Sistema de recomendacion de peliculas y series

Santiago García Tirado, Yohel Pérez, Jonatan Restrepo
Departamento de Ingeniería de Sistemas y Computación
Universidad de Antioquia
Proyecto de Curso: Modelos y Simulación II
https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset

I. Descripción del problema

I-A. Contexto y motivación

En plataformas de contenidos (streaming, catálogos multimedia) hay una gran cantidad de títulos disponibles; ayudar a cada usuario a encontrar películas y series relevantes mejora la experiencia, retención y, en algunos casos, ingresos.

El objetivo del proyecto es desarrollar un sistema de recomendación que prediga la calificación que un usuario daría a una película o serie y, a partir de ello, sugerir títulos que probablemente le gusten.

En este contexto, los sistemas de recomendación basados en **filtrado colaborativo** permiten aprovechar las similitudes entre usuarios o entre ítems, ofreciendo una forma eficiente de personalizar la experiencia y reducir la sobrecarga de información.

I-B. Composición de la base de datos

El proyecto utilizará el conjunto de datos **The Movies Dataset** de Kaggle, un dataset que combina información de *The Movie Database (TMDB)* y *MovieLens*. Los archivos principales que se utilizarán son:

- **movies_metadata.csv:** Contiene información sobre 45,000 películas, incluyendo nombre, IDs, carteles, fondos, presupuesto, ingresos, fechas de lanzamiento, idiomas, países de producción y empresas.
- **ratings.csv:** Cada línea de este archivo, después de la fila del encabezado, representa una calificación de una película por parte de un usuario. Las calificaciones se realizan en una escala de 5 estrellas, con incrementos de media estrella (0.5–5.0 estrellas).
- **links.csv:** Contiene los IDs de las películas en las bases de datos TMDB (*The Movie Database*) e IMDB (*Internet Movie Database*). Esta base de datos relaciona `ratings.csv` con `movies_metadata.csv`.

Análisis exploratorio inicial: Un análisis preliminar sobre una muestra del archivo `ratings.csv` (la versión "small") revela:

- Número de muestras: aproximadamente 100,000 calificaciones.
- Número de usuarios únicos: 671.
- Número de películas únicas calificadas: 9,066.

Significado de variables:

- `userId`: identificador numérico único para cada usuario.
- `movieId`: identificador numérico único para cada película.
- `rating`: valor numérico (flotante) que representa la calificación.

Datos faltantes: Se ha identificado que un pequeño porcentaje de las películas en `ratings.csv` no tienen una correspondencia directa en `movies_metadata.csv`. La estrategia inicial para manejar esta inconsistencia será eliminar los registros de calificaciones para los cuales no se encuentren metadatos, ya que representan una fracción mínima del total.

Codificación de variables: Las variables `userId` y `movieId` son categóricas por naturaleza, pero se tratarán como identificadores numéricos. La variable objetivo, `rating`, es numérica y continua dentro de su escala, lo que define la naturaleza del problema.

Variables principales:

- `ratings.csv`: userId, movieId, rating, timestamp.
- `movies_metadata.csv`: id, adult, belongs_to_collection, budget, genres, homepage, imdb_id, original_language, original_title, overview, popularity, poster_path, production_companies, production_countries, release_date, revenue, runtime, spoken_languages, status, tagline, title, video, vote_coverage, vote_count.
- `links.csv`: movieId, imdbId, tmdbId.

Análisis de distribución: Se realizó un análisis de los datos para entender cómo están distribuidos. Los resultados mostrados a continuación se detallan en el repositorio del proyecto.

A partir de las gráficas y estadísticas se concluye que:

- No todas las películas tienen una cantidad parecida de reseñas; la película con más calificaciones es la ID 356, con 350 calificaciones.
- Las calificaciones no están distribuidas uniformemente entre los usuarios: el usuario 547 ha dado alrededor de 2,400 calificaciones.
- La mayoría de usuarios otorga calificaciones cercanas a 4 estrellas.

- El 47 % (4,265) de las películas tienen entre 1 y 2 calificaciones.
- El 53 % restante de las películas tienen más de 3 calificaciones.
- El histograma muestra que la mayoría de usuarios (427, equivalente al 63 % del total de 671) tienen 50 o más calificaciones.

I-C. Limpieza de datos

La relación entre los archivos del dataset es la siguiente:

- `Ratings (movieId)` → `Links (movieId)`.
- `Links (tmdbId)` → `Movies (id)`.

Se observó que la columna `id` en `movies_metadata.csv` era de tipo `object`, por lo que se convirtió a tipo `Int64` para poder realizar correctamente la relación entre `Links` y `Movies`.

Para obtener un dataset más limpio se realizó el siguiente proceso:

- Se analizaron duplicados, registros faltantes y datos vacíos.
- Se encontraron 29 registros duplicados en `movies_metadata` con el mismo ID; luego de eliminarlos quedaron 45,433 películas únicas.
- No se encontraron duplicados en `ratings.csv` ni en `links.csv`.
- Se identificaron 148 calificaciones que hacen referencia a películas sin información válida en `movies_metadata`.
- Se descartaron las películas que no han sido calificadas por ningún usuario.

I-D. Paradigma de aprendizaje

El problema de predecir la calificación que un usuario le dará a una película se enmarca dentro del paradigma de **aprendizaje supervisado** y, específicamente, se aborda como un problema de **regresión**.

Dado que se busca predecir una variable continua (`rating`) en función de variables como el ID de usuario, ID de película, género, idioma, etc., y la relación entre ellas no es lineal ni simple, se requiere un modelo capaz de capturar relaciones no lineales sin sobreajustar.

Entre los enfoques clásicos de filtrado colaborativo, el algoritmo **k-vecinos más cercanos (k-NN)** ofrece una alternativa interpretable y eficiente, ya que estima la calificación de un usuario para una película en función de las calificaciones de los usuarios más similares.

Estudios recientes (Bohra et al., 2023; Alsekait et al., 2024) destacan que las estrategias basadas en k-NN o su integración con factorización de matrices (SVD, ALS) siguen siendo competitivas frente a modelos más complejos, especialmente en datasets con dispersión de ratings o usuarios con pocas interacciones.

En este proyecto, se toma k-NN como punto de partida para implementar un modelo de recomendación colaborativo que sirva como base para posibles extensiones híbridas en futuras iteraciones.

Métricas de evaluación:

- **Error Cuadrático Medio Raíz (RMSE - Root Mean Squared Error):** métrica principal. Se calcula como la raíz cuadrada del promedio de los errores al cuadrado. Penaliza severamente los errores grandes, lo cual es deseable para evitar predicciones que se desvíen drásticamente de la calificación real.
- **Error Absoluto Medio (MAE):** métrica adicional, más robusta ante valores atípicos.

II. Estado del arte

II-A. Artículo A — Leveraging Hybrid Deep Learning for Enhanced Movie Recommendation (D. M. Alsekait et al., 2024)

Este estudio compara SVD, ALS, Autoencoders, NCF y otros métodos, evaluándolos en varios datasets incluyendo MovieLens y *The Movies Dataset* [1].

El paradigma empleado es **comparativo experimental**: una comparación empírica entre modelos tradicionales de factorización (SVD/ALS), métodos basados en redes neuronales (Autoencoders, Neural Collaborative Filtering) y enfoques híbridos.

Técnicas aplicadas:

- **Matrix Factorization (ALS, SVD).**
- **Neural Collaborative Filtering (NCF).**
- **Autoencoders** para extracción de embeddings de usuario/ítem.
- **Modelos híbridos** que incorporan características de contenido (género, metadata) en capas adicionales.

Metodología de validación: División en conjuntos *train/validation/test*; métricas RMSE/MAE para predicción de calificaciones y Precision/Recall/NDCG para top-k. Se realizaron repeticiones con diferentes semillas (*seeds*) y se reportaron promedios y desviaciones para asegurar robustez.

Resultados clave: No existe un "mejor modelo absoluto"; cada algoritmo tiene ventajas según la densidad del dataset y la disponibilidad de atributos de contenido.

Conclusión: SVD sobresale en MovieLens, mientras que los Autoencoders y NCF pueden mejorar el rendimiento en datasets con características ricas como *The Movies Dataset*. Los autores recomiendan usar factorización (SVD/ALS) como baseline fuerte, y luego probar modelos híbridos para mejoras específicas.

II-B. Artículo B — Hybrid Machine Learning Based Recommendation Algorithm for Multiple Movie Dataset (Bohra et al., 2023)

El trabajo de Bohra, Gaikwad y Singh *Hybrid Machine Learning Based Recommendation Algorithm for Multiple Mo-*

vie Dataset [2] propone un enfoque **híbrido** que combina filtrado colaborativo, factorización de matrices y señales de contenido, y lo evalúa en múltiples datasets de películas (incluyendo TMDB 5000 y MovieLens).

Paradigma: híbrido supervisado con evaluación comparativa experimental entre enfoques individuales (SVD/MF, k-NN, K-means) y su combinación ponderada.

Técnicas aplicadas:
- **Matrix Factorization (MF) / SVD**: representación usuario–ítem en factores latentes.
- **k-Nearest Neighbors (k-NN)**: similitud local entre usuarios/películas para refinar vecindarios.
- **K-means**: agrupamiento para capturar estructuras globales y diversidad en preferencias.
- **Modelo híbrido ponderado**: combina puntuaciones de SVD/MF, similitud k-NN y popularidad; los *weights* se ajustan empíricamente.

Metodología de validación:
- Partición 80/20 en entrenamiento/prueba (y validación separada cuando aplica).
- Métricas de regresión: **RMSE** y **MAE** para predicción de *ratings*.
- Repeticiones en múltiples datasets para evaluar consistencia y robustez.

Resultados clave:
- El **enfoque híbrido** reduce RMSE/MAE respecto a modelos individuales (SVD, k-NN o K-means por separado).
- La combinación de **factores latentes (SVD)** con **similitud local (k-NN)** mejora la precisión, en especial ante *cold-start* y usuarios con pocas calificaciones.
- Recomiendan ajustar los *weights* del híbrido según densidad del dataset y dispersión de *ratings*.

Conclusión: Los modelos híbridos son una alternativa práctica y efectiva en escenarios reales: aprovechan la capacidad de generalización de SVD/MF y la personalización fina de k-NN, ofreciendo un balance entre precisión y cobertura. Para este proyecto, el artículo respalda usar **SVD como baseline** y, posteriormente, **integrarlo con k-NN** (y señales de popularidad o contenido) para mejorar la calidad de las recomendaciones.

II-C. Artículo C — Matrix Factorization Techniques for Recommender Systems (Koren, Bell & Volinsky, 2009)

Este artículo es uno de los trabajos más influyentes en sistemas de recomendación contemporáneos. Los autores presentaron una serie de modelos basados en **factorización de matrices (MF)**, específicamente **SVD y SVD++**, diseñados para mejorar la predicción de calificaciones en plataformas como Netflix y MovieLens [3].

Paradigma: aprendizaje supervisado enfocado en regresión, donde la variable objetivo es el `rating` y los factores latentes se aprenden mediante modelos paramétricos regularizados.

Técnicas aplicadas:
- **SVD y SVD++**: incorporan información explícita e implícita del usuario.
- **Modelos temporales**: ajustes dinámicos basados en la variación del comportamiento del usuario en el tiempo.
- **Regularización avanzada**: evita sobreajuste y mejora la generalización incluso en bases de datos dispersas.

Metodología de validación: Utilizan divisiones *train/test* similares a las del *Netflix Prize* y MovieLens, incorporando validación cruzada y análisis comparativo entre SVD, k-NN y modelos basados en contenido.

Métricas empleadas:
- **RMSE**: métrica principal, utilizada como estándar en competencias de recomendación.
- **MAE**: métrica secundaria para robustez.

Resultados clave: Los modelos basados en factorización logran reducciones sustanciales en RMSE respecto a métodos tradicionales como k-NN. En particular, SVD++ muestra mejoras significativas cuando se incorporan señales implícitas. Además, los autores demuestran que los modelos híbridos MF + contenido pueden superar limitaciones de *cold-start*.

Conclusión: Este trabajo respalda firmemente el uso de SVD como baseline sólido y demuestra que los modelos con factores latentes pueden capturar mejor las relaciones no lineales entre usuarios y películas frente a métodos clásicos como k-NN.

II-D. Artículo D — Recommender Systems Survey (Bobadilla et al., 2013)

Bobadilla, Ortega, Hernando y Gutiérrez presentan una revisión exhaustiva de los **sistemas de recomendación**, con énfasis en técnicas de **filtrado colaborativo**, modelos basados en contenido y enfoques híbridos [4]. Aunque se trata de un artículo de tipo survey, incluye resultados experimentales y comparaciones directas sobre datasets estándar como MovieLens, altamente relacionados con el problema abordado en este proyecto.

Paradigma: revisión comparativa de métodos supervisados y no supervisados aplicados a recomendación de ítems, principalmente películas y productos en plataformas en línea.

Técnicas analizadas:
- **Filtrado colaborativo basado en memoria**: k-NN usuario-usuario e ítem-ítem, con diversas medidas de similitud (Pearson, Coseno, etc.).
- **Filtrado colaborativo basado en modelo**: factorización de matrices (SVD, modelos probabilísticos).
- **Sistemas basados en contenido**: perfiles de usuario construidos a partir de características de ítems.
- **Modelos híbridos**: combinaciones CF + contenido o CF + conocimiento contextual.

Metodología de validación y métricas: El artículo resume prácticas estándar de validación, incluyendo particiones train/test y validación cruzada, y destaca el uso de:

- **RMSE y MAE**: para evaluar predicción de ratings.
- **Precision@k, Recall@k y F1**: para evaluar listas de recomendación top-N.
- **Cobertura y novedad**: como métricas complementarias de calidad del sistema.

Resultados y conclusiones relevantes: Los autores concluyen que los métodos basados en modelo, en particular la factorización de matrices, suelen superar a los enfoques basados en memoria (k-NN tradicional) en precisión de predicción cuando se dispone de suficientes datos. Sin embargo, resaltan que k-NN sigue siendo competitivo en escenarios con menos datos por usuario y que los enfoques híbridos ofrecen un mejor compromiso entre precisión, cobertura y capacidad de manejar *cold-start*. Para este proyecto, el survey de Bobadilla et al. sirve como soporte teórico global: justifica el uso de k-NN como punto de partida y de técnicas de factorización de matrices como posibles extensiones futuras del sistema de recomendación.

III. Metodología

Esta sección describe el proceso de preparación de datos, la selección del modelo, la estrategia de validación y las métricas empleadas para evaluar el desempeño del sistema de recomendación.

III-A. Modelo de aprendizaje seleccionado y justificación

Tras una fase exploratoria inicial se comparó el desempeño de los algoritmos k-Vecinos más Cercanos (k-NN) y la Descomposición en Valores Singulares (SVD). Con base en los resultados obtenidos, se seleccionó **SVD** como el modelo principal para este proyecto.

Justificación: SVD, una técnica de factorización de matrices, mostró un rendimiento superior en las métricas de error (**RMSE** y **MAE**) y menores tiempos de entrenamiento y predicción en comparación con k-NN. El modelo opera bajo la premisa de que las preferencias de los usuarios pueden explicarse mediante un número reducido de factores latentes (relacionados con elementos como géneros, directores, actores o estilos narrativos), los cuales se infieren automáticamente a partir de la matriz de calificaciones.

Además, SVD descompone la matriz original de interacciones Usuario–Ítem en dos matrices de menor dimensión, lo que le permite:

- **Generalizar mejor:** puede predecir calificaciones para pares usuario–película inexistentes en el dataset, utilizando los factores latentes aprendidos.
- **Manejar la escasez de datos (*sparsity*):** a diferencia de k-NN, que depende de vecinos con películas en común, SVD es robusto en matrices dispersas.
- **Ser computacionalmente eficiente:** una vez entrenado, la predicción es rápida al reducirse a un producto punto entre vectores latentes.

III-B. Estrategia de validación y optimización de hiperparámetros

Para garantizar una evaluación objetiva del modelo se usó la siguiente estrategia:

División de datos (*Hold-Out*): El conjunto `ratings_small.csv` se dividió en dos subconjuntos:

- **Entrenamiento (80 %):** utilizado para ajustar el modelo y aprender factores latentes.
- **Prueba (20 %):** utilizado exclusivamente para evaluar desempeño final en datos no vistos.

Optimización de hiperparámetros (*Grid Search*): Se realizó una búsqueda en malla mediante `GridSearchCV` con validación cruzada de 3 pliegues sobre el conjunto de entrenamiento. Los hiperparámetros explorados fueron los siguientes:

Cuadro I
HIPERPARÁMETROS ANALIZADOS PARA EL MODELO SVD

Hiperparámetro	Descripción
`n_factors`	Número de factores latentes a aprender.
`n_epochs`	Número de iteraciones del entrenamiento.
`lr_all`	Tasa de aprendizaje del optimizador.
`reg_all`	Regularización para prevenir sobreajuste.

Configuración óptima: Tras el proceso de búsqueda en malla, se identificó la siguiente combinación de hiperparámetros que minimiza el RMSE:

Cuadro II
HIPERPARÁMETROS ÓPTIMOS PARA EL MODELO SVD

Hiperparámetro	Valor Óptimo
`n_factors`	200
`n_epochs`	50
`lr_all`	0.01
`reg_all`	0.1

Métrica de evaluación: La métrica principal para seleccionar hiperparámetros y evaluar el modelo final fue **RMSE**, complementada por **MAE** para una interpretación más directa del error.

III-C. Experimentos y resultados iniciales

En la fase inicial se aplicó una validación cruzada de 5 pliegues para comparar k-NN y SVD usando sus configuraciones por defecto. Los resultados promedio obtenidos se resumen en la Tabla III.

Cuadro III
COMPARACIÓN INICIAL ENTRE K-NN Y SVD (CONFIGURACIÓN POR DEFECTO)

Algoritmo	RMSE promedio	MAE promedio
k-NN	0.928	0.711
SVD	0.897	0.690

Como se observa, **SVD obtiene un error menor** en ambas métricas y un tiempo de entrenamiento considerablemente más

bajo. Este resultado respalda la elección de SVD como modelo principal para las siguientes etapas del proyecto.

Finalmente, tras la optimización de hiperparámetros, el modelo SVD alcanzó un desempeño robusto con **RMSE = 0.875** y **MAE 0.672** en el conjunto de prueba, confirmando la eficacia de la factorización de matrices para este problema de recomendación basado en calificaciones, el MAE por su lado nos da que en promedio, las predicciones del modelo se desvían en 0.67 estrellas de la calificación real otorgada por los usuarios.

IV. Análisis de resultados

El modelo SVD optimizado logró un rendimiento robusto con un **RMSE de 0.875** y un **MAE de 0.672** en el conjunto de prueba. Este resultado es competitivo para un problema de recomendación basado en filtrado colaborativo y confirma la eficacia de la factorización de matrices para predecir calificaciones en un dataset disperso como *The Movies Dataset*.

La superioridad de SVD frente a k-NN se evidenció desde los experimentos iniciales. En la comparación con validación cruzada de 5 pliegues utilizando configuraciones por defecto, SVD obtuvo menor error (RMSE promedio de 0.896 y MAE de 0.690) frente a k-NN (RMSE promedio de 0.927 y MAE de 0.710). Esta ventaja se explica porque SVD aprende representaciones densas mediante factores latentes, capturando patrones complejos de preferencia usuario–película, mientras que k-NN depende directamente de coincidencias locales entre usuarios o ítems, las cuales son escasas cuando la matriz de calificaciones es altamente dispersa.

Tras el proceso de optimización de hiperparámetros, el modelo mejoró su capacidad de generalización. El RMSE final indica que, en promedio, la desviación típica del error de predicción es cercana a 0.87 estrellas, lo cual representa una predicción razonablemente cercana a la calificación real dentro de la escala [0.5, 5.0]. Por su parte, el MAE permite interpretar el error de forma más directa: las predicciones difieren en promedio alrededor de 0.67 estrellas de la calificación real otorgada por los usuarios.

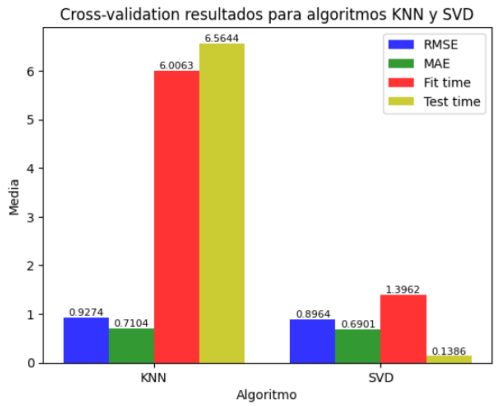

Figura 1. Curva de aprendizaje del modelo SVD.

IV-A. Limitaciones

A pesar del buen desempeño, el modelo presenta una limitación inherente a los métodos colaborativos: el problema de *cold-start*. Esto ocurre cuando se intenta recomendar para usuarios nuevos (sin historial de calificaciones) o ítems nuevos (películas sin ratings), casos en los que el modelo no dispone de suficiente información para inferir factores latentes confiables. Esta limitación abre la posibilidad de explorar modelos híbridos que combinen contenido y señales colaborativas en futuras etapas del proyecto.

Referencias

[1] D. M. Alsekait *et al.*, "Leveraging hybrid deep learning for enhanced movie recommendation," *Natural Sciences Publishing*, 2024. [Online]. Available: https://www.naturalspublishing.com/files/published/3nw1w641kciz46.pdf

[2] S. Bohra, A. Gaikwad, and G. Singh, "Hybrid machine learning based recommendation algorithm for multiple movie dataset," *Indian Journal of Science and Technology*, vol. 16, no. 35, 2023. [Online]. Available: https://indjst.org/articles/hybrid-machine-learning-based-recommendation-algorithm-for-multiple-movie-dataset

[3] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Computer*, vol. 42, no. 8, pp. 30–37, 2009. [Online]. Available: https://datajobs.com/data-science-repo/Recommender-Systems-[Netflix].pdf

[4] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-Based Systems*, vol. 46, pp. 109–132, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705113001044