

# TRABAJO FINAL MINERÍA DE DATOS PARA GRANDES VOLÚMENES

PRESENTADO POR:

Álvaro Villa Vélez

Edgar Leandro Jiménez Jaimes

Jorge Luis Rentería Roa

Luis Vesga Vesga

Santiago Echeverri Calderón

MAESTRÍA EN CIENCIAS DE LOS DATOS Y ANALÍTICA

UNIVERSIDAD EAFIT

MEDELLÍN

SEPTIEMBRE DE 2020

## **1. Introducción**

### **1.1. Pregunta de investigación**

¿Cuáles son las canciones similares basados en los gustos de grupos y/o individuos parecidos a través de la aplicación de sistemas de recomendación sobre grandes volúmenes de datos.?

### **1.2. Objetivos**

#### **Objetivo general:**

Identificar las canciones similares basados en los gustos de grupos y/o individuos parecidos a través de la aplicación de sistemas de recomendación sobre grandes volúmenes de datos.

#### **Objetivos específicos:**

- Adecuar los ambientes técnicos de modelación y almacenamiento en AWS.
- Analizar el comportamiento histórico de los datos y su distribución estadística.
- Seleccionar una métrica de similaridad entre canciones.
- Implementar un modelo para recomendar canciones relevantes para un usuario, dada la música que el usuario escucha.
- Medir el desempeño del sistema de recomendación.

## **2. Revisión de la literatura, estado del arte y bibliografía**

### **2.1. Bibliografía**

[1] Song, Y., Dixon, S., & Pearce, M. (2012, June). A survey of music recommendation systems and future perspectives. In 9th International Symposium on Computer Music Modeling and Retrieval (Vol. 4, pp. 395-410).

[2] Chen, H. C., & Chen, A. L. (2001, October). A music recommendation system based on music data grouping and user interests. In Proceedings of the tenth international conference on Information and knowledge management (pp. 231-238).

## **2.2. Estado del arte**

En una revisión rápida de literatura se encontró que los métodos más usados en sistemas de recomendación de música son los Filtros Colaborativos, los Filtros Basados en Contenidos, y los Modelos Híbridos que combinan ambos. Estos últimos han mostrado tener mejor desempeño que un solo modelo, ya que incorporan las ventajas de ambos métodos [1].

La música sin embargo, al ser expresión artística y cultural, tiene un carácter subjetivo y hace que los sistemas tradicionales de recomendación se queden cortos para determinar las preferencias personales de un usuario.

Para solucionar este problema algunos autores han propuesto métodos de recomendación basados en emociones o en contextos sociales, pero estos aún se encuentran en etapas tempranas y requieren datos con los que no contamos en el dataset de entrenamiento, como emociones percibidas del usuario, comentarios, reviews y etiquetas de redes sociales que escriba el usuario.

Por este motivo el presente proyecto se centrará en los métodos tradicionales de Filtros Colaborativos.

## **3. Metodología de investigación**

El proyecto se desarrollará bajo la metodología CRISP-DM. Sin embargo, el alcance llegará hasta la fase de evaluación. Se realizará un despliegue en un ambiente de Spark sobre un cluster de máquinas que permitan escalar a un volumen de datos mayor, pero no se espera llevar el proyecto a un ambiente de producción.

En la etapa de modelación planteamos 3 métodos para el sistema de recomendación:

- i. Un modelo base mediante clustering sobre el contenido de las canciones. En este modelo el objetivo es encontrar canciones similares de acuerdo con su contenido, de manera que cuando un usuario esté reproduciendo una canción se le puedan recomendar canciones de contenido similar.
- ii. Un modelo de Filtros Colaborativos Item-Item, que nos permita identificar canciones similares con base en los usuarios en común que las escuchan.
- iii. Un modelo híbrido que combine los 2 métodos anteriores.

Para la fase de evaluación se dividirán los datos en 2 subconjuntos, train (75%) y test (25%), procurando que en ambos conjuntos se encuentren interacciones de todos los usuarios, de manera que se pueda validar si las recomendaciones arrojadas por el sistema fueron reproducidas por el usuario.

La métrica de evaluación para comparar el desempeño de los modelos será la precisión. Esta métrica se seleccionó debido a que el dataset cuenta con el número de veces que un usuario ha reproducido una canción, pero no tiene una valoración explícita que permita medir cuánto le gusta la canción al usuario.

Esta métrica se definirá entonces de la siguiente manera:

$$precisión = \frac{|recomendaciones \cap test|}{|test|}$$

## 4. Análisis de los datos

### 4.1. Fuente de datos

Para el proyecto se utilizaron 2 datasets del proyecto *Million Songs Dataset* (MSD, disponible en: <http://millionsongdataset.com>). El primero es el dataset principal MSD, el cual contiene los metadatos para 1 millón de canciones. El segundo consiste en el conjunto “*Taste Profile subset*”, el cual contiene información de reproducciones de canciones por usuario en tripletas (usuario-cancion-reproducciones).

Originalmente considerábamos que podíamos obtener más información del dataset de canciones, pues en la documentación se referenciaban 46 atributos. Pero la distribución del dataset es a través de un snapshot de disco para una instancia EC2 de AWS y si bien pudimos adjuntar el snapshot a una instancia, no pudimos extraer la información para ponerla disponible en S3 o en otro medio que nos permitiera tener el almacenamiento distribuido para consumirlo desde el cluster.

Finalmente encontramos una distribución no oficial del dataset en SQLite. Ésta cuenta con el millón de canciones, pero está limitada a los 9 atributos que se muestran a continuación:

Campo	Descripción
song_id	Id de la canción según The Echo Nest
song_name	Título de la canción
release	Álbum al que pertenece la canción
artist_name	Nombre del artista
artist_mbid	Id del artista en musicbrainz.org
artist_hotness	Popularidad del artista
duration	Duración en segundos
year	Año de lanzamiento
tempo	Pulsaciones por minutos (BPM)

También contamos con las siguientes variables del dataset de reproducciones “*Taste Profile subset*”:

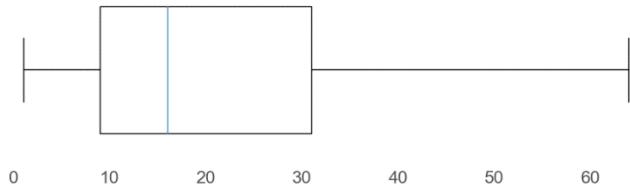
Campo	Descripción
user	Id del usuario
song	Id de la canción de The Echo Nest
play count	Cantidad de reproducciones

Esta reducción en la cantidad de los atributos nos cambió el alcance del proyecto, pues las variables con las que contábamos para describir el contenido de una canción no eran suficientes. Los clusters que se obtenían con estos atributos agrupaban canciones del mismo año y el mismo artista, grupos que resultaban insuficientes para hacer una recomendación basada en contenidos. Otros campos como la duración y el tempo eran muy comunes y genéricos, por ende no eran útiles en el modelo de clustering.

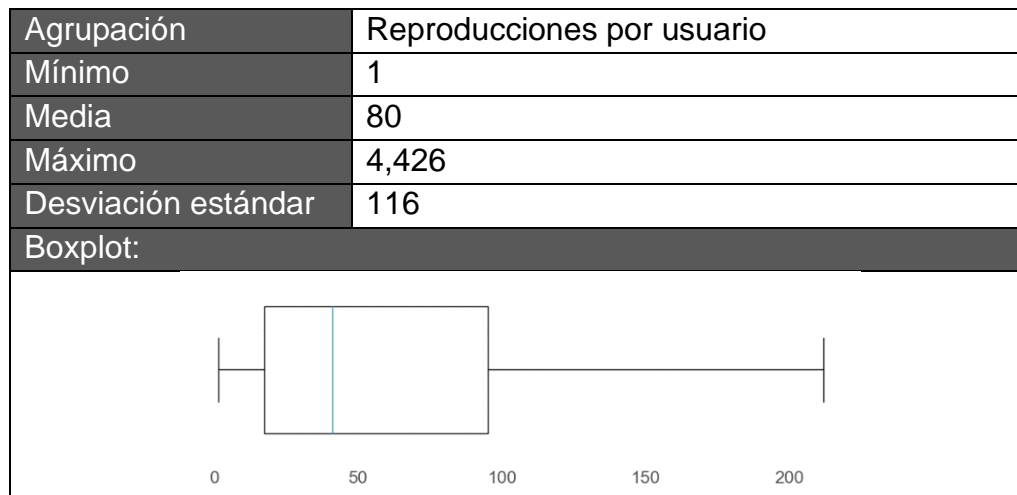
Por este motivo tuvimos que trabajar exclusivamente con el conjunto de datos de reproducciones por usuario y centrarnos en el modelo de Filtro Colaborativo (item-item).

#### 4.2. Análisis descriptivo de los datos del dataset “*Taste Profile subset*”

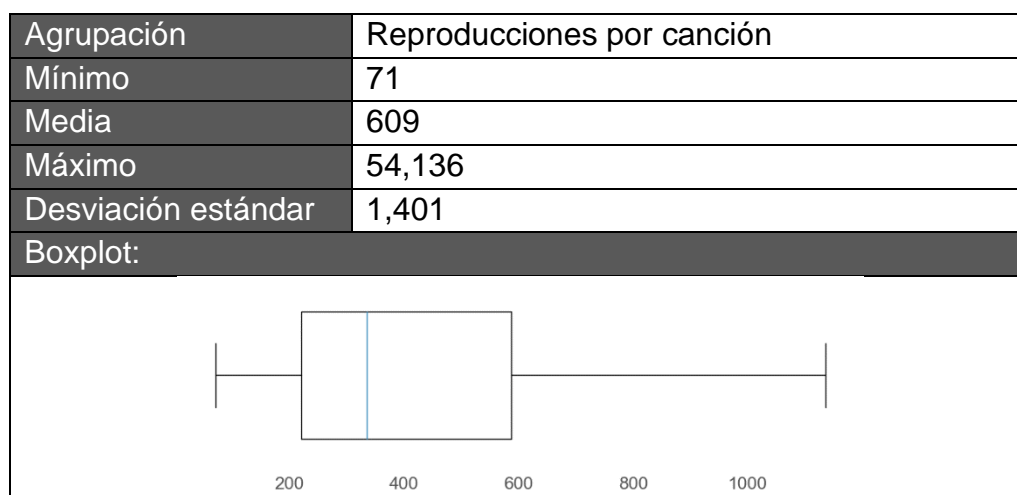
Del conjunto de datos se tomó una muestra de 2 millones de registros o tripletas, estos se agruparon para entender el comportamiento de los usuarios y los resultados fueron los siguientes:

Agrupación	Cantidad de canciones que escucha un usuario
Mínimo	1
Media	26
Máximo	711
Desviación estándar	31
Boxplot:	
	
Nota: se eliminaron los outliers del boxplot por comodidad en su visualización.	

En la anterior agrupación (Cantidad de canciones que escucha un usuario) se puede observar que el 50% de los usuarios escuchan más de 17 canciones, por lo que consideramos que hay información suficiente para poder medir la similitud de las canciones con base en los usuarios comunes que las escuchan.



En esta segunda agrupación es importante notar que hay usuarios con 1 sola reproducción. Estos usuarios deben borrarse del dataset antes de ejecutar el modelo de recomendación pues no aportan suficiente información y no permiten determinar otra canción similar. Adicionalmente, al tener 1 sola canción no es posible usarlos para la etapa de validación.



En esta tercera agrupación de reproducciones por canción observamos que la mayoría de las canciones tienen reproducciones suficientes como para que varios usuarios las hayan reproducido. Se puede concluir que el total de canciones es útil para el modelo.

## 5. Modelo de Filtro Colaborativo

En el modelo de recomendación partimos del supuesto de que, si 2 canciones son escuchadas por una gran porción de usuarios comunes del total de oyentes, se puede decir que las 2 canciones son similares entre sí.

El dataset no cuenta con una valoración explícita de la canción, ni información adicional que nos permita inferir una valoración implícita (como por ejemplo el tiempo que el usuario reprodujo la canción). Entonces decidimos considerar únicamente los registros con más de 2 reproducciones, asumiendo que si un usuario reproduce una canción una segunda es porque le gustó.

En este mismo orden de ideas, es importante aclarar que al no contar con la valoración explícita de la canción debimos usar una métrica binaria que simplemente calculara la similaridad mediante la coocurrencia de usuarios. Para esto se utilizó la distancia de Jaccard, en donde la similitud entre 2 canciones  $i$  y  $j$  estaría dada por:

$$sim_{jaccard}(i, j) = \frac{|usuarios_i \cap usuarios_j|}{|usuarios_i \cup usuarios_j|}$$

Una vez definido esto, empezamos a calcular la similaridad de cada canción con el resto y almacenar estas similaridades en una matriz de coocurrencia. Para esto debíamos consultar el listado de usuarios de cada canción y calcular la similaridad entre los 2 vectores usuarios. Sin embargo, notamos que este proceso no sería escalable pues era computacionalmente muy costoso y la matriz resultante no nos cabría en memoria.

Para resolver este problema tomamos 2 aproximaciones:

- i. En vez de calcular una matriz única de co-ocurrencia, calcular una matriz precomputada por usuario y escribirla en el disco. De esta manera reduciríamos el número de filas de la matriz al número de canciones que hubiera escuchado el usuario. Sin embargo, esto implicaba hacer el de forma iterativa para todos los usuarios.
- ii. En vez de calcular la similaridad de las canciones del usuario con todas las canciones de la colección, calcularíamos su similaridad con una porción menor de canciones relacionadas. Estas canciones relacionadas vendrían de uno o varios clusters de canciones definidos a priori, en los cuales la similaridad no estaría basada en coocurrencia de usuario sino en contenido. Y estos cluster se seleccionaría midiendo la distancia entre las 2 canciones con más reproducciones del usuario y los clusters definidos previamente.

Como no pudimos hacer el modelo de clustering propuesto inicialmente, decidimos (para efectos del prototipo del modelo) utilizar las 5,000 canciones más populares (según número de reproducciones) de la colección. Este concepto podría aplicarse también sobre charts de canciones, por ejemplo: mismo género, misma región o mismo mes de lanzamiento.

Después de contar con la matriz de coocurrencia de las canciones del usuario se calculó un score de recomendación. Este score consiste en un promedio ponderado de las similitudes entre todas las canciones del usuario y cada canción de la colección reducida. Si se tiene la matriz de coocurrencia del usuario, en donde  $i$  son las canciones del usuario y  $j$  las canciones de la colección el score de recomendación esta dado por:

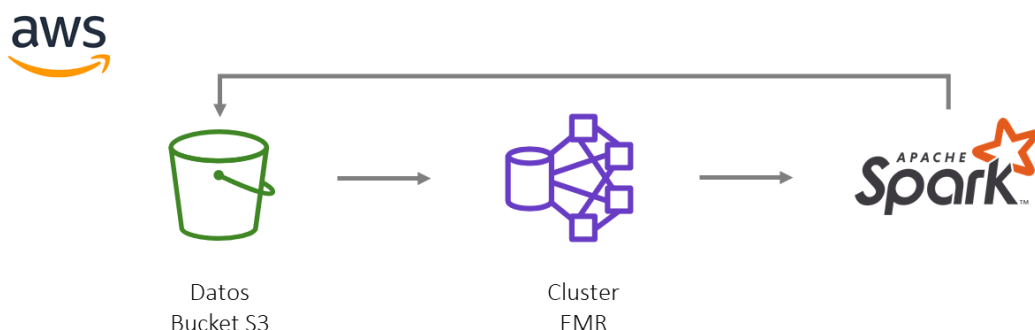
$$score\ de\ recomendación_j = \frac{\sum_{i=1}^n a_{ij}}{n}$$

Así que obtenemos un vector con el score de recomendación para cada canción de la colección reducida y podemos retornar la recomendación o recomendaciones más pertinentes para el usuario.

## 6. Herramientas de Big Data

Las pruebas del modelo se realizaron en un ambiente AWS. Los datasets se almacenaron en un bucket de S3 y fueron procesados en un cluster de 4 nodos tipo m5.2xlarge (1 maestro y 3 principales).

Para aprovechar el procesamiento paralelo se utilizó Apache Spark.





## **7. Entregables**

Adicional al presente informe se entrega el código del proyecto en el siguiente repositorio de Github:

<https://github.com/eljimenezj/Mineria-grandes-volumenes>

El código está presentado en 3 Notebooks de Jupyter, uno para la partición de los datos en train y test, otro para el modelo de recomendación, y un tercer notebook para la evaluación de los resultados.

## **8. Resultados**

El sistema de recomendación se probó en una muestra de 3,000 usuarios. Para estos usuarios se generaron las 30 recomendaciones con mayor score y se midió la métrica de precisión para cada usuario. La precisión media en los 3,000 usuarios fue del 41%.

## **9. Conclusiones y trabajo futuro**

### **9.1. Conclusiones**

- Se implementó un sistema de recomendación mediante Filtros Colaborativos y se obtuvo una precisión del 41% en el conjunto de pruebas.
- Consideramos que el nivel de precisión es bueno dado que el modelo utiliza el número de reproducciones y no una valoración del usuario con la cual el modelo tendría más información para conocer los gustos personales del usuario.
- La solución es práctica como ejercicio académico, pero no es escalable hasta no implementar el modelo de clustering. Pues se necesitan los grupos reducidos de canciones para no tener que calcular la matriz de coocurrencia sobre la colección completa.

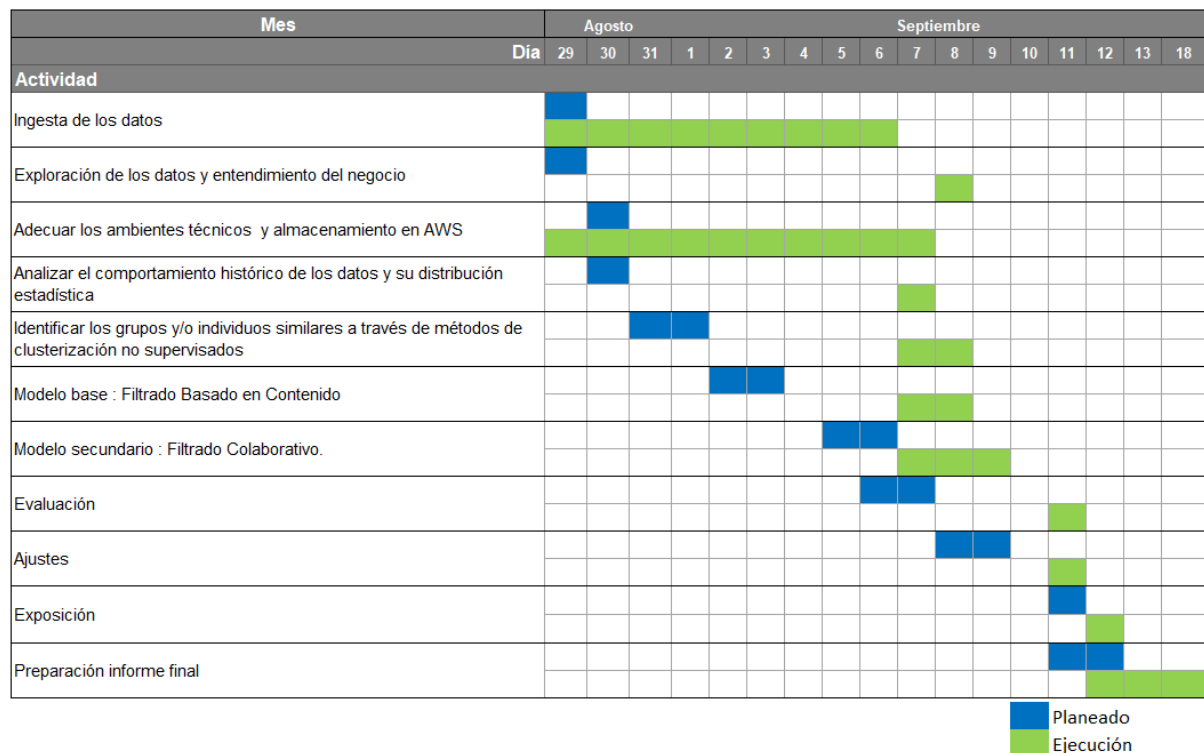
### **9.2. Trabajo futuro**

Consideramos que todavía hay mucho espacio para mejora en el sistema de recomendación. El trabajo futuro deberá centrarse en:

- Conseguir más atributos de las canciones y hacer una agrupación de las canciones por contenido.
- Explorar opciones para calcular una valoración implícita basada en el número de reproducciones o conseguir un dataset que además del número de reproducciones cuente con las valoraciones del usuario para trabajar con una métrica de similaridad que nos permita hacer una predicción del rating.
- Implementar un método híbrido que incluya el sistema de filtro colaborativo buscando mejorar la precisión de las recomendaciones.

## 10. Ejecución del plan

La ejecución del proyecto tuvo varios contratiempos. Inicialmente se estaba contemplando un proyecto diferente, y cuando se decidió ejecutar el presente proyecto se perdió mucho tiempo en el acceso al conjunto de datos. Finalmente el proyecto se desarrolló de acuerdo con el siguiente cronograma:



## 11. Implicaciones éticas

Durante los últimos años, los sistemas de recomendación han sido implementados a lo largo de las ¿industrias? buscando optimizar el tiempo de decisión de los usuarios y como una alternativa de menor fricción en el uso de sus servicios, pese

a ello, el hecho de dar una recomendación lleva consigo la responsabilidad ética de eliminar la mayor cantidad de sesgos discriminatorios y contenido inapropiado que podría ir inmerso en la solución. Con base en esto, es importante considerar que:

- Es posible que las recomendaciones favorezcan injustamente a algunos artistas. Por ejemplo, los discos muy recientes o de artistas pequeños no serían altamente recomendados (sesgo hacia artistas o genero).
- Se deben filtrar recomendaciones con contenidos inapropiados. Por ejemplo, canciones con contenido explícito no se deben sugerir a menores.
- Se debe mitigar el riesgo de que se filtre información personal a agentes externos, que podría revelar orientaciones religiosas, políticas o sexuales del usuario.

## **12. Aspectos legales y comerciales**

En producción, el sistema capturará información personal de los usuarios y su historial de reproducción de canciones, datos pueden que deben custodiarse cuidadosamente pues podrían afectar su intimidad o privacidad. Por lo tanto, al usuario se le debe informar, previo a la suscripción del servicio, el uso que se le dará a su información personal.

Se deberá también solicitar el consentimiento del usuario y se debe dar cumplimiento a lo dispuesto en la Ley Estatutaria 1581 de 2012, conocida como ley de protección de datos personales o Habeas Data.