# Memoria: Sistema de Recomendación de Libros

## 1. Introducción

En este proyecto, se desarrolla un sistema de recomendación de libros combinando dos enfoques principales: recomendaciones basadas en contenido (content-based) y colaborativas (collaborative filtering).

El objetivo es aprovechar datos sobre libros, usuarios y valoraciones para construir un sistema que sugiera lecturas relevantes para un usuario dado, ya sea basado en las características de los libros que le han gustado o en las preferencias de usuarios similares. El proyecto incluye pasos de limpieza de datos, análisis exploratorio y evaluación de métricas.

## 2. Preparación y limpieza de datos

### 2.1 Fuentes de datos

Los datos provienen de varios archivos:

- books.csv: Información detallada de libros (títulos, autores, géneros).
- ratings.csv: Valoraciones de usuarios sobre libros.
- book_tags.csv: Asociación de libros con etiquetas de género.

### 2.2 Identificación de problemas

Se detectó que los book_id en el dataset de valoraciones no coincidían completamente con los presentes en el dataset de libros. Además, se identificaron duplicados en los títulos de los libros. Para resolver esto, se consolidaron los duplicados manteniendo el título único y calculando el promedio de las valoraciones asociadas. Asimismo, se detectó que las columnas de género y número de páginas no estaban presentes y se generaron utilizando modelos de IA para complementar los datos. Por último, se observó que el 91.90% de los book_id en df_ratings no coincidían con los presentes en la biblioteca, lo que significaba que la mayoría de las recomendaciones podrían apuntar a libros desconocidos. Para abordar esto, los book_id desconocidos fueron reemplazados por book_id existentes basados en las preferencias de lectura predominantes del usuario, lo que podría influir significativamente en la calidad de las recomendaciones. Esto implicaba que algunos libros valorados por los usuarios no tenían información asociada en books.csv.

### 2.3 Solución

Se creó un nuevo dataframe (df_ratings_modified) en el cual los book_id desconocidos fueron reemplazados por valores válidos basados en el género predominante de lecturas del usuario. Para los títulos duplicados, se calculó un promedio de las valoraciones asociadas y se mantuvo un único título representativo. Esto aseguró que todas las valoraciones estuvieran alineadas con libros del dataset principal.

Pasos realizados:

- Identificar los book_id válidos en books.csv.
- Crear un mapeo para asignar a cada book_id desconocido un ID válido, basado en la afinidad del usuario hacia ciertos géneros.
- Generar df_ratings_modified como versión actualizada y consistente. Se detectó que el 91.90% de los book_id en df_ratings no coincidían inicialmente con los presentes en la biblioteca, lo que hacía esencial este paso de corrección. Esto aseguraba que todas las recomendaciones se realizaran sobre libros disponibles, utilizando un enfoque basado en las preferencias de género de cada usuario para asignar IDs válidos.

Tras la corrección, todos los book_id en el dataset de valoraciones tenían información correspondiente en books.csv. Este proceso no solo permitió alinear las valoraciones con los libros disponibles, sino que también influyó directamente en la calidad de las recomendaciones, ya que se basaron en géneros preferidos por los usuarios, lo que incrementa la relevancia de las sugerencias generadas por el sistema.

### 2.4 Análisis y filtrado de df_ratings

Al analizar la distribución de las valoraciones por usuario en el dataframe df_ratings, se observa que **hay muchos usuarios con pocas reviews (pico en el mínimo), y la frecuencia decrece a medida que aumenta el número de reviews.** Esta distribución sugiere la necesidad de aplicar un filtrado para eliminar usuarios con un número insuficiente o excesivo de valoraciones, que podrían no aportar información útil o ser considerados *outliers*.

Se aplicaron los siguientes cortes:

- **Corte inferior:** El 25% de los usuarios tienen 3 reviews o menos. Esto justifica un corte mínimo en 3, ya que los usuarios con menos reviews no aportan suficiente información para patrones útiles.
- **Corte superior:** El máximo es 200, y el 75% de los usuarios tienen 22 o menos reviews. Esto sugiere que los usuarios con más de 50 reviews podrían ser

considerados *outliers*.

Tras aplicar el filtrado, los resultados fueron los siguientes:

- Total usuarios antes del filtrado: 53424
- Total usuarios después del filtrado: 40304
- Porcentaje de usuarios eliminados: 24.56%
- Porcentaje de usuarios mantenidos: 75.44%

**Tan solo eliminamos el 25% de los datos, quedándonos con 40.000 usuarios, un número todavía significativo** que mantiene una base de datos robusta para el entrenamiento de los modelos.

## 3. Análisis Exploratorio de Datos (EDA)

Aunque el EDA no es directamente determinante para los modelos de recomendación, permitió extraer insights clave sobre los datos:

- Distribución de valoraciones por usuario.
- Libros más leídos.
- Libros mejor valorados.

Este análisis proporcionó información contextual para entender los patrones de comportamiento de los usuarios y preparar los datos para los modelos.

## 4. Modelos de Recomendación

### 4.1 Recomendación basada en contenido

Se empleó un enfoque basado en las características de los libros, como géneros, autores y descripciones, utilizando técnicas avanzadas como TF-IDF (Term Frequency - Inverse Document Frequency) y Cosine Similarity. TF-IDF permitió representar las descripciones textuales de los libros como vectores numéricos, asignando un peso mayor a términos específicos que diferencian un libro de otro. Posteriormente, Cosine Similarity fue utilizada para calcular la similitud entre los vectores de características de los libros, permitiendo identificar aquellos que son más similares entre sí en función de su contenido. Esto facilitó recomendaciones personalizadas basadas en el perfil de intereses del usuario.

Flujo del modelo:

- Construcción de un vector de características para cada libro, incluyendo géneros y número de páginas generados mediante IA. Estas características enriquecieron los datos existentes y tuvieron un impacto significativo en la calidad del modelo, ya que proporcionaron información adicional clave para identificar similitudes

más precisas entre los libros.
- Cálculo de la similitud (coseno) entre el libro favorito del usuario y el resto.
- Recomendación de libros con mayor similitud al perfil del usuario.

### 4.2 Filtrado colaborativo (User-Based)

El **filtrado colaborativo basado en usuarios (User-Based Collaborative Filtering)** se basa en la idea de que si dos usuarios tuvieron gustos similares en el pasado (calificaron de manera similar un conjunto de ítems), probablemente tendrán gustos similares en el futuro. Es decir, si el usuario A y el usuario B calificaron de forma parecida varios libros, es probable que un libro que le gustó al usuario B (pero que el usuario A aún no ha leído) también le guste al usuario A.

Este enfoque utiliza un modelo que compara las preferencias de un usuario con las de otros para identificar usuarios "similares".

Pasos generales:

- Representar los datos de usuarios y libros como una matriz dispersa.
- Usar un algoritmo para encontrar usuarios con gustos similares (vecinos cercanos).
- Generar recomendaciones basadas en los libros valorados por esos usuarios similares que el usuario objetivo aún no ha calificado.

#### Esparsidad de la matriz

Al trabajar con datos de valoraciones de usuarios y libros, es común encontrarse con una **matriz de valoraciones muy dispersa (sparse)**. Un alto porcentaje de esparsidad significa que la gran mayoría de las celdas de esa tabla están vacías (o contienen un 0, indicando que el usuario no ha calificado ese libro). En nuestro dataset, se observó un **99.87% de esparsidad**, lo que significa que solo el 0.11% de todas las posibles calificaciones en nuestro dataset han sido realmente dadas por los usuarios. **De cada 10,000 posibles combinaciones de usuario y libro, ¡solo 11 tienen una calificación real!** Esta alta esparsidad es un desafío común en sistemas de recomendación y requiere algoritmos capaces de manejarla eficientemente.

#### Evaluación del modelo y métricas

Inicialmente, se exploró el uso del algoritmo k-Nearest Neighbors (kNN) para encontrar usuarios similares. Para una primera aproximación, se utilizó neg_mean_squared_error como métrica de scoring sin realizar una división explícita en conjuntos de entrenamiento y prueba. Sin embargo, **neg_mean_squared_error no es la mejor métrica aquí**.

**¿Por qué neg_mean_squared_error no es la mejor métrica aquí?**

El modelo NearestNeighbors no es un regresor; su propósito principal es encontrar los "vecinos" más cercanos en un espacio de datos, basándose en una métrica de distancia (como coseno, euclidiana, etc.). No predice un valor numérico (como una calificación específica) para un ítem. neg_mean_squared_error se usa para evaluar modelos que sí predicen valores numéricos continuos.

Para evaluar la calidad de las recomendaciones en un sistema de filtrado colaborativo, que busca generar una lista de ítems que probablemente le gusten al usuario, son más apropiadas las métricas que se centran en la precisión de la lista de recomendaciones, como **Precision@k** y **Recall@k**.

Para una evaluación más rigurosa y adecuada, se utilizó la librería **Surprise**. Surprise es una librería de Python diseñada específicamente para construir y analizar sistemas de recomendación de ítems. Proporciona herramientas para cargar datasets de valoraciones, implementar diversos algoritmos de recomendación (incluyendo factorización matricial y métodos basados en vecinos), y evaluar su rendimiento utilizando técnicas como validación cruzada y métricas relevantes para recomendadores.

Se emplearon las métricas **Precision@k** y **Recall@k** para evaluar el rendimiento del modelo, donde k representa el número de recomendaciones consideradas (por ejemplo, los top 10 libros).

1. **Precision@k** se centra en la "pureza" de la lista: ¿cuántas "basura" hay entre tus recomendaciones? Una alta precisión significa que la lista es muy relevante y el usuario no tendrá que "buscar" mucho para encontrar algo que le guste. *De los k (ej. 10) libros que tu sistema le recomendó a un usuario, ¿qué porcentaje de ellos fueron realmente relevantes para ese usuario (es decir, que el usuario valoró positivamente en el conjunto de prueba, por encima de un umbral definido)?*
2. **Recall@k** se centra en la "exhaustividad" o "cobertura": ¿está tu sistema descubriendo todas las cosas que le gustarían al usuario? Un alto recall es importante para la "sorpresa" y para asegurar que el usuario no se pierda nada importante. *De todos los libros que un usuario realmente consideró relevantes (según tu umbral de rating, ej. 4.0 o más en el conjunto de prueba), ¿qué porcentaje de esos libros fueron capturados por tu sistema en el top k (ej. 10) de recomendaciones?*

Precision y Recall, al enfocarse en el "top-k", evalúan directamente la calidad de la lista final que el usuario ve, lo cual es el objetivo principal de un sistema de

recomendación.

**Modelo SVD**

Dentro del enfoque de filtrado colaborativo, se utilizó el algoritmo **Singular Value Decomposition (SVD)** implementado en la librería Surprise. SVD es una técnica de factorización matricial que descompone la matriz de valoraciones usuario-ítem en matrices latentes de menor dimensión. Estas matrices latentes representan características ocultas o factores que influyen en las preferencias de los usuarios y las características de los ítems. Al reducir la dimensionalidad, SVD puede capturar patrones subyacentes en los datos y hacer predicciones de valoraciones para pares usuario-ítem que no han interactuado previamente. SVD es especialmente efectivo para manejar la esparsidad de la matriz de valoraciones.

Los valores de **Precision@10 promedio: 0.8907** y **Recall@10 promedio: 0.9388** obtenidos con el modelo SVD son excepcionalmente altos, especialmente para un dataset con una esparsidad del 99.89%. Esto significa:

1. **Precision@10 (0.8907)**: Aproximadamente el 89% de los 10 libros recomendados por el modelo SVD son realmente libros que el usuario valoró positivamente en el conjunto de prueba (considerando un umbral de relevancia). Esto indica una alta precisión en las recomendaciones principales.
2. **Recall@10 (0.9388)**: El modelo está logrando identificar y recomendar casi el 94% de todos los libros que un usuario valoró positivamente en el conjunto de prueba. Esto indica que el modelo es muy bueno encontrando la mayoría de los "libros relevantes ocultos" para el usuario.

Estos resultados tan elevados sugieren que el modelo SVD, combinado con la preparación de datos y el filtrado aplicado, fue muy efectivo para capturar los patrones de preferencia de los usuarios y generar recomendaciones de alta calidad en este dataset.

Resultados y observaciones

- Variación en número de recomendaciones: Dependiendo del perfil del usuario (número y variedad de libros valorados), el sistema podía sugerir más o menos libros. Esto es consistente con la naturaleza del filtrado colaborativo.
- Interpretación práctica: Los libros recomendados son aquellos que los usuarios similares calificaron altamente, pero que el usuario objetivo no ha explorado.

Posibilidad de integrar datos demográficos
Los datos adicionales como edad y sexo podrían ser utilizados para influir en la asignación de book_id basados en las preferencias de género de los usuarios, lo que podría mejorar la

personalización. Además, podrían contribuir a:

- Crear clusters iniciales de usuarios.
- Ajustar recomendaciones para reflejar patrones demográficos.

## 5. Flujo de trabajo del recomendador interactivo

Para crear el recomendador de libros interactivo, se siguió un flujo de trabajo que integra los modelos desarrollados con una interfaz de usuario simple. El proceso general implica:

1. **Carga y preparación de datos:** Se cargan los datasets (books.csv, ratings.csv, etc.) y se aplican los pasos de limpieza y filtrado descritos en la Sección 2. Se prepara el dataset de valoraciones en el formato requerido por la librería Surprise.
2. **Entrenamiento del modelo:** Se entrena el modelo SVD utilizando el dataset de valoraciones preparado. Este modelo aprende las matrices latentes que representan las preferencias de usuarios y las características de los libros.
3. **Implementación de la lógica de recomendación:** Se desarrolla la función o el código que, dado un user_id, utiliza el modelo SVD entrenado para predecir las valoraciones que ese usuario daría a los libros que aún no ha leído.
4. **Generación de recomendaciones:** Se ordenan las predicciones de valoraciones de forma descendente y se seleccionan los top-N libros con las predicciones más altas como recomendaciones para el usuario. Se excluyen los libros que el usuario ya ha valorado.
5. **Integración con la interfaz de usuario:** Se crea una interfaz simple (por ejemplo, utilizando *streamlit* o una aplicación web básica) donde el usuario puede ingresar su user_id. Al enviar el ID, la aplicación ejecuta la lógica de recomendación y muestra la lista de libros recomendados, posiblemente incluyendo información adicional como título, autor y género obtenida del dataset books.csv.
6. **Visualización de resultados:** La interfaz presenta la lista de libros recomendados de manera clara y legible para el usuario.

Este flujo permite que el usuario interactúe con el sistema, reciba recomendaciones personalizadas basadas en el modelo SVD entrenado y explore nuevos libros.

## 6. Conclusiones y mejoras futuras

El sistema desarrollado demuestra que es posible generar recomendaciones efectivas utilizando datos existentes de valoraciones y descripciones de libros.

Principales logros

- Limpieza de datos para garantizar consistencia.
- Implementación de dos enfoques complementarios de recomendación.
- Evaluación del modelo colaborativo utilizando métricas adecuadas como Precision@k y Recall@k.
- Obtención de resultados prometedores con el modelo SVD.

Mejoras futuras

- Considerar un modelo híbrido que combine el enfoque basado en contenido y el filtrado colaborativo para aprovechar las fortalezas de ambos.
- Explorar otros algoritmos de recomendación avanzados.
- Integrar datos demográficos y de descripciones de libros de manera más profunda en los modelos.
- Realizar pruebas de usuario para obtener feedback cualitativo sobre la relevancia de las recomendaciones.

El proyecto establece una base sólida para extender y personalizar sistemas de recomendación en contextos prácticos.