**Presentación: Sistema de Recomendación de Libros**

**Diapositiva 1: Presentación**

*Bienvenidos a la presentación del proyecto de ML, que es un sistema de recomendación de libros. En esencia, buscamos cómo podemos ayudar a los lectores a descubrir su próxima gran lectura, utilizando los algoritmos de recomendación y el análisis de datos.*

**Diapositiva 2: Contenido**

*………………………………………….*

**Diapositiva 3: Recomendador Basado en Contenido**

**¿Qué es?**

*El modelo basado en contenido recomienda libros analizando las características de los libros que un usuario ya conoce y le gustan, buscando similitudes entre ellos. Se enfoca exclusivamente en los libros y no considera las preferencias de otros usuarios.*

*La lógica es: si te gustó un libro con ciertas características (género, autor, palabras clave en su descripción), es probable que te interesen otros libros que comparten esas mismas características.*

**Técnicas Utilizadas:**

***TF-IDF (Term Frequency - Inverse Document Frequency):*** *Esta* ***técnica*** *nos ayuda a convertir el texto de las descripciones o géneros de los libros en vectores numéricos. Le da más importancia a las palabras que son relevantes para un libro específico.*

*Si estás analizando la sinopsis de un libro, palabras comunes como "el" o "un" tendrán menos importancia, mientras que palabras específicas como "fantasía" o "dragón" tendrán más peso.*

***Cosine Similarity:*** *Una vez que tenemos los libros representados como vectores, usamos la similitud de coseno para medir qué tan "cercanos" o similares son dos libros en ese espacio vectorial. Si el ángulo entre los vectores es pequeño (similaridad alta), significa que los libros tienen características parecidas*

**Diapositiva 4: Flujo de trabajo**

1. ***Representación de Libros:*** *Primero, procesamos la información de cada libro (géneros, número de páginas, autor, título) y creamos un vector de características para cada uno, utilizando TF-IDF para el texto.*
2. ***Perfil del Usuario:*** *Identificamos las características del libro.*
3. ***Cálculo de Similitud:*** *Calculamos la similitud de coseno entre el vector del libro y los vectores de todos los demás libros disponibles en nuestra colección.*
4. ***Generación de Recomendaciones:*** *Ordenamos los libros según su puntuación de similitud de mayor a menor. La lista de recomendados serán los libros con mayor similitud.*

**Diapositiva 5: Filtrado Colaborativo (User-Based)**

1. ***¿Qué es?***

* *El objetivo es encontrar usuarios con gustos similares. Si dos usuarios califican libros de manera parecida, se asume que tienen gustos similares. Recomendamos libros que un usuario similar haya calificado alto, pero que el usuario objetivo no haya leído aún. La premisa es: "Si a personas con gustos similares a los tuyos les gustó un libro, es muy probable que a ti también te guste, aunque no lo conozcas."*
* *Mira los patrones de comportamiento y las preferencias de los usuarios.*

***El Gran Desafío: La Esparsidad de los Datos***

* *Nuestros datos de valoraciones son extremadamente dispersos. La matriz donde las filas son usuarios y las columnas son libros, y las celdas son las valoraciones, está casi vacía.*
* *En nuestro caso, solo el 0.11% de todas las posibles combinaciones usuario-libro tienen una calificación. Esta alta esparsidad hace que sea difícil encontrar usuarios con suficientes valoraciones en común para determinar su similitud de forma directa.*

1. ***Midiendo lo que Realmente Importa***

* *Inicialmente, podríamos pensar en usar métricas como el neg\_mean\_squared\_error, que mide la precisión de una predicción numérica exacta.*
* ***¿Por qué neg\_mean\_squared\_error no es la mejor métrica aquí?*** *Porque nuestro objetivo principal no es predecir si un usuario dará un 4.2 o un 4.5 a un libro, sino generar una lista de libros que probablemente le gusten. El error numérico exacto no refleja bien la calidad de esa lista.*

**Diapositiva 6: Filtrado Colaborativo (User-Based)**

1. ***Métricas clave para recomendaciones***

***Precision@k:*** *"De los top-k libros que recomendé, ¿cuántos fueron realmente relevantes para el usuario?" Mide la "pureza" de la lista.*

***Recall@k:*** *"De todos los libros que al usuario le gustaron, ¿cuántos logré incluir en mi lista top-k de recomendaciones?" Mide la "exhaustividad" o "cobertura".*

*Estas métricas son cruciales porque evalúan directamente la experiencia final del usuario: la calidad de la lista de libros que ve.*

1. **Modelos para Encontrar Patrones en Datos Dispersos:**

***k-Nearest Neighbors (kNN):*** *Inicialmente, se consideró usar kNN directamente. kNN busca los 'k' usuarios más cercanos basándose en la similitud de sus valoraciones. Sin embargo, con la alta esparsidad, encontrar vecinos con suficientes valoraciones en común es complicado.*

***SVD (Singular Value Decomposition):*** *Este es el modelo principal que utilizamos. SVD es una técnica de factorización matricial muy potente para datos dispersos. Lo que hace es 'descomponer' nuestra matriz de valoraciones en matrices más pequeñas que representan factores latentes o características ocultas tanto de los usuarios como de los libros.*

**Diapositiva 7: Filtrado Colaborativo (User-Based)**

Los resultados fueron muy positivos, especialmente considerando la esparsidad.

**Precision@10 promedio: 0.8907**. Esto significa que, en promedio, casi 9 de cada 10 libros que recomendamos en el top 10 fueron libros que al usuario le gustaron en el conjunto de prueba.

**Recall@10 promedio: 0.9388**. Esto es impresionante; el modelo logró identificar y recomendar casi el 94% de todos los libros que al usuario le gustaron en el conjunto de prueba.

Estos valores tan altos demuestran que SVD fue muy efectivo para capturar los patrones de preferencia de los usuarios a pesar de la esparsidad.

**Diapositiva 8: Filtrado Colaborativo (User-Based)**

**Flujo de Trabajo Detallado:**

1. **Preparación de la Matriz de Valoraciones:** Organizamos los datos de usuario, libro y calificación en una estructura que el modelo pueda entender, conceptualmente una matriz dispersa.
2. **Entrenamiento del Modelo SVD:** Aplicamos el algoritmo SVD a esta matriz. El modelo aprende los vectores latentes para cada usuario y libro, capturando las características subyacentes de sus preferencias e ítems.
3. **Inferencia de Similitud:** El modelo SVD nos permite inferir qué usuarios son similares basándonos en la cercanía de sus vectores latentes.
4. **Predicción de Valoraciones para Libros No Leídos:** Para un usuario específico, utilizamos su vector latente y los vectores latentes de todos los libros que *no* ha leído para predecir qué calificación les daría.
5. **Generación de la Lista de Recomendaciones:** Ordenamos estas predicciones de valoraciones de mayor a menor.
6. **Filtrado y Presentación:** Seleccionamos los top-N libros con las predicciones más altas. **Un paso CRUCIAL aquí es asegurarnos de que la lista final solo contenga libros que el usuario aún no ha leído.** Presentamos esta lista al usuario.

**Manejo del Cold Start (Usuarios Nuevos):**

* Un desafío importante para el filtrado colaborativo es el problema del 'cold start', especialmente con usuarios completamente nuevos de los que no tenemos ninguna valoración. Sin datos, el modelo no puede encontrarles vecinos ni aprender sus factores latentes.
* Para abordar esto, una estrategia común es pedir a los usuarios nuevos que proporcionen una **pequeña valoración inicial** de algunos libros populares o de diferentes géneros al registrarse. Esto le da al sistema una base mínima para empezar a inferir sus gustos y generar las primeras recomendaciones, que irán mejorando a medida que el usuario interactúe más con la plataforma y califique más libros.

**Diapositiva 9: Conclusiones**

**Complicaciones:**

* La **calidad inicial de los datos** y la necesidad de limpieza profunda.
* La **alta esparsidad** de la matriz de valoraciones, un desafío común.

**Limitaciones**

* El problema del **'cold start'** para usuarios y libros nuevos.

**Mejoras Futuras:**

* Implementar un **modelo híbrido** que combine contenido y filtrado colaborativo para aprovechar lo mejor de ambos mundos y mitigar el 'cold start'.
* Explorar algoritmos más avanzados o técnicas de *deep learning* para capturar patrones más complejos.
* Integrar **más datos contextuales** (historial de navegación, tiempo de lectura, reseñas) para enriquecer los perfiles de usuario y libro.
* Realizar **estudios de usuario** para validar la calidad de las recomendaciones en un entorno real y obtener feedback cualitativo.
* Mejorar la **interfaz de usuario** para una experiencia más intuitiva.

**Naturaleza de SVD:** SVD aprende patrones latentes de las calificaciones de los usuarios. No entiende explícitamente "género". Si el modelo ha aprendido que los usuarios a los que les gusta la ciencia ficción también suelen calificar bien libros de fantasía, ensayo o romance (porque hay esos patrones en el conjunto de datos de entrenamiento), SVD seguirá esas correlaciones, incluso si el nuevo usuario solo ha calificado ciencia ficción. SVD busca maximizar la similitud de los factores latentes entre usuarios y libros, no la correspondencia explícita de atributos como el género.