Proyecto Final

Inteligencia Artificial 2

Sistema de recomendación híbrido

Alumno:
Andrés Maglione

Profesores:
Dr. Rodrigo González
Dr. Jorge Guerra

Introducción

Los sistemas de recomendación se han convertido en herramientas clave para personalizar la experiencia del usuario en plataformas de música, películas y comercio electrónico, entre otros. Estos sistemas permiten sugerir contenido relevante al usuario basándose en su comportamiento pasado, el contenido que consume y la similitud con otros usuarios. Sin embargo, el problema del *cold start* —donde nuevos usuarios o elementos no tienen suficiente información inicial— sigue siendo un desafío significativo.

Este proyecto aborda el desarrollo de un sistema de recomendación híbrido que combina técnicas de filtrado colaborativo y modelos basados en contenido, utilizando datos reales de la red social Last.fm. A través de este enfoque, se busca generar recomendaciones de artistas musicales basadas en ratings implícitos, afrontando los problemas que suelen surgir en sistemas de este tipo.

Objetivos

1. **Objetivo general:**
   Desarrollar un sistema de recomendación híbrido que combine modelos de filtrado colaborativo y modelos basados en contenido para sugerir artistas a partir de ratings implícitos.

2. **Objetivos específicos:**
   - Implementar un modelo de *matrix factorization* para predecir interacciones usuario-artista.
   - Incorporar un modelo basado en contenido para abordar el problema del *cold start*.
   - Evaluar el desempeño de los modelos utilizando métricas como Precision@k y AUC.
   - Explorar iteraciones adicionales como el uso de relaciones de amistad entre usuarios y datos temporales para recomendaciones más dinámicas.

Dataset

El dataset utilizado será el **hetrec2011-lastfm-2k**, que incluye información sobre redes sociales, etiquetado, y datos de escucha musical de 1,892 usuarios de Last.fm.
Características principales del dataset:

- **Usuarios**: 1,892.
- **Artistas**: 17,632.

- **Relaciones de amistad**: 12,717 bidireccionales.
- **Relaciones usuario-artista**: 92,834 con datos de recuentos de escucha.
- **Etiquetas**: 11,946 únicas, con 186,479 asignaciones de etiquetas.
- **Archivos clave**:
    - `user_artists.dat` para relaciones usuario-artista y conteos de escucha.
    - `user_friends.dat` para conexiones sociales.
    - `user_taggedartists.dat` para asignaciones de etiquetas.
    - `tags.dat` y `artists.dat` para información adicional.

Es importante resaltar que este dataset contiene ratings implícitos.

Desarrollo

Análisis exploratorio de datos

Para comenzar con el proyecto, se realizó un análisis exploratorio de datos para poder observar la estructura de los mismos. Como se mencionó anteriormente, el conjunto de datos cuenta con dos archivos principales de interacciones de usuarios (cantidad de escuchas a artistas y *tags* asignados a un artista) y dos archivos con datos específicos de estos artistas y tags. También se incluye un archivo con datos de amistad entre usuarios, que por motivos de alcance no fue utilizado en este proyecto.

El proyecto tiene como objetivo predecir si a un usuario le gusta un artista o no (es decir, se centra en los datos de cantidad de reproducciones), por lo que la mayor parte del análisis datos se centra en dicho dataset. El dataset de reproducciones incluye los 50 artistas más escuchados de cada usuario, por lo que se puede observar un claro desbalance entre los artistas más populares, mientras que existen algunos con pocas reproducciones.

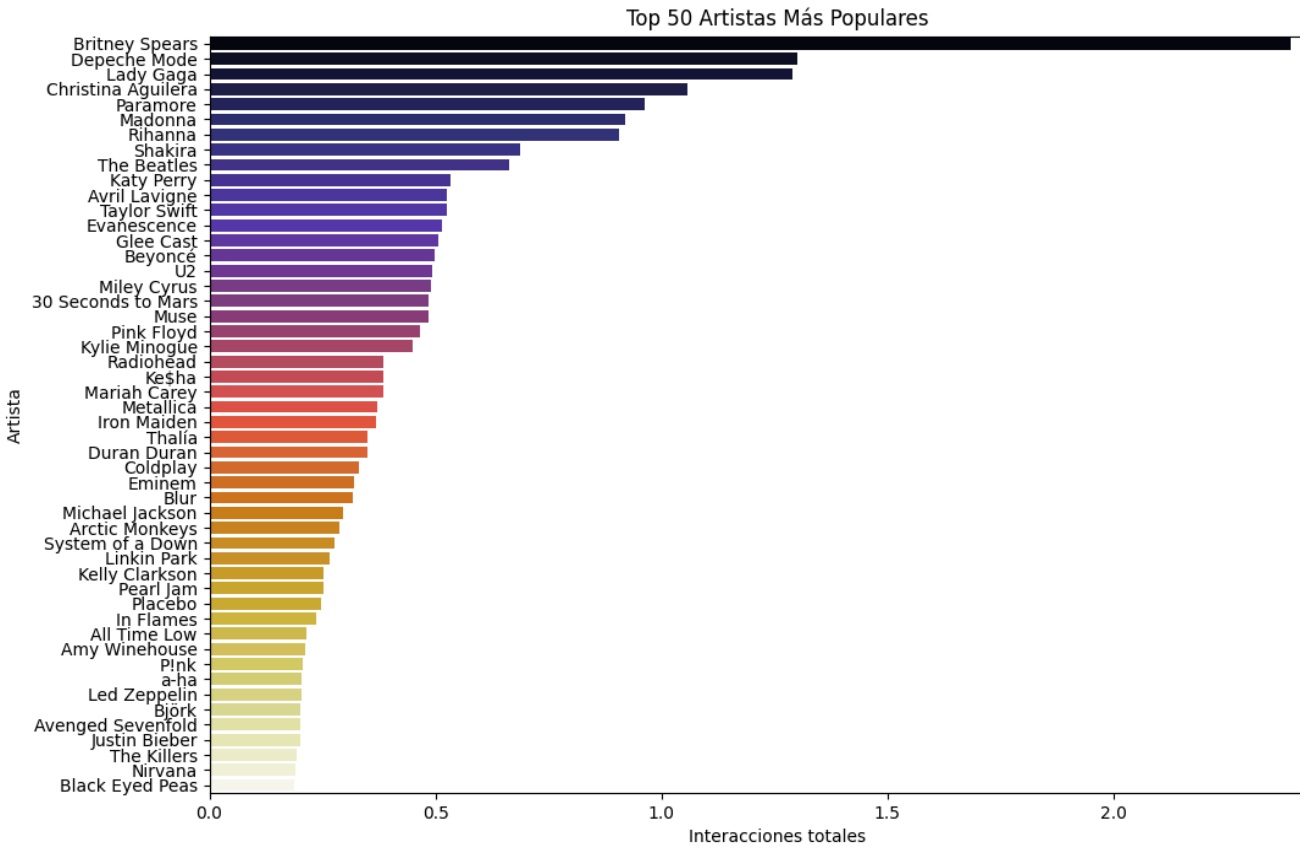

Figura 1. Artistas más populares

El siguiente gráfico permite observar más fácilmente ese desbalance, donde se observa que la enorme mayoría de los artistas tienen muy pocas reproducciones

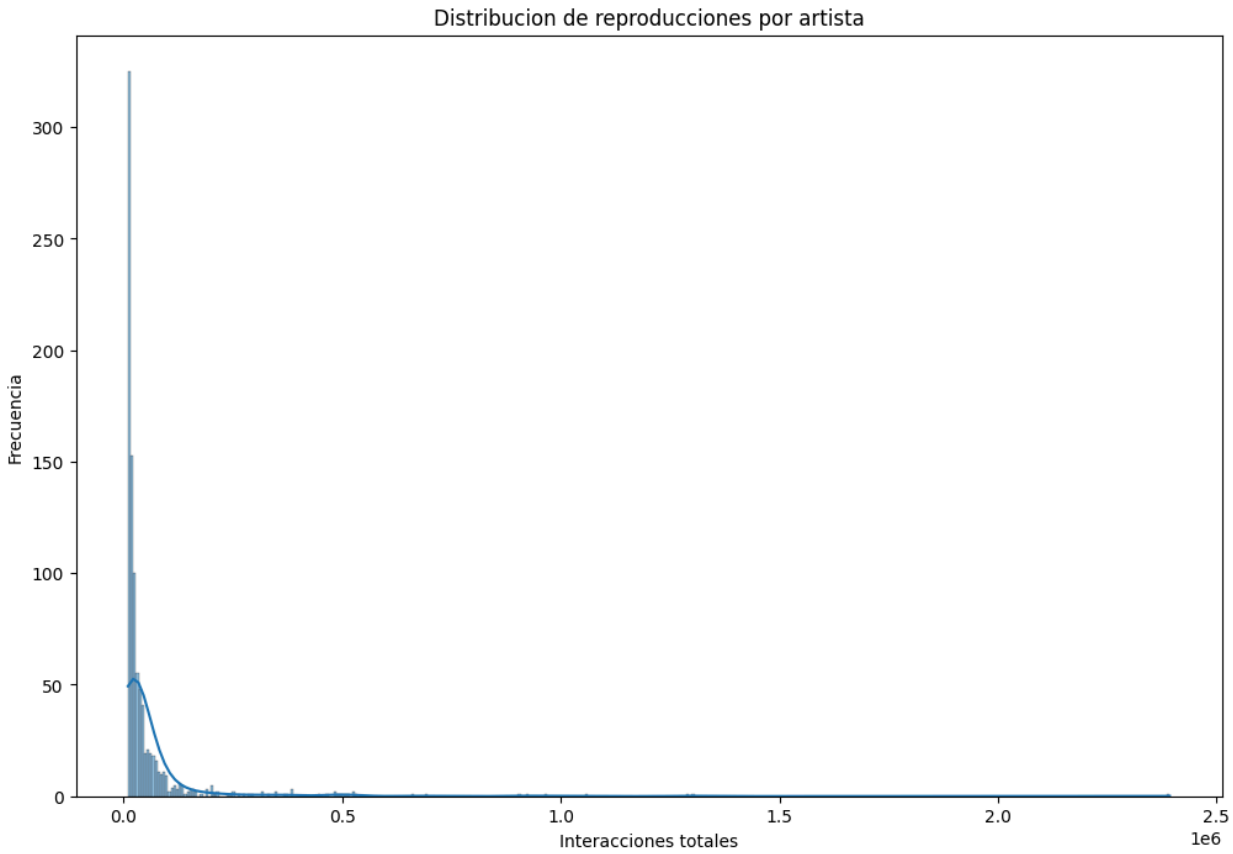

Figura 2. Distribución de reproducciones por artista

Si contrastamos esto con la distribución de reproducciones por usuario, observamos que la mayoría de los usuarios tiene entre 5000 y 50000 reproducciones, con muchos *outliers*

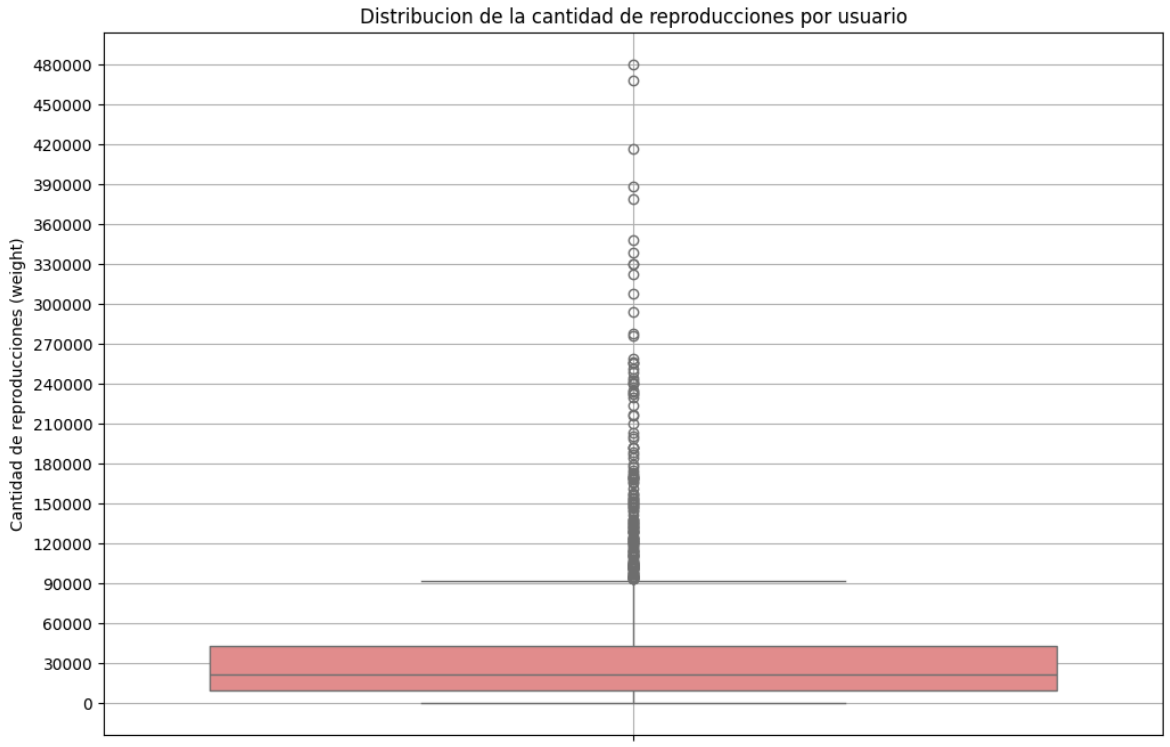

Figura 3. Distribución de la reproducciones por usuario

También se puede realizar un análisis sobre los *tags*. El siguiente gráfico muestra los tags más populares

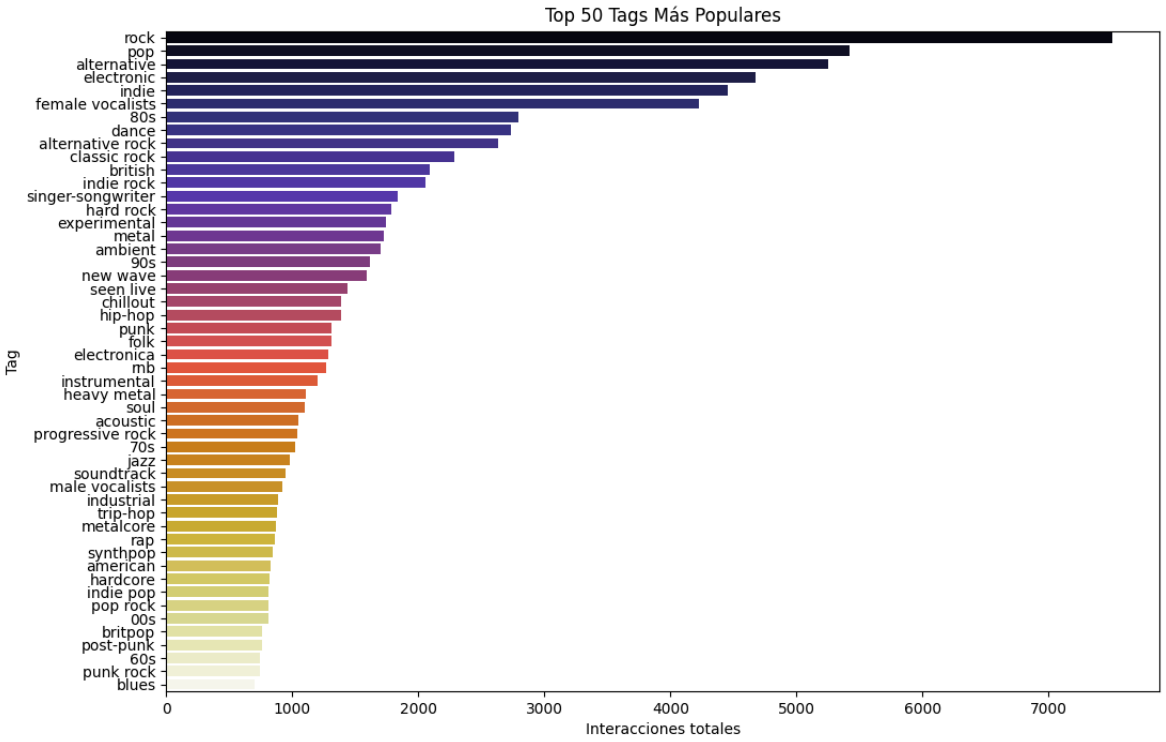

Figura 4. Distribución de la popularidad de tags

A su vez, estos tags se distribuyen de manera muy similar a las reproducciones. La mayoría de los artistas tienen muy pocos tags asignados, mientras que los más populares llegan a tener hasta 7000 tags

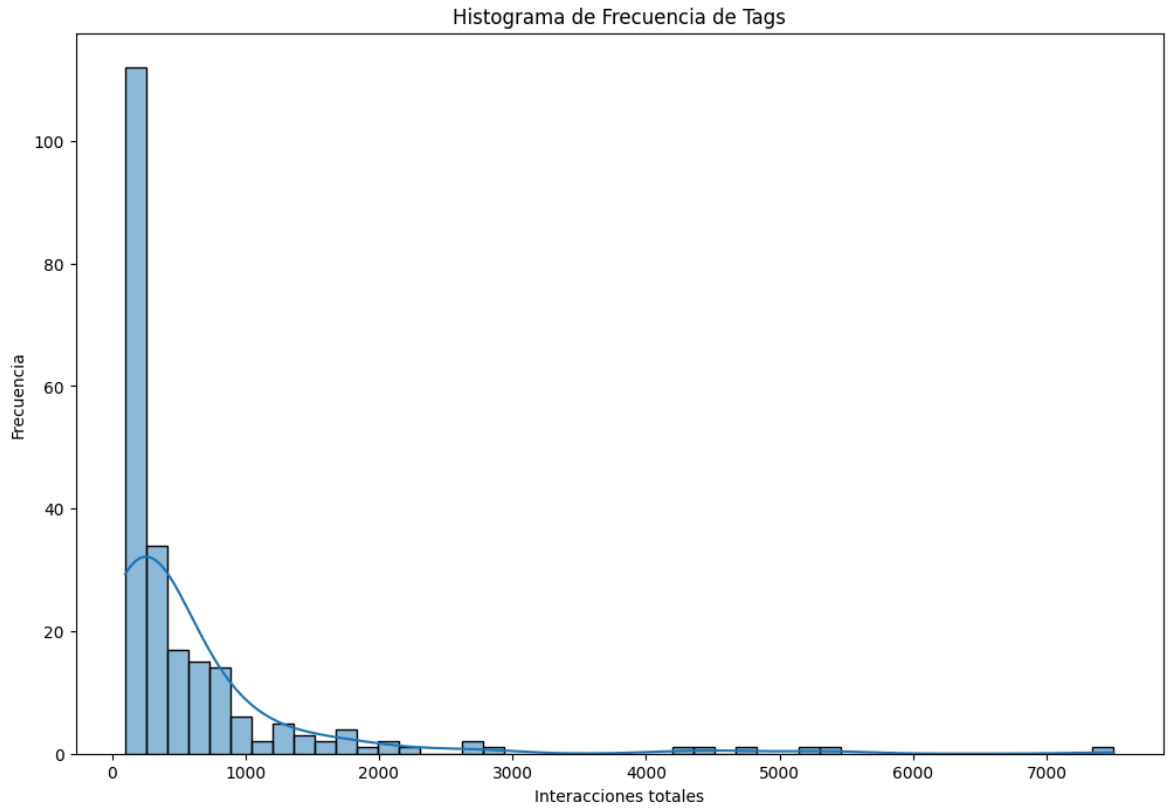

Figura 5. Distribución de la cantidad de tags por artista

Modelo base

Antes de comenzar a experimentar con modelos avanzados para sistemas de recomendación, se implementó un modelo base que permite medir la performance de los modelos siguientes.

Este modelo sencillo consiste en recomendar, para todos los usuarios, los *k* artistas más populares.

Sistema de recomendación

A la hora de implementar un sistema de recomendación utilizando técnicas de machine learning, se decidió usar la librería de Python *LightFM*. Esta librería implementa un modelo híbrido que combina factorización de matrices con técnicas basadas en el contenido. El modelo fue desarrollado por la empresa Lyst y se describe en un paper como una alternativa a modelos existentes que incorpora embeddings de los ítems de entrada y perfiles de usuario para mejorar la performance

Esta librería permite entrenar modelos basados en ratings implícitos y opcionalmente se pueden agregar metadatos de los ítems y de los usuarios, principalmente con el objetivo de eliminar el problema de *cold start*.

En este proyecto, se hizo uso de los metadatos de ítems, a través de un modelo *bag of words*, que cuenta la cantidad de apariciones de cada tag para cada artista y permite encontrar similitudes semánticas entre artistas. Sin embargo, ya que no se tiene un conjunto de datos de artistas más rico (por ejemplo, incluyendo perfil del artista o datos sobre sus canciones), no se pudo aprovechar al máximo el poder expresivo de este tipo de modelos.

A partir de esto, el sistema funciona como un modelo clásico de collaborative filtering, utilizando métodos de machine learning para descomponer la matriz *sparse* de interacciones entre usuarios e ítems en dos matrices separadas de rango arbitrario, que describen tanto a los usuarios como a los ítems como vectores compactos.

Entre los experimentos realizados con *LightFM*, se puede resaltar el entrenamiento de diversas combinaciones de modelos incluyendo y eliminando elementos como los embeddings de ítems, embeddings de usuarios y también probando escalar la matriz de pesos para las interacciones. Finalmente, se realizó un barrido de hiperparámetros para maximizar la performance del modelo.

Cold start

Una de las premisas de este proyecto, y un gran motivo para escoger *LightFM* como base para realizar el modelo es la capacidad para abordar el problema del cold start.

Como última etapa del proyecto, se elaboró un sistema que utiliza el modelo entrenado en la etapa anterior y permite describir a un artista a partir de una lista de tags. Utilizando la matriz de ítems de la etapa anterior (aprendida a partir de los embeddings de ítems y sus interacciones), se obtiene una "representación latente" de cada artista. Si luego aplicamos la misma transformación al nuevo artista creado, podemos utilizar métodos de similitud de vectores para hallar aquellos artistas más similares.

Comparación de métricas

Antes de analizar los resultados obtenidos, es importante identificar cuáles son las métricas más importantes a la hora de evaluar sistemas de recomendación. Existen diversas métricas aplicables a este tipo de modelos, siendo las principales la Precisión@k (extensible a Recall@k y F1@k), la AUC ROC y métricas específicas para rankings como Discounted Cumulative Gain o el cubrimiento del conjunto de datos.

La métrica a maximizar depende de cuál es nuestro objetivo con el sistema

- Precisión: mide la *eficiencia* del sistema de recomendación. La Precisión@k mide, de *k* recomendaciones, cuántas de ellas son relevantes. Se utiliza sobre todo cuando el usuario verá muy pocas recomendaciones.

- AUC: es la métrica más general y representa la probabilidad de, dados un elemento relevante A y uno no relevante B, que A esté rankeado más alto que B. Un área bajo la curva de 0.5 representa un modelo aleatorio.
- DCG: es una métrica comúnmente utilizada cuando el orden en el que aparecen las recomendaciones es realmente importante.
- Coverage: porcentaje del conjunto de datos que es recomendado a al menos un usuarios

Para este proyecto, se decidió utilizar AUC como la métrica principal, ya que es una métrica general de performance sin tener mayor conocimiento del contexto en el que se utiliza el sistema

Resultados

Antes de comenzar con la evaluación de cualquier modelo, se implementó el modelo baseline, obteniendo un AUC de 0.48 (peor que un modelo aleatorio). Este modelo presenta un punto de partida a mejorar para determinar si se están realizando avances sobre el problema.

Luego de experimentar con el modelo de LightFM hasta alcanzar cierta performance base, se realizó un barrido de hiperparámetros para maximizar la performance del modelo.

Hiperparámetros obtenidos:
- Epochs: 30
- Learning rate: 0.01
- Función de loss: `bpr`
- Número de componentes: 64

Utilizando los hiperparámetros finales, se obtuvieron los siguientes resultados

Modelo	Conjunto de datos	Train	Test
Modelo base (solo interacciones)		0.75	0.71
Base + Embeddings de items		0.79	0.75
Metadatos de items + pesos de interacciones		0.88	0.86

Posibilidades de mejora

- La falta de conocimiento previo sobre librerías y técnicas comunes utilizadas para este tipo de sistemas trajo algunos problemas de metodología en la implementación inicial que debieron ser resueltos posteriormente. Particularmente, LightFM no tiene soporte para correr en GPU, por lo que el entrenamiento y la evaluación fueron bastante costosos.

- El conjunto de datos elegido no tenía buenos metadatos sobre artistas, lo que hizo que el modelo basado en el contenido sea mejorable. Este problema fue de la mano con el mencionado en el ítem anterior, ya que el hardware fue una limitación a la hora de entrenar el modelo y hacer experimentos.

- Por cuestiones de tiempo, no se alcanzó a utilizar los datos de amistad entre usuarios ni a probar técnicas de regularización.

- Como se eligió LightFM como sistema de recomendación que implementa nativamente un modelo híbrido, no se intentó combinar otros modelos con una red neuronal a modo de ensemble

Conclusiones

- **Importancia de los sistemas de recomendación híbridos**: Desarrollar sistemas de recomendación híbridos desde un comienzo permiten tener un mayor poder expresivo que si solo se utiliza filtrado colaborativo y nos da herramientas para afrontar el cold start. También nos permite comenzar a explicar el porqué de algunas recomendaciones, que de otra forma son tratadas como una caja negra

- **Justificación del modelo:** El modelo final considera una mejora significativa por sobre el modelo base, lo que nos lleva a pensar que con una mayor cantidad de datos y un mejor preprocesamiento de los mismos se puede alcanzar una excelente performance.

- **Métricas**: LightFM resultó ser una herramienta adecuada para este proyecto, que permitió realizar el proyecto sin experiencia previa con la plataforma. Aunque presentó limitaciones como la falta de soporte para GPU, su capacidad para trabajar con ratings implícitos y metadatos mejoró significativamente el rendimiento en comparación con un modelo base.

- **Sobre el conjunto de datos:** el conjunto de datos utilizado puede ser considerado apropiado como un enfoque inicial para obtener un sistema de recomandación usable. Sin embargo, el dataset tiene ya 15 años y no cuenta con una gran cantidad de datos de perfil de artista ni de usuario, lo que nos da una cantidad limitada de opciones para seguir refinando el modelo.

Bibliografía

- Recommending Products: Matrix Factorization
- Recommender Systems (Hwanjo Yu)
- LightFM source code
- Learning to Rank Sketchfab Models with LightFM
- Metadata Embeddings for User and Item Cold-start Recommendations
- Recommender Systems — A Complete Guide to Machine Learning Models