439	0,1439	0,1449	0,1442	0,0004	5	46,8	0,4	7,2	0,6	5	7	20
17	0,2784	0,2786	0,2788	0,2786	0,0001	18	0,1448	0,1447	0,1445	0,1447	0,0001	10	57,8	1,1	7,2	0,6	5	7	25
18	0,2756	0,2763	0,2768	0,2762	0,0005	1	0,1453	0,1467	0,1458	0,1459	0,0006	19	33,7	0,4	7,2	0,6	7	3	15
19	0,2751	0,2775	0,2772	0,2766	0,0011	5	0,1452	0,1466	0,1462	0,1460	0,0006	21	43,4	0,5	7,1	0,6	7	3	20
20	0,2746	0,2767	0,2781	0,2764	0,0014	4	0,1450	0,1464	0,1474	0,1463	0,0010	23	54,1	0,7	6,8	0,5	7	3	25
21	0,2763	0,2786	0,2787	0,2779	0,0011	14	0,1451	0,1449	0,1456	0,1452	0,0003	14	35,4	0,4	7,1	0,5	7	5	15
22	0,2770	0,2768	0,2787	0,2775	0,0008	12	0,1431	0,1436	0,1448	0,1438	0,0007	1	46,5	0,6	7,1	0,6	7	5	20
23	0,2775	0,2771	0,2787	0,2778	0,0007	13	0,1441	0,1443	0,1441	0,1442	0,0001	3	57,7	0,5	7,2	0,6	7	5	25
24	0,2768	0,2786	0,2809	0,2787	0,0017	23	0,1437	0,1438	0,1450	0,1442	0,0005	4	37,4	0,4	7,1	0,5	7	7	15
25	0,2783	0,2783	0,2794	0,2786	0,0005	19	0,1437	0,1437	0,1444	0,1440	0,0003	2	49,1	0,6	6,8	0,5	7	7	20
26	0,2771	0,2785	0,2806	0,2787	0,0014	22	0,1439	0,1445	0,1445	0,1443	0,0003	6	60,5	0,6	7,1	0,5	7	7	25

Anexo 2. Definición del tamaño *L* de la lista de recomendación y del umbral *U* para la clasificación del interés

Tabla 26. *Impacto de **L** y **U** en las listas de recomendación por KNN*

L	U	Precision	Recall	F1-Score
8	0,5	0,9384	0,6821	0,7900
8	0,6	0,9348	0,6778	0,7858
8	0,7	0,9317	0,6524	0,7674
8	0,8	0,9286	0,6273	0,7488
8	0,9	0,9131	0,5674	0,6999
10	0,5	0,9421	0,7323	0,8241
10	0,6	0,9301	0,6983	0,7977
10	0,7	0,9223	0,6855	0,7865
10	0,8	0,9279	0,6595	0,7710
10	0,9	0,9144	0,5952	0,7211
12	0,5	0,9333	0,7423	0,8269
12	0,6	0,9303	0,7268	0,8161
12	0,7	0,9315	0,7093	0,8054
12	0,8	0,9249	0,6767	0,7816
12	0,9	0,9133	0,6110	0,7322
15	0,5	0,9283	0,7676	0,8403
15	0,6	0,9285	0,7534	0,8318
15	0,7	0,9299	0,7340	0,8204
15	0,8	0,9276	0,7006	0,7983
15	0,9	0,9100	0,6275	0,7428

Tabla 27. *Impacto de L y U en las listas de recomendación por NP*

L	U	Precision	Recall	F1-Score
8	0,5	0,4722	0,3601	0,4086
8	0,6	0,478	0,3635	0,4130
8	0,7	0,4542	0,2722	0,3404
8	0,8	0,4346	0,2247	0,2962
8	0,9	0,4099	0,1791	0,2493
10	0,5	0,4748	0,3844	0,4248
10	0,6	0,4647	0,3357	0,3898
10	0,7	0,4537	0,2863	0,3511
10	0,8	0,4342	0,2332	0,3034
10	0,9	0,4165	0,1885	0,2595
12	0,5	0,4734	0,3997	0,4334
12	0,6	0,4668	0,3462	0,3976
12	0,7	0,4549	0,2943	0,3574
12	0,8	0,434	0,2391	0,3083
12	0,9	0,4104	0,1877	0,2576
15	0,5	0,4754	0,4181	0,4449
15	0,6	0,4639	0,3587	0,4046
15	0,7	0,4491	0,2993	0,3592
15	0,8	0,4338	0,2452	0,3133
15	0,9	0,4106	0,193	0,2626

Tabla 28. *Impacto de L y U en las listas de recomendación por SVD*

L	U	Precision	Recall	F1-Score
8	0,5	0,9362	0,6992	0,8005
8	0,6	0,9328	0,6974	0,7981
8	0,7	0,9335	0,6684	0,7790
8	0,8	0,9301	0,6421	0,7597
8	0,9	0,9215	0,5893	0,7189
10	0,5	0,9307	0,734	0,8207
10	0,6	0,9347	0,7206	0,8138
10	0,7	0,9348	0,701	0,8012
10	0,8	0,9301	0,6711	0,7797
10	0,9	0,9223	0,609	0,7336
12	0,5	0,9307	0,762	0,8379
12	0,6	0,9319	0,7466	0,8290
12	0,7	0,9312	0,7223	0,8136
12	0,8	0,9293	0,6911	0,7927
12	0,9	0,9202	0,6183	0,7396
15	0,5	0,9277	0,7907	0,8537
15	0,6	0,9294	0,7711	0,8429
15	0,7	0,9295	0,7455	0,8274
15	0,8	0,928	0,7108	0,8050
15	0,9	0,9191	0,6284	0,7464

Tabla 29. *Impacto de L y U en las listas de recomendación por NMF*

L	U	Precision	Recall	F1-Score
8	0,5	0,93	0,683	0,7876
8	0,6	0,9289	0,6795	0,7849
8	0,7	0,924	0,6384	0,7551
8	0,8	0,9110	0,5896	0,7159
8	0,9	0,7087	0,3764	0,4917
10	0,5	0,9241	0,7121	0,8044
10	0,6	0,9231	0,6958	0,7935
10	0,7	0,921	0,6664	0,7733
10	0,8	0,9068	0,6091	0,7287
10	0,9	0,7097	0,3885	0,5021
12	0,5	0,9257	0,7437	0,8248
12	0,6	0,9198	0,7179	0,8064
12	0,7	0,9212	0,6897	0,7888
12	0,8	0,9058	0,6281	0,7418
12	0,9	0,7096	0,3895	0,5029
15	0,5	0,9181	0,7659	0,8351
15	0,6	0,9213	0,7465	0,8247
15	0,7	0,9193	0,7132	0,8032
15	0,8	0,9069	0,6459	0,7545
15	0,9	0,7117	0,3889	0,5030

Tabla 30. *Impacto de L y U en las listas de recomendación por KM*

L	U	Precision	Recall	F1-Score
8	0,5	0,9218	0,6805	0,7830
8	0,6	0,9159	0,6768	0,7784
8	0,7	0,9194	0,6486	0,7606
8	0,8	0,8977	0,6016	0,7204
8	0,9	0,8247	0,4904	0,6151
10	0,5	0,9174	0,7139	0,8030
10	0,6	0,9146	0,6971	0,7912
10	0,7	0,9123	0,6724	0,7742
10	0,8	0,8977	0,6326	0,7422
10	0,9	0,8145	0,4999	0,6196
12	0,5	0,915	0,7394	0,8179
12	0,6	0,9135	0,7207	0,8057
12	0,7	0,9096	0,6925	0,7863
12	0,8	0,8975	0,6533	0,7562
12	0,9	0,8268	0,5281	0,6445
15	0,5	0,908	0,7644	0,8300
15	0,6	0,911	0,7427	0,8183
15	0,7	0,9114	0,7210	0,8051
15	0,8	0,8977	0,6777	0,7723
15	0,9	0,8221	0,5409	0,6525