Portada oficial

Acta

**Abstract**

Contenido:

* Planteamiento del problema o de la necesidad a cubrir.
* Objetivo del trabajo.
* Resultados obtenidos que dan respuesta a dicho problema o necesidad.
* Conclusiones obtenidas.

Longitud máxima de una página.

• Resumen de 200 palabras máximo en Castellano y de 5 a 10 palabras clave

• Resumen de 200 palabras máximo en Inglés y de 5 a 10 palabras clave

**Agradecimientos**

Este capítulo es opcional.

**Contenido**

[1 Introducción 1](#_Toc173585620)

[1.1 Título 2 1](#_Toc173585621)

[1.1.1 Título 3 1](#_Toc173585622)

[2 Sistemas de recomendación 5](#_Toc173585623)

[2.1 5](#_Toc173585624)

[3 Modelos de recomendación 7](#_Toc173585625)

[3.1 Modelo de filtrado por contenido 7](#_Toc173585626)

[3.2 Modelo de filtrado colaborativo 7](#_Toc173585627)

[3.2.1 Historia del modelo colaborativo 7](#_Toc173585628)

[3.2.2 CF basado en memoria 8](#_Toc173585629)

[3.2.3 CF basado en modelo 9](#_Toc173585630)

[3.2.4 CF de múltiples aspectos 9](#_Toc173585631)

[3.2.5 CF de dominio cruzado 10](#_Toc173585632)

[4 Evaluación de modelos de recomendación 10](#_Toc173585633)

[4.1 Métricas predictivas 10](#_Toc173585634)

[4.2 Métricas clasificatorias 11](#_Toc173585635)

[4.3 Métricas de rango 11](#_Toc173585636)

[5 Retos de un modelo de recomendación 13](#_Toc173585637)

[5.1 Arranque en frío(*Cold* *start*) 13](#_Toc173585638)

[5.2 Esparcidad *(Sparsity)* 14](#_Toc173585639)

[5.3 Escalabilidad 15](#_Toc173585640)

[5.4 Sesgo de calificaciones 16](#_Toc173585641)

[6 Conclusiones 18](#_Toc173585642)

[7 Referencias 19](#_Toc173585643)

[8 Referencias 19](#_Toc173585644)

[9 LEYENDA SUBRALLADOS 20](#_Toc173585645)

**Acrónimos**

CF: *Collaborative Filtering*

ROC: *Reciever Operating Characteristic*

En este capítulo se listan los acrónimos que aparecen en el documento, por orden alfabético. Ya no se desplegarán de nuevo en el resto del documento. Si los acrónimos provienen de palabras escritas en un idioma diferente al del documento, se escribirá en cursiva.

# Introducción

Contenido:

* Antecedentes o puesta en contexto.
* Propósito o objetivos.
* Resultados.
* Organización o estructura del documento.

Se aprovecha también este primer capítulo para exponer algunas normas de elaboración en lo que a la forma del documento se refiere.

En primer lugar, en lo que concierne a los capítulos del documento. Este documento debe tener los capítulos tal y como se muestran en esta plantilla, a excepción del capítulo de agradecimientos que es opcional. Evidentemente, para los capítulos intermedios entre el capítulo Introducción y el capítulo Conclusiones, mostrados en esta plantilla a modo de ejemplo como Capítulo 1 y Capítulo 2, el alumno tiene libertad para decidir el número de estos capítulos intermedios así como el nombre de cada uno de ellos.

Cada vez que se crea un capítulo, sección o subsección dentro de un capítulo, se debe utilizar el estilo correspondiente. Para los nombres de capítulo, como es el caso de Introducción, el estilo a utilizar es Título 1. Para los títulos de menor jerarquía, los estilos son los que siguen:

## Título 2

Texto.

### Título 3

Texto.

#### Título 4

Texto.

Por otro lado, en el caso de elaborar listados dentro del texto, los estilos permitidos son los siguientes. Para listados consistentes en elementos breves, menores de 4 líneas aproximadamente, se utiliza el estilo Listado 1.

* Elemento 1.
  + Subelemento.
* Elemento 2.
* Elemento 3.
* Elemento 4.

Para listados consistentes en una sucesión de elementos con tamaño considerable, se utilizará el estilo Listado 2, que aumenta el espaciado entre párrafos facilitando así la lectura:

Elemento 1.

* + Subelemento.

Elemento 2.

Elemento 3.

Elemento 4.

Queda a criterio del alumno la decisión última de utilizar un estilo de listado u otro.

A grandes rasgos, el alumno debe tener en cuenta en la elaboración del documento que sólo puede utilizar los estilos creados en este documento, a saber:

* Listado 1.
* Listado 2.
* Normal.
* Título 1.
* Título 2.
* Título 3.
* Título 4.

Queda a criterio del alumno el uso de **negrita**.

La letra en cursiva se utilizará en aquellas palabras escritas en un idioma diferente al que se ha escrito el documento. Por ejemplo:

* Para realizar un mezclado digital se utilizará un *down-converter*.
* En este caso se utiliza un esquema de *routing* híbrido.
* En este caso se utiliza un esquema de encaminado híbrido.

El escrito del documento debe realizarse en modo impersonal. En el caso de querer enfatizar un alto nivel de implicación en alguna tarea o punto crítico, se podrá utilizar en última instancia la primera persona del plural.

Es muy común que en documentos de TFG-TFM o similares se utilicen elementos de apoyo como figuras, tablas o fórmulas. Cada vez que aparezca uno de estos tres elementos se debe asegurar:

* La asignación de un título para dicho elemento.
  + Referencias 🡪 Insertar título.
* Una vez creado el título, éste debe ser llamado o referenciado desde el texto mediante una referencia cruzada.
  + Referencias 🡪 Referencia cruzada.

Por ejemplo: Tal y como se puede observar en la Figura 1, en la Ecuación 1 y de manera resumida en la Tabla 1.



Figura 1. Sensor de piezoeléctrico de Pro-Wave.

Ecuación 1. *Fast Fourier Transform*.

Los elementos ecuación deben escribirse con el editor de ecuaciones.

Tabla 1. Control de cambios.

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Autor | Fecha | Cambios respecto la versión anterior |
| 00 | RA | 29/09/11 | No aplica. |
| 01 | RA | 04/11/11 | * Definición tipología de receptores. * Inclusión aspectos normativos. * Particularización arquitectura de interfaz radio. * Se desestima posicionamiento. |

Se debe notar que los títulos asociados a figuras y ecuaciones se sitúan en la parte inferior de dichos elementos mientras que los títulos asociados a tablas se sitúan en la parte superior de éstas. La forma de la tabla queda a criterio del alumno siempre que se mantenga un estilo limpio y que no dificulte su lectura y comprensión.

Finalmente se aborda el tema de las referencias a otros documentos. Es muy usual y no puede ser de otra manera que un documento como este se apoye en otros documentos (libros, artículos de congreso, artículos de revista, etc.) para poder conseguir un discurso científico y/o técnico coherente. La estructura de este documento ya dispone de un capítulo de referencias (último capítulo). La manera de elaboración de referencias es la siguiente. Cuando se necesite hacer referencia a un documento de apoyo, en primer lugar se acudirá al capítulo de referencias y se irán añadiendo referencias (estilo Referencia) a los documentos que el alumno considere oportunos. Suponiendo que se desea hacer una referencia a la última referencia añadida se debe ir a Referencias 🡪 Referencia cruzada 🡪 Elemento numerado y Número de párrafo. Aquí la referencia cruzada al último elemento [3] .

# Sistemas y modelos de recomendación

Hoy en día con el crecimiento masivo de los contenidos online, los usuarios están inundados de elecciones por lo que es crucial ofrecer recomendaciones para aumentar la satisfacción y retención del usuario. (1) Es por eso que los sistemas de recomendación cada vez son más comunes en plataformas de internet. Listando algunos de los grandes ejemplos se incluyen la recomendación de películas de Netflix (2), el ranking de páginas web de Google (3), la recomendación de productos de Amazon (4) o la recomendación social de Facebook (5).

Los sistemas de recomendación se usan para recomendar objetos o productos a un usuario en base a sus interacciones previas: compras, visitas, intereses, calificaciones, lista de deseos, reseñas, etc que demuestran el interés de usuario a partir del cual se es capaz de presentarle productos adecuados. Recomendar es una forma particular de filtrar información y está dentro del mundo de la minería de datos y del aprendizaje automático. (6)

Entonces, está claro que cualquier sistema de recomendación necesita recopilar retroalimentación de los usuarios. Dicha retroalimentación puede ser explícita o implícita.

• La **retroalimentación explícita** son esas interacciones donde el usuario explícitamente expresa su satisfacción de un objeto. Ejemplos de información explícita son: calificaciones numéricas dadas después de comprar un producto, pulgar arriba o abajo después de ver un vídeo, reseñas escritas después de leer un libro, etc. Este tipo de información suele ser una información detallada sobre cuánto el usuario le gustó un objeto pero a menudo es difícil de recoger ya que la mayoría de usuarios usualmente no escriben reseñas o dan calificaciones explícitas de cada producto que compran. (1)

• La **retroalimentación implícita** son esas interacciones que realiza el usuario de las cuales se puede inferir cierta información sobre sus gustos pero de forma indirecta. Serían ejemplos: la lista de los vídeos o canciones reproducidas, los objetos comprados, el historial de navegación, el tiempo por página o los clicks realizados, etc. Este tipo de información suele ser información menos detallada y más ruidosa (que se compre un producto no implica que sea para esa persona, podría ser un regalo. Tampoco implica que le haya gustado. Etc.). No obstante, este tipo de información es recopilada de forma mucho más abundante. Por esa razón el ruido se vuelve despreciable cuando se compara con la inmensa cantidad de datos disponibles de ese tipo. De hecho la mayoría de sistemas de recomendación tienden a depender de la información implícita. (1)

Tabla 2. Ejemplo de matriz de calificación explícita e implícita.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Matriz rij de retroalimentación explícita  rij = Calificación del usuario *i* de la película visualizada *j* de 0 a 5 estrellas | | | | |  | Matriz rij de retroalimentación explícita  rij = El usuario *i* ha visto la película *j*? (0=No, 1=Sí) | | | | |
|  | Death Race | Need for Speed | Driven | Fast & Furious 7 |  |  | Death Race | Need for Speed | Driven | Fast & Furious 7 |
| Maria | Nulo | Nulo | 3 | 0 |  | Maria | 1 | 0 | 1 | 1 |
| Joan | Nulo | Nulo | Nulo | Nulo |  | Joan | 1 | 1 | 0 | 0 |
| Pere | Nulo | 1 | 3 | 5 |  | Pere | 0 | 1 | 1 | 1 |
| Laura | 5 | Nulo | 4 | 4 |  | Laura | 1 | 1 | 1 | 1 |
| Arnau | 3 | 4 | Nulo | 2 |  | Arnau | 1 | 1 | 0 | 1 |

Recommender systems represent a specific type of software that are designed to predict user behavior and suggest items that are of most interest for a user at a given time (items can be movies, articles, music, etc.). As a result, user satisfaction grows and so does the service’s income. The income does not have to be direct revenue; the service can capitalize on gaining user loyalty and retention, that in a long run might turn into revenue from ads, subscriptions, or others [1]. With giants like Netflix, YouTube, Amazon and many others, recommender systems (RecSys) gained influence in the in- dustry and became an important part of service-user interaction. Depending on the business model, RecSys may be the core of the service (think TripAdvisor or Booking) or can be a complimentary service that eases navigation and decision making. (7)

## Modelo de filtrado por contenido

El modelo de recomendación por contenido predice las preferencias de un usuario por un producto u otro basado únicamente en calificaciones explícitas del mismo usuario y en las características del producto. (7) Para este tipo de filtrado es necesario tener una descripción de todos los productos disponibles y una opinión explícita del usuario. (8)

Es un método simple que funciona bien delante de problemas de arranque en frío y no requiere trabajar con excesivas cantidades de datos. Usualmente el modelo de filtrado por contenido es aplicado a películas, libros o artículos donde las características de esos objetos suelen estar bien detalladas y basarse en valorar la opinión del usuario suele ser bastante satisfactorio. (8)

Sin embargo, el método carece al ser muy influenciable tanto por el propio usuario como por la descripción de los objetos por lo que acaba sufriendo en diversidad de recomendaciones. (8) Por ese motivo es interesante introducir informaciones de segundos usuarios, método en el que se enfoca el modelo de filtrado colaborativo.

KNN model

## Modelo de filtrado colaborativo

El modelo de recomendación por CF predice las preferencias de un usuario por un producto u otro basado en preferencias previas de otros usuarios. A diferencia del modelo de filtrado por contenido, el CF no solo tiene en cuenta las interacciones del propio usuario a recomendar con los diferentes objetos, sino que también tiene en cuenta decisiones similares de otros usuarios. La base lógica del modelo es la de asumir que los usuarios que estaban de acuerdo en el pasado, también lo estarán en el futuro. (7) (9) (10)

Por tanto, en CF la recomendación se construye basada en las interacciones de otros usuarios de características similares que interactuaron con el mismo objeto. Al requerir usar informaciones de otros usuarios para decidir, CF necesita conocer el conocimiento grupal, o sea identificar qué representa similitud entre los usuarios. Por este motivo las recomendaciones se basan en filtrar usuarios con patrones de comportamiento similares. (7)

Dado que para funcionar el método requiere un vasto conocimiento previo de los hábitos e interacciones de los usuarios, el CF presenta limitaciones en una situación de arranque en frío cuando se carece de ese conocimiento previo. Por otro lado, el modelo CF, a diferencia del modelo basado en contenido, no está exclusivamente ligado a las características del usuario y objeto lo que genera que el alcance de las recomendaciones sea más amplio y variado. (7)

Matrix factorization –> Librería “surprise”

Info básica de la matemática y conceptos de Matrix factorization en: https://towardsdatascience.com/recommender-systems-a-complete-guide-to-machine-learning-models-96d3f94ea748

### Historia del modelo colaborativo

CF fue propuesto por primera vez por Goldberg et al. en 1992 (10). Propusieron un sistema de calificación de correos electrónicos como “buenos” o “malos” que permitiría filtrar los correos. (11) Su trabajo demostró la efectividad incorporar la actividad humana en el proceso filtrado en contraste a un filtrado por contenido. (9)

Tres años más tarde, Hill et al. (12) compararon calificaciones de vídeos entre usuarios para encontrar personas con intereses similares y dar recomendaciones basadas en calificaciones de usuarios similares. (9)En el mismo año, Shardanand et al. (13) usaron CF para hacer recomendaciones personalizadas de música y artistas basándose en las similitudes entre el perfil de interés y perfiles similares. (9)

Finalmente, en los primeros años del CF fue destacable que Grouplens (14) desarrolló un sistema CF que demostró poder implementar el CF para predecir las calificaciones de los usuarios. En concreto Grouplens creó un sistema para Usernet news que permitía a los lectores calificar los artículos. (9)

En los últimos treinta años, CF se ha convertido en el modelo más usado para generar recomendaciones. Ya han sido varios los algoritmos que han sido propuestos y evaluados tanto en el mundo real como con datos artificiales de testeo. (9) Estudios como (15) o (16) acaban categorizando los algoritmos de CF en dos tipologías: basados en memoria y basados en modelo. Se detallan dichos tipos en los siguientes dos apartados.

### CF basado en memoria

Los algoritmos de CF basados en memoria predicen los gustos de los usuarios basándose enteramente en la colección de calificaciones previas por usuarios de los objetos. De esta forma la predicción se computa como una agregación de las calificaciones de otros usuarios escogidos por su similaridad. Investigaciones iniciales se enfocaron en métodos de vecinos más cercanos para determinar la similaridad entre usuarios u objetos. Los algoritmos basados en memoria se pueden clasificar en: basados en usuario o basado en objeto según si el proceso de búsqueda de vecinos se enfoca en usuarios o en objetos similares. (9)

#### CF basado en memoria de usuarios

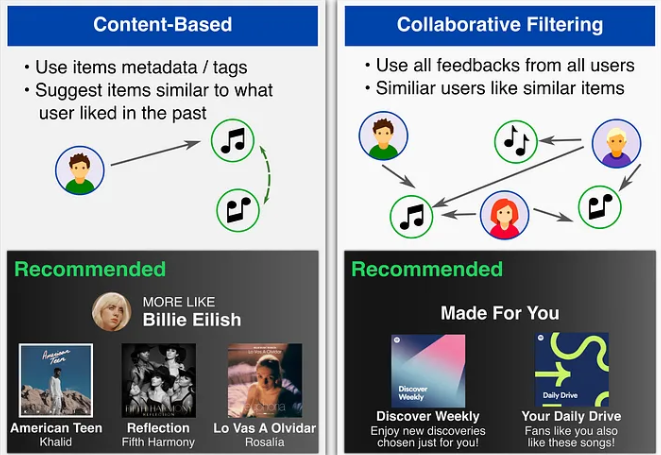
La base lógica de este modelo es la de asumir que usuarios con gustos similares tienden a preferir los mismos productos. Por eso, este método filtra para usar solamente informaciones de usuarios con comportamientos similares al usuario que se quiere recomendar. (9) Recomienda al usuario productos que gustaron a usuarios similares. (7) El CF basado en memoria de usuarios fue propuesto por primera vez Resnick et al. (17) que usaron el coeficiente de regresión de Pearson como métrica de similitud para seleccionar los vecinos del usuario a recomendar.

Más info en: **Guan, Xin.** *On Reducing the Data Sparsity in Collaborative Filtering Recommender Systems. Pags 17-18-19*

#### CF basado en memoria de objetos

La base lógica de este modelo es la de asumir que productos similares al producto que gusta al usuario, también será de interés por el usuario. (9) Por eso, en este modelo se escoge un objeto que gustó al usuario y entonces hay dos opciones: se busca filtrar objetos similares (filtrado por contenido) o bien se identifica otros usuarios a quien también les gustó el producto y se recomienda al usuario original objetos de interés de estos otros usuarios (filtrado colaborativo). (7) El algoritmo comúnmente más usado en este caso es el de la similitud del coseno. (9)

Elaborar una imagen similar a la siguiente para ilustrar las explicaciones del texto:



https://towardsdatascience.com/recommender-systems-a-complete-guide-to-machine-learning-models-96d3f94ea748

Más info en: **Guan, Xin.** *On Reducing the Data Sparsity in Collaborative Filtering Recommender Systems. Pags 19-20-21*

#### Similaridad calculada a partir de *cosine-similarity*

#### Similaridad calculada a partir de *Pearson correlation coeficient*

### CF basado en modelo

Los algoritmos de CF basados en modelo predicen los gustos de los usuarios basándose en modelos de aprendizaje que aprendieron de una colección de calificaciones.

Ejemplos de modelos que se hicieron en: **Guan, Xin.** *On Reducing the Data Sparsity in Collaborative Filtering Recommender Systems. Pag 22 a 27*

Diversos estudios han sugerido que el CF basado en modelo dota de una mayor precisión predictiva que el CF basado en memoria gracias al uso de técnicas como la factorización matricial y la reducción de dimensiones. (11) (18) (19)

Más información de matrix factorization, clustering y deep learning en **Alieva O. A., Gangan E. S., Ilyushin E. A., Kachalin A. I.** *Automatic Evaluation of Recommendation Models* pag 3 de pdf abajo derecha: Los algoritmos de aprendizaje automático usados se pueden clasificar en: factorización matricial (*matrix* *factorization*), agrupamiento (*clustering*) y aprendizaje profundo (*deep learning*).

Más información de collaborative filtering algorithms (Top-N Recommendations, SVD, Autoencodersm, Restricted Boltzmann Machines) en **Alieva O. A., Gangan E. S., Ilyushin E. A., Kachalin A. I.** *Automatic Evaluation of Recommendation Models* pag 4 y 5 de pdf

### CF de múltiples aspectos

La mayoría de los sistemas de recomendaciones por CF actuales mantienen las matrices de análisis restringidas a calificaciones de un único criterio. Sin embargo, estudios más recientes indican que considerar múltiplos aspectos de los objetos puede mejorar la predicción y niveles de precisión de las recomendaciones. Esto da lugar al sistema de recomendación por filtrado colaborativo de criterio múltiple (*Multi-Criteria Collaborative Filtering*). En este sistema se recogen calificaciones de múltiples aspectos de los objetos aumentando así el tamaño de la matriz e influenciando en los problemas de la escasez de datos y su escalabilidad. Este sistema de recomendación requiere técnicas para reducir las dimensiones de la matriz que mejoran la precisión y eficiencia del recomendador. (11)

Más info del planteamiento técnico de MULTI-CRITERIA ITEM-BASED COLLABORATIVE FILTERING en: **Girase, S, Mukhopadhyay, D y Bokde, D.** *An Item-Based Collaborative Filtering using Dimensionality Reduction Techniques on Mahout Framework.* s.l. : arXiv preprint arXiv, 2015. 1503.06562. APARTADO II y IV

### CF de dominio cruzado

A diferencia de los sistemas de recomendación de un solo dominio que tratan la información independientemente en su único campo de aplicación (pongamos por ejemplo la recomendación de películas), los sistemas de recomendación de dominio cruzado buscan mejorar la información de recomendación explotando datos de otros dominios de conocimiento (como por ejemplo la recomendación de libros o música). (9) Y, es que, si lo pensamos tiene lógica. Los gustos de un usuario por una película o un género de esa están relacionados con los gustos del usuario por un libro o tipología de libro. Pongamos por ejemplo Harry Potter o Señor de los Anillos. A un usuario que le gustaron esas películas, es más propenso a que le gusten los libros o música similar a la de la película frente a un usuario que no le gustó.

Los sistemas de dominio cruzado normalmente se apoyan de fuentes auxiliares que contienen abundantes datos de preferencia de usuario. Hay dos tipos de aproximaciones según como se explota esta información auxiliar: ya sea agregándola (usado junto la principal) o bien transfiriéndola (mezclando indistintamente con la principal). (9)

Más info sobre como se agrega o transfiere información en: **Guan, Xin.** *On Reducing the Data Sparsity in Collaborative Filtering Recommender Systems. Pag 33 a 40*

# Retos de un modelo de recomendación

En el mundo real el CF sufre el problema de la falta de datos ya que a menudo se tienen escasos datos sobre las preferencias de los usuarios hecho que dificulta la tarea de comprender las verdaderas preferencias del usuario y realizar una recomendación satisfactoria. (9) O no solo en si escasos datos sino también datos esparcidos. Es decir, con muchos valores nulos (que la cantidad de usuarios por campo es mucho mayor a la cantidad de datos).

Las mismas soluciones se aplican para el problema de falta de datos. Al final, el aumento del conocimiento de las preferencias del usuario siempre pasa por recoger más datos de los usuarios. Sin embargo, la utilidad de cada información recogida puede variar significativamente. Por eso es importante obtener datos que sean capaces de reflejar de la mejor forma posible las preferencias de los usuarios. Aquí aparecen dos estrategias de recopilación de datos: (9)

• Las estrategias personalizadas que piden o se recogen unos datos u otros según las características de cada usuario. (9)

• Las estrategias no personalizadas donde se recopilan los mismos datos para todos los usuarios. (9)

## Arranque en frío(*Cold* *start*)

El problema de arrancar el sistema en frío sin información (conocido en inglés como *cold start*) es sufrido por la mayoría de los algoritmos CF cuando un usuario u objeto entra por primera vez al sistema. Al no haber información inicial del usuario u objeto, el CF falla al generar recomendaciones. (9) El problema del arraque frío también se conoce como problema del usuario nuevo (como el caso de Joan en la Tabla 1) o como problema del objeto nuevo (como el caso de Fast & Furious 8 en la Tabla 1)

Tabla 3. Ejemplo del problema del arranque frío con un usuario nuevo (Joan) y una película nueva (Fast & Furious 8) en una matriz de calificación del 0 al 5.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Death Race | Need for Speed | Driven | Fast & Furious 7 | Fast & Furious 8 |
| Maria | Nulo | Nulo | 3 | 0 | Nulo |
| Joan | Nulo | Nulo | Nulo | Nulo | Nulo |
| Pere | Nulo | 1 | 3 | 5 | Nulo |
| Laura | 5 | Nulo | 4 | 4 | Nulo |
| Arnau | 3 | 4 | Nulo | 2 | Nulo |

Las soluciones más comunes aplicadas para resolver el problema del usuario u objeto nuevo son: (9)

• Hacer contestar unas preguntas iniciales al usuario o añadir información inicial del objeto.

• Aplicar un sistema de aprendizaje activo (Active Learning) para aprender de ese usuario u objeto.

• Encarar una aproximación con un modelo de recomendación híbrido que combine el CF con el contenido. Sin embargo, esta aproximación tiene sus limitaciones y complicaciones. (9)

## Esparcidad *(Sparsity)*

Como se indicaba al principio del punto, a menudo hay una falta de datos sobre las preferencias del usuario. Y no es tanto la falta absoluta de datos (bajo número de datos) sino sobre todo la cantidad de datos esparcidos. O sea que la cantidad de usuarios por campo es mucho mayor a la cantidad de datos, hecho que hace difícil cruzar y usar bien los datos. Se detalla y ejemplifica más concretamente el concepto de espacdidad (*sparsity*).

En el ejemplo de la Tabla 1 se tiene 5 usuarios (Maria, Joan, Pere, Laura y Arnau) y 5 campos (Death Race, Need for Speed, Driven, Fast & Furious 7, Fast & Furious 8). A pesar de que el número de datos en valor absoluto es no es bajo respecto a la cantidad de personas (se tienen **11 datos** entre los 5 usuarios), sí que lo es respecto a toda la cantidad de datos posibles que se podrían tener (se podrían tener **25 datos**; 5 usuarios por 5 campos). Es por eso que la esparcidad es baja. En concreto la esparcidad se define según (20) como:

Ecuación 2. Expresión para calcular la esparcidad donde # denota la cantidad total.

En el ejemplo que se hablaba de la tabla 1 se cuenta con y con por lo que la fracción de datos que se tiene [11] respecto a los que se pueden tener [25] es de y la fracción contraria a esa, o sea la cantidad de datos faltantes [14] respecto a los que se pueden tener [25], eso es la esparcidad es de: .

Por lo tanto, se entiende que la esparcidad significa el porcentaje de valores faltantes en la matriz usuario-objeto. Y es que el problema de la esparcidad representa uno de los mayores cuellos de botella para los algoritmos de CF. En la mayoría de los sistemas de recomendación, el número de calificaciones obtenidas para cada usuario suele ser muy pequeño en comparación a la cantidad disponible de objetos en el conjunto de datos. Por esa razón las matrices usadas para CF suelen tener una esparcidad muy alta (como se muestra en la tabla 2).

Tabla 4. Esparcidad de diferentes conjuntos de datos de sistemas de recomendación. (21)

|  |  |
| --- | --- |
| ***Dataset*** | ***Sparsity*** |
| Movielens 100K | 93,70% |
| Movielens 1M | 95,75% |
| Movielens 10M | 98,69% |
| Book-Crossing | 99,998% |
| Jester v1 | 45,52% |
| Jester v2 | 80,83% |
| Yahoo! Music | 99,96% |

While most research in the field of recommender systems focus on improving prediction algorithms, even the best algorithm will fail without sufficient data. Take movie recommendation as an example, the movies that have been rated with only few ratings would be recommended rarely, even with high ratings. Also, users with special tastes for movies usually suffer from poor recommendations since similar users are rare in the system.

Problems of sparsity in machine learning and how to solve them with small Python code: https://www.kdnuggets.com/2023/04/best-machine-learning-model-sparse-data.html

Efectos del Multi-Criteria Collaborative Filtering en el problema de sparsity:

**Improvement:**

1. **Additional Information**: MC-CF uses multiple criteria for rating items, which can provide more detailed user profiles and item descriptions. This additional information can help fill in gaps where single-criterion ratings are sparse.
2. **Better Matching**: With more criteria, there are more opportunities to find commonalities between users and items, potentially leading to better matches and recommendations even in the presence of sparse data.

**Worsening:**

1. **Increased Dimensionality**: The inclusion of multiple criteria increases the dimensionality of the data. If users do not rate items across all criteria, it can lead to sparsity in these additional dimensions.
2. **Complex Data Collection**: It may be more challenging to collect sufficient ratings for all criteria, exacerbating sparsity if users are not motivated to provide detailed feedback.

**Summary:**

**Sparsity**: MC-CF can both mitigate and exacerbate sparsity. It mitigates it by providing richer user profiles but can worsen it due to the increased data requirements for multiple criteria.

## Escalabilidad

A Data-set and user rating increases with time gradually which may take long processing times. More processing power need highly configured CPU and RAM consumption and all the clusters are so busy that extra hardware is added at runtime to maintain the processing accuracy which is costly and need maintenance. (6)

Compared to memory-based algorithms, model-based algorithms scale better in terms of resource requirements. (11)

Efectos del Multi-Criteria Collaborative Filtering en la escalabilidad:

**Improvement:**

1. **More Accurate Predictions**: With more criteria, the recommendations can be more accurate, potentially reducing the need for complex algorithms to improve recommendation quality.
2. **Enhanced Clustering**: The richer data can improve clustering of users and items, making the recommendation process more efficient.

**Worsening:**

1. **Increased Computational Complexity**: The increased dimensionality from multiple criteria means more data to process, which can lead to higher computational costs and greater challenges in scaling the algorithms.
2. **More Storage Requirements**: Handling additional criteria requires more storage, which can become problematic as the number of users and items grows

**Summary:**

**Scalability**: MC-CF generally worsens scalability due to increased computational complexity and storage requirements. However, it can improve the efficiency of the recommendation process through better user-item matching and clustering.

## Sesgo de calificaciones

Users rate their liked products and brands they use and rate them on the basis of their experiences and some of them gives low rating and reviews to their opponents (6)

# Evaluación de modelos de recomendación

La calidad de un sistema de recomendación se basa en los resultados de su evaluación. En la evaluación se seleccionan métricas según el tipo y aplicación del CF. Las métricas usadas en la evaluación se dividen en tres clases: predictivas, clasificatorias y de rango. (9)

Estas tipologías de métricas miden cosas diferentes. Ya que, por ejemplo, aunque el sistema clasifique correctamente los objetos para un usuario (acierte en recomendar 10 objetos que serían los que el mismo usuario escogería) quizás los esté ordenando incorrectamente (el objeto que más quiere el usuario sea el décimo recomendado) o quizás aunque los ordene correctamente el sistema esté fallando en predecir correctamente las calificaciones (el usuario pondría un 8 al objeto top 1 y el sistema lo calificó con un 10). (9) Por esa razón es importante distinguir estos distintos tipos de métricas y saber diferenciar el significado de lo que indica cada una de ellas.

## Métricas predictivas

Las métricas predictivas miden, para cada objeto recomendado, la diferencia entre la predicción del sistema de recomendación y las calificaciones reales hechas por un usuario. (9)

• Error Absoluto Medio (*Mean Absolute Error*). Mide la desviación entre la calificación predicha y la calificación dada por el usuario. Donde es la calificación dada por el usuario en el objeto , es la calificación predicha para ese usuario y objeto y es el número total de muestras. (9)

Ecuación 3. Error Absoluto Medio.

• Error cuadrático medio (*Root Mean Square Error*). Mide las desviaciones de los errores entre las predicciones y los valores reales amplificando las contribuciones de errores grandes en comparación al Error Absoluto Medio. (9)

Ecuación 4. Error cuadrático medio.

## Métricas clasificatorias

Las métricas clasificatorias miden, dado un usuario, la frecuencia con la que el sistema recomendador hace recomendaciones relevantes o irrelevantes de un objeto. Las métricas relacionadas con la clasificación se suelen usar cuando el recomendador realiza un número determinado de recomendaciones a la vez para el usuario en cuestión. Lo que se conoce como *top-N recomendation*. (9)

• La precisión (*Accuracy*) mide la proporción o fracción de recomendaciones correctas.

Ecuación 5. Precisión o *accuracy*.

Así, una precisión alta significa que hay pocos falsos positivos, es decir, indica que la probabilidad de que la recomendación sea correcta es alta. Una precisión baja implica que hay bastantes falsos positivos, es decir, muchas recomendaciones marcadas como positivas que son incorrectas.

• La recuperación (*Recall*) o también llamada sensibilidad o tasa de verdaderos positivos es la proporción de recomendaciones positivas correctas respecto al total de recomendaciones positivas en los datos de prueba.

Ecuación 6. Recuperación, sensibilidad, tasa de verdaderos positivos o *recall.*

La sensibilidad es útil cuando el objetivo principal es identificar todos los casos positivos, incluso a costa de indicar incorrectamente algunos ejemplos negativos como positivos. Una recuperación alta indica que el modelo “recupera” correctamente todos los casos positivos. Es decir, indica que el modelo es capaz de detectar todas las recomendaciones que serían positivas, se pierden pocas instancias positivas. Una recuperación baja indica que pocos casos que serían positivos fueron detectados correctamente por el modelo. Es decir, el modelo no es capaz de detectar las recomendaciones que serían positivas por el usuario, se pierden bastantes instancias positivas.

* El F1 score es una medida que combina precisión y recuperación en un solo valor. Es especialmente útil cuando hay un desequilibrio entre las recomendaciones.

Ecuación 7. F1 Score.

Un F1 score alto indica que tanto la precisión como la recuperación son altas y están bien equilibradas. Un F1 score bajo indica que tanto la precisión como la recuperación son bajos o bien que están desequilibradas. O sea, el modelo tiene muchos falsos positivos y/o muchos falsos negativos.

• La Precisión Media Promedio (*Mean Average Precision*) mide la calidad de la clasificación de los resultados recomendados en términos de precisión, teniendo en cuenta el orden en el que se presentan. Esta métrica es particularmente útil en contextos donde el orden de los resultados es crucial como en los sistemas de recomendación. (22)

Para calcularla primero se calcula la Precisión Media para cada usuario, que es la media de la precisión en los puntos donde hay recomendaciones relevantes. Por eso es la precisión considerando los *k* primeros elementos y es la función relevancia que vale 1 si el elemento de la posición *k* es relevante y 0 en caso contrario. Una vez calculados las Precisiones Medias para las *N* recomendaciones relevantes de cada usuario *u,* se calcula la Precisión Media Promedio haciendo el promedio para los *U* usuarios existentes en el sistema. (22)

Ecuación 8. Cálculo de la Precisión Media Promedio a través de la Precisión Media.

Para interpretar el resultado de la métrica de se puede tener en cuenta que si esa vale 1 significa que todas las recomendaciones hechas por el modelo para todos los usuarios son perfectas (todas las recomendaciones relevantes están presentes y en el orden correcto). Si esa vale 0 significa que el modelo no logró recomendar ningún elemento relevante. (22)

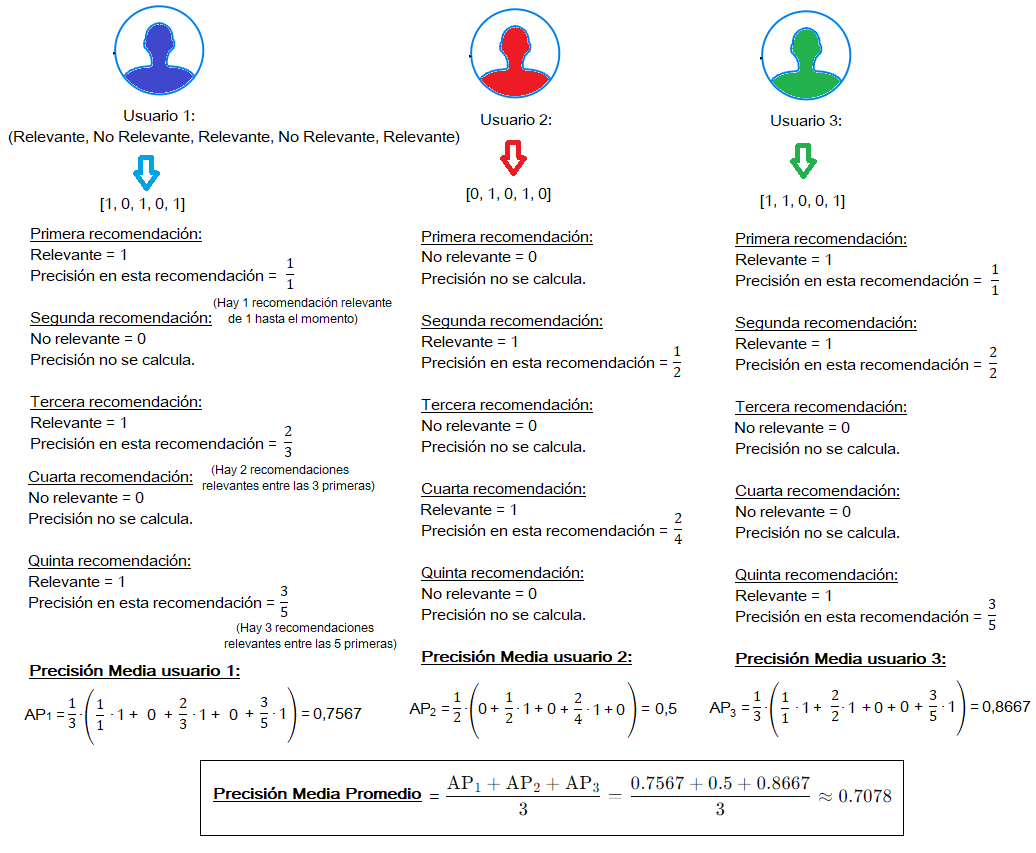


Ilustración 1. Ejemplo de cálculo de Precisión Media y Precisión Media Promedio para 5 recomendaciones a 3 usuarios.

• La Característica Operativa del Receptor (*Reciever Operating Characteristic, ROC*) es una herramienta gráfica de evaluación de los modelos de recomendación binarios. La curva ROC ilustra la relación entre la tasa de verdaderos positivos (o sensibilidad o recuperación) con la tasa de falsos positivos. Recordemos lo que son estas tasas: (23)

Ecuación 9. La tasa de verdaderos positivos, sensibilidad o recuperación mide la proporción de elementos relevantes (Verdaeros positivos + Falsos negativos) que han sido correctamente identificados (Verdaderos positivos).

Ecuación 10. La tasa de falsos positivos mide la proporción de elementos no relevantes (Falsos positivos + Verdaderos negativos) que han sido incorrectamente clasificados como relevantes (Falsos positivos).

Entonces la curva ROC se dibuja asignando al eje x la tasa de falsos positivos y en el eje y la tasa de verdaderos positivos y variando el umbral de decisión de las recomendaciones. Así, la curva empieza en el (0, 0) donde no hay verdaderos positivos ni falsos positivos y termina en el (1, 1) donde todos los elementos son positivos. La curva ROC proporciona una representación visual y cuantitativa del modelo para distintos umbrales, ayudando a decidir si el modelo es adecuado o necesita ajustes adicionales. (23)

• El Área Bajo la Curva (*Area Under the Curve*) es una métrica asociada a la curva ROC que mide el área que queda debajo suya y sirve para cuantificar la capacidad discriminativa de relevante/no relevante del modelo. (23)

Un área alta indica que el modelo es efectivo en distinguir los elementos relevantes de los no relevantes. En concreto si el área es de 1 el modelo es perfecto en separar los elementos relevantes. Si el área es de 0,5 el modelo no tiene capacidad discriminativa, clasifica los elementos aleatoriamente. Si el área es menor a 0,5 el modelo clasifica peor que un modelo aleatorio, invirtiendo las clases.

For other collaborative filtering evaluation metrics, see:

• J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating collaborative filtering recommender systems,” ACM Transactions on Information Systems (TOIS), vol. 22, no. 1, pp. 5–53, 2004.

• G. Shani and A. Gunawardana, “Evaluating recommendation systems,” in Recommender systems handbook. Springer, 2011, pp. 257–297. Métricas predictivas

## Métricas de rango

Las métricas de rango son una extensión de las métricas de clasificación que no solo tienen en cuenta la relevancia o no de los objetos que se han recomendado, sino que también toman en cuenta el rango en el que se han recomendado. Es decir, su posición relativa respecto a más o menos recomendado. Entonces las métricas de rango miden, dado un usuario, la habilidad del sistema recomendador para producir una recomendación clasificada en orden de los objetos de la misma forma que lo ordenaría el usuario. (9)

• La Utilidad de Vida Media (*Half life utility*) se basa en la idea de que la relevancia de un elemento para un usuario disminuye exponencialmente a medida que se aleja de las primeras posiciones de la lista de recomendaciones. Se considera que los usuarios tienden a prestar más atención a los elementos recomendados que aparecen en las primeras posiciones, y que es menos probable que interactúen o encuentren útiles los elementos que se encuentran más abajo en la lista. Es por eso que se da más peso a los elementos en las primeras posiciones y exponencialmente menos peso a los que están más alejados. (24)

La **Utilidad de Vida Media** se formula utilizando la función de decaimiento exponencial basada en la constante de **vida media** (*h*), que representa la posición a partir de la cual la probabilidad de que el usuario considere el elemento relevante se reduce a la mitad. (24)

Ecuación 11. Formulación de la Utilidad de Vida Media donde es la relevancia del elemento ( si es relevante, si no es relevante), es la constante de vida media y es el número total de recomendaciones.

Esta métrica evalúa la calidad de un conjunto de recomendaciones considerando tanto la relevancia de los elementos recomendados como su posición en la lista. Un valor alto de Utilidad de Vida Media indica que el sistema de recomendación coloca los elementos más relevantes en posiciones tempranas, que es lo deseable y útil para el usuario. El valor de la Utilidad de Vida Media se encuentra entre 0 y el número de elementos recomendados *N*. (24)

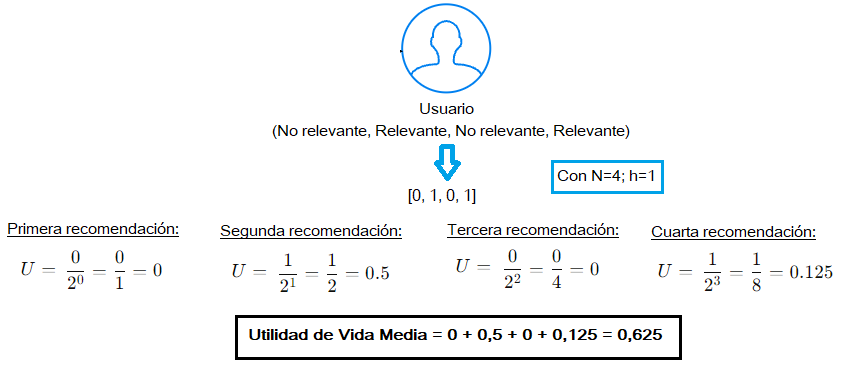


Ilustración 2. Ejemplo de cálculo de Utilidad de Vida Media para un usuario con 4 recomendaciones y vida media h=1 que refleja que dado que los elementos relevantes están en las posiciones 2 y 4, el valor final de la utilidad es bajo debido a la disminución exponencial de la utilidad.

• La Ganancia Acumulativa Descontada (*Discounted Cumulative Gain*) también considera que los elementos relevantes que aparecen en posiciones más altas de la lista deben contribuir más a la puntuación final que los elementos relevantes ubicados en posiciones más bajas. Para considerarlo lo que hace es descontar valor en las posiciones más bajas a través de una relación logarítmica en base 2. (25)

Ecuación 12. Formulación de la Ganancia Acumulativa Descontada donde es la relevancia del elemento ( si es relevante, si no es relevante) y es el número total de recomendaciones.

• La Ganancia Acumulativa Descontada Normalizada (*Normalized Discounted Cumulative Gain*) es la normalización de la Ganancia Acumulativa Descontada según su valor ideal o máximo que se obtendría si todos los elementos relevantes estuvieran perfectamente ordenados. (25)

Ecuación 13. Formulación de la Ganancia Acumulativa Descontada Normalizada donde el máximo de la Ganancia Acumulativa Descontada corresponde al caso en que todos los elementos están perfectamente ordenados.

Con la normalización se consigue que los valores de la métrica vayan de 0 a 1. Donde 1 indica que todos los elementos relevantes están perfectamente ordenados desde la primera posición en adelante y un valor cercano a 0 indica que los elementos están desordenados o que no hay elementos relevantes en las primeras posiciones. (25)

For other collaborative filtering evaluation metrics, see:

• J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, “Evaluating collaborative filtering recommender systems,” ACM Transactions on Information Systems (TOIS), vol. 22, no. 1, pp. 5–53, 2004.

• G. Shani and A. Gunawardana, “Evaluating recommendation systems,” in Recommender systems handbook. Springer, 2011, pp. 257–297. Métricas predictivas

# Implementación

## Esquema de la implementación

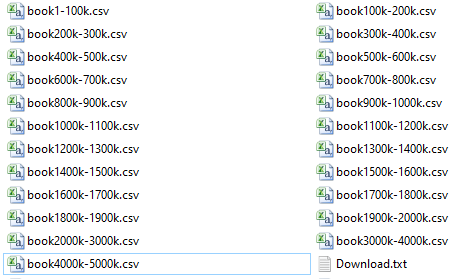
1. Captura de datos
   * Alimentación de Magento con las Scripts de los datos
   * Infraestructura de particionamento
2. Aplicativo de consultas de similitud
   * Organización de los datos según el estado del sistema:
     + Sistema vacío (usuario nuevo)
     + Sistema no vacío (usuario del data set)
     + Sistema inexistente (libro fuera del data set) -> Pedir características del libro para poder dar la recomendación.
     + Sistema existente (libro del data set)
   * Filtrado de la data
     + Otros usuarios recomiendan estos otros títulos similares
     + Otros usuarios han comprado estos títulos
   * Aplicación de métricas estadísticas de evaluación
3. Extensión de Magento que muestre dentro del e-commere las recomendaciones seleccionadas (fuera del alcance del trabajo)

## Conjunto de datos

Los datos con los que se va a realizar la implementación se llaman “Goodreads Books Datasets With User Rating 2M” pertenecientes al autor Bahram Jannesary Soroush Ghaderi y se descargan de la web Kaggle a través del siguiente enlace: <https://www.kaggle.com/datasets/bahramjannesarr/goodreads-book-datasets-10m/data>

### Datos de libros

Este conjunto de datos cuenta con 50 .csv de 100.000 libros cada uno, llegando a un total de 5 millones de libros.



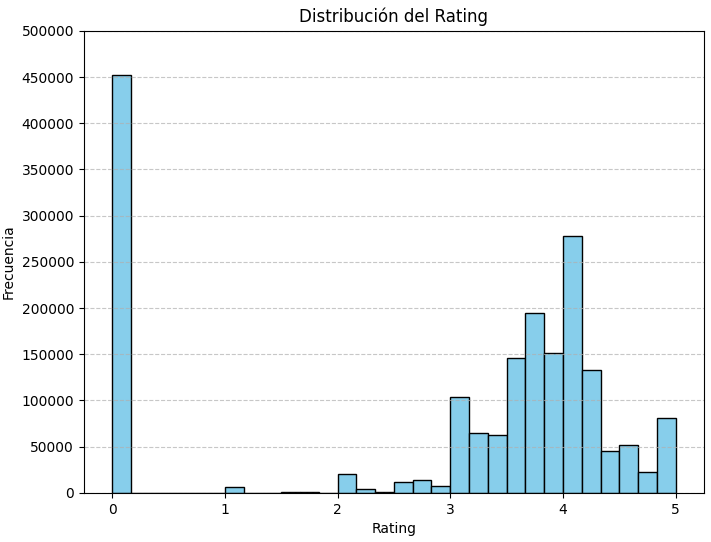
Las columnas de dichos libros son:

* Id: Identificador único del libro en la base de datos.
* Name: El título del libro.
* RatingDist1: Es la cantidad de usuarios que valoró con 1 estrella el libro.
* RatingDist2: Es la cantidad de usuarios que valoró con 2 estrellas el libro.
* RatingDist3: Es la cantidad de usuarios que valoró con 3 estrellas el libro.
* RatingDist4: Es la cantidad de usuarios que valoró con 4 estrellas el libro.
* RatingDist5: Es la cantidad de usuarios que valoró con 5 estrellas el libro.
* RatingDistTotal: Es la cantidad total de votos de los libros (la suma de RatingDist 1, 2, 3, 4 y 5).
* Rating: Es la calificación media del libro.
* pagesNumber: Es la cantidad de páginas que tiene el libro.
* PagesNumber: Es la cantidad de páginas que tiene el libro.
* PublishDay: Es el entero de 1 a 31 que representa el día de publicación.
* PublishMonth: Es el entero de 1 a 12 que representa el mes de publicación.
* PublishYear: Es el entero de 0 a 2020 que representa el año de publicación.
* Publisher: Es el nombre del publicador.
* Description: Descripción del libro.
* CountsOfReview:
* Count of text reviews:
* Language: Es el idioma en que el libro está escrito.
* Authors: Son los autores de los libros.
* ISBN: Es el código ISBN (International Standard Book Number), una versión de 10 dígitos que identifica el libro a nivel internacional.

Explorando los datos df\_book.describe()se observa que hay 1,85·106 Id de libros distintos y 4,85·106 en total, que hay algún mes de publicación incorrecto ya que el máximo es 31, que el contaje de text reviews en el 75% de los casos llega hasta 3 pero que el máximo es un estatrosférico 43652, que el máximo año de publicación es un incorrecto 65 535.

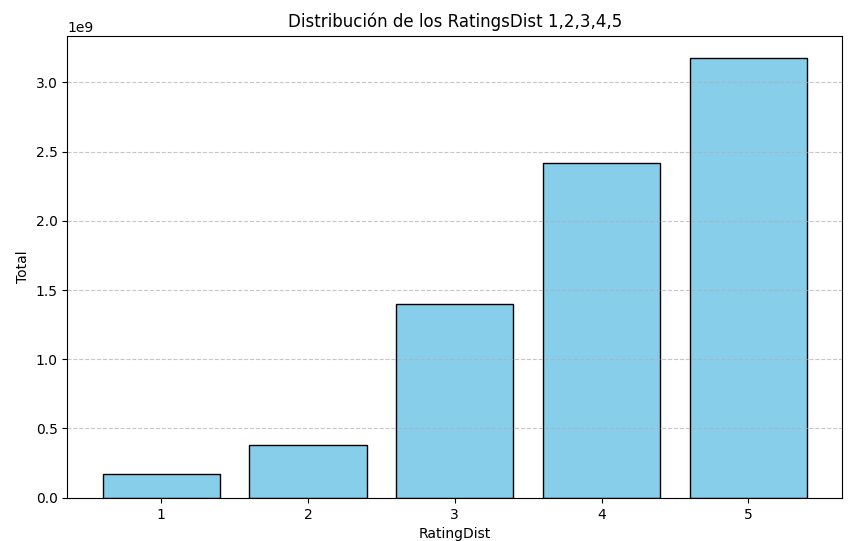
#### Rating

Los valores de ‘Rating’ son tipo float64, no hay ningún valor nulo y ningún valor que quede fuera de los rangos esperados entre 0 y 5. Estos valores se distribuyen de la siguiente forma donde el 24% de los datos tienen 0 como puntuación y no hay ningún valor de 0,algo. Eso hace pensar que la puntuación esperada era entre 1 y 5 y el valor 0 es no esperado o nulo.



Por lo que respecta a los datos de ‘RatingDist1’ , ‘RatingDist2’ , ‘RatingDist3’ , ‘RatingDist4’ , ‘RatingDist4’ , ‘RatingDist5’ y ‘RatingDistTotal’ se ve que son todos de tipo string con ningún valor nulo. De hecho estas columnas tienen el formato string siguiente 1:y el número entero de votos con una estrella, 2: y el número entero de votos con dos estrellas y así sucesivamente. Por lo tanto lo que se hace es convertir esa columna a simplemente el entero que representa el número de votos. Con esa conversión queda un valor de RatingDist5 negativo (un -2) y un valor de RatingDistTotal negativo (un -2). Esos valores son erróneos.

Los RatingDist tienen una distribución correcta y esperable.



Se comprueba por código que efectivamente RatingDist1 + RatingDist2 + RatingDist3 + RatingDist4 + RatingDist5 = RatingDistTotal para todos los datos.

#### Id y Name

No hay valores nulos de Id ni Name.

Se observa que hay 195 libros que tienen la Id duplicada a lo largo del conjunto de datos.

Se observa que 152 752 libros tienen el título repetido con otra fila del conjunto de datos hasta 24 veces.

#### Language

El 86,39% de los libros tienen un valor nulo de language. No sería una columna a utilizar por el gran volumen de falta de valores ya que solo hay 251 809 valores de 120 idiomas diferentes.

#### Authors

No hay ningún valor nulo en la columna de autores. Hay 675 289 autores únicos.

#### Description

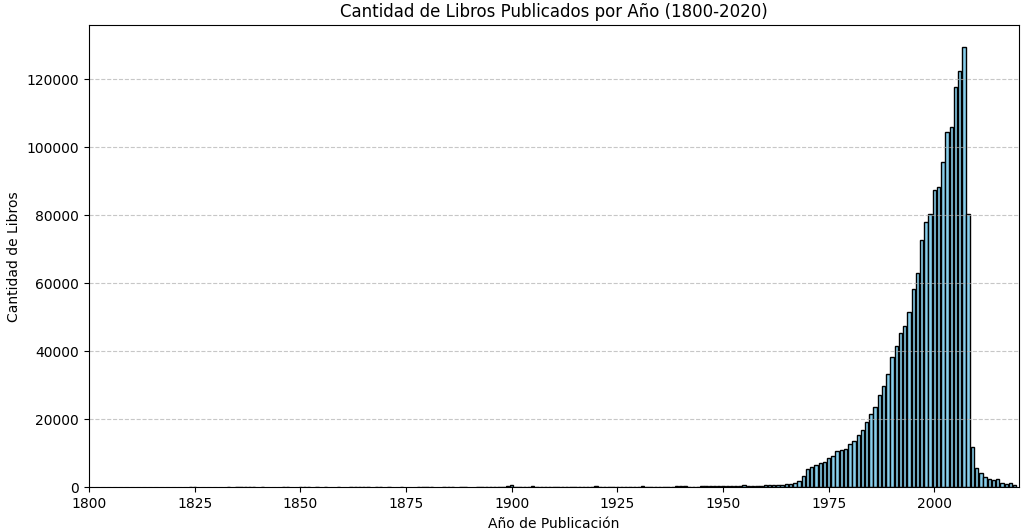
El 36,70% de los libros tienen un valor nulo de Description. Esto no ayudará cuando se tengan que recomendar libros en base a su descripción para los que no tienen descripción.

#### pagesNumber y PagesNumber

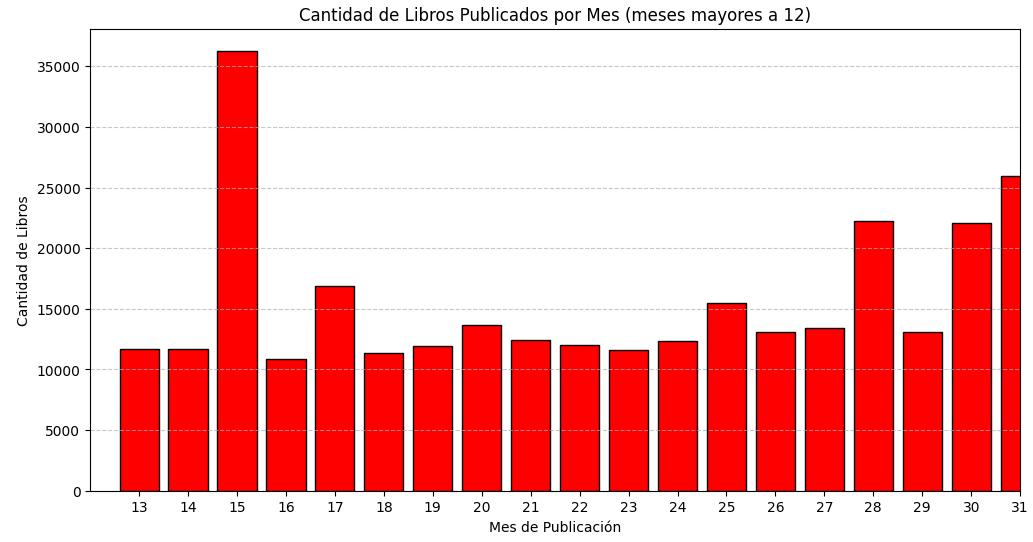
Se observa que hay dos columnas destinadas para el número de páginas. Ambas en formato float64 y cumpliendo que cuando una es nula, la otra tiene valor y viceversa. Lo que se hace es unir ambas columnas para que haya una sola columna con los datos combinados de los dos sin ningún valor nulo.

#### Fecha (Año, Mes y Día)

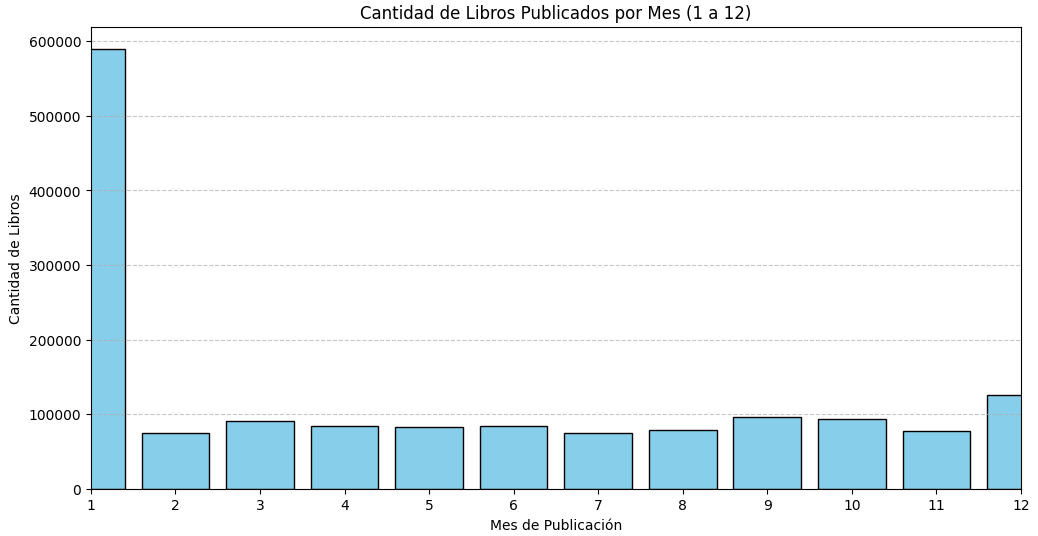
Los valores de año son int64. No existen valores de año nulos, negativos o ceros. Se observan 72 libros publicados un año más adelante que la creación del conjunto de datos, o sea valores erróneos. Solo hay 27 libros publicados antes del 1800. Esta es la distribución anual de los libros.



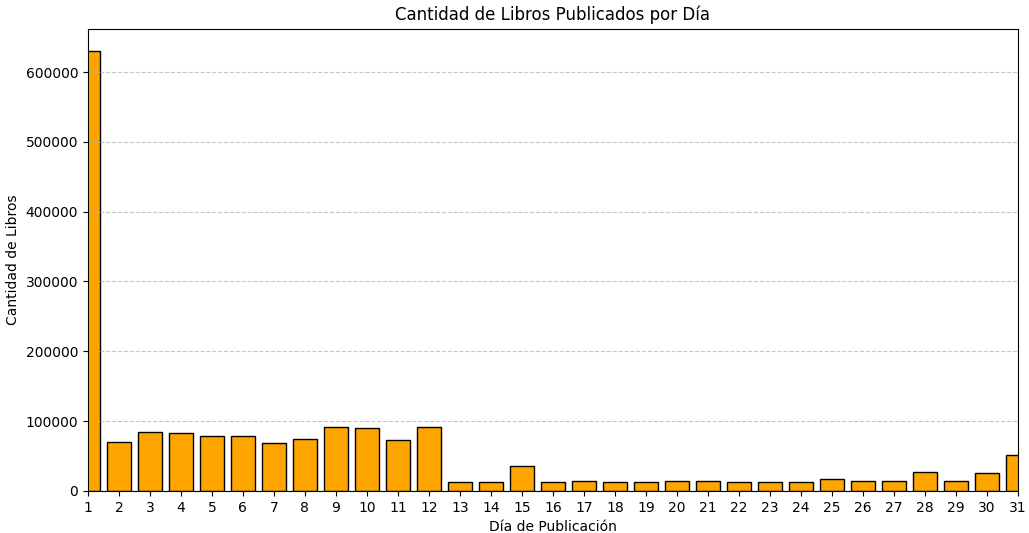
Los valores de mes son int64. No existen valores de mes nulos, negativos o ceros. Pero si que hay 297 991 casos donde el mes es superior a 12. Por lo tanto existen todos estos valores erróneos de mes:



Y la distribución en los meses de 1 a 12 es la siguiente:



Los valores de día son int64. No existen valores de día nulos, negativos, ceros o mayores de 31. Es decir, todos los valores están entre 1 y 31.



La gran cantidad de valores de 1 de día y de 1 de mes parece indicar que los libros en los que se desconocía su día y mes de publicación se asignó por defecto el día 1 y mes 1, lo que serían valores nulos algunos de ellos.

### Datos de valoraciones

A su vez el conjunto de datos cuenta con otros 11 .csv con 1000 valoraciones en cada uno, llegando a un total de 11.000 valoraciones de usuarios.



Las columnas de dichas valoraciones de usuario son:

* ID: El identificador del usuario que valoró.
* Name: Título del libro que se valoró.
* Rating: Selección de 5 posibles valores con los que se calificó: 'it was amazing', 'really liked it', 'liked it', 'it was ok', 'did not like it', "This user doesn't have any rating".

#### ID

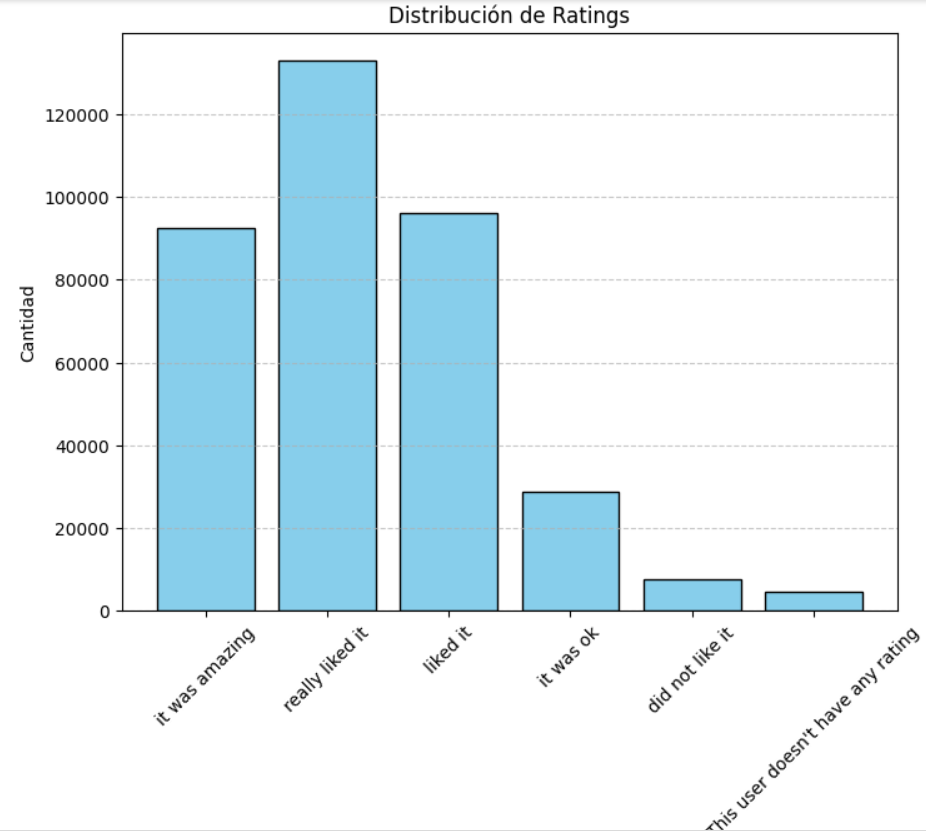
Es un valor entero, sin ningún nulo que va del 1 al 10 999 con 8919 valores únicos.

#### Name

Es un valor objeto, sin ningún nulo que tiene 103 533 valores únicos.

#### Rating

Es un valor objeto, sin ningún NaN pero con 4 765 valores sin calificación ‘This user doesn't have any rating’. Los valores calificados se distribuyen de la siguiente forma:



A partir de la data se podrían que generar las siguientes matrices de salida:

* Matriz de similitud del coseno
* Matriz de similitud del coseno suave
* Matriz de similitud de la distancia euclídea (de Euler)
* Matriz de similitud de la distancia euclídea del punto medio

Clasificación de estos tipos de algoritmos y ver fortalezas y debilidades de estos tipos de algoritmos. Buscar situaciones en que algoritmo va mejor y en qué algoritmo va peor.

# Conclusiones

Contenido:

* Conclusiones del trabajo realizado.
* Puntos fuertes y puntos débiles del trabajo realizado.
* Limitaciones del trabajo realizado.
* Líneas de continuación del trabajo.

# Referencias

Bibliografía con las reseñas bibliográficas completas según la ISO‐690.

1. Referencia al documento 1.
2. Referencia al documento 2.
3. Última referencia añadida.

# Referencias

1. **Casalegno, Francesco.** Recommender Systems — A Complete Guide to Machine Learning Models. *Towards Data Science.* 2022.

2. **J. Bennett, S. Lanning et al.** *The netflix prize.* Nueva York : in Proceedings of KDD cup, 2007. pág. 35. .

3. **A. S. Das, M. Datar, A. Garg, and S. Rajaram.** *Google news personalization:.* ACM : in Proceedings of the 16th international, 2007. págs. 271–280. .

4. **G. Linden, B. Smith, and J. York.** *Amazon.com recommendations: Item-to-item collaborative filtering.* s.l. : IEEE, 2003. págs. 76-80. Vol. 7. 1.

5. **E.-A. Baatarjav, S. Phithakkitnukoon, and R. Dantu.** *Group recommendation system for facebook.* OTM : Springer, 2008. págs. 211-219. .

6. **Abdul Samad, Saif-ur-Rahman.** *Comparative Analysis of Collaborative Filtering on GraphLab, MLlib and Mahout.* [ed.] Institute of Science and Technology (SZABIST). Karachi : Journal of Independent Studies and Research, 2015. Vol. 13.

7. **Alieva O. A., Gangan E. S., Ilyushin E. A., Kachalin A. I.** *Automatic Evaluation of Recommendation Models.* Moscow : s.n., 2020. DOI: 10.25559/SITITO.16.202002.398-406.

8. **T., Hofmann.** *Latent semantic models for collaborative filtering.* ACM Transactions on Information Systems : s.n., 2004. págs. 89-115. DOI: https://doi.org/10.1145/963770.963774.

9. **Guan, Xin.** *On Reducing the Data Sparsity in Collaborative Filtering Recommender Systems.* Coventy : University of Warwick, 2017. págs. 1-40.

10. **D. Goldberg, D. Nichols, B. M. Oki, and D. Terry.** *Using collaborative filtering to weave an information tapestry.* s.l. : Communications of the ACM, 1992. págs. 61-70. Vol. 35. 12.

11. **Girase, S, Mukhopadhyay, D y Bokde, D.** *An Item-Based Collaborative Filtering using Dimensionality Reduction Techniques on Mahout Framework.* s.l. : arXiv preprint arXiv, 2015. 1503.06562.

12. **W. Hill, L. Stead, M. Rosenstein, and G. Furnas.** *Recommending and evaluating choices in a virtual community of use.* in Proceedings of the SIGCHI conference on Human factors in computing systems : ACM Press/AddisonWesley Publishing Co, 1995. págs. 194–201.

13. **Maes, U. Shardanand and P.** *Social information filtering: algorithms for automating word of mouth.* in Proceedings of the SIGCHI conference on Human factors in computing systems : ACM Press/Addison-Wesley Publishing Co, 1995. págs. 210-217.

14. **J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl.** *Grouplens: applying collaborative filtering to usenet news.* s.l. : Communications of the ACM, 1997. págs. 77-87. Vol. 40. 3.

15. **Tuzhilin, G. Adomavicius and A.** *Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions.* s.l. : IEEE transactions on knowledge and data engineering, 2005. págs. 734–749. Vol. 17. 6.

16. **J. S. Breese, D. Heckerman, and C. Kadie.** *Empirical analysis of predictive algorithms for collaborative filtering.* in Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence : Morgan Kaufmann Publishers, 1998. págs. 43-52.

17. **P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl.** *Grouplens: an open architecture for collaborative filtering of netnews.* in Proceedings of ACM conference on Computer supported cooperative work : ACM, 1994. págs. 175–186.

18. **Francesco Ricci, Lior Rokach, Bracha Shapira and Paul B. Kantor.** *Recommender Systems Handbook.* Springer : Springer Science + Business Media LLC, 2011. ISBN: 978-0-387-.

19. **Koren, Yehuda.** *Matrix Factorization Techniques fo rRecommender Systems.* s.l. : Published by the IEEE Computer Society,, August 2009. págs. 42-49. IEEE 0018-9162/09.

20. **J. Friedman, T. Hastie, and R. Tibshirani.** *The elements of statistical learning.* New York : Springer series in statistics, 2001. Vol. 1.

21. **Aggarwal, Rajesh.** *Neighborhood-Based Collaborative Recommendations: An Introduction.* 2020. págs. 91-110. ISBN 978-981-15-3356-3.

# LEYENDA SUBRALLADOS

Información añadible útil

Información añadible discutible

Más información

Información seguramente mal añadida

**ALBERT:**

• Proponer un modelo de recomendación de filtrado por contenido o de filtrado colaborativo o lo que sea que me parezca adecuado para un ecomerce

• Justificar porque ese modelo

• Hacer pasos para implementarlo para que se junte con la parte de Edwin

* + Captura de datos, procesamiento, modelo, propuesta de solución al cold-start (por contenido) + recom. colaborativa, función de python

• Definir qué tests hacer para saber si las recomendaciones son buenas. ✅

• Implementar la evaluación sobre como ejecutar los tests. (Leyendo periódicamente la base de datos)

* Limpiar la base de datos y dejarla lista para el consumo por la api.

Tareas:

1. Convertir el dataset en payloads (JSON) para cargar API:
2. Productos del Catálogo (libros)
3. Categorías x Producto
4. Imágenes x Producto
5. Recomendaciones x Producto
6. Usuarios
7. Recomendaciones x Usuario
8. Definir el Algoritmo de Recomendación

**PARA ACTUALIZAR 100% EL DOCUMENTO:**

* + En el texto citas numéricas (1) -> Actualizar citas i bibliografía
  + En el texto Ilustraciones, Tablas y Ecuaciones -> Actualizar campos
  + En referencias -> Actualizar campos
  + En índice -> Actualizar campos