UNIVERSIDAD PERUANA
CAYETANO HEREDIA

Sistema de Recomendación

Curso: Métodos Numéricos y Optimización para Machine Learning

Docentes:

- Nelson Enrique Castro Zarate
- Josué Angel Mauricio Salazar

Integrantes:

- Josue Eduardo Huarauya Fabian
- Vanesa Nelsy Morales Taipe
- Nohereily Kimberly Salazar Berrios

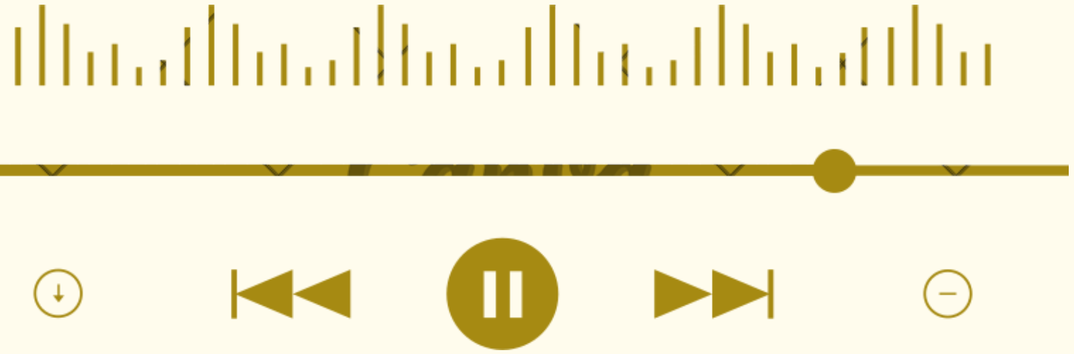


Tabla de Contenido

Introducción

El desarrollo de los sistemas de recomendación se inició a partir de una observación bastante simple: los individuos a menudo se basan en recomendaciones proporcionadas por otros al tomar decisiones diarias y de rutina [1,2].

En la música esta herramienta tecnológica permite a los usuarios descubrir nuevas canciones o artistas basándose en sus preferencias y patrones de escucha anteriores. Estos sistemas utilizan diferentes técnicas y algoritmos para proporcionar recomendaciones precisas y personalizadas.

Nuestro objetivo: Proporcionar a los usuarios recomendaciones de música personalizadas y efectivas basándose en sus patrones de escucha.

Estado del ARte

Music Recommender System using Autorec Method for Implicit Feedback [1]

El sistema ayuda a los usuarios a encontrar música que se adapte a sus gustos, utilizando el paradigma de filtrado colaborativo. El método utilizado fue el 'Autoencoder', que mejora el rendimiento de la factorización de matrices para predecir las valoraciones de los usuarios. Autorec supera al método de descomposición en valores singulares (SVD) en un conjunto de datos de música, con una diferencia de RMSE de 0.7.

Collaborative Filtering-Based Music Recommendation in View of Negative Feedback System [2]

El método propio que combina el filtrado colaborativo basado en ítems con un sistema de retroalimentación negativa (NFS) que permite a los usuarios rechazar las canciones que no quieren escuchar. Esto resultó en una nueva serie de recomendaciones basadas solo en las canciones que le gustan al usuario. Gracias al NFS, el usuario puede reconocer fácilmente las recomendaciones con una precisión del 16,78%.

A Music Recommendation System Based on logistic regression and eXtreme Gradient Boosting [3]

El sistema de recomendación de música que utiliza la regresión logística y el eXtreme Gradient Boosting (xgboost) para predecir las preferencias musicales del usuario. El sistema propuesto es un algoritmo híbrido llamado LX que integra la regresión logística y el xgboost. La regresión logística, que es un modelo lineal, se utiliza como clasificador para predecir las preferencias musicales del usuario. Sin embargo, la regresión logística no maneja muy bien las características de datos no lineales complejas. Para resolver este problema, se propone el uso de xgboost, que es capaz de manejar características no lineales.

Descripción

- **song_data**

song_id: Identificador único de la canción.
title: Título de la canción.
release: Nombre del álbum al que pertenece la canción.
artist_name: Nombre del artista.
year: Año de lanzamiento de la canción.

- **count_data**

user_id: Identificador único del usuario.
song_id: Identificador único de la canción.
play_count: Número de veces que el usuario ha escuchado la canción.

Características de la data

song_data.csv

- **Número total de registros:** 1,000,000
- **Número de artistas únicos:** 72665
- **Número de canciones únicas:** 999056 (considerando que cada registro es único)
- **Valores nulos:** Se identificaron 15 valores nulos en la columna “title” y 5 en la columna “release”.
- **Años de lanzamiento:** El conjunto de datos abarca canciones de diferentes años, incluyendo algunos registros sin año especificado (indicado como 0).

count_data.csv

- **Número total de registros:** 2,000,000
- **Número de usuarios únicos:** 76353
- **Valores nulos:** No se identificaron valores nulos.
- **Duplicados:** 498 registros duplicados

Unir CSVs

1.Proceso de Combinación

```
#Combinar ambos conjuntos de datos
#usando 'song_id' como clave
combined_data = pd.merge(count, song, on='song_id',
how='left')
```

2.Análisis del Conjunto de Datos Combinado

- **Valores nulos:**

```
nulos = combined_data.isnull().sum()
```

- **Búsqueda de Artistas Específicos**
- **Cantidad de Artistas Antes de la Combinación:**

```
valores_unicos = song['artist_name'].unique()
cantidad_artistas = len(valores_unicos)
```

- **Cantidad de Artistas Después de la Combinación:**

```
valores_unicos = combined_data['artist_name'].unique()
cantidad_artistas = len(valores_unicos)
```

FILTRACIÓN DE LA DATA

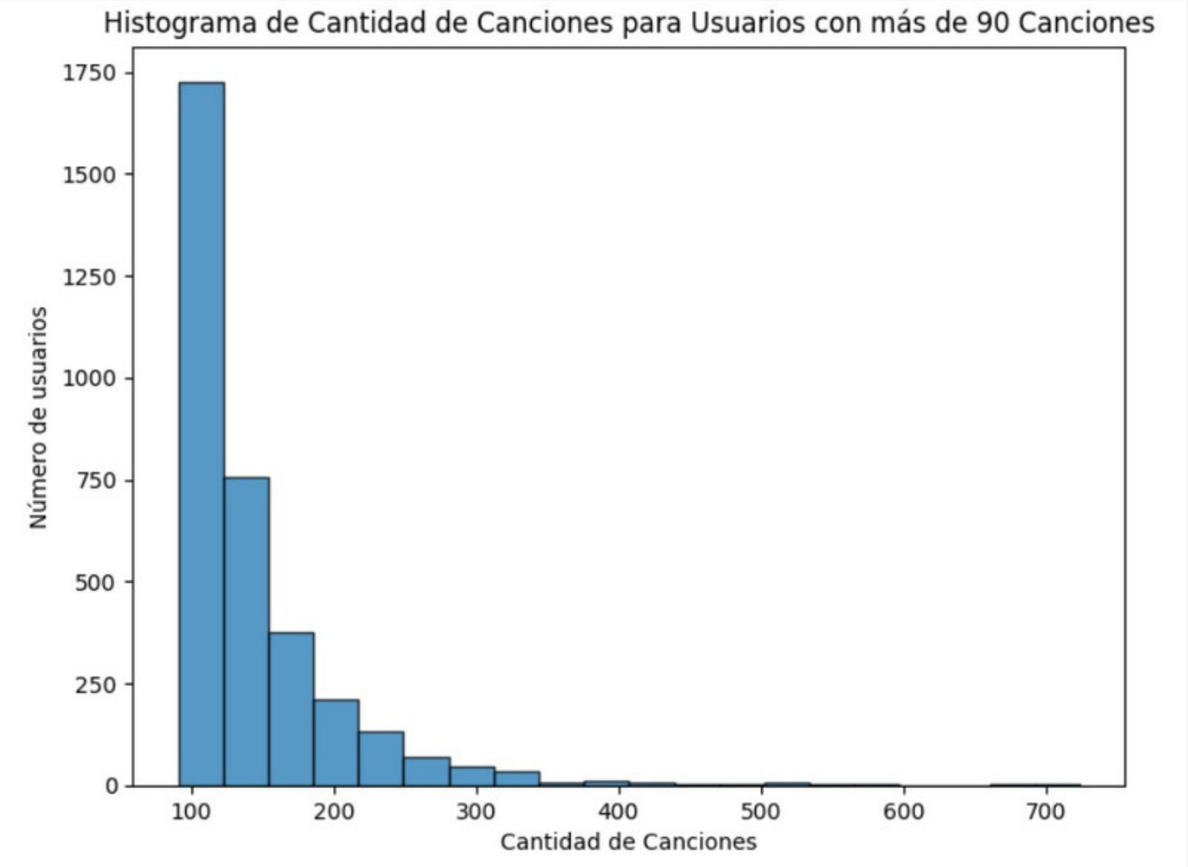
Idea clave:

1. **Identificar los usuarios hayan escuchado más de 90 canciones.**
   - **Se encontraron 3,390 usuarios que cumplen con este criterio.**
   - **Eliminamos los usuarios que tienen menos de 90 canciones escuchadas.**
2. **Identificar las canciones que hayan sido escuchadas por más de 120 usuarios.**
   - **Se encontraron 5,256 canciones que cumplen con este criterio.**
   - **Eliminamos las canciones que tienen menos de 120 usuarios.**

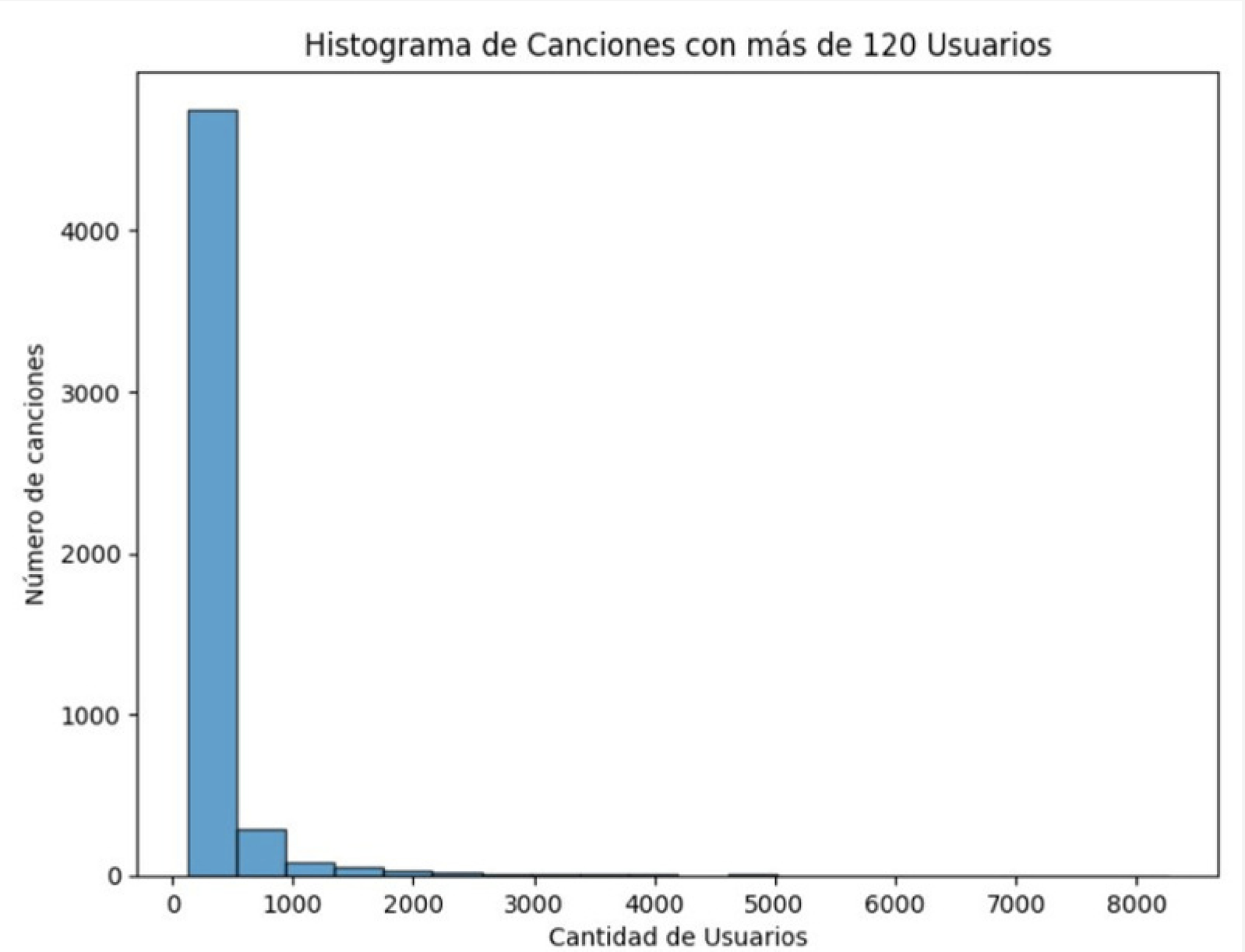
Métodos para obtener la cantidad:

- **Sin GroupBy: Se utilizó un enfoque iterativo para determinar cuántas canciones escuchó cada usuario**
- **Con GroupBy: Se utilizó la función groupby para agrupar los datos por user_id y por song_id.**

HISTOGRAMAS DEL LOS FILTROS



El número de cantidad de canciones que tienen los usuarios está muy sesgado a la derecha.



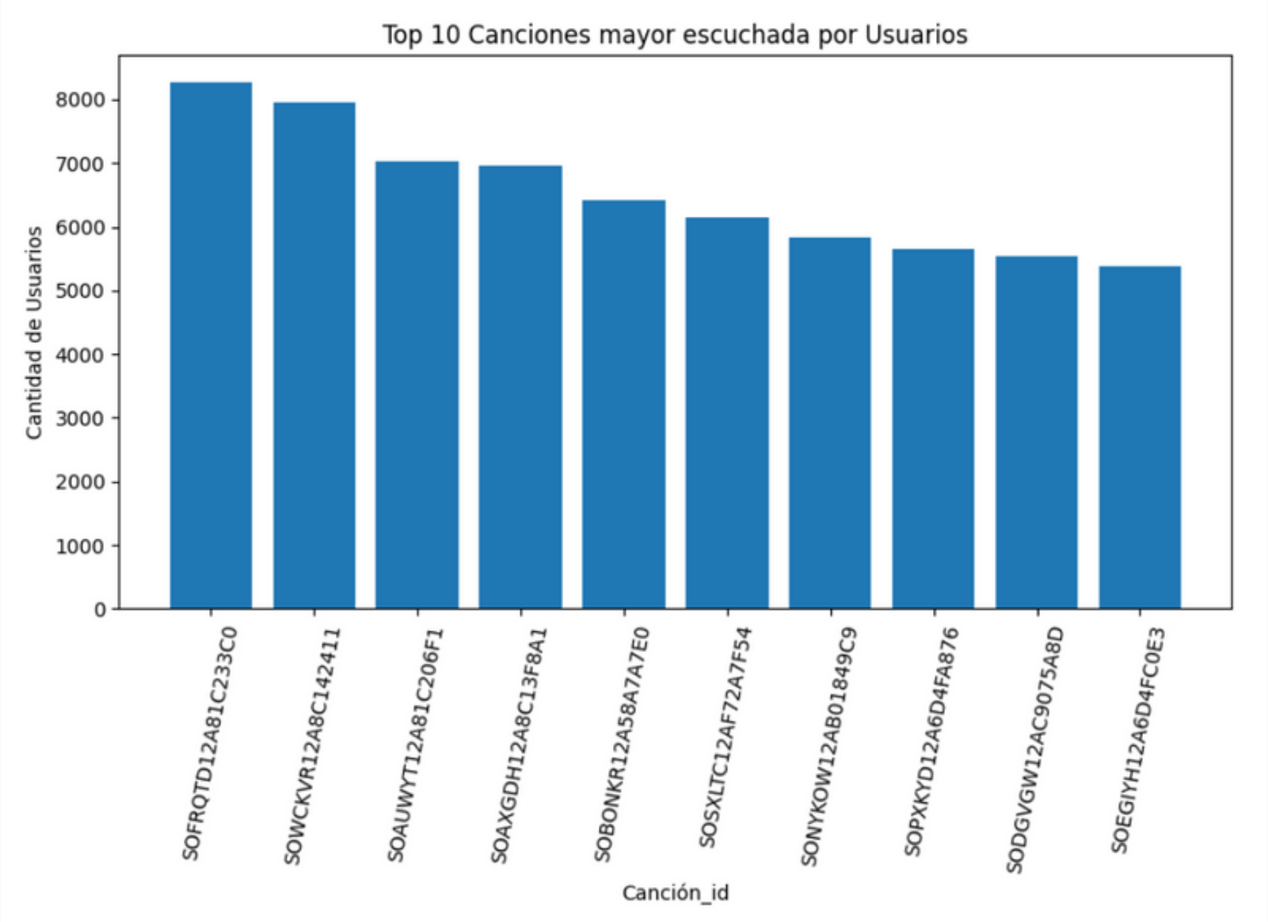
El número de cantidad de usuarios que tienen las canciones está muy sesgado a la derecha.

Características de la data filtrada

- Número total de registros: 386848
- Número de artistas únicos: 1959
- Número de usuarios únicos: 3390
- Número de canciones únicas: 5256
- Valores nulos: No hay valores nulos.

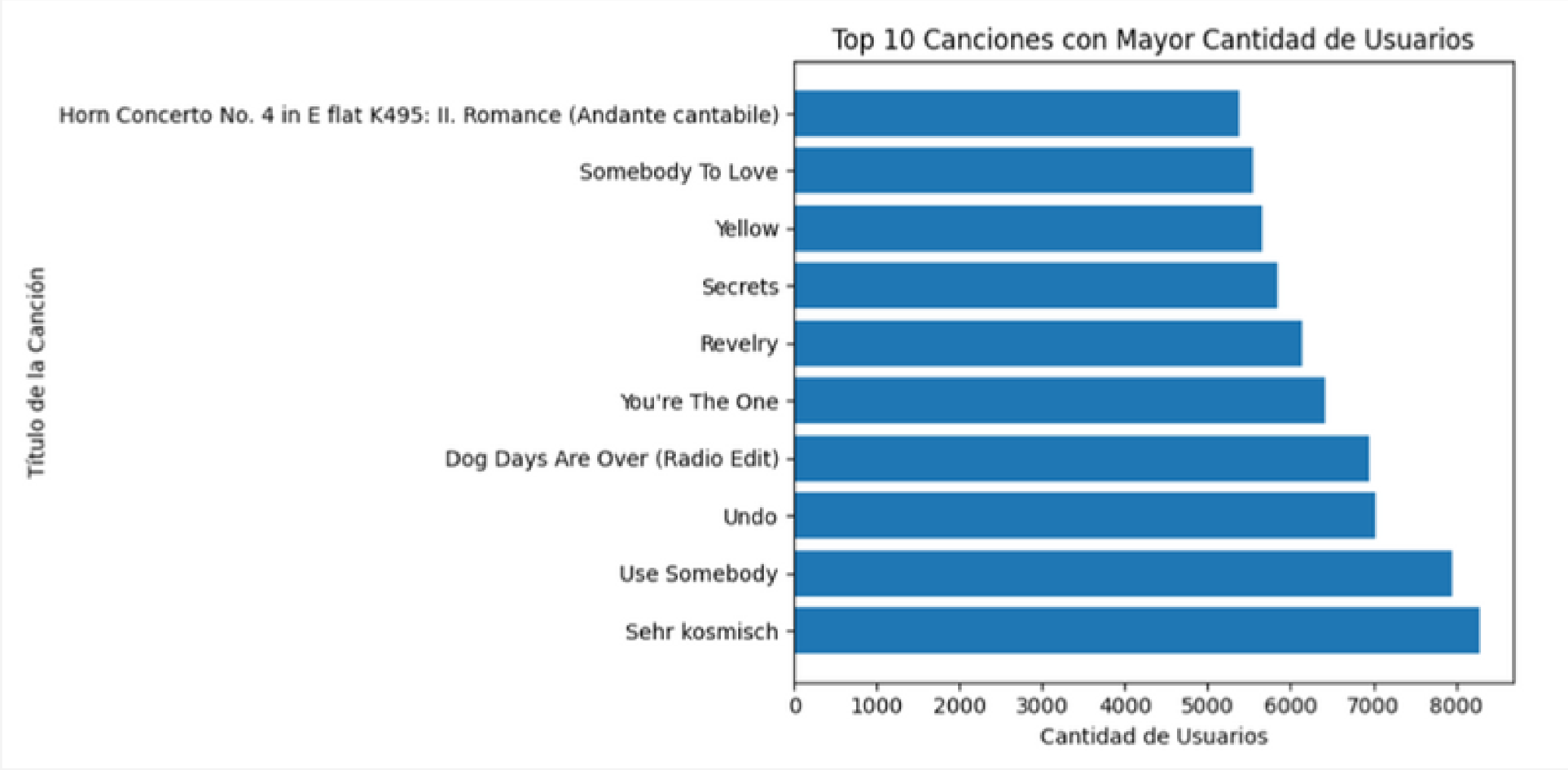
Top 10 Canciones Más Escuchadas

Top 10 canciones mayor escuchada por usuarios



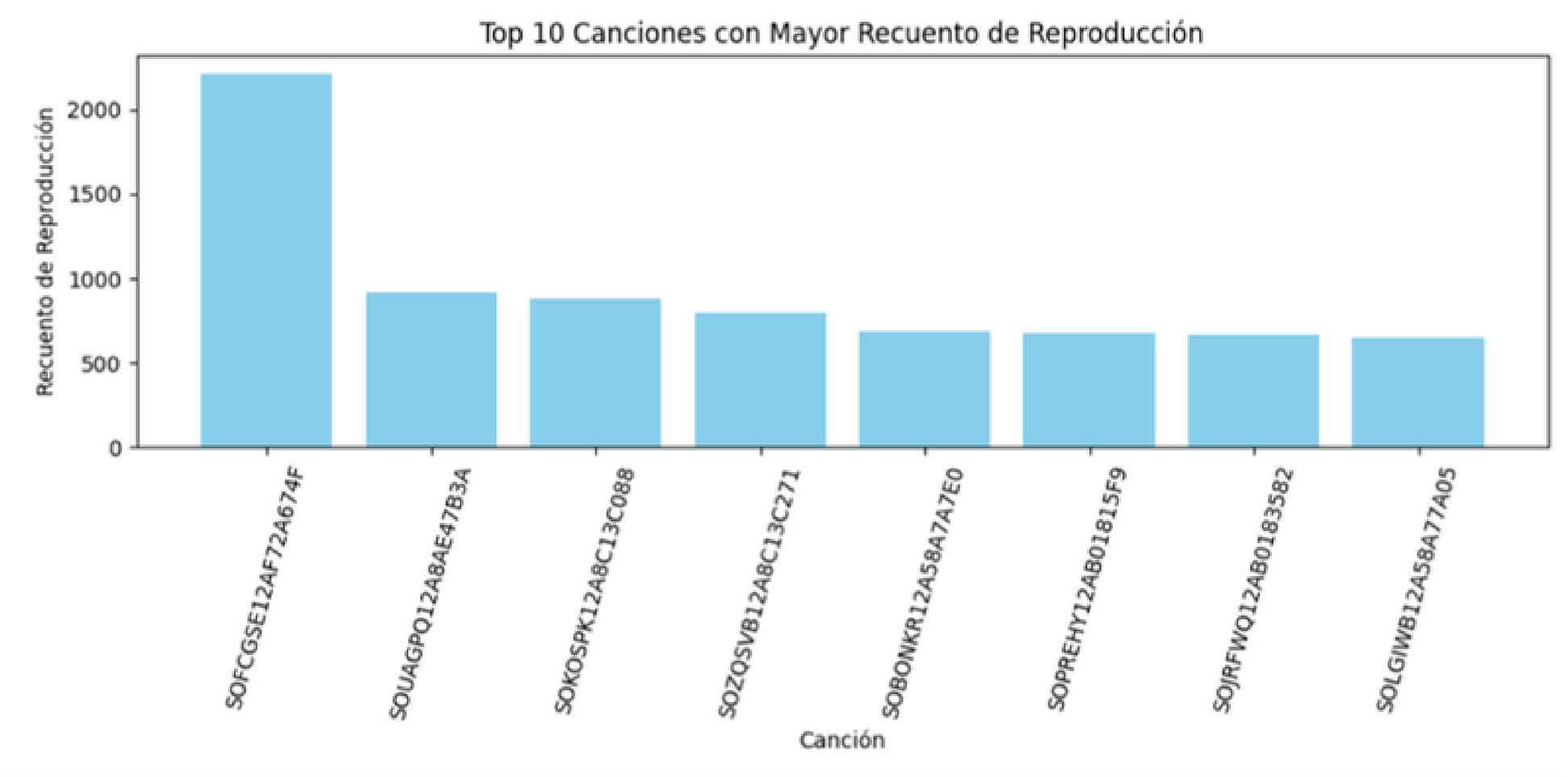
Top 10 Canciones Más Escuchadas

Top 10 canciones con mayor cantidad de usuario



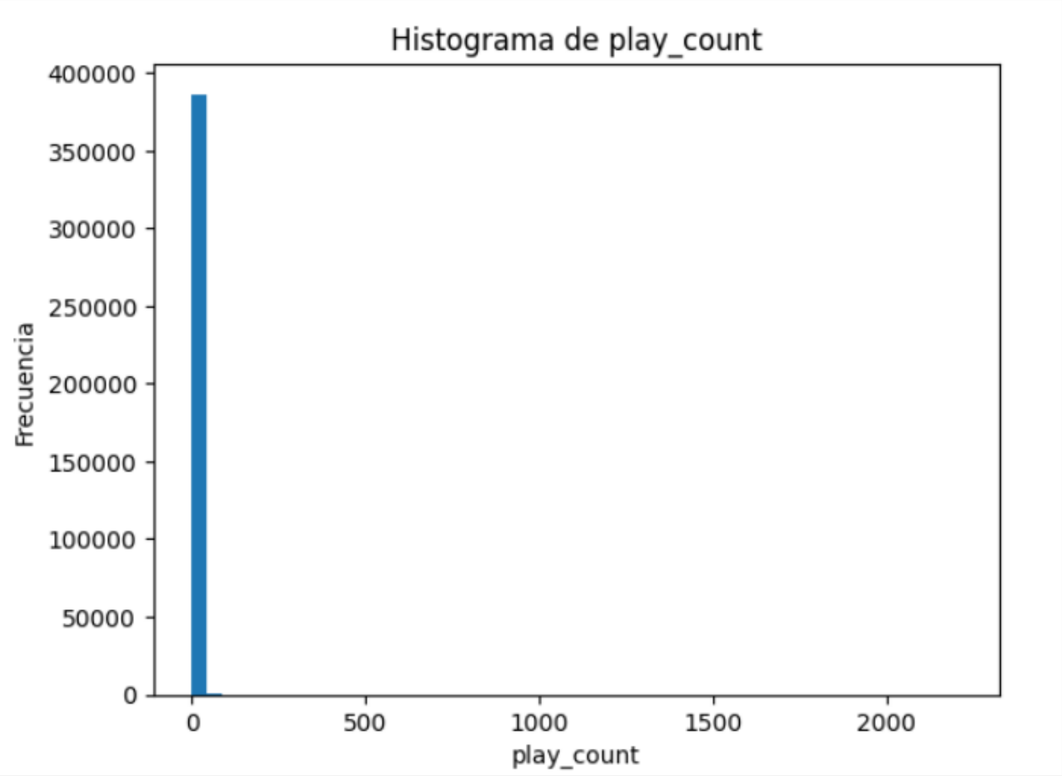
Top 10 Canciones Más Escuchadas

Top 10 canciones con mayor frecuencia de reproducción

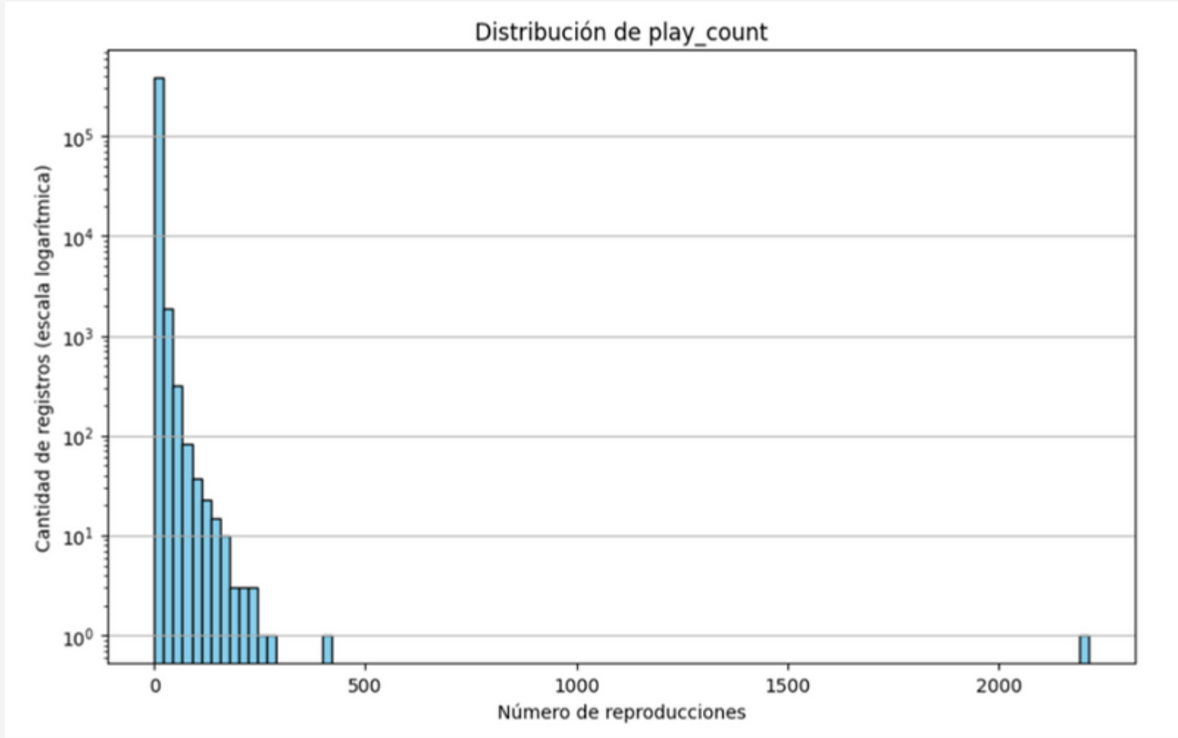


Distribución de play_count

Sin escala logarítmica



Con escala logarítmica

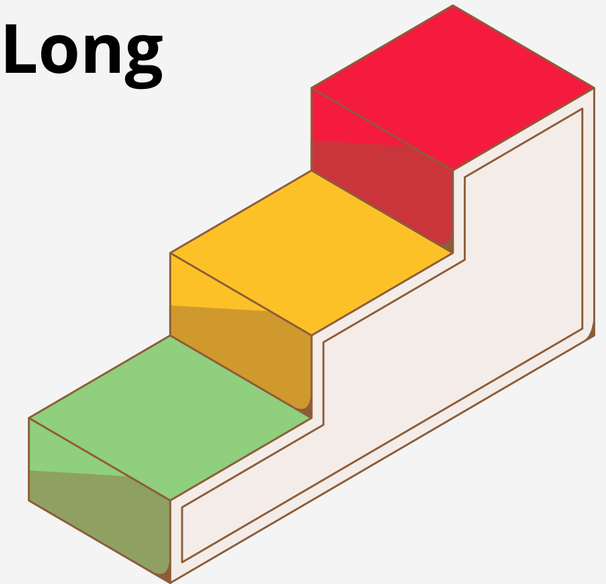


El número de reproducciones que hay en la cantidad de registros está muy sesgado a la derecha.

Pasos para construir el sistema de recomendación

Al tener la data ya tratada, realizaremos a continuación:

1. **La reducción de Dimensionalidad usando técnicas:**
    - **Factorización de Matrices(SVD, PCA).**
2. **Obtener una solución al problema Long Tail en recomendación musical.**



Experimentación :

Filtrado Colaborativo:

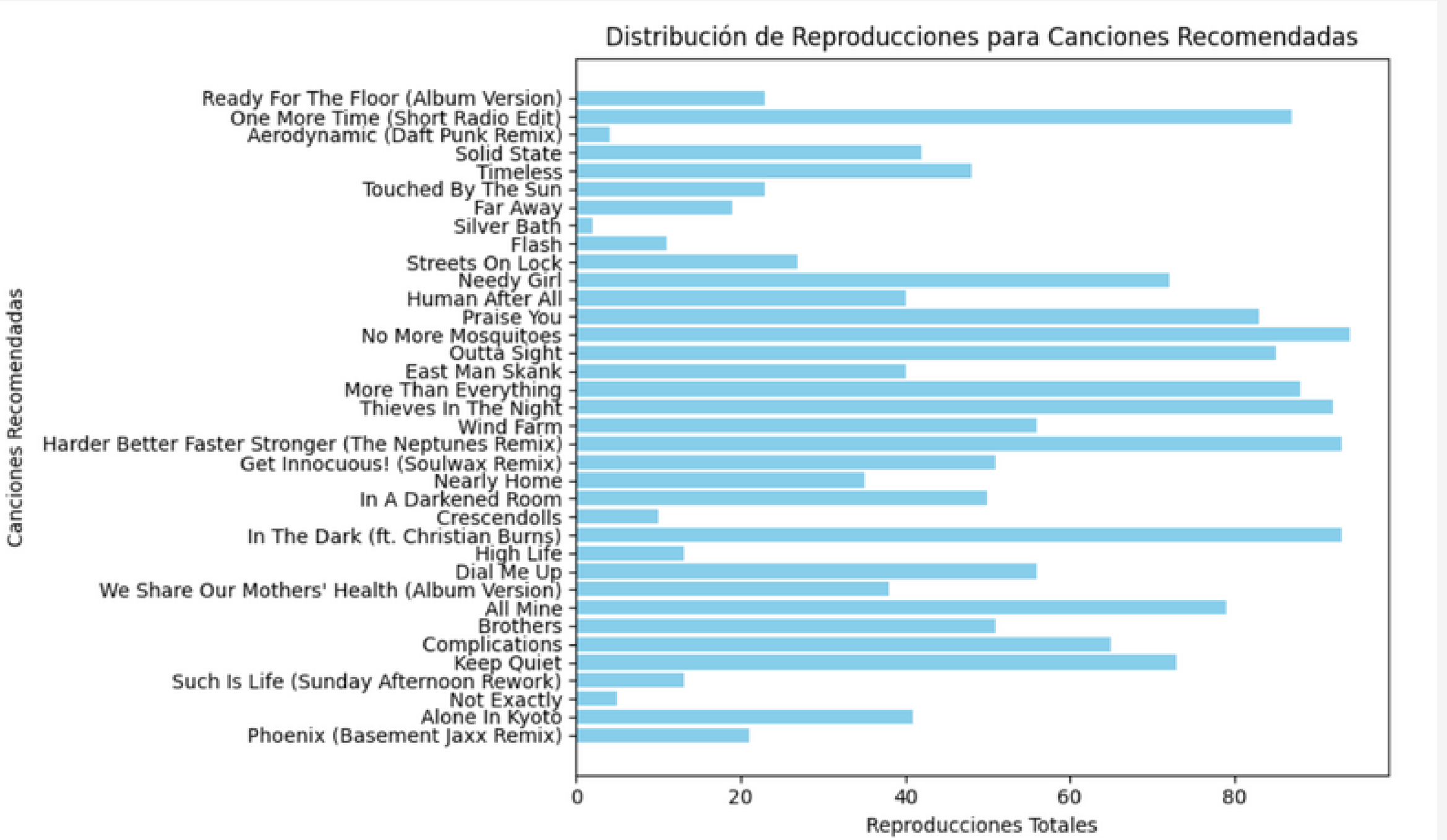
El filtrado colaborativo es una técnica común en sistemas de recomendación. En el caso específico de la música, se puede usar para sugerir canciones basadas en la cantidad de veces que han sido reproducidas. Esto se hace mediante un método llamado filtrado colaborativo basado en memoria.

Cálculo de Similitud de Coseno

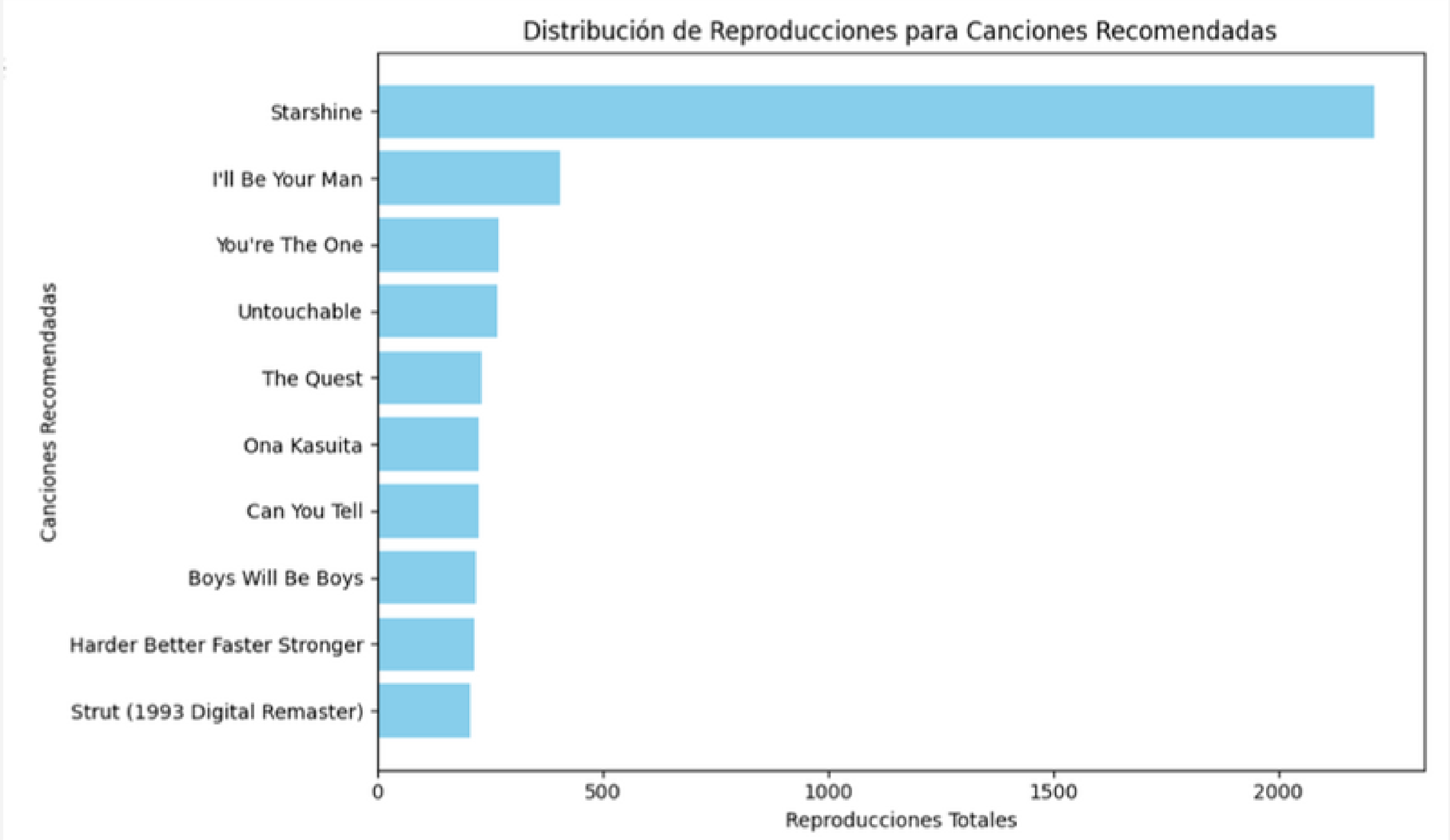
La similitud entre usuarios o canciones se mide para identificar patrones de escucha similares. Utilizando la matriz de reproducciones, se calcula la similitud de coseno entre los usuarios mediante la función cosine_similarity del paquete sklearn.metrics.pairwise.

Similitud de Coseno

$$\text{Similitud} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$


Distribución de Reproducciones para Canciones Recomendadas
Canciones Recomendadas
Ready For The Floor (Album Version)
One More Time (Short Radio Edit)
Aerodynamic (Daft Punk Remix)
Solid State
Timeless
Touched By The Sun
Far Away
Silver Bath
Flash
Streets On Lock
Needy Girl
Human After All
Praise You
No More Mosquitoes
Outta Sight
East Man Skank
More Than Everything
Thieves In The Night
Wind Farm
Harder Better Faster Stronger (The Neptunes Remix)
Get Innocuous! (Soulwax Remix)
Nearly Home
In A Darkened Room
Crescendolls
In The Dark (ft. Christian Burns)
High Life
Dial Me Up
We Share Our Mothers' Health (Album Version)
All Mine
Brothers
Complications
Keep Quiet
Such Is Life (Sunday Afternoon Rework)
Not Exactly
Alone In Kyoto
Phoenix (Basement Jaxx Remix)
0
20
40
60
80
Reproducciones Totales

SVD CON CUR PARA DATA FILTRADA



SVD CON CUR PARA DATA FILTRADA

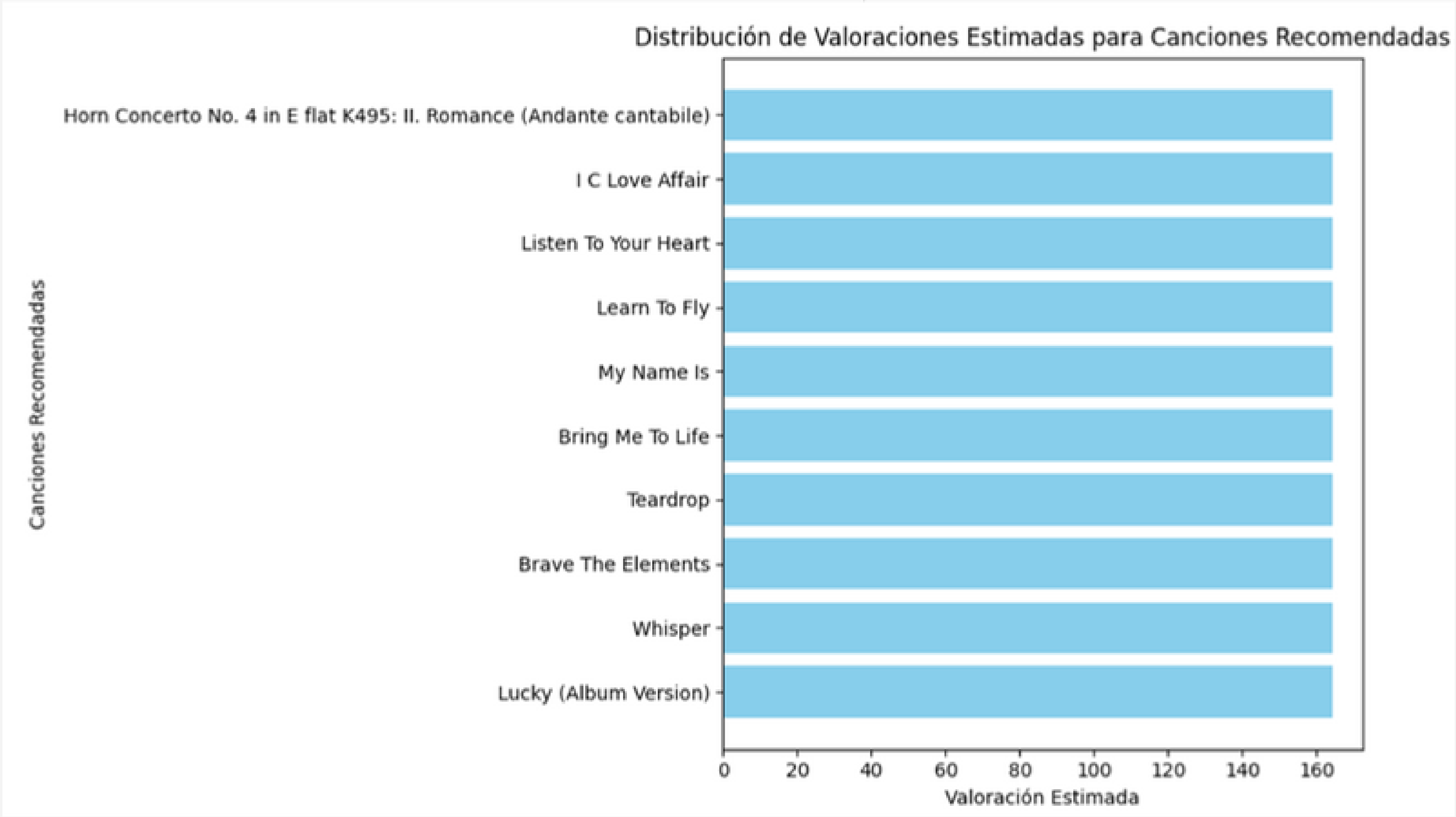
- El SVD es una técnica que descompone una matriz de dato en que representan las preferencias de los usuarios y las características de las canciones.
- El CUR es una técnica que reduce la complejidad de la matriz de datos al seleccionar solo las filas y columnas más relevantes para el modelado.
- El uso de CUR mejora la escalabilidad del sistema de recomendación al permitir que el SVD trabaje con un subconjunto de datos más pequeño y eficiente.

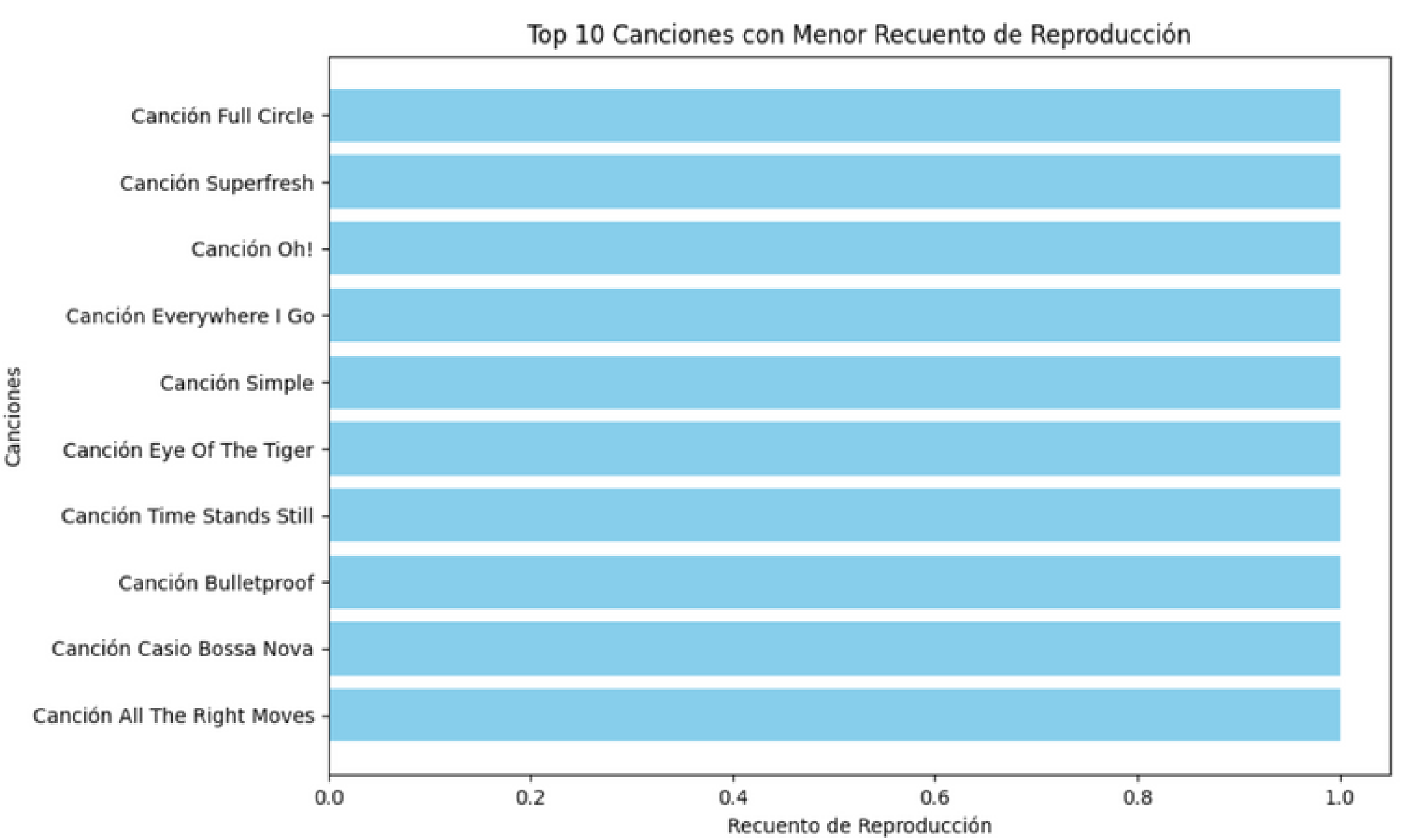
Data de entrenamiento y test

En el contexto de modelos de recomendación, el sesgo se refiere a una tendencia sistemática en las predicciones del modelo. En el caso del modelo SVD (Descomposición en Valores Singulares), el sesgo puede referirse a la incorporación de términos adicionales en la estimación de las valoraciones de los usuarios para corregir o ajustar la predicción.

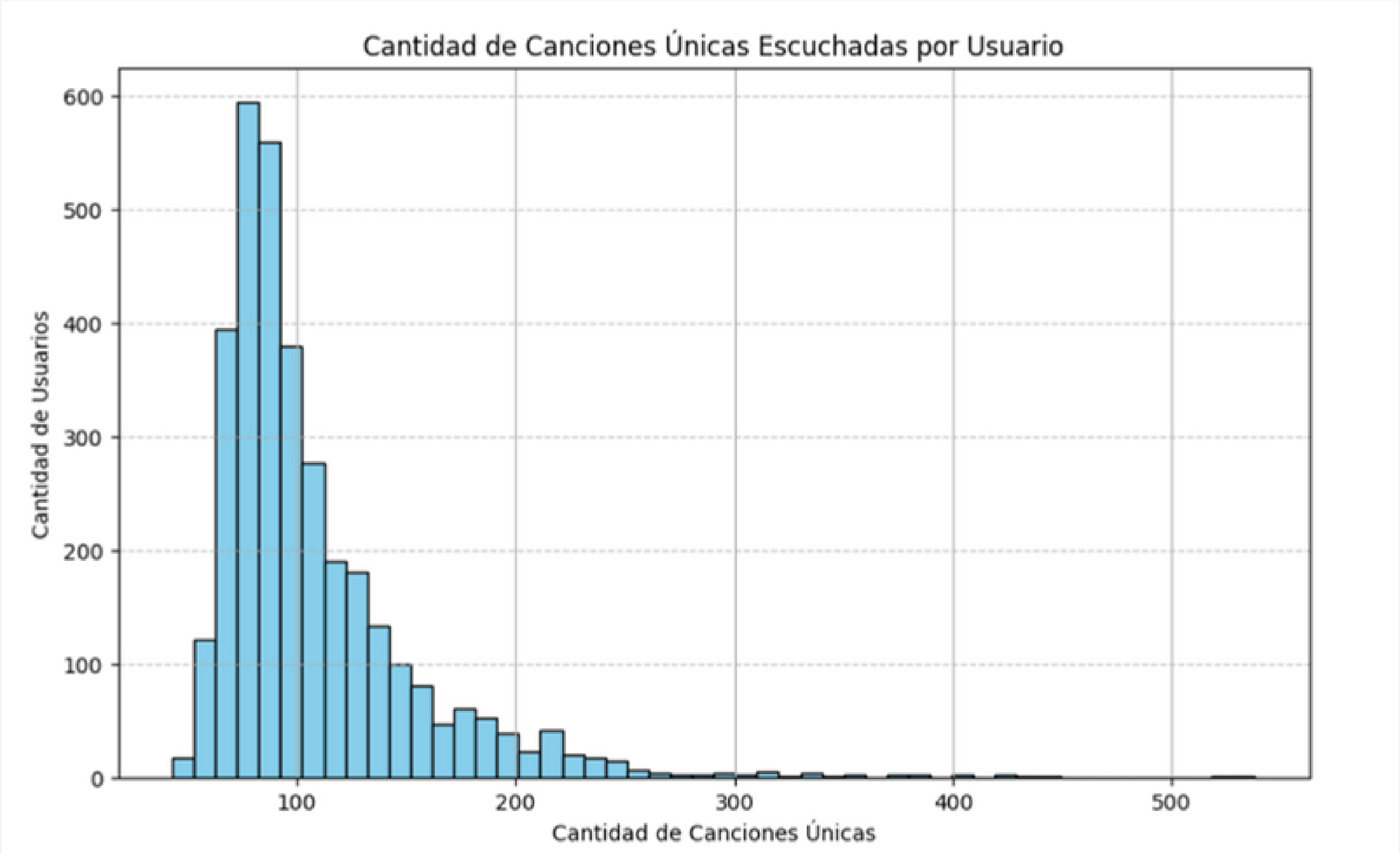
Libreria “Surprise ”

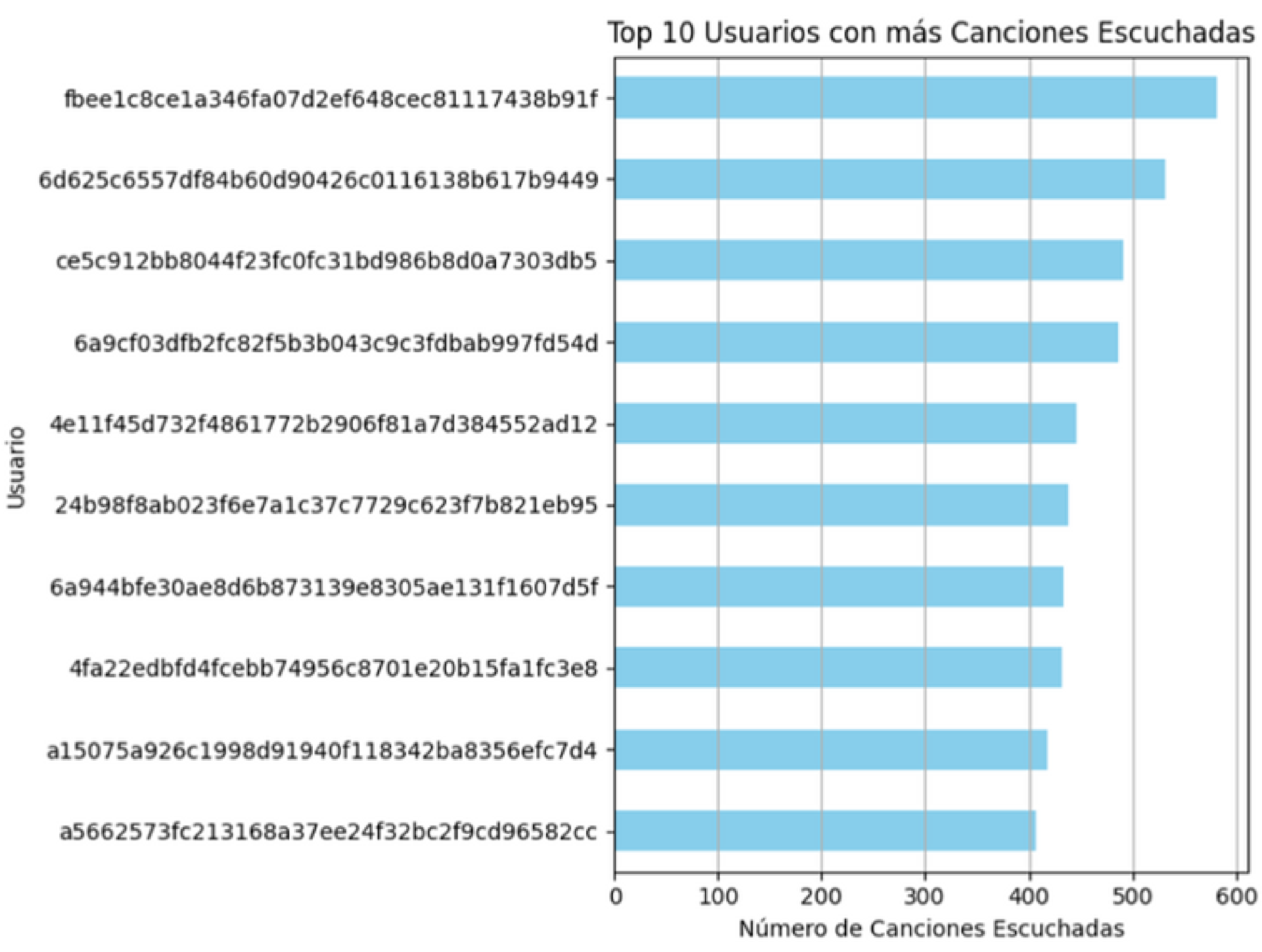
- Surprise es un scikit de Python fácil de usar para sistemas de recomendaciónDar a los usuarios un control perfecto sobre sus experimentos. Para ello, se hace hincapié en la documentación, que se ha tratado de hacer lo más clara y precisa posible señalando cada detalle de el SVD


Distribución de Valoraciones Estimadas para Canciones Recomendadas
Horn Concerto No. 4 in E flat K495: II. Romance (Andante cantabile)
I C Love Affair
Listen To Your Heart
Learn To Fly
My Name Is
Bring Me To Life
Teardrop
Brave The Elements
Whisper
Lucky (Album Version)
Canciones Recomendadas
0
20
40
60
80
100
120
140
160
Valoración Estimada


Top 10 Canciones con Menor Recuento de Reproducción
Canción Full Circle
Canción Superfresh
Canción Oh!
Canción Everywhere I Go
Canción Simple
Canción Eye Of The Tiger
Canción Time Stands Still
Canción Bulletproof
Canción Casio Bossa Nova
Canción All The Right Moves
Canciones
0.0
0.2
0.4
0.6
0.8
1.0
Recuento de Reproducción

Cantidad de Canciones Únicas Escuchadas por Usuario




Top 10 Usuarios con más Canciones Escuchadas
fbee1c8ce1a346fa07d2ef648cec81117438b91f
6d625c6557df84b60d90426c0116138b617b9449
ce5c912bb8044f23fc0fc31bd986b8d0a7303db5
6a9cf03dfb2fc82f5b3b043c9c3fdbab997fd54d
4e11f45d732f4861772b2906f81a7d384552ad12
24b98f8ab023f6e7a1c37c7729c623f7b821eb95
6a944bfe30ae8d6b873139e8305ae131f1607d5f
4fa22edbfd4fcebb74956c8701e20b15fa1fc3e8
a15075a926c1998d91940f118342ba8356efc7d4
a5662573fc213168a37ee24f32bc2f9cd96582cc
Usuario
0
100
200
300
400
500
600
Número de Canciones Escuchadas

Conclusiones:

- La aplicación de técnicas como filtrado colaborativo y SVD con CUR demuestra la versatilidad y eficacia de los sistemas de recomendación en el ámbito musical. El **uso de similitud de coseno** permite identificar patrones de **escucha similares entre usuarios**, mejorando la personalización de las recomendaciones.
- La **implementación de SVD sin sesgo** simplifica el modelo y puede ser beneficioso e**n escenarios con datos densos y representativos**. El análisis visual y la aplicación de umbrales(por ejemplo, 90 reproducciones) pueden ser útiles para filtrar datos y mejorar la calidad de las recomendaciones. resumen en viñetas

Recomendaciones:

- Experimentar con la integración de otros algoritmos podría ofrecer mejoras significativas en la precisión y la diversidad de las recomendaciones
- Experimentar con diferentes umbrales para el filtrado de datos.
- Ajustar el modelo de SVD con CUR

Bibliografía:

1. Irawan, M. F., & Baizal, Z. K. A. (2023). Music Recommender System using Autorec Method for Implicit Feedback. JURNAL MEDIA INFORMATIKA BUDIDARMA, 7(2), 609-614.
2. Verma, J. P., Bhattacharya, P., Rathor, A. S., Shah, J., & Tanwar, S. (2022, July). Collaborative Filtering-Based Music Recommendation in View of Negative Feedback System. In Proceedings of Third International Conference on Computing, Communications, and Cyber-Security: IC4S 2021 (pp. 447-460). Singapore: Springer Nature Singapore.
3. Tian, H., Cai, H., Wen, J., Li, S., & Li, Y. (2019, July). A music recommendation system based on logistic regression and eXtreme gradient boosting. In 2019 international joint conference on neural networks (IJCNN) (pp. 1-6). IEEE.
4. [1] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl, Grouplens: an open architecture for collaborative filtering of netnews, in Proceedings ACM Conference on Computer-Supported Cooperative Work (1994), pp. 175–186
5. [2] P. Resnick, H.R. Varian, Recommender systems. Commun. ACM 40(3), 56–58 (1997)

Muchas Gracias