Herramienta para el análisis de big data aplicado a un sistema de recomendación utilizando MapReduce

Anthony José Vega Mohalem

Director:

Henry Lamos Díaz

Ph.D. En Física – Matemáticas

Codirector:

Daniel Orlando Martínez Quezada

Magister En Ingeniería Industrial

Trabajo De Grado Para Optar Título De Ingeniero Industrial

Universidad Industrial De Santander

Facultad De Ingenierías Físco-Mecánicas

Escuela De Estudios Industriales Y Empresariales

Bucaramanga

2018

DEDICATORIA

A mis padres por darme la oportunidad y apoyo durante estos años para poder realizar mis estudios, por el sacrificio y empeño.

A mi mamá por enseñarme el significado de la que es ser humilde y sencillo con cada persona y por enseñarme lo que en realidad es el respeto al prójimo

A mi papá por ser una persona trabajadora y dedicada a su negocio, me hizo entender lo que es el trabajo duro y empeño en cada cosa propuesta.

CONTENIDO

Pág.

Introducción 15

1 Planteamiento del problema 17

2 Justificación del Proyecto 19

3 Objetivos 21

3.1 Objetivo general 21

3.2 Objetivos específicos 21

4 Revisión de la literatura 22

4.1 Sistemas de recomendación 22

4.2 MapReduce 31

5 Marco de Teórico 32

5.1 Machine Learning. 32

5.2 Big Data 35

5.3 Tipos de datos 39

5.4 MapReduce: 40

5.5 Sistemas de Recomendación. 43

5.6 Filtrado colaborativo: 44

5.6.1 Similitud..... ……..45

5.6.2 Filtrado colaborativo basado en el usuario. 53

5.6.3 Filtrado colaborativo basado en ítems. 55

5.6.4 Basados en contenido. 57

5.6.5 Basados en híbridos. 58

5.6.6 Ejemplo de Filtrado colaborativo 59

5.7 Métodos que utilizan los sistemas de recomendación. 64

5.8 Retos de los sistemas de recomendación. 67

5.9 Bases de datos. 68

5.9.1 MovieLens 69

5.9.2 Wiki Lens 70

5.9.3 ook-Crossing 70

5.9.4 Jester……. 70

5.9.5 EachMovie 71

5.9.6 HetRec 2011 71

6 Sistema de recomendación 72

6.1 Carga y exploración de los datos 72

6.1.1 Valoraciones de los usuarios. 73

6.1.2 Atributos de las películas. 75

6.2 Sistema de recomendación basado en contenido. 76

6.3 Sistema de recomendación basado en usuarios. 78

6.3.1 Similitud entre usuarios 79

6.4 Filtrado colaborativo basado en ítems 81

6.5 Sistema de recomendación utilizando la herramienta MapReduce con filtrado colaborativo. 84

7 Conclusiones 88

8 Recomendaciones 89

Referencias bibliográficas 91

Lista de Figuras

Pág.

Figura 1. Tipos de aprendizaje automático con algoritmos comúnmente adoptados.. 35

Figura 2. Características de las 5 V's de Big Data.. 38

Figura 3. Descripción general de una operación MapReduce.. 43

Figura 4. Aislamiento de los elementos calificados y cálculo de similitud.. 48

Figura 5. Valoración de los usuarios vs las películas. 73

Figura 6. Distribución de las valoraciones. 74

Figura 7. Películas disponibles por año. 75

Figura 8. Temáticas más frecuentes. 76

Figura 9. Representación de las predicciones para el usuario *329*. 77

Figura 10. grafico del top 10 de película predichas. 80

Figura 11 grafico del top 10 de película predichas. 82

Figura 12. Frecuencia de artículo para los 20 ítems principales. 84

Figura 13. Frecuencia de los ítems vs al número de ítems.. 85

Figura 14. Distribución del top 5 de la matriz de confianza y soporte. 86

Figura 15. Distribución del top 5 de la matriz de elevación y convicción. 86

Lista de Tablas

Pág.

Tabla 1. *Cumplimiento de objetivos del proyecto* 14

Tabla 2 *Métodos de combinación en híbridos.* 59

Tabla 3. *Ejemplo de filtrado colaborativo.* 60

Tabla 4. *Cálculo de similitud entre usuarios (ux, u1).* 60

Tabla 5. *Cálculo de similitud entre ítems (i5, i1)* 63

Tabla 6 *Base de datos de HetRec 2011* 71

Tabla 7. *Resumen de las valoraciones ilustradas en la figura 6.* 75

Tabla 8. *Predicción de las películas no vistas para el usuario 329, basado en contenido.* ... 77

Tabla 9. *Resumen de las similitudes entre el usuario 329 y los otros usuarios.* 79

Tabla 10. *Predicción de las películas no vistas por el usuario 329, filtrado colaborativo basado en usuario.* 80

Tabla 11. *Predicción de las películas no vistas por el usuario 329, filtrado colaborativo basado en ítem.* 82

Tabla 12. *Comparación del peso de las predicciones según el filtrado colaborativo.* 83

Tabla 13. *Resumen de los elementos de frecuencia.* 85

Tabla 14. *Precisión en el sistema de recomendación, Filtrado colaborativo, asociación de roles de soporte, de confianza, de elevación y de convicción* 87

Lista de Apéndices

(Ver apéndices adjuntos en el CD y los pueden visualizarlos en la Base de Datos de la Biblioteca UIS)

Pág.

Apéndice A. Ejemplo de Filtrado colaborativo basado en usuario (formato de lenguaje R). …59

Apéndice B. Ejemplo de Filtrado colaborativo basado en ítems (formato de lenguaje R). …. 62

Apéndice C. Filtrado colaborativo basado en contenido (formato de lenguaje R).

Apéndice D. Filtrado colaborativo basado en ítems (formato de lenguaje R).

Apéndice E. Filtrado colaborativo basado en usuario (formato de lenguaje R).

Apéndice F. MapReduce Filtrado colaborativo (formato de lenguaje R). ……...87

Apéndice G. Artículo.

Apéndice H. Música – Artista.

Resumen

TÍTULO DEL PROYECTO: HERRAMIENTA PARA EL ANÁLISIS DE BIG DATA APLICADO A UN SISTEMA DE RECOMENDACIÓN UTILIZANDO MAPREDUCE[*]

AUTOR: ANTHONY JOSÉ VEGA MOHALEM[**]

PALABRAS CLAVE: BIG DATA, FILTRADO COLABORATIVO, MAPREDUCE, SISTEMA DE RECOMENDACIÓN

DESCRIPCIÓN:

El propósito de este proyecto es realizar un sistema de recomendación basado en la herramienta de MapReduce que permita obtener resultados con mayor precisión en base a los contenidos vistos por los usuarios. Los sistemas de recomendación pueden definirse como herramientas diseñadas para interactuar con grandes conjuntos de información y complejos, determinando la facilidad de interacción con el usuario.

A través de los sistemas de recomendación basados en modelos estadísticos, se busca adecuar la información y dar una mejor experiencia al nuevo usuario cuando interactúe con los ítems ya calificados por otros usuarios, los cuales pueden interesar. Para lograr relacionar los ítems con otros, se hace el filtrado colaborativo con la herramienta MapReduce.

Los datos han ido constituyendo los grandes volúmenes de datos y crecen de modo exponencial, tanto así que las bases de datos de organizaciones y empresas han ido creciendo, pasando de volúmenes de datos de Terabytes a Petabytes; sin embargo, los datos de la web son los que tienen mayor porcentaje en lo que hoy en día se le atribuye con el nombre de Big Data, siendo esta la fuente de datos más utilizada y reconocida en la actualidad. Esto lo podemos ver en Amazon, Netflix, eBay, YouTube, entre otros.

---

[*] Trabajo de grado

[**] Facultad de Ingenierías Físico-Mecánicas. Escuela de Estudios Industriales y Empresariales. Director: PhD. Henry Lamos Díaz, Codirector: Msc. Daniel Orlando Martínez Quezada.

Por lo anterior, en este proyecto de grado se implementó un algoritmo bajo la metodología de filtrado colaborativo, utilizando MapReduce, para realizar mejor la precisión en las recomendaciones en la interacción de los usuarios e ítems, productos o artículos.

Abstract

PROJECT TITLE: TOOL FOR BIG DATA ANALYSIS APPLIED TO A RECOMMENDATION SYSTEM USING MAPREDUCE*

AUTOR: ANTHONY JOSÉ VEGA MOHALEM**

KEYWORDS: BIG DATA, COLLABORATIVE FILTERING, MAPREDUCE, RECOMENDATION SYSTEM

DESCRIPTION:

The purpose of this project is to make a recommendation system based on the MapReduce tool that allows to obtain results with greater precision based on the contents seen by the users, the recommendation systems can be defined as tools designed to interact with large information sets and complex, determining the ease of interaction with the user.

Through the recommendation systems based on statistical models, the aim is to adapt the information and give a better experience to the new user when interacting with the items already qualified by other users, which may be of interest. In order to relate the items with others, collaborative filtering is done with the MapReduce tool,

Data has been constituting large volumes of data and growing exponentially, so much so that the databases of organizations and companies have been growing, from data volumes from Terabytes to Petabytes, however, the data on the web are those that have a greater percentage in what is nowadays attributed with the name of Big Data, being this the most used and recognized data source at present. We can see this in Amazon, Netflix, eBay, YouTube, among others.

Due to the above, in this project an algorithm was implemented under the methodology of collaborative filtering using MapReduce, in order to improve the accuracy of the recommendations in the interaction of users and items, products or articles.

---

* Degree work

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Estudios Industriales y Empresariales. Director: PhD. Henry Lamos Díaz, Codirector: Msc. Daniel Orlando Martínez Quezada.

Tabla 1.

Cumplimiento de objetivos del proyecto

Objetivos Específicos	Cumplimiento
• **Realizar una revisión de literatura de Big Data y los sistemas de recomendación.**	Capítulo 4.
• **Identificar las bases de datos del benchmarking para la aplicación en el sistema de recomendación.**	Capítulo 5.
• **Revisar las medidas de desempeño en los sistemas de recomendación con respecto a las bases de datos del benchmarking.**	Capítulo 6.
• **Proponer y evaluar un sistema de recomendación utilizando filtrado colaborativo distribuido.**	Capítulo 6
• **Elaborar un artículo de carácter publicable**	Apéndice G. Artículo

Introducción

La gran revolución de los datos está transformando la forma en que se entienden los procesos económicos o sociales circundantes. Ya no se puede ignorar el enorme volumen de datos que se producen todos los días. El término "grandes datos" se definió como conjuntos de datos en aumento de volumen, velocidad y variedad (Oancea & Dragoescu, 2014).

Hoy en día los datos proceden de numerosas fuentes, desde datos de videojuegos hasta las innumerables cantidades de datos de operaciones en los grandes almacenes, en bancos, la administración pública, sensores, teléfonos inteligentes, entre otros. Todos estos datos han ido constituyendo los grandes volúmenes de datos y crecen de modo exponencial, tanto así que las bases de datos de organizaciones y empresas han ido creciendo, pasando de volúmenes de datos de Terabytes a Petabytes, sin embargo, los datos de la web son los que tienen mayor porcentaje en lo que hoy en día se le atribuye con el nombre de Big Data, siendo esta la fuente de datos más utilizada y reconocida en la actualidad (Aguilar, 2013). En el 2009, la tasa de crecimiento del universo digital alcanzó el 62%, lo que resulta en 1,2 zettabits de datos (¡es decir, 1,2 millones de terabytes!). Se estima que, en 2020, esta cantidad será 44 veces más grande, mientras que el 80% del universo de datos califica como no estructurada(Yejas, Zhuang, & Pannu, 2014).

El análisis de estas cantidades de datos resulta un tanto tediosa y difícil, por tal razón, la tendencia en el avance de la tecnología abre las puertas hacia el entendimiento y toma de decisiones, la cual es utilizada para describir enormes cantidades de datos (estructurados, no estructurados y semiestructurados).

Al igual que la mayoría de las aplicaciones de datos grandes, la gran tendencia de los datos también plantea fuertes impactos en los sistemas de recomendación de servicios. Con el creciente

número de servicios alternativos, la recomendación efectiva de servicios preferidos por los usuarios, se ha convertido en un importante tema de investigación. Los sistemas de recomendación de servicio, se han demostrado como herramientas valiosas para ayudar a los usuarios a soportar la sobrecarga de servicios y proporcionarles recomendaciones apropiadas (Meng, Dou, Zhang, & Chen, 2014).

Las dificultades más habituales vinculadas a la gestión de los grandes volúmenes de datos se centran en la recolección y el almacenamiento, así como la búsqueda, la distribución, el análisis y la visualización. Con el desarrollo del proyecto, se quiere proponer una herramienta de análisis de grandes volúmenes de datos, con la importancia de seguir consolidando la línea de análisis de Big Data, en este caso, los sistemas de recomendación aplicado a grandes volúmenes de datos, es una valiosa herramienta en el cálculo estadístico de numerosas fuentes de donde se generan datos cada día, con el fin que pronostique y recomiende, dejando así una herramienta para el análisis estadístico, utilizando la programación R e incluyendo el marco MapReduce de Hadoop. Donde Hadoop es un framework de software que soporta aplicaciones distribuidas bajo una licencia libre que permite a las aplicaciones trabajar con miles de nodos y petabytes de datos.

El trabajo se encuentra organizado de la siguiente forma: sección 1. Planteamiento del problema sección 2. Justificación de problema, sección 3. Objetivos (Objetivo general y Objetivos específicos), sección 4. Revisión de literatura, sección 5. Marco teórico, sección 6. Sistema de recomendación, sección de conclusiones y recomendaciones.

1 Planteamiento del problema

En la actualidad la sobrecarga de información y el aumento masivo de datos, es conocido como Big Data. Los datos generados rápidamente por Internet, la bioinformática, sensores, entre otras fuentes, conlleva a que la visualización de datos y las grandes cantidades de datos con procesamiento distribuido, se han convertido un tema popular hoy en día.

Para abordar el problema de la sobrecarga de información y procesar conjuntos de datos cada vez más grandes, es importante resaltar el marco de MapReduce que es una técnica que aborda el procesamiento distribuido de datos, encontrando muchas aplicaciones en la nube y entornos de grandes datos, catalogándolo como un medio potencial para hacer frente al problema de rendimiento. MapReduce ha demostrado ser altamente escalable, eficiente y fiable en escenarios de computación de grandes datos, siendo una tecnología importante para apoyar a los analistas de datos hacer frente a la creciente sobrecarga de información y el creciente volumen de datos(Chen et al., 2013).

En la actualidad uno de los problemas es encontrar un producto que se acomode a la necesidad según su criterio de compra de un producto o servicio, todo a partir de datos proporcionados por ciertos clientes, usuarios y/o proveedores, esto para reducir los tiempos de revisión de elementos con el fin de dar una respuesta rápida. Actividades como la recomendación de productos, por lo general son repetitivas y conllevan a una cotidianidad, por lo tanto, los sistemas de recomendación están acogiendo este problema, el cual, proporciona sugerencias de artículos y servicios a través de análisis estadístico de datos, seleccionando automáticamente productos que se adapten a las preferencias e intereses del usuario. La investigación en este campo se centra en mejorar la calidad

de las recomendaciones, adoptando técnicas de aprendizaje automático, procesamiento del lenguaje natural, minería de texto, multimedia, etc. (Manzato et al., 2016).

Páginas web como Amazon, eBay, Netflix, entre otras ofrecen variedad de productos y tienen demasiados clientes, los cuales generan mucha información, haciendo las tareas de recomendación de producto a los clientes no apta para que sea desarrollada por humanos, siendo la labor de recomendación de gran importancia para el éxito comercial de estas compañías. Por lo tanto, es necesario la construcción de máquinas a partir de modelos estadísticos que hagan la tarea de recomendación de forma automática, entregando resultados adecuados de los productos de interés para un cliente especifico.

En los sistemas de recomendación también hay retos que asumir, dentro de estos se encuentran: lidiar con usuarios con solo un producto calificado, ya que conlleva a que la efectividad de las recomendaciones se reduzca, generando malas predicciones; la forma en cómo manejar la creciente información es otro reto que afrontan los sistemas de recomendación. Por lo tanto, el presente proyecto propone el desarrollo de una arquitectura de sistema de recomendación, que permita la sugerencia de productos a un conjunto de clientes, a partir de un enfoque de filtrado colaborativo. Además, con el fin de analizar los grandes volúmenes de datos, se busca implementar en un marco distribuido de MapReduce, verificando la eficiencia del sistema a través comparaciones con bases de datos del benchmarking.

2 Justificación del Proyecto

El término Big Data se ha extendido rápidamente en el marco de Data Mining and Business Intelligence (Minería de Datos e Inteligencia de negocios). Este escenario puede definirse por medio de aquellos problemas que no pueden ser abordados eficaz o eficientemente, utilizando los recursos informáticos estándar que se conocen actualmente. Big Data no sólo implica grandes volúmenes de datos, también la necesidad de escalabilidad; es decir, asegurar una respuesta en un tiempo transcurrido aceptable. La explosión actual de datos que se está generando, se debe a que cientos de aplicaciones tales como sensores móviles, servicios de redes sociales y otros dispositivos relacionados, están recolectando información de forma continua; la capacidad de almacenamiento ha mejorado tanto que la recolección de datos es más barata que nunca, haciendo preferible comprar más espacio de almacenamiento en lugar de decidir qué eliminar; y el aprendizaje automático y los enfoques de recuperación de información han alcanzado una mejora significativa en los últimos años, permitiendo así la adquisición de un mayor grado de conocimiento a partir de los datos (García, Ra-Mírez-Gallego, Luengo, Herrera, & Ramírez-Gallego, 2016).

Tecnologías como Internet generan datos a un ritmo exponencial, gracias al gran desarrollo de almacenamiento y los recursos de red. El volumen actual de datos ha superado las capacidades de procesamiento de los sistemas clásicos de minería de datos (García et al., 2016).

Los avances en la tecnología de Internet han dado lugar a que mucha información esté disponible en línea. Los sistemas de recomendación aplican técnicas estadísticas y de descubrimiento de conocimientos, resolviendo el problema de sobrecarga de información. Su objetivo es identificar las preferencias de los usuarios y filtrar los datos que les son

desfavorables. Como resultado, los sistemas de recomendación pueden ahorrar tiempo a los usuarios en la búsqueda de información (Hahsler, 2011).

3 Objetivos

3.1 Objetivo general

Desarrollar un sistema de recomendación utilizando filtrado colaborativo aplicando la herramienta MapReduce a partir de repositorios del benchmarking.

3.2 Objetivos específicos

- Realizar una revisión de literatura de Big Data y los sistemas de recomendación.
- Identificar las bases de datos del benchmarking para la aplicación en el sistema de recomendación.
- Revisar las medidas de desempeño en los sistemas de recomendación con respecto a las bases de datos del benchmarking.
- Proponer y evaluar un sistema de recomendación utilizando filtrado colaborativo distribuido.
- Elaborar un artículo de carácter publicable.

4 Revisión de la literatura

El análisis de grandes volúmenes de datos (Big Data) es un tema en constante evolución, dado al crecimiento de internet y la sobrecarga de información que esto conlleva, la tarea de seleccionar aquello que realmente se necesita dentro de una cantidad masiva de datos, resulta un tanto difícil y poco precisa. Por tanto, uno de los métodos útiles para esto son los sistemas de recomendación como una herramienta que pronostica y facilita la selección de información de manera rápida. Para ello se apoyan en el aprendizaje automático (Machine Learning) que es una máquina que aprende, donde los modelos pueden ser predictivos o descriptivos para obtener conocimiento, a partir de grandes datos, donde se identifica artículos o productos que el usuario o cliente requiere, y de esa forma entrega resultados que se puedan ajustar a su necesidad.

4.1 Sistemas de recomendación

En un comienzo los sistemas de recomendación eran conocidos tan sólo como filtrados colaborativos, y los primeros trabajos datan a principios de los años 90 (Galán, 2007). El pionero en los sistemas de recomendación fue Tapestry. Este es un sistema de correo experimental desarrollado en el Centro de Investigación Xerox Palo Alto en el año de 1992. La motivación para Tapestry provenía del creciente uso del correo electrónico, lo que provocaba que los usuarios fueran inundados por una enorme corriente de documentos entrantes.

Grouplens es un sistema distribuido para hacer recomendaciones automatizadas, el cual proporcionaba un mecanismo para ayudar a los usuarios a encontrar artículos en la enorme corriente de artículos disponibles, su idea era basada en las personas que estuvieron de acuerdo en

su evaluación subjetiva de artículos pasados, lo cual, era probable que estuviera de acuerdo en el futuro. Después de leer los artículos, los usuarios les asignaban calificaciones numéricas. GroupLens utiliza las clasificaciones de dos maneras. En primer lugar, correlaciona las calificaciones con el fin de determinar qué calificaciones de los usuarios son más similares entre sí. En segundo lugar, predice qué tan bien a los usuarios les gustarán los nuevos artículos, basados en las calificaciones de usuarios similares (Resnick, Iacovou, Suchak, Bergstrom, & Riedl, 1994). A mediados de la década de los 90's, desarrollaron Ringo un algoritmo de filtrado de información social para automatizar el boca a boca en inglés "Word of Mouth". El enfoque común utilizado para abordar el problema de filtrado de información era el basado en contenido y el basado en palabras claves y de indexación semántica latente, el objetivo principal de Ringo era hacer recomendaciones personalizadas de álbumes de música y artistas (Shardanand & Maes, 1995).

Los sistemas de recomendación de filtrado colaborativo utilizan una base de datos sobre las preferencias del usuario, para predecir temas o productos adicionales a los que un nuevo usuario desea. A finales de la década de los 90's, realizaron una comparación empírica de dos algoritmos de filtrado colaborativo: algoritmo basado en memoria y algoritmo basado en modelo. Se concluyó que el algoritmo preferido depende de la naturaleza del conjunto de datos, la naturaleza de la aplicación (presentación clasificada o una por una) y la disponibilidad de votos para hacer predicciones. Otras consideraciones incluyen el tamaño de la base de datos, la velocidad de las predicciones y el tiempo de aprendizaje (Breese, Heckerman, & Kadie, 1998).

En el año 2001 desarrollaron un sistema de recomendación personalizado de productos de supermercados, el sistema se basó en compras a distancias que permitía a los clientes hacer las compras sin recorrer los pasillos del supermercado, en el sistema utilizaban dos fuentes de información, primero aplicaban la minería de asociación a los datos de compra del cliente, para

derivar las relaciones entre las clases de productos y las subclases; en segundo lugar, usamos el agrupamiento para asignar clientes a grupos con intereses similares, basados en patrones de compra previos. Para ser claros, el sistema de recomendación personalizada en productos utiliza, el filtrado basado en contenido, con las ideas del filtrado colaborativo (Lawrence, Almasi, Kotlyar, Viveros, & Duri, 2001).

En el mismo año, se apoyaron en algoritmos de recomendación de filtrado colaborativo basado en artículos (ítems), para abordar restos que mejorara la escalabilidad de los algoritmos de filtrado colaborativo. Estos algoritmos se basan en la búsqueda de miles de vecinos (usuarios) potenciales en tiempo real para realizar la recomendación. Otro desafío es mejorar la calidad de las recomendaciones para los usuarios. Por lo tanto, lo que buscaban con el filtrado colaborativo basado en artículos, era reducir el cuello de botella en la búsqueda de vecinos potenciales y explorar las relaciones entre los primeros artículos. De esta forma, demostraron que el filtrado colaborativo basado en artículos se comportaba mejor que la búsqueda de vecinos potenciales (Sarwar, Karypis, Konstan, & Reidl, 2001).

Por otro lado, en el año 2002 plantearon un sistema de recomendación personalizada basado en la minería de uso de la web y la inducción de árboles de decisión, una estrategia de recomendación personalizada que ayudaba a los clientes a encontrar los productos que le gustaría comprar, mediante la producción de una lista de productos recomendados para cada cliente, siendo un mecanismo habilitador para superar la sobrecarga de información que se producía, cuando se comprara en un mercado de Internet (Cho, Kim, & Kim, 2002).

Un importante acontecimiento dentro de los sistemas de recomendación fue el conocido premio Netflix, en este premio una de las disciplinas predominantes fue el aprendizaje automático, en el que se centraban en algoritmos, donde el foco estaba en la calidad de las predicciones (Taylor,

Bell, Koren, & Volinsky, 2013). En octubre del 2006 Netflix lanzó un concurso para mejorar su sistema de recomendación en un 10% o más, donde el premio era de 1 millón de dólares (Bennett & Lanning, 2007), se creía que era un trabajo de unas cuantas semanas, sin embargo, fue hasta el 2009 que se dio a conocer al ganador de este premio, siendo AT&T quien tuvo la mayor mejora sobre el algoritmo interno de Netflix, llamado Cine match (Netflix, 2009).

Después del premio Netflix para mejorar las predicciones, los trabajos para los sistemas de recomendación aumentaron, y siguieron el curso de mejorar la experiencia del cliente al momento de comprar artículos que se encuentran en internet, basándose en los algoritmos que comúnmente utilizan estos sistemas de recomendación.

En el año 2011 se desarrolló un sistema de recomendación de programas basado en la nube para plataformas de televisión digital, el cual proporcionaba sugerencias de programas de televisión basadas en los resultados estadísticos, obtenidos al analizar datos a gran escala. La frecuencia y la duración de los programas que los usuarios han observado, se recogen y ponderan mediante técnicas de minería de datos. Un conjunto grande de datos produce resultados que representan mejor las preferencias de un espectador de programas de TV en un área específica. Para procesar una cantidad tan grande de datos de preferencia de espectador, debe eliminarse el cuello de botella de escalabilidad y potencia de cálculo. Se introdujo y se aplicó una arquitectura para un sistema de recomendación de programas de TV, basado en la computación en la nube y el marco de MapReduce, el algoritmo de k-means y el algoritmo vecino más cercano. La arquitectura propuesta soportaba la demanda de procesamiento de datos a gran escala para un sistema de recomendación de programas (Lai, Chang, Hu, Huang, & Chao, 2011).

En general, los sistemas de recomendación pueden sugerir información relacionada o los productos que son de interés para los usuarios, y pueden aumentar las relaciones entre los sitios

web comerciales y sus clientes. En consecuencia, se cree que proporcionar a los usuarios un cierto nivel de satisfacción y las recomendaciones de alta calidad, pueden mejorar la lealtad del cliente. Existen dos enfoques comúnmente utilizados en los sistemas, para recomendar artículos relevantes a los usuarios, que son el filtrado basado en contenido y el filtrado colaborativo. En el filtrado basado en contenido, la información del usuario se recoge para crear su perfil, y luego se recomiendan todos los artículos que han comprado. Por lo tanto, el filtrado basado en contenido recomienda artículos a los que tienen adjunto un perfil similar al de los artículos recomendados. Con el fin de recoger las opiniones de los usuarios, la clasificación se utiliza como método universal. La ventaja del filtrado basado en contenido, es que puede recomendar los elementos previamente no clasificados a usuarios con intereses únicos, y proporcionar explicaciones para las recomendaciones. Por otra parte, el filtrado colaborativo es aplicado ampliamente en los sistemas de recomendación, y es la técnica más exitosa de recomendación. A diferencia del filtrado basado en contenido, el filtrado colaborativo se basa en la suposición de que las personas que comparten las mismas preferencias sobre algunos artículos tienden a compartir las mismas preferencias en otros artículos. Por lo general, para cada usuario se encuentra un conjunto de "vecinos más cercanos" cuyas clasificaciones pasadas tienen un alto nivel de correlación. Las puntuaciones para los artículos no vistos, se predicen en la combinación de las puntuaciones conocidas a partir de los vecinos más cercanos. Por lo tanto, la tarea principal de este enfoque es encontrar usuarios con afinidades similares, y se basa en las preferencias de estos “vecinos similares” para proporcionar recomendaciones (Tsai & Hung, 2012).

Aunque existan diferentes aplicaciones de los sistemas de recomendación, al final el método de uso y su procesamiento de las cantidades masivas de datos para recolectar información es similar, basándose en la información del usuario, la información del artículo e información de las

transacciones. Los consumidores hoy en día tienen una cantidad creciente de experiencias con los sistemas de recomendación en línea, como cuando se compran libros de Amazon, ven películas de Netflix o crean círculos de amigos en Facebook. En comparación con los motores de búsqueda, los sistemas de recomendación aplican mecanismos de filtrado de información, que pueden llevar a los usuarios a elementos que no conocen o que no pueden acceder, mediante una búsqueda por palabra clave. Los sistemas de recomendación bien diseñados ahorran tiempo a los usuarios, mejoran la satisfacción del cliente y promueven las ventas. Para captar mejor los intereses de los usuarios y hacer recomendaciones efectivas, es necesario combinar múltiples modelos y hacer uso efectivo de diferentes tipos de datos, como información de usuarios, información de artículos e información de transacciones (transacciones comerciales, actividades de navegación, actividades de revisión, etc.). El paradigma de recomendación de filtrado colaborativo, modela los comportamientos de colaboración de los usuarios que se reflejan en las transacciones y recomienda los productos a los usuarios. Hay generalmente dos tipos de tareas de la recomendación, predicción de la compra contra la predicción de la calificación. La transacción / compra es esencialmente una calificación implícita y aproximada al preferir un artículo. Además, no diferencia los estados de "desconocido" frente a "diferencia". Por lo tanto, las dos tareas son muy diferentes entre sí de acuerdo a su naturaleza computacional. En las aplicaciones de comercio electrónico del mundo real, existe generalmente más información de transacciones, que la información de calificación explícita (Li & Chen, 2013).

En la era del "Internet de las cosas", existe una creciente importancia de los sistemas de recomendación personalizada, estos se han convertido en herramientas para el comercio electrónico con el fin de promover los negocios y ayudar a los clientes a descubrir nuevos productos. SingCF desarrollado en el año 2014, el cual que intenta incorporar calificaciones

singulares (clientes con una única calificación), además de las calificaciones duales (clientes con más de un producto calificado), para implementar el filtrado colaborativo, con el objetivo de mejorar la precisión de la recomendación. En primer lugar, procesaron las puntuaciones no clasificadas de las calificaciones singulares, encontrando las correlaciones y las transformaron en duplas. Luego, realizaron un proceso de filtrado colaborativo, para descubrir a los usuarios del vecindario y hacer predicciones para cada usuario objetivo. Además, proporcionaron un marco distribuido basado en MapReduce de Hadoop, para la mejora significativa en la eficiencia (Xu, Wang, Zheng, & Chen, 2014). Los experimentos en comparación con los métodos del estado de la técnica, demuestran las mejoras de rendimiento de sus enfoques. Para ello las principales contribuciones en su investigación incluyen:

- SingCF, un algoritmo de filtrado colaborativo que incorpora valoraciones singulares para mejorar la precisión de recomendación, donde primero estimaron las puntuaciones no clasificadas de las calificaciones singulares y las transformaron en dos, como datos adicionales para el entrenamiento y la predicción.
- Demostraron que eran equivalentes mostrando que los algoritmos orientados a la clasificación, podrían obtenerse a partir de las técnicas orientadas a la calificación. Sobre la base del mismo marco, han implementado dos versiones de SingCF para la validación, una calificación orientada y un ranking orientado a fines de verificación.
- Proporcionaron DSingCF, un algoritmo SingCF distribuido basado en MapReduce de Hadoop, con el objetivo de mejorar significativamente la eficiencia de SingCF. Los experimentos en comparación con los métodos del estado de la técnica, demostraron las ganancias de rendimiento a sus enfoques.

Por el contrario, ese mismo año, implementaron un algoritmo de recomendación basado en contenido, y cómo se puede paralelizar y distribuir eficientemente a través de muchas máquinas homogéneas en un entorno de memoria distribuida. Al centrarse en los datos en paralelo y construir la definición de trabajo en el contexto de los sistemas de recomendación, son capaces de dividir el proceso de cálculo completo en cualquier número de trabajos independientes y de igual tamaño (Dooms, Audenaert, Fostier, De Pessemier, & Martens, 2014).

En un sistema de recomendación, queremos aprender un modelo de datos de calificación incompletos pasados, de tal manera que la preferencia de cada usuario sobre todos los elementos se puede estimar con el modelo. La factorización de la matriz se demostró empíricamente como un modelo mejor que los enfoques basados en los vecinos más cercanos tradicionales. La factorización de la matriz se da cuando se tienen valores perdidos, convertido en una de las técnicas principales para los sistemas de recomendación. Para manejar conjuntos de datos de web-escala con millones de usuarios y miles de millones de calificaciones, la escalabilidad se convierte en un tema importante. Dos enfoques populares son los mínimos cuadrados alternos (Alternate Least Squares por sus siglas en inglés ALS), y el descenso de gradiente estocástico (Stochastic Gradient Descent por sus siglas en inglés SGD), para calcular la factorización de la matriz, y ha habido una ráfaga de actividad reciente para paralelizar estos algoritmos (Yu, Hsieh, Si, & Dhillon, 2014).

Los sistemas de recomendación siempre se han centrado en el cálculo de un conjunto de recomendaciones para un usuario individual. Sin embargo, hay ciertos escenarios en los que recomendar un conjunto de elementos a un grupo de varios usuarios, es más apropiado que proporcionar varios conjuntos de recomendaciones a cada usuario individual del grupo (por ejemplo, recomendar un destino de vacaciones a una familia o recomendar una película a un grupo de amigos). Los Sistemas de Recomendación de Grupo, tienen como objetivo proporcionar un

conjunto de recomendaciones que satisfagan las preferencias de todos los usuarios de un grupo (Ortega, Hernando, Bobadilla, & Kang, 2016).

En resultado, los sistemas de recomendación sugieren una lista de artículos interesantes, para los usuarios basados en su anterior compra o comportamiento de navegación en las plataformas de comercio electrónico. Los sistemas de recomendación, se ha centrado principalmente en el desarrollo de algoritmos para la tarea de predicción de calificación. Sin embargo, la mayoría de las plataformas de comercio electrónico proporcionan "top-k", lista de artículos de interés para cada usuario. En línea con esta idea, proponen un nuevo algoritmo de aprendizaje automático para predecir una lista de ítems 'top-k', optimizando los factores latentes de usuarios y elementos con las puntuaciones asignadas de las calificaciones. La idea básica es aprender los factores latentes basados en la similitud entre los usuarios y los elementos de las características latentes, que luego se utiliza para predecir las puntuaciones de artículos no vistos para cada usuario. Las evaluaciones empíricas exhaustivas sobre conjuntos de datos de referencia disponibles públicamente, revelan que el modelo propuesto supera a los algoritmos de punta en la recomendación (Kumar, Bala, & Srivastava, 2016).

El servicio de recomendación es una forma alternativa de ayudar a los usuarios a buscar y encontrar artículos que les lleguen a interesar, dado al aumento de los datos en internet los sistemas de recomendación son una herramienta de apoyo tanto para el usuario como para la empresa, mejorando las recomendaciones de artículos, apoyándose en el aprendizaje automático.

Los datos provenientes de la web originan tareas difíciles, como encontrar el interés del usuario para los sistemas de recomendación personalizada y no personalizada. El intercambio de conocimientos entre los usuarios de la web, se ha convertido en crucial para determinar el uso de los datos web y personalizar el contenido en varios sitios web, según el deseo del usuario.

En la actualidad los enfoques de Inteligencia Artificial, están apareciendo a la vanguardia de la investigación en sistemas de recuperación de información y de filtrado de información. Los sistemas de recomendación, son un buen ejemplo de un enfoque de Inteligencia Artificial. Tales sistemas han sido desarrollados para recomendar activamente información relevante a los usuarios, normalmente sin necesidad de una consulta de búsqueda explícita. Han surgido en el dominio del comercio electrónico y son una forma de abordar este problema. Basado en las necesidades de los individuos, los sistemas de recomendación les ayudan a encontrar los elementos adecuados. En el año 2017 se desarrolló un sistema de recomendación multi criterios para el dominio del turismo, utilizando técnicas de aprendizaje automático para la predicción y el conjunto de agrupamiento (clúster) (Nilashi, Bagherifard, Rahmani, & Rafe, 2017).

4.2 MapReduce

Antes del desarrollo de MapReduce, en Google implementaron cientos de cálculos de propósito especial que procesan grandes cantidades de datos en bruto. Para soportar el aumento de los datos y el problema de almacenamiento, implementaron El Sistema de Archivos de Google (en inglés, The Google File System(GFS)), para satisfacer las crecientes demandas de las necesidades de procesamiento de datos en Google. GFS comparte las metas de rendimientos, escalabilidad, fiabilidad y disponibilidad, pero su diseño fue impulsado por las observaciones de las cargas de trabajo de aplicaciones y entorno tecnológico, el cual soporta aplicaciones distribuidas (Ghemawat, Gobioff, & Leung, 2003).

Sin embargo, los datos de entrada eran grandes y los cálculos tenían que ser distribuidos a través de cientos o miles de máquinas, con el fin de terminar en una cantidad razonable de tiempo. Los

problemas de cómo paralelizar el cálculo, distribuir los datos y manejar los fallos, conspiraban para oscurecer el cálculo simple original, con grandes cantidades de código complejo para tratar estos problemas. Como acción a esta complejidad el año 2008 se desarrolló un modelo programación llamado MapReduce, que permite expresar los cálculos simples ocultando los desordenados detalles de la paralelización, tolerancia a fallos, distribución de datos y equilibrio de carga en una nueva biblioteca. Se dieron cuenta que la mayoría de los cálculos implicaba aplicar una operación de *map* a cada registro lógico en sus entradas, para calcular un conjunto de pares intermedios de clave/valor y luego aplicar la operación de *reduce* a todos los valores que compartieron la misma clave, para combinar los datos derivados apropiadamente. MapReduce se desarrolló con unas razones específicas, en primer lugar, el modelo es fácil de usar, incluso para programadores sin experiencia con sistemas paralelos y distribuidos, ocultando los detalles de paralelización, tolerancia a fallos, optimización de la localidad y equilibrio de carga. En segundo lugar, una gran variedad de problemas es fácilmente expresado como cálculos de MapReduce. En tercer lugar, hace uso eficiente de los recursos de la máquina y por tanto, es adecuado para el uso de los grandes problemas computacionales (Dean & Ghemawat, 2008).

5 Marco de Teórico

5.1 Machine Learning.

El aprendizaje automático (Machine Learning), es un subcampo importante en la inteligencia artificial, que se ocupa del diseño y desarrollo de algoritmos para identificar patrones complejos, a partir de datos experimentales, sin asumir una ecuación preestablecida como modelo y tomar

decisiones inteligentemente. Los modelos basados en el aprendizaje automático pueden ser predictivos para realizar predicciones, o descriptivos para obtener conocimiento a partir de datos, o ambos. Aunque el aprendizaje automático surgió a partir de la búsqueda de la inteligencia artificial, su alcance y potencial es mucho más general. Contiene pensamientos de un conjunto variado de disciplinas, incluyendo la Teoría de la Información, Probabilidad y Estadística, Psicología y Neurobiología, Control de complejidad computacional, teoría y filosofía (Taffese & Sistonen, 2017).

Las tareas de aprendizaje automático se clasifican normalmente en tres amplias categorías; estos son: a) el aprendizaje supervisado, en el que el sistema deduce una función de datos de entrenamiento etiquetados, b) aprendizaje no supervisado, en el que el sistema de aprendizaje intenta deducir la estructura de los datos no etiquetados, y c) el aprendizaje reforzado, en el que interactúa el sistema con un entorno dinámico (Kavakiotis et al., 2017).

a) *Aprendizaje supervisado:* a partir de una base de datos de entrenamiento que contiene instancias de entrada y salidas deseadas, su objetivo es construir una función (o modelo), para predecir con precisión la salida objetivo desconocida de instancias futuras. La característica clave del aprendizaje supervisado, es la existencia de un "maestro" y los datos de entrada-salida del entrenamiento. Si el objetivo es predecir variables de destino continuas, la tarea se conoce como regresión. Pero, si el objetivo es predecir variables objetivo discretas, se dice que la tarea es la clasificación.

b) *Aprendizaje no supervisado:* A partir de una base de datos de entrenamiento que involucra instancias de entrada, su objetivo es dividir los ejemplos de entrenamiento en grupos de modo

que los datos de cada clúster demuestren un alto nivel de proximidad. A diferencia del aprendizaje supervisado, las etiquetas de los datos no están disponibles en el aprendizaje sin supervisión.

c) *Aprendizaje reforzado:* El algoritmo se basa en la idea de cómo aprender interactuando con un entorno y adaptando el comportamiento, para maximizar una función objetivo específica a este entorno. El mecanismo de aprendizaje se basa en el ensayo y el error de las acciones y la evaluación de la recompensa. El objetivo es encontrar el comportamiento óptimo: aquel cuyas acciones maximizan el refuerzo a largo plazo. El aprendizaje del refuerzo se utiliza a menudo en agentes inteligentes (Kordon, 2010).

El aprendizaje supervisado y no supervisado, son los tipos más ampliamente estudiados del aprendizaje automático en varios campos de aplicación, incluyendo la ingeniería. Algunos de los algoritmos más influyentes que han sido ampliamente utilizados en los tipos de aprendizaje supervisado y no supervisión, se ilustran en la Figura 1.

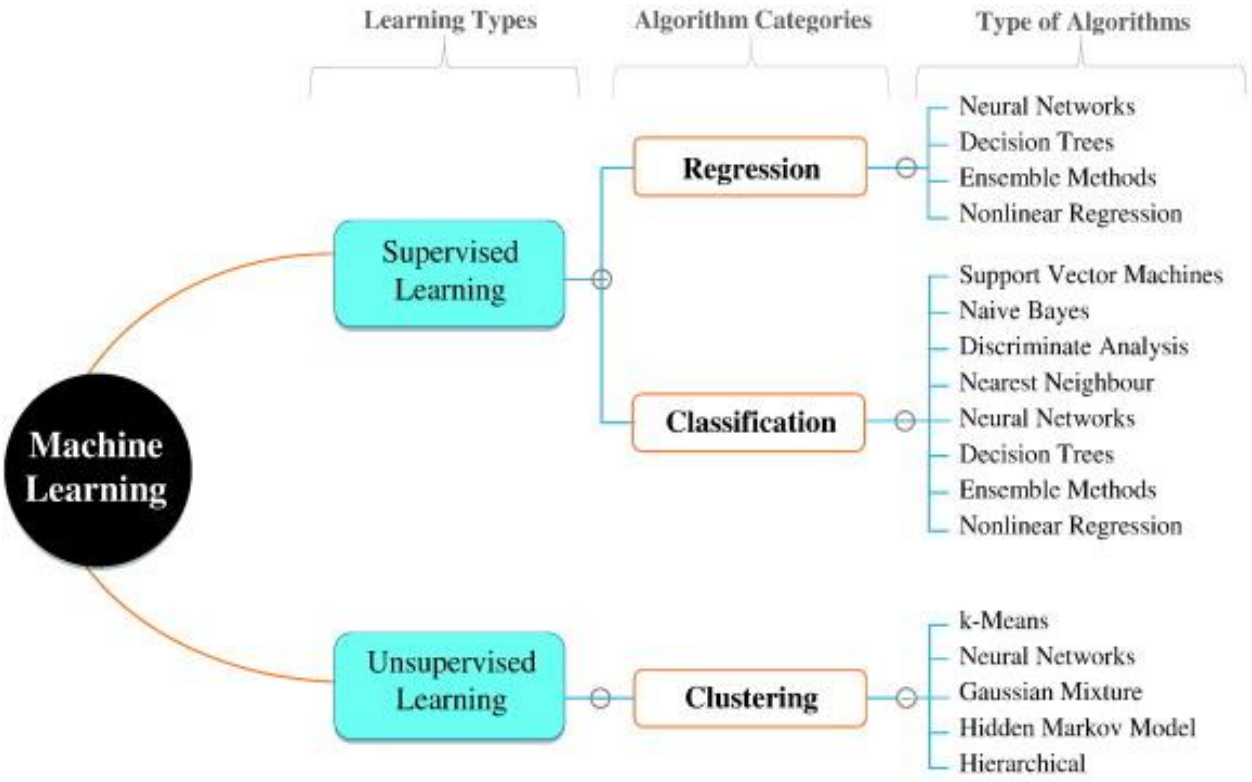

Figura 1. Tipos de aprendizaje automático con algoritmos comúnmente adoptados. Adaptado de Taffese, W. Z., & Sistonen, E. (2017).

5.2 Big Data

Big Data es el tratamiento informatizado de grandes cantidades de información, la definición de lo que es “Big Data” no ha cambiado con el tiempo, puesto que los sistemas informáticos son cada vez más potentes, y cada vez pueden almacenar y procesar más datos de lo que se podía antes. Además, dependen de la capacidad del procesador, para algunos el problema está en procesar cientos de gigabytes, mientras que para otros se trata de peta bytes, cuando se encuentran con problemas.

"Big Data" es un término aplicado a conjuntos de datos que superan la capacidad del software habitual para ser capturados, gestionados y procesados en un tiempo razonable. Los tamaños del "Big Data" se encuentran constantemente en movimiento creciente, de esta forma en 2012 se encontraba dimensionada en un tamaño de una docena de terabytes hasta varios petabytes de datos, en un único conjunto de datos.

Una forma útil de caracterizar las dimensiones de Big Data son las 5V's, volumen, velocidad, variedad, veracidad y valor; y si bien estas dimensiones engloban los principales atributos de Big Data. Esta información requiere nuevas formas de procesamiento para permitir una mejor toma de decisiones, descubrimiento de información y optimización de procesos, La convergencia de estas cinco dimensiones ayuda a definir y a distinguir que es Big Data:

- *Volumen*: la cantidad de datos. Siendo quizá la característica que se asocia con mayor frecuencia a Big Data, el volumen hace referencia a las cantidades masivas de datos que las organizaciones intentan aprovechar, para mejorar la toma de decisiones en toda la empresa. Los volúmenes de datos continúan aumentado a un ritmo sin precedentes. No obstante, lo que constituye un volumen verdaderamente "alto", varía en función del sector e incluso de la ubicación geográfica, y es más pequeño que los petabytes y zetabytes a los que a menudo se hace referencia. Algo más de la mitad de los encuestados consideran que conjuntos de datos de entre un terabyte y un petabytes ya son Big Data, mientras que otro 30% simplemente no sabía cuantificar este parámetro para su empresa. Aun así, todos ellos estaban de acuerdo en que sea lo que fuere que se considere un "volumen alto" hoy en día, mañana lo será más.

- *Variedad:* diferentes tipos y fuentes de datos. La variedad tiene que ver con gestionar la complejidad de múltiples tipos de datos, incluidos los datos estructurados, semiestructurados

y no estructurados. Las organizaciones necesitan integrar y analizar datos de un complejo abanico de fuentes de información, tanto tradicional como no tradicional procedentes tanto de adentro como de afuera de la empresa. Con la profusión de sensores, dispositivos inteligentes y tecnologías de colaboración social, los datos que se generan presentan innumerables formas entre las que se incluyen texto, datos web, tuits, datos de sensores, audio, vídeo, secuencias de clic, archivos de registro y mucho más.

- *Velocidad:* los datos en movimiento. La velocidad a la que crea, procesa y analizan los datos continúa aumentando. Contribuir a una mayor velocidad es la naturaleza en tiempo real de la creación de datos, así como la necesidad de incorporar el flujo continuo de datos a los procesos de negocio y la toma de decisiones. La velocidad afecta a latencia: el tiempo de espera entre el momento en el que se crean los datos, el momento en el que se captan y el momento en el que están accesibles. Hoy en día, los datos se generan de forma continua a una velocidad a la que a los sistemas tradicionales les resulta imposible captarlos, almacenarlos y analizarlos. Para los procesos en los que el tiempo resulta fundamental, tales como la detección de fraude en tiempo real o el marketing “instantáneo” multicanal, ciertos tipos de datos deben analizarse en tiempo real para que resulten útiles para el negocio.

- *Veracidad:* la incertidumbre de los datos. La veracidad hace referencia al nivel de fiabilidad asociado a ciertos tipos de datos. Esforzarse por conseguir unos datos de alta calidad es un requisito importante y un reto fundamental de Big Data, pero incluso los mejores métodos de limpieza de datos no pueden eliminar la imprevisibilidad inherente de algunos datos, como el tiempo, la economía o las futuras decisiones de compra de un cliente. La necesidad de

reconocer y planificar la incertidumbre es una dimensión de Big Data, que surge a medida que los directivos intentan comprender mejor el mundo incierto que les rodea. (ver Figura 2).

- *Valor:* se refiere a la característica importante de los datos que se define por el valor añadido que los datos recopilados pueden aportar al proceso, la actividad o el análisis predictivo / hipótesis previstas. El valor de los datos dependerá de los eventos o procesos que representan, tales como estocásticos, probabilísticos, regulares o aleatorios. Dependiendo de esto, pueden imponerse los requisitos para recopilar todos los datos, y almacenarlos durante un período más largo (para algún posible evento de interés), etc., el valor de los datos está estrechamente relacionado con el volumen y la variedad de los datos (Jasim Hadi, Hameed Shnain, Hadishaheed, & Haji Ahmad, 2014).

En la figura 2 muestra las características de las 5v's provenientes de los grandes volúmenes de datos denominado Big Data.

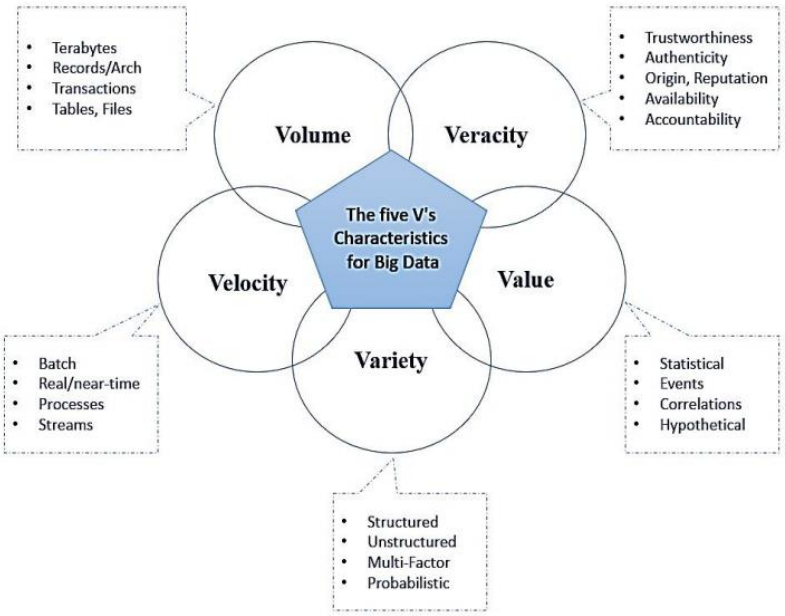

Figura 2. Características de las 5 V's de Big Data. Adaptado de Jasim Hadi, H., Hameed Shnain, A., Hadishaheed, S., & Haji Ahmad, A.

5.3 Tipos de datos

Los Big Data son diferentes de las fuentes de datos tradicionales que almacenan datos estructurados en las bases de datos relacionales. Es frecuente dividir las categorías de datos en dos grandes tipos: *estructurados* (datos tradicionales) y *no estructurados* (datos de Big Data). Sin embargo, las nuevas herramientas de manipulación de Big Data han originado unas nuevas categorías dentro de los tipos de datos no estructurados: *datos semiestructurados* y *datos no estructurados*, propiamente dichos (Aguilar *et. al.*, 2013):

- *Datos estructurados:* La mayoría de las fuentes de datos tradicionales son datos estructurados, datos con formatos o esquemas que poseen campos fijos. En estas fuentes, los datos vienen de un formato bien definido que se especifica en detalle. Los datos estructurados se componen de piezas de información que se conocen previamente, vienen en formatos especificado, y se producen en un orden especificado. Estos formatos facilitan el trabajo con dichos datos, formatos conocidos como: fecha de nacimiento (DD, MM, AAAA). Datos con formato o esquema fijo que poseen campos fijos. Son los datos de las bases de datos relacionales, hojas de cálculo y los archivos, fundamentalmente.
- *Datos semiestructurados:* Estos tipos de datos tienen un flujo lógico y un formato que puede ser definidos, pero no es fácil su comprensión por el usuario. Datos que no tienen formatos fijos, pero contienen etiquetas y otros marcadores que permiten separar los elementos dato. La lectura de datos semiestructurados requiere el uso de reglas complejas que determinan como proceder después de la lectura de cada pieza de información. Un ejemplo de estos datos, son

los registros de *Web logs* de las conexiones a internet; un *Web log* se compone de diferentes piezas de información, cada una de las cuales sirve para un propósito en específico.

- *Datos no estructurados:* Son datos sin tipos predefinidos. Se almacenan como documentos u objetos sin estructura uniforme, y se tiene poco o ningún control sobre ellos. información de textos, video, audio y fotografía son datos no estructurados. Sin duda, los datos más difíciles de dominar por los analistas son los datos no estructurados, con su continuo crecimiento se crean herramientas para la manipulación como es el caso de MapReduce, Hadoop, etc.

5.4 MapReduce:

MapReduce es un modelo de programación y una implementación asociada, para procesar y generar grandes conjuntos de datos que son susceptibles de una amplia variedad de tareas del mundo real. Los usuarios especifican el cálculo en términos de un *map* y una función de *reduce*, y el sistema de tiempo de ejecución subyacente paraleliza automáticamente el cálculo a través de grandes grupos de máquinas, maneja fallas de máquina y programa la comunicación entre máquinas para hacer un uso eficiente de la red y los discos. Los programadores encuentran el sistema fácil de usar: más de diez mil programas MapReduce han sido implementados internamente en Google en los últimos años, y un promedio de cien mil empleos de MapReduce se ejecutan en los clústeres de Google cada día, procesando un total de más de Veinte petabytes de datos por día.

Las invocaciones de mapeo se distribuyen a través de múltiples máquinas por partición automática de los datos de entrada en un conjunto de *M* divisiones. Las divisiones de entrada pueden ser procesadas en paralelo por diferentes máquinas. Las invocaciones de reducción (reduce), se distribuyen dividiendo el espacio de clave intermedio en pedazos *R* usando una

función de partición. El usuario especifica el número de particiones (*R*) y la función de partición. En la Figura 3, muestra el flujo general de una operación MapReduce. Cuando el programa de usuario llama a la función MapReduce, se produce la siguiente secuencia de acciones (las etiquetas numeradas en la Figura 3 corresponden a los números de la siguiente lista) (Dean & Ghemawat, 2008).

1. La biblioteca de MapReduce en el programa de usuario divide primero los archivos de entrada en M pedazos típicamente 16-64MB por pedazo (controlable por el usuario vía un parámetro opcional). A continuación, inicia muchas copias del programa en un grupo de máquinas.
2. Una de las copias del programa -el maestro- es especial. El resto son trabajadores a los que el amo asigna trabajo. Existen M tareas de mapeo y R tareas de reducir que asignar. El maestro selecciona a los trabajadores inactivos y asigna a cada uno una tarea de mapeo o una tarea de reducción.
3. Un trabajador al que se le ha asignado una tarea de mapeo lee el contenido de la división de entrada correspondiente. Analiza pares de clave/valor de los datos de entrada y pasa cada par a la función de mapeo definida por el usuario. Los pares intermedios de clave/valor producidos por la función de mapeo están almacenados en memoria intermedia.
4. Periódicamente, los pares almacenados en memoria intermedia se escriben en disco local, dividido en regiones R por la función de partición. Las ubicaciones de estos pares almacenados en el disco local se devuelven al maestro que es responsable de reenviar estas ubicaciones a los trabajadores de reducción.
5. Cuando un trabajador de reducción es notificado por el maestro acerca de estas ubicaciones, utiliza llamadas de procedimiento remoto para leer los datos almacenados en el búfer de los discos locales de los trabajadores de mapeo. Cuando un trabajador de reducción ha leído todos

los datos intermedios de su partición, lo ordena mediante las teclas intermedias de modo que todas las apariciones de la misma clave se agrupan. La clasificación es necesaria porque normalmente muchas claves diferentes se asignan a la misma tarea de reducción. Si la cantidad de datos intermedios es demasiado grande para caber en la memoria, se utiliza un tipo externo.

6. El trabajador de reducción itera sobre los datos intermedios clasificados y para cada clave intermedia única encontrada, pasa la clave y el conjunto correspondiente de valores intermedios a la función de reducción del usuario. La salida de la función de reducción se anexa a un archivo de salida final para esta partición de reducción.
7. Cuando todas las tareas de mapa y tareas de reducción se han completado, el maestro despierta el programa de usuario. En este punto, la llamada MapReduce en el programa de usuario devuelve al código de usuario.

Las funciones que incluyen las fases de un cálculo de MapReduce son las siguientes:

$map: (k_1, v_1) \rightarrow list((k_2, v_2)$

$reduce: \big(k_2, list(v_2)\big) \rightarrow list(v_2)$

Es decir, las claves de entrada y los valores se extraen de un dominio diferente de las claves de salida y los valores. Además, las claves y valores intermedios son del mismo dominio que las claves y valores de salida.

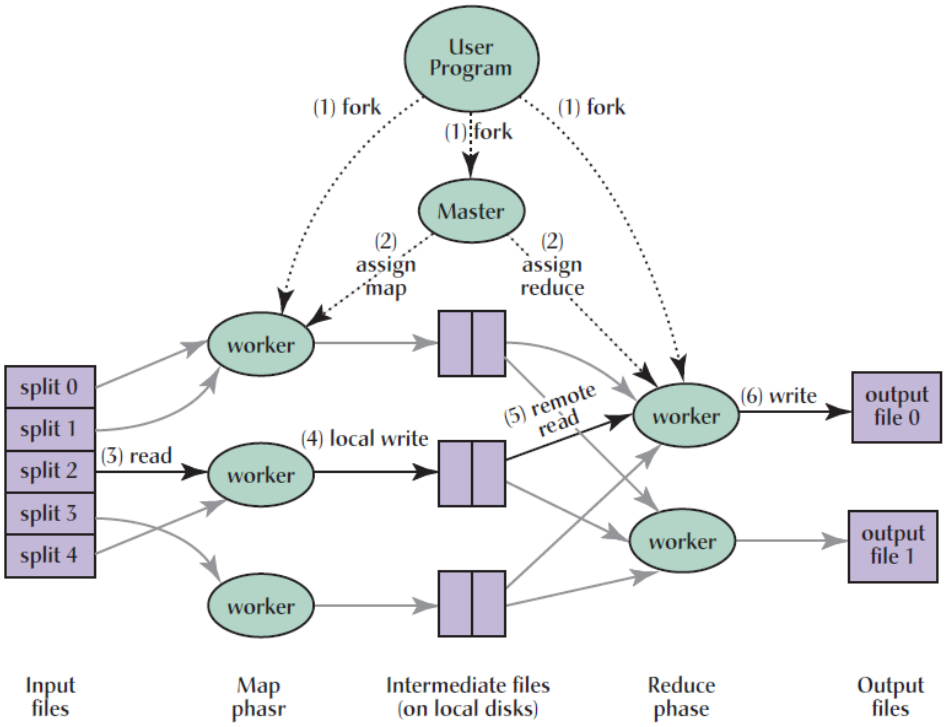

Figura 3. Descripción general de una operación MapReduce. Adoptado de Dean, J., & Ghemawat, S. 2004.

5.5 Sistemas de Recomendación.

Los sistemas de recomendación se desarrollaron como un área de investigación independiente a mediados de la década de 1990, cuando los problemas de recomendación comenzaron a centrarse en los modelos de calificación. Los sistemas de recomendación son un tipo específico de filtro de información individualizada como salida, o tiene el efecto de guiar al usuario de manera personalizada a servicios interesantes o útiles en un amplio espacio de opciones posibles (Meng et al., 2014).

Debido a la gran variedad de ámbitos aplicables a la tarea de recomendación, existe un gran espectro de sistemas, cada uno tratando de optimizar dicha tarea en base a las particularidades que presenta. Como es obvio, la recomendación de películas y la de restaurantes puede afrontarse de

muy diferentes modos. El tipo de información disponible sobre usuarios, objetos y la relación entre ambos, determina el tipo de sistemas de recomendación aplicables a un dominio. En general, se distinguen las siguientes familias (Ráez, 2014):

5.5.1 Filtrado colaborativo: Utilizan los datos de interacción de usuarios con los ítems, para encontrar patrones y filtrar los ítems interesantes para cada usuario. El modelo puede ser construido únicamente a partir del comportamiento de un solo usuario o más efectivamente, también del comportamiento de otros usuarios que tienen rasgos similares. En esencia, las recomendaciones se basan en una colaboración automática de múltiples usuarios, y se filtran en aquellos que exhiben similares preferencias o comportamientos.

Por ejemplo, tiene un sitio web para recomendar blogs. Al utilizar la información de muchos usuarios que se suscriben y leen blogs, puede agrupar a esos usuarios según sus preferencias. Por ejemplo, puede agrupar usuarios que lean varios de los mismos blogs. A partir de esta información, identifica los blogs más populares que son leídos por ese grupo. Entonces, para un usuario en particular en el grupo, se recomienda el blog más popular que él o ella no lee ni se suscribe.

El filtrado colaborativo trabaja en dos fases: (I) el descubrimiento de usuarios del vecindario y (II) la predicción para la recomendación. Para cada usuario, la Fase I descubre un conjunto de usuarios más similares, como los usuarios del vecindario, y luego, la Fase II predice los puntajes de calificación o las preferencias en los ítems para propósitos de recomendación (Xu et al., 2014).

La mayoría de los algoritmos de filtrado colaborativo, se basan en el concepto de similitud. Algunos algoritmos calculan la similitud entre los usuarios, otros miran la similitud entre los elementos, otros la similitud entre las categorías de los elementos. Antes de que podamos entender cómo funcionan los algoritmos de filtrado colaborativo, se necesita entender esta similitud.

5.5.2 Similitud. Todos los sistemas de recomendación tienen una cosa en común, para poder llevar a cabo las predicciones, necesitan definir y cuantificar la similitud entre ítems o usuarios. El término distancia, se emplea dentro del contexto de sistemas de recomendación, como cuantificación de la similitud o diferencia entre observaciones. La similitud se define mediante el análisis de datos en término de una función de distancia, es decir, cuanto más distantes son los objetos, menos similares se vuelven, como la distancia Euclidiana (1) y la distancia de Manhattan (2).

$$d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + \cdots + (x_{in} - x_{jn})^2} \qquad (1)$$

$$d(i,j) = |x_{i1} - x_{j1}| + \cdots + |x_{in} - x_{jn}| \qquad (2)$$

En esas funciones de distancia, se toma la diferencia entre los valores correspondientes de los atributos, en las tuplas de i y j. Típicamente los atributos se normalizan para que los atributos con valores grandes no superen a los atributos con valores más pequeños. La distancia euclidiana y Manhattan trivialmente funcionan bien, y pueden ayudarnos a calcular la similitud para las tuplas que tienen atributos con valores numéricos. Sin embargo, si el atributo es categórico, como el color, se necesita métodos para diferenciar la clasificación (por ejemplo, color azul vs negro). Algunos de estos métodos son similitudes basadas en cosenos, similitud basada en probabilidad condicional y similitud de correlación de Pearson (Almazro et al., 2010).

5.5.2.1 Similitud basada en el coseno: En este enfoque, los ítems se consideran como vectores en el espacio de usuario m-dimensional donde la dimensión es el atributo por el cual el elemento está clasificado. El coseno del ángulo entre los vectores que representan dos elementos es su similitud. Se sabe por cálculo que la fórmula del producto punto es:

$\vec{\imath} \cdot \vec{\jmath} = \left|\left|\vec{\imath}\right|\right| \cdot \left|\left|\vec{\jmath}\right|\right| \cdot \cos\theta$

$\Longrightarrow sim(i,j) = \cos\theta = \frac{\vec{\imath}\cdot\vec{\jmath}}{||\vec{\imath}||\cdot||\vec{\jmath}||}$ (3),

5.5.2.2 Similitud condicional basada en la probabilidad: Una alternativa de calcular la similitud entre cada par de ítems (i y j) es usar una medida que se basa en la probabilidad condicional de comprar uno de los ítems, dado que el otro ya ha sido comprado. En particular, la probabilidad condicional de comprar j, dado que i ya se ha comprado P(j|i), no es más que el número de clientes que compran los ítems i y j divididos por el número total de clientes que compraron i, es decir,

$$P(j|i) = \frac{Freq(ij)}{Freq(i)},$$

Donde *Freq(X)* es el número de clientes que han comprado los artículos en el conjunto X. Obsérvese que, en general, $P(j|i) \neq P(i|j)$ y usando esto como una medida de similitud conduce a relaciones asimétricas.

Como se mencionó anteriormente, una de las limitaciones de usar una función de similitud asimétrica, es que cada ítem tenderá a tener probabilidades condicionales altas con ítems que se compran con frecuencia. Este problema ha sido reconocido en los sistemas de recuperación de información y recomendación. El problema se puede corregir dividiendo *P(j|i),* con una cantidad que depende de la frecuencia de ocurrencia del ítem *j.* Hay dos métodos diferentes propuestos. El primero, inspirada en la escala de frecuencia de documentos inversos, multiplica *P(j|i)* por –

$log_2(P(j))$. En este último método conduce a una función similitud simétrica. Se ha demostrado que este escalamiento afecta en gran medida al experimento del sistema de recomendación, y que el grado de escalabilidad depende del problema; por tal razón, se utiliza la siguiente fórmula para calcular la similitud entre dos elementos.

$$sim(i,j) = \frac{Freq(ij)}{Freq(i) \times (Freq(j))^{\alpha}} \quad (4),$$

donde α es un parámetro que toma un valor entre 0 y 1. Tenga en cuenta que, cuando $\alpha = 0$, La ecuación (4) se convierte en idéntica a *P(j|i)*, mientras que si $\alpha = 1$, se vuelve similar a la formulación en la que *P(j|i)* está dividido por *P(j)*. La función de similitud definida en la ecuación (4), no discrimina entre los clientes que han comprado el número diferente de artículos. Para lograr esta discriminación y dar mayor peso a los clientes que han comprado menos artículos, hemos ampliado la medida de similitud de la ecuación (4), de la siguiente manera: Primero, normalizamos cada fila de la matriz R para que sea de longitud unitaria, y luego, definimos la similitud entre los ítems *i* y *j* como:

$$sim(i,j) = \frac{\sum_{\forall\, q:R_{q,j>0}} R_{q,j}}{Freq(i) \times (Freq(j))^{\alpha}} \quad (5).$$

La única diferencia entre la ecuación (5) y la ecuación (4), es que, en lugar de utilizar la frecuencia de ocurrencia, usamos la suma de las entradas no nulas correspondientes de la j-ésima columna en la matriz de objetos del usuario. Dado que las filas se normalizan para ser de longitud unitaria, los clientes que ha comprado más artículos tenderán a contribuir menos a la similitud general. Por lo tanto, esto da énfasis a las decisiones de compra de los clientes que han comprado menos artículos (Deshpande & Karypis, 2004).

5.5.2.3 Similitud de Correlación de Pearson: La similitud viene dada por la cantidad de correlación entre los ítems o usuarios. La correlación se calcula con la fórmula de Pearson. Si el conjunto de usuarios que han calificado i y j están denotados por U, entonces la similitud de correlación viene dada por:

$$sim(i,j) = corr_{ij} = \frac{\sum_{u \in U}(R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R}_i)^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R}_j)^2}} \quad (6).$$

Donde, $R_{u,i}$ denota la calificación del usuario *u* en el ítem *i*, $\bar{R}_i$ es la calificación media del i-ésimo ítem. En la figura 4, se muestra la semejanza de ítem-ítem y se calcula mirando sólo los elementos calificados. En el caso de los ítems *i* y *j*, la semejanza $S_{i,j}$ se calcula mirándolos. Cada uno de estos pares calificados, se obtienen de diferentes usuarios, en este ejemplo proceden de los usuarios 1, *u y m*-1(Almazro et al., 2010).

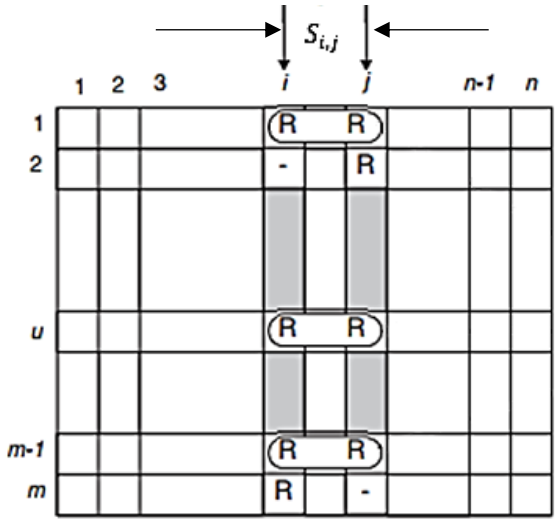

Figura 4. Aislamiento de los elementos calificados y cálculo de similitud. Modificado de: Sarwar, Badrul, Karypis George, Konstan Joseph, Reidl John, (2001).

5.5.2.4 Correlación Jackknife: El coeficiente de correlación de Pearson resulta efectivo en ámbitos muy diversos. Sin embargo, tiene la desventaja de no ser robusto frente a valores atípicos, a pesar de que se cumpla la condición de normalidad. Si dos variables tienen un pico o un valle común en una única observación, por ejemplo, por un error de lectura, la correlación va a estar dominada por este registro, a pesar de que entre las dos variables no haya correlación real alguna. Lo mismo puede ocurrir en la dirección opuesta. Si dos variables están altamente correlacionadas excepto para una observación, en la que los valores son muy dispares, entonces la correlación existente quedará enmascarada. Una forma de evitarlo es recurrir a la correlación Jackknife, que consiste en calcular todos los posibles coeficientes de correlación entre dos variables, si se excluye cada vez una de las observaciones. El promedio de todas las correlaciones Jackknife calculadas, atenúa en cierta medida el efecto de los valores atípicos.

$$\bar{\theta}(A,B) = Promedio\ Jackknife\ correlation\ (A,B) = \frac{1}{n}\sum_{i=1}^{n} \widehat{r_i}$$

Donde *n* es el número de observaciones y $\widehat{r_i}$ es el coeficiente de correlación entre las variables *A* y *B*, habiendo excluido la observación *i*.

Además del promedio, se puede estimar su error estándar *(SE)* y así obtener intervalos de confianza para la correlación Jackknife y su correspondiente $p - value$.

$$SE = \sqrt{\frac{n-1}{n}\sum_{i=1}^{n}(\widehat{r_i} - \bar{\theta})^2}$$

Intervalo de confianza del 95% (*Z=1.96*)

$$Promedio\ Jackknife\ correlation\ (A,B) \pm\ 1.96 * SE$$

$$\bar{\theta} \pm 1{,}96 \sqrt{\frac{n-1}{n}\sum_{i=1}^{n}(\hat{r_i} - \bar{\theta})^2}$$

$P - value$ para la hipótesis nula de que $\bar{\theta} = 0$

$$Zcalculada = \frac{\bar{\theta} - H_0}{SE} = \frac{\bar{\theta} - H_0}{\sqrt{\frac{n-1}{n}\sum_{i=1}^{n}(\hat{r_i} - \bar{\theta})^2}}$$

$$Pvalue = P(Z > Zcalculada)$$

Cuando se emplea este método, es conveniente calcular la diferencia entre el valor de correlación obtenido por la correlación Jackknife ($\bar{\theta}$), y el que se obtiene si se emplean todas las observaciones ($\bar{r}$). A esta diferencia se le conoce como Bias. Su magnitud es un indicativo de cuanto está influenciada la estimación de la correlación entre dos variables, debido a un valor atípico.

$$Bias = (n-1) * (\bar{\theta} - \hat{r})$$

Si se calcula la diferencia entre cada correlación ($\hat{r_i}$) estimada en el proceso de Jackknife, y el valor de correlación ($\hat{r}$) obtenido si se emplean todas las observaciones, se puede identificar que observaciones son más influyentes.

Cuando el estudio requiere minimizar al máximo la presencia de falsos positivos, a pesar de que se incremente la de falsos negativos, se puede seleccionar como valor de correlación el menor de entre todos los calculados en el proceso de Jackknife.

$$Correlacion = min\{\hat{r_1}, \hat{r_2}, \dots, \hat{r_n}\}$$

A pesar de que el método de Jackknife permite aumentar la robustez de la correlación de Pearson, si los valores atípicos son muy extremos su influencia seguirá siendo notable.

5.5.2.5 Simple matching coefficient: Cuando las variables con las que se pretende determinar la similitud entre observaciones son de tipo binario, a pesar de que es posible codificarlas de forma numérica como 1 o 0, no tiene sentido aplicar operaciones aritméticas sobre ellas (media, suma…). Por ejemplo, si la variable sexo se codifica como 1 para mujer y 0 para hombre, carece de significado decir que la media de la variable sexo en un determinado set de datos es 0.5. En situaciones como esta, no se pueden emplear medidas de similitud basadas en distancia euclídea, manhattan o en la correlación.

Dado dos objetos *A* y *B*, cada uno con n atributos binarios, *el simple matching coefficient (SMC)* define la similitud entre ellos como:

$$SMC = \frac{número\ coincidencias}{número\ total\ de\ atributos} = \frac{M_{00} + M_{11}}{M_{00} + M_{01} + M_{10} + M_{11}}$$

Donde M_{00} y M_{11} son el número de variables para las que ambas observaciones tienen el mismo valor (ambas 0 o ambas 1), y M_{01} y M_{10} el número de variables que no coinciden. El valor de distancia simple matching distance (*SMD*) se corresponde con $1 - SMC$.

5.5.2.6 Índice Jaccard: El índice Jaccard o coeficiente de correlación Jaccard es similar al simple matching coefficient (SMC). La diferencia radica en que el SMC tiene el término $\boldsymbol{M_{00}}$ en el numerador y denominador, mientras que el índice de Jaccard no. Esto significa que, SMC, considera como coincidencias tanto si el atributo está presente en ambos sets, como si el atributo no está en ninguno de los sets, mientras que Jaccard solo cuenta como coincidencias cuando el atributo está presente en ambos sets.

$$J = \frac{M_{11}}{M_{01} + M_{10} + M_{11}}$$

o en términos matemáticos de sets:

$$J(A, B) = \frac{A \cap B}{A \cup B}$$

La distancia de Jaccard $(1 - J)$ supera a *la simple matching distance* en aquellas situaciones en las que la coincidencia de ausencia no aporta información. Para ilustrar este hecho, supóngase que se quiere cuantificar la similitud entre dos clientes de un supermercado en base a los artículos comprados. Es de esperar que cada cliente solo adquiera unos pocos artículos de los muchos disponibles, por lo que el número de artículos no comprados por ninguno (M_{00}) será muy alto. Como la distancia de Jaccard ignora las coincidencias de tipo M_{00}, el grado de similitud dependerá únicamente de las coincidencias entre los artículos comprados.

5.5.3 Filtrado colaborativo basado en el usuario. Filtrado colaborativo basado en el usuario, es un algoritmo basado en la memoria que trata de imitar la palabra de la boca, mediante el análisis de los datos de calificación de muchas personas. La suposición es que los usuarios con preferencias similares clasificarán los artículos de manera similar. Así, las calificaciones que faltan para un usuario, pueden predecirse encontrando primero un vecindario de usuarios similares y luego agregando las calificaciones de estos usuarios, para formar una predicción. El vecindario se define en términos de similitud entre usuarios, ya sea tomando un número dado de usuarios más similares (k vecinos más cercanos), o todos los usuarios dentro de un umbral de similitud dado. Se puede usar dos enfoques para calcular la similitud entre los usuarios; explícita e implícitamente.

Antes de describir los algoritmos se introducen las siguientes definiciones para facilitar el proceso de explicación:

- Un conjunto de *m* usuarios $U = \{u_x: x = 1, 2, \ldots, m\}$;
- Un conjunto de *n* ítems $I = \{i_x: x = 1, 2, \ldots, n\}$;
- Un conjunto de *p* categorías $C = \{c_x: x = 1, 2, \ldots, p\}$;
- Un conjunto de *q* calificaciones explícitas $R = \{r_x: x = 1, 2, \ldots, q \wedge q \leq m*n\}$;
- Un conjunto de *t* calificaciones implícitas $R = \{r'_x: x = 1, 2, \ldots, t \wedge t \leq m*p\}$;
- La calificación explícita de un usuario u_x con referencia a un ítem i_x como r_{u_x,i_h};
- La calificación explícita promedio de un usuario u_x como $\overline{r_{u_x}}$.

5.5.3.1 Algoritmo explícito de calificaciones basado en usuarios. En este caso, los usuarios expresan sus calificaciones en los artículos. Si el conjunto de elementos que los usuarios u_x y u_y han calificado se define como I'= $\{i_x: x=1, 2, \ldots, n' \wedge n' \leq n\}$, donde n es el número total de elementos en la base de datos, entonces la similitud entre dos usuarios se define como el coeficiente de correlación de Pearson de sus filas asociadas en la matriz de usuario y es dada por (Papagelis & Plexousakis, 2005):

$$k_{x,y} = sim(u_x, u_y) = \frac{\sum_{h=1}^{n'}\left(r_{u_x,i_h} - \overline{r_{u_x}}\right)\left(r_{u_y,i_h} - \overline{r_{u_y}}\right)}{\sqrt{\sum_{h=1}^{n'}\left(r_{u_x,i_h} - \overline{r_{u_x}}\right)^2}\sqrt{\sum_{h=1}^{m'}\left(r_{u_y,i_h} - \overline{r_{u_y}}\right)^2}}$$

5.5.3.2 Algoritmo implícito de calificaciones basado en usuarios. La clasificación implícita no significa que un usuario no muestre su aprecio hacia un artículo, simplemente significa que no lo hace directa o explícitamente como con el enfoque anterior. La calificación de cada elemento se captura de forma implícita. Por ejemplo, si un usuario pasa más tiempo buscando en un elemento, que recibe una calificación alta o también obtendrá una calificación alta si un usuario repetidamente viene a buscarlo (Almazro et al., 2010).

Sea $C = \{c_x: x = 1, 2, \ldots, p\}$, donde p es el número total de categorías de la base de datos que los usuarios u_x y u_y han calificado. La similitud entre u_x y u_y está dada por:

$$\lambda_{x,y} = sim(u_x, u_y) = \frac{\sum_{h=1}^{n'}\left(r_{u_x,c_h} - \overline{r_{u_x}}\right)\left(r_{u_y,c_h} - \overline{r_{u_y}}\right)}{\sqrt{\sum_{h=1}^{n'}\left(r_{u_x,c_h} - \overline{r_{u_x}}\right)^2}\sqrt{\sum_{h=1}^{m'}\left(r_{u_y,c_h} - \overline{r_{u_y}}\right)^2}},$$

donde esta preferencia $r'_{u_x,c_x} \in R'$ se considera como calificación implícita y esa categoría se calcula como $r'_{u_x,c_x} = \left({c_{x_{pos}}}\Big/{c_{x_{pos}} + c_{x_{neg}}}\right) * 10$, donde $c_{x_{pos}}, c_{x_{neg}}$ son, respectivamente, el

número de calificaciones positivas y negativas que el usuario u_x ha dado implícitamente a la categoría x. Después de que los usuarios se han agrupado, los algoritmos persiguen por encontrar artículos populares entre esos usuarios y recomendarlos (Papagelis & Plexousakis, 2005).

5.5.4 Filtrado colaborativo basado en ítems. Filtrado colaborativo basado en ítems es un enfoque apoyado en modelos, que genera recomendaciones en relación con los ítems deducidos de la matriz de calificación. La suposición detrás de este enfoque, es que los usuarios prefieren artículos que son similares a otros artículos que les gusta (Hahsler, 2011). El análisis de la información histórica puede hacerse de forma explícita, observando las evaluaciones explícitas de los usuarios realizadas sobre los ítems, o implícitamente, a través de la información de navegación del usuario o la clasificación de las categorías de elementos.

Los algoritmos basados en ítems son algoritmos de dos pasos; en el primer paso, los algoritmos exploran las informaciones pasadas de los usuarios, las calificaciones que dieron a los elementos se recogen en este paso. A partir de estas calificaciones, la similitud entre ítems se construye y se insertan en una matriz *M* de ítem a ítem. El elemento $x_{i,j}$ de la matriz *M*, representa la similitud entre el ítem de la fila *i* y el ítem de la columna *j*. después, en el paso final, los algoritmos seleccionan los ítems que son más similares al ítem particular que un usuario está calificando. La similitud en el filtrado colaborativo basado en ítems, también se puede calcular explícita o implícitamente.

5.5.4.1 Algoritmo explícito de calificaciones basado en usuarios. Como se ha dicho antes, este enfoque requiere que los usuarios evalúen específicamente (expresen su opinión) los ítems. Sea U' = {u_x: x = 1, 2, . . ., m' ∧ m' ≤ m}, donde m es el número total de usuarios en la base de datos, el conjunto de usuarios que han calificado tanto el ítem i como el ítem j, el coeficiente de correlación de Pearson de sus columnas asociadas en la matriz usuario-ítem siguiendo la fórmula:

$$sim(i,j) = \frac{\sum_{h=1}^{m\prime}\left(R_{u_h,i} - \overline{R_i}\right)\left(R_{u_h,j} - \overline{R_j}\right)}{\sqrt{\sum_{h=1}^{m\prime}(R_{u_h,i} - \overline{R_i})^2}\sqrt{\sum_{h=1}^{m\prime}(R_{u_h,j} - \overline{R_j})^2}}$$

$R_{u_h,i}$ es la calificación explícita dada por un usuario u_h a un ítem *i*, y R_i es la media de las calificaciones dadas en el ítem *i*.

5.5.4.2 Algoritmo explícito de calificaciones basado en ítems. Al igual que con el algoritmo implícito basado en el usuario, las calificaciones asignadas a los ítems pueden captarse implícitamente. Se calcula la similitud entre dos ítems, como el coeficiente de correlación de Pearson de sus filas asociadas en la matriz de bitmap de categoría de ítems, y está dada por:

$$v_{x,y} = sim\left(i_x, j_y\right) = \frac{\sum_{h=1}^{p}\left(v_{c_h,i_x} - \overline{v_{i_x}}\right)\left(v_{c_h,j_x} - \overline{v_{j_x}}\right)}{\sqrt{\sum_{h=1}^{p}(v_{u_h,i_x} - \overline{v_{i_x}})^2}\sqrt{\sum_{h=1}^{p}(v_{u_h,j_x} - \overline{v_{j_x}})^2}},$$

donde *p* es el número de categorías y v_{c_h,i_x} es un valor booleano* que equivale a 1 si el ítem *x* pertenece a la categoría *h* o es igual a 0 de lo contrario.

* Booleano se aplica a todo símbolo que se usa para establecer relaciones entre términos matemáticos o variables admitiendo solo dos respuestas posibles.

5.5.5 Basados en contenido. Tratan de analizar el contenido de los ítems para realizar las recomendaciones, tratando de acercar ítems con un contenido más relevante basados en los gustos del usuario. Este enfoque normalmente puede utilizar información histórica de navegación, como los blogs que el usuario lee y las características de esos blogs. Por ejemplo, si un usuario suele leer artículos sobre tecnología, aunque sea probable que deje comentarios en blogs sobre ingeniería de software. El filtrado basado en contenido puede usar este historial para identificar y recomendar contenido similar (artículos sobre tecnologías u otros blogs sobre ingeniería de software). Este contenido puede definirse manualmente o extraerse automáticamente, basándose en otros métodos de similitud. Es decir, la utilidad u (c, s) del ítem s para el usuario c se estima basándose en las utilidades de u (c, s_i) asignadas por el usuario c a los elementos $s_i \in S$ que son similares a los ítems s (Adomavicius & Tuzhilin, 2005).

Más formalmente, deje que el contenido sea un perfil del ítem, es decir, un conjunto de atributos que caracterizan el ítem s. Normalmente se calcula extrayendo un conjunto de características del ítem s (su contenido), y se utiliza para determinar la conveniencia del elemento con fines de recomendación. Los sistemas basados en contenido están diseñados principalmente para recomendar elementos basados en texto, el contenido de estos sistemas se describe normalmente con palabras clave, es decir, la "importancia" de la palabra k_j en el documento d_j se determina con alguna medida de ponderación w_{ij} , que puede definirse de varias maneras diferentes.

Una de las medidas más conocidas para especificar pesos de palabras clave en recuperación de información, es el término frecuencia / frecuencia inversa de documento (TF-IDF), medida que se define como sigue: Supongamos que N es el número total de documentos que se pueden recomendar a los usuarios y esa palabra clave kj aparece en ni de ellos. Además, supongamos que fi, j es el número de veces que la palabra clave ki aparece en el documento dj. Entonces, TFi, j, el

término frecuencia (o frecuencia normalizada) de la palabra clave *ki* en el documento dj, se define como:

$$TF_{i,j} = \frac{f_{i,j}}{max_z f_{z,j}},$$

Donde el máximo se calcula sobre las frecuencias $f_{z,j}$ de todas las palabras clave k_z que aparecen en el documento d_j. Sin embargo, las palabras clave que aparecen en muchos documentos no son útiles para distinguir entre un documento relevante y otro no pertinente. Por lo tanto, la medida de la frecuencia del documento inverso (IDF_i), se utiliza a menudo en combinación con la frecuencia de término simple ($TF_{i,j}$). La frecuencia del documento inverso para la palabra clave k_i suele definirse como:

$$IDF_i = \log \frac{N}{n_i}.$$

Entonces, el peso TF-IDF para la palabra clave *ki* en el documento dj se define como:

$$w_{i,j} = TF_{i,j} \times IDF_i.$$

5.5.6 Basados en híbridos. Los sistemas de recomendación híbridos combinan dos o más técnicas de recomendación para obtener un mejor rendimiento con menos de los inconvenientes de cualquier individuo. Más comúnmente, el filtrado colaborativo se combina con alguna otra técnica en un intento de evitar el problema de aumento. En la tabla 2 se muestra algunos métodos de combinación que se han empleado (Tran & Cohen, 2000).

Tabla 2

Métodos de combinación en híbridos.

Métodos de híbridos	Descripción
Ponderado	Las puntuaciones (o votos) de varias técnicas de recomendación se combinan para producir una sola recomendación.
Conmutación	El sistema conmuta entre las técnicas de recomendación dependiendo de la situación actual.
Mixto	Las recomendaciones de varios sistemas diferentes se presentan al mismo tiempo.
Combinación de funciones	Las características de las fuentes de datos de recomendación diferentes, se combinan en un solo algoritmo de recomendación.
Cascada	Un sistema refina las recomendaciones dadas por otro.
Aumento de la característica	La salida de una técnica se utiliza como característica de entrada a otra.
Meta-nivel	El modelo aprendido por un sistema de recomendación se utiliza como entrada a otro

Nota: Adaptado de "Hybrid Recommender Systems: Survey and Experiments". Tran & Cohen, 2000.

5.5.7 Ejemplo de Filtrado colaborativo

5.5.7.1 Filtrado colaborativo basado en usuario. Supóngase que se dispone del historial de valoraciones que 4 usuarios (u1, u2, ..., u4) han hecho sobre 5 ítems (i1, i2, ..., i5). Un nuevo usuario (ux) no ha valorado el ítem (i5). Se pretende aplicar un sistema de recomendación colaborativo, para predecir la valoración del usuario (ux) sobre el ítem i5.

Tabla 3.

Ejemplo de filtrado colaborativo.

	i1	***i2***	***i3***	***i4***	***i5***
ux	5	3	4	4	NA
u1	3	1	2	3	3
u2	4	3	4	3	5
u3	3	3	1	5	4
u4	1	5	5	2	1

Para este caso se calcula la similitud entre el usuario ***ux*** con los demás usuarios, de tal manera que se sepa cual usuario se asemeja más, para poder dar la calificación del ítem ***i5***.

En este caso, el cálculo de la similitud entre usuarios, se utiliza la ecuación de similitud de correlación de Pearson.

Tabla 4.

Cálculo de similitud entre usuarios ***(ux, u1).***

	R	***R***				
	ux	***u1***	$R_{ux,i} - \bar{R}_{ux}$	$R_{u1,\mathbf{i}} - \bar{R}_{u1}$	$(R_{ux,i} - \bar{R}_{ux})^{\mathbf{2}}$	$(R_{u1,\mathbf{i}} - \bar{R}_{\boldsymbol{u1}})^{\mathbf{2}}$
i1	5	3	1	0,75	1	0,5625
i2	3	1	-1	-1,25	1	1,5625
i3	4	2	0	-0,25	0	0,0625
i4	4	3	0	0,75	0	0,5625

$\boldsymbol{\bar{R}_{ux} = 4}$

$\boldsymbol{\bar{R}_{u1} = 2,25}$

Se calcula la similitud entre el usuario ***ux*** y ***u1***.

$$sim(u_x, u_1) = \frac{(1)(0{,}75) + (-1)(-1{,}25)}{\sqrt{(1) + (1)} * \sqrt{(0{,}5625) + (1{,}5625) + (0{,}0625) + (0{,}5625)}}$$

$$sim(u_x, u_1) = \frac{2}{\frac{\sqrt{22}}{2}} = \frac{4}{\sqrt{22}}$$

Se hace el mismo proceso para los otros usuarios y se calcula la similitud entre ellos.

- $sim(u_x, u_1) = 0{,}85280287$
- $sim(u_x, u_2) = 0{,}70710678$
- $sim(u_x, u_3) = 0$
- $sim(u_x, u_2) = -0{,}79211803$

Obtenidos los valores de las similitudes y ordenados de mayor a menor similitud respecto al usuario ***ux***, se procede a calcular la predicción de la valoración que posiblemente pueda asignar al ítem ***i5***. El cálculo se puede hacer mediante el promedio de las valoraciones de los *n* usuarios más cercanos, y calculando el ponderado de la media con los valores de similitud.

En el promedio de las valoraciones de los ***n*** usuarios más cercanos, se evita tener en cuenta la valoración de usuarios que tienen un perfil diferente al de interés; ***n*** se debe considerar como un parámetro cuyo valor óptimo se identifica, por ejemplo, mediante validación cruzada. En este caso el valor para *n* = 3. Donde los 3 usuarios más similares a ***ux*** son: ***u1, u2, u3***. La predicción de la valoración que hace el usuario ***ux*** sobre el ítem ***i5***, se obtiene como la media de las valoraciones que cada uno de los usuarios seleccionados tiene sobre el ítem 5 (*Rusuario*, item5).

$$Predicción(u_x\ , i_5) = \frac{Ru1, i5 + Ru2, i5 + Ru3, i5}{n = 3}$$

$$Predicción(u_x, i_5) = \frac{3 + 5 + 4}{3} = 3$$

El inconveniente de la aproximación anterior es que, los ***n*** usuarios seleccionados tienen el mismo peso en la predicción, sin embargo, no todos se parecen en la misma medida al usuario de interés. Para ello se calcula el ponderado de la media con los valores de similitud. Solo puede aplicarse cuando la similitud toma valores en el rango [0, número positivo], ya que, la media aritmética ponderada, no está definida para pesos negativos y, al menos uno de los pesos, debe ser mayor de cero.

$$Predicción(u_x, i_5) = \frac{sim(u_x, u_1)(Ru_1, i_5) + sim(u_x, u_2)(Ru_2, i_5) + sim(u_x, u_3)(Ru_3, i_5)}{sim(u_x.u_1) + sim(u_x.u_2) + sim(u_x.u_3)}$$

$$Predicción(u_x, i_5) = \frac{(0.85 * 3)\ + (0.71 * 5)\ + (\ 0 * 4)}{(0.85\ +\ 0.71\ +\ 0)} = 3.910256$$

5.5.7.2 Filtrado colaborativo basado en ítem. La idea es muy similar al método basado en usuarios, pero en este caso, se identifican ítems similares (empleando el perfil de valoraciones que han recibido) en lugar de usuarios similares. Además, los ítems que participan en el proceso tienen que haber sido valorados por el usuario de interés ux.

En primer lugar, se calcula la similitud entre el ítem ***i5*** y los demás ítems. Ejemplo de filtrado colaborativo, se corresponde con similitud entre columnas.

Tabla 5.

Cálculo de similitud entre ítems (i5, i1)

	i5	***i1***	$\boldsymbol{R_{i5,u} - \bar{R}_{i5}}$	$\boldsymbol{R_{i1,u} - \bar{R}_{i1}}$	$\boldsymbol{(R_{i5,u} - \bar{R}_{i5})^2}$	$\boldsymbol{(R_{i1,u} - \bar{R}_{i1})^2}$
ux		5				
u1	3	3	-0,25	0,25	0,0625	0,0625
u2	5	4	1,75	1,25	3,0625	1,5625
u3	4	3	0,75	0,25	0,5625	0,0625
u4	1	1	-2,25	-1,75	5,0625	3,0625

$\boldsymbol{\bar{R}_{i5}} = 3{,}25$

$\boldsymbol{\bar{R}_{i1}} = 2{,}75$

$$sim(i_5, i_1) = \frac{(-0{,}25)(0{,}25) + (1{,}75)(1{,}25) + (0{,}75)(0{,}25) + (-2{,}25)(-1{,}25)}{\sqrt{(0{,}0625) + (3{,}0625) + (0{,}5625) + (5{,}0625)} * \sqrt{(0{,}0625) + (1{,}5625) + (0{,}0625) + (3{,}0625)}}$$

$$sim(i_5, i_1) = \frac{6{,}25}{\frac{\sqrt{665}}{4}} = \frac{25}{\sqrt{665}}$$

Se calcula de igual manera para los otros ítems y se ordena de mayor a menor la similitud entre ellos, se seleccionan los ***n*** ítems más parecidos entre ellos y, se obtienen la predicción a partir de las valoraciones que el usuario ***ux*** ha hecho de esos ***n*** ítems.

- $sim(i_5, i_1) = 0{,}969458418$
- $sim(i_5, i_4) = 0{,}581675051$
- $sim(i_5, i_3) = -0{,}427617987$
- $sim(i_5, i_2) = -0{,}478091444$

- Cálculo de la predicción de la valoración para el ítem ***i5*** del usuario ***ux***. Predicción basada en el promedio de los ***n*** = 3 ítems más precedidos.

$$Predicción(u_x\ , i_5) = \frac{Ru_x, i_1 + Ru_x, i_4 + Ru_x, i_3}{n = 3}$$

$$Predicción(u_x\ , i_5) = \frac{3 + 1 + 4}{3}$$

$$Predicción(u_x\ , i_5) = 2{,}666666667.$$

- Predicción basada en el promedio ponderado por similitud, los valores de similitud negativos se sustituyen por 0 para poder calcular:

$$Predicción(u_x, i_5) = \frac{sim(i_5, i_1)(Ru_x, i_1) + sim(i_5, i_4)(Ru_x, i_4) + sim(i_5, i_3)(Ru_x, i_3)}{sim(i_5, i_1) + sim(i_5, i_4) + sim(i_5, i_3)}$$

$$Predicción(u_x, i_5) = \frac{(0{,}969458418)(3) + (0{,}581675051)(1) + (0)(4)}{(0{,}969458418) + (0{,}581675051) + (0)}$$

$$Predicción(u_x, i_5) = 2{,}25.$$

5.6 Métodos que utilizan los sistemas de recomendación.

Muchos de los algoritmos provienen del campo del aprendizaje automático (Machine Learning), un sub-campo de inteligencia artificial que produce algoritmos para el aprendizaje, la predicción y la toma de decisiones (Jones, 2013).

- *Algoritmos de Agrupamiento:* Son una forma de aprendizaje no supervisado que se pueden encontrar en la estructura de un conjunto de datos aparentemente al azar (o no marcado). En general, funcionan mediante la identificación de similitudes entre los elementos, tales como

los lectores del blog, mediante el cálculo de su distancia de otros objetos en un espacio de características (las características en un espacio de características podrían representar el número de artículos leídos en un conjunto de blogs). El número de características independientes define la dimensión del espacio. Si los artículos se "cierran" juntos, pueden estar unidos en un clúster.

Existen muchos algoritmos de agrupamiento. El más sencillo es *k* -means, que divide los elementos en *k* grupos. Inicialmente, los elementos se colocan al azar en grupos. Entonces, un *centroide* (o *centro*), se calcula para cada grupo en función de sus miembros. A continuación, se comprueba la distancia de cada elemento de los centroides. Si se encuentra un elemento para estar más cerca de otro grupo, se traslada a ese grupo. Los centroides se recalculan cada vez que se comprueban todas las distancias de los artículos. Cuando se alcanza la estabilidad (es decir, cuando no se mueven elementos durante una iteración), el conjunto se agrupa correctamente y el algoritmo termina.

El cálculo de la distancia entre dos objetos puede ser difícil de visualizar. Un método común es tratar a cada elemento como un vector multidimensional y calcular la métrica de distancia Euclídea. Otras variantes incluyen la familia de agrupamiento adaptativo Teoría de Resonancia (ART), Fuzzy, k-means, y la expectativa de maximización (agrupación probabilística), para nombrar unos pocos.

- *Otros Algoritmos:* Existen muchos algoritmos y un conjunto aún mayor de variaciones de esos algoritmos para motores de recomendación:

  - ✓ *Las redes Bayesianas de creencias*, que se puede visualizar como un gráfico a cíclico dirigido, con arcos que representan las probabilidades asociadas entre las variables.

- ✓ *Las cadenas de Markov,* que tienen un enfoque similar a las redes de creencias bayesianas, pero tratan el problema de la recomendación como la optimización secuencial en lugar de simplemente la predicción.
- ✓ *Clasificación Rocchio,* Desarrollado con el modelo de espacio vectorial), que explota la retroalimentación de la relevancia, elemento para mejorar la precisión de recomendación.

Los sistemas de recomendación se basan en diferentes tipos de datos de entrada, que a menudo se colocan en una matriz con una dimensión que representa a los usuarios, y la otra dimensión que representa los elementos de interés. Algunas de las realizaciones de mayor éxito de los modelos de factores latentes, están basados en la factorización de la matriz (Koren, Bell, & Volinsky, 2009). Para manejar conjuntos de datos de escala web con millones de usuarios y miles de millones de calificaciones, la escalabilidad se convierte en un tema importante. Alternar los mínimos cuadrados y el gradiente descendente estocástico, son dos enfoques populares para la factorización de la matriz computacional (Yu, Hsieh, Si, & Dhillon, 2014b).

- *Matriz Factorización*: Se utiliza básicamente para factorizar una matriz, es decir, para determinar dos (o más) matrices, de manera que, la multiplicación es igual a la matriz original. Puede ser utilizado para descubrir rasgos latentes que subyacen a las interrelaciones entre los usuarios y los artículos, con el fin de predecir calificaciones desconocidas de las clasificaciones conocidas en las calificaciones de la matriz. Por lo tanto, si se pudieran descubrir esas características latentes, las calificaciones desconocidos podrían predecirse con respecto a un determinado usuario y un cierto elemento, debido a que las características asociadas con el usuario deben coincidir con las características asociadas con el artículo (Christensen & Schiaffino, 2013).

5.7 Retos de los sistemas de recomendación.

Los sistemas de recomendación también hay retos que se deben tomar en cuenta para manejar la información lo mejor posible y dar óptimos resultados. Dentro de los principales retos que se pueden encontrar son (Pilares et al., 2015):

- *Escasez de datos:* Cuando llega un usuario o producto nuevo al sistema de recomendación, estos no cuentan con información previa para poder obtener y realizar la recomendación, presentándose así el problema de escasez de datos. En este caso, la tarea de encontrar sus similares, se vuelve más complicada, ya que, un nuevo producto no puede ser recomendado hasta que un usuario lo haya calificado y, a nuevos usuarios no se les darán buenas recomendaciones por la falta de calificaciones en su historial de compras. Esto puede reducir la efectividad de los sistemas de recomendación y, por lo tanto, generar malas predicciones.
- *Escalabilidad:* La escalabilidad dentro de un sistema de recomendación se refiere a la forma en la cual crece la información dentro de éste. Cuando la información tanto de usuarios como de productos crece rápidamente, decimos que se presenta la escalabilidad.
- *Sinónimos:* En ocasiones se encuentran sinónimos dentro de los identificadores de un producto y por tal motivo, algunos pueden no ser tomados en cuenta para la recomendación. Por ejemplo, se pueden tener una película que dentro de su descripción tenga películas para niños y otra muy similar que tenga película infantil, sin embargo, si no se tienen considerados los sinónimos, no se encontrarán dentro del mismo grupo, aunque tenga características similares.

- *Oveja gris:* En muchas ocasiones los usuarios no ayudan a la realización de las recomendaciones ya que no están de acuerdo con algún grupo de personas, es decir, el perfil del usuario pertenece a diferentes grupos de usuarios y en muchas ocasiones grupos opuestos. Cuando esto sucede, se dice que el usuario es una oveja gris. Este tipo de usuarios no ayuda a dar buenas recomendaciones y es difícil determinar para ellos una recomendación adecuada.
- *Diversidad vs Precisión:* Cuando la tarea es recomendar productos que sean apreciados para un usuario en particular, es más sencillo recomendar productos populares o con mayor calificación, sin embargo, esta recomendación no siempre es útil para el usuario, ya que, las opciones más populares son más fáciles de encontrar, incluso difíciles de evitar sin necesidad de utilizar un sistema de recomendación. Una lista de buenas recomendaciones debe contener productos que no sean fáciles de localizar para los usuarios y que le sean de utilidad, tratando así el reto de diversidad vs precisión.
- *El valor del tiempo:* Es importante que, al realizar una recomendación, esta se dé en el menor tiempo posible, encontrando así el reto el valor del tiempo. Entre mayor sea la cantidad de datos que se tengan, mayor es la dificultad de tratar este reto.

5.8 Bases de datos.

Las bases de datos que se encuentran disponibles para realizar análisis de datos gratuitamente, estas bases datos presentadas tienen características particulares. Las características comparadas corresponden a las cantidades de productos, la cantidad de usuarios y tipo de base de datos. A continuación, se presentan las bases de datos que se encuentran en el benchmarking (“GroupLens,” n.d.).

5.8.1 MovieLens. GroupLens Research ha recopilado y puesto a disposición, conjuntos de datos de calificación del sitio web MovieLens (http://movielens.org). Los conjuntos de datos se recopilaron durante varios períodos de tiempo, dependiendo del tamaño del conjunto.

5.8.1.1 MovieLens 20M Dataset. Conjunto de datos de referencia estable, 20 millones de calificaciones y 465,000 aplicaciones de etiquetas, aplicadas a 27,000 películas por 138,000 usuarios. Incluye datos del genoma de etiqueta con 12 millones de puntajes de relevancia en 1.100 etiquetas. Lanzado el 4/2015; actualizado 10/2016 para actualizar links.csv y agregar datos de genoma de etiqueta.

5.8.1.2 MovieLens Latest Datasets. Estos conjuntos de datos cambiarán con el tiempo y no son apropiados para informar resultados de investigación. 100.000 clasificaciones y 1.300 aplicaciones de etiquetas, aplicadas a 9.000 películas por 700 usuarios. Última actualización 10/2016, por tal razón, estos datos no son confiables, ya que mantienen un cambio en el tiempo.

5.8.1.3 MovieLens 100k Datasets. Conjunto de datos de referencia estable. 100.000 calificaciones de 1.000 usuarios en 1.700 películas.

5.8.1.4 MovieLens 1M Dataset. Conjunto de datos de referencia estable. 1 millón de calificaciones de 6.000 usuarios en 4.000 películas.

5.8.1.5 ***MovieLens 10M Dataset.*** Conjunto de datos de referencia estable. 10 millones de calificaciones y 100.000 aplicaciones de etiquetas aplicadas a 10,000 películas por 72,000 usuarios.

5.8.1.6 ***MovieLens Tag Genome Dataset.*** 11 millones de puntajes de relevancia de película de un conjunto de 1.100 etiquetas aplicadas a 10.000 películas.

5.8.2 Wiki Lens: Era un sistema de recomendación colaborativa generalizada que permitía a su comunidad definir tipos de elementos (por ejemplo, cerveza) y categorías (por ejemplo, micro cervezas, cervezas pálidas, stouts), y luego calificaba y obtenía recomendaciones para los artículos. Fue desconectado en 2009 debido a la falta de mantenimiento y soporte del sistema.

5.8.3 Book-Crossing: El conjunto de datos Book-Crossing (BX), fue recopilado por Cai-Nicolas Ziegler en un rastreo de 4 semanas (agosto / septiembre de 2004) de la comunidad Book-Crossing. Contiene 278.858 usuarios (anónimos, pero con información demográfica), que proporcionan 1.149.780 calificaciones (explícitas / implícitas) de 271.379 libros.

5.8.4 Jester: Este conjunto de datos contiene 4.1 millones de calificaciones continuas (-10.00 a +10.00), de 100 bromas de 73,496 usuarios.

5.8.5 EachMovie HP/Compaq Research (anteriormente DEC Research) ejecutó la recomendación de películas EachMovie. Cuando se cerró EachMovie, el conjunto de datos estaba disponible para el público para su uso en la investigación. MovieLens se basó originalmente en este conjunto de datos. Contiene 2,811,983 calificaciones ingresadas por 72,916 usuarios, para 1628 películas diferentes, y se ha utilizado en numerosas publicaciones de Filtrado Colaborativo. A partir de octubre de 2004, HP retiró el conjunto de datos EachMovie.

5.8.6 HetRec 2011. El II Taller Internacional sobre Heterogeneidad y Fusión de la Información en los Sistemas de Recomendación (HetRec, 2011) ha publicado conjuntos de datos de Delicious, Last.fm Web 2.0, MovieLens, IMDb y Rotten Tomatoes. Estos conjuntos de datos contienen información de redes sociales, etiquetado y consumo de recursos (marcadores de página web y escucha de artistas musicales) de conjuntos de alrededor de 2.000 usuarios.

Los conjuntos de datos fueron generados por el Grupo de Recuperación de Información de la Universidad Autónoma de Madrid.

Tabla 6

Base de datos de HetRec 2011

Nombre BD HetRec 2011	**Contenido**
Delicious Bookmarks	105,000 marcadores de 1867 usuarios.
Last FM	92.800 registros de artistas que escuchan de 1892 usuarios.
MovieLens + IMDb / Rotten Tomatoes	86,000 calificaciones de 2113 usuarios.

Nota: Adaptado de (“GroupLens,” n.d).

6 Sistema de recomendación

Con el fin de tener resultados que se ajusten a una medida de recomendación adecuada para el usuario y así permita una mejor experiencia a la hora de buscar un producto, que se ajuste a la necesidad en la que está deseando, se hacen pruebas con filtrado colaborativo de MovieLense.

El conjunto de datos MovieLense del paquete recommenderlab, contiene información sobre más de 1000 películas, tanto variables descriptivas de cada largometraje como las valoraciones de más de 900 usuarios. Empleando este conjunto de datos, se generan 3 tipos de sistemas de recomendación con el objetivo de recomendar 10 nuevas películas a un determinado usuario.

6.1 Carga y exploración de los datos

Las valoraciones de los usuarios se encuentran almacenadas en un objeto de tipo realRatingMatrix llamado MovieLense, y la descripción de las películas en un dataframe (marco de datos) llamado MovieLenseMeta, esto en el software de programación R. Para facilitar su manejo, se almacenan ambos conjuntos de datos en formato de dataframe y se renombran como valoraciones y atributos. Las valoraciones son las calificaciones que los usuarios han hecho a determinado ítem y los atributos son las características que tienen los ítems, como lo son el título, el año de publicación y el género al que pertenece, se le puede llamar datos descriptivos de los ítems.

6.1.1 Valoraciones de los usuarios. En la figura 5 se aprecia las valoraciones realizadas por los usuarios sobre diferentes películas, se observa también que hay espacios en blanco, lo que quiere decir es que, hay películas o ítems que los usuarios no dieron valoración, es decir, que esos espacios en blancos se consideraron como NA.

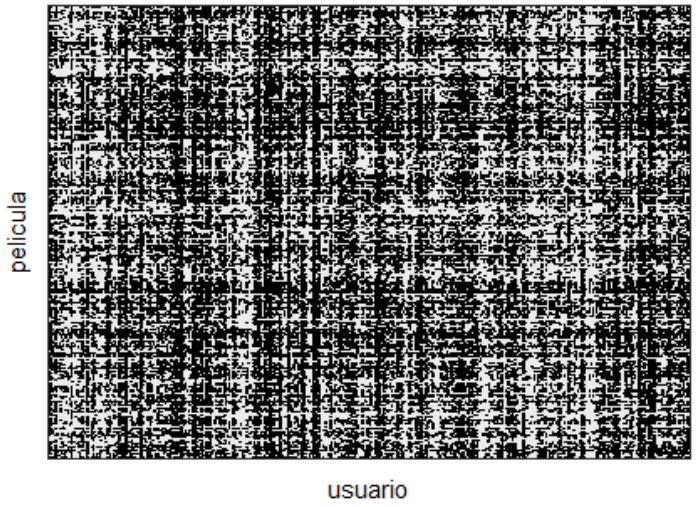

Figura 5. Valoración de los usuarios vs las películas. Adaptado de autor.

6.1.1.1 Calculo del Porcentaje de valores NA. Para realizar el cálculo del porcentaje es necesario aplicar MapReduce, se puede apreciar en el código realizado en el lenguaje de programación R. Es decir, calcular los valores NA que se encuentran en el marco de datos de MovieLense.

$$total\ de\ elementos = 943\ usuarios * 1664\ peliculas$$

$$total\ de\ elementos = 1569152$$

$$porcentaje\ de\ NA = \left(\frac{total\ de\ NA}{total\ de\ elementos}\right) * 100.$$

$$porcentaje\ de\ NA = \left(\frac{1469760}{1569152}\right) * 100.$$

$$porcentaje\ de\ NA = 93{,}66588\ \%.$$

El conjunto de datos contiene las valoraciones de 943 usuarios sobre un total de 1664 películas. Sin embargo, hay que tener en cuenta que se trata de una matriz incompleta (aproximadamente el 94% de valores ausentes), cada película ha sido valorada únicamente por una pequeña fracción de los usuarios. La mediana de valoraciones por usuario es de 64 películas.

6.1.1.2 Distribución de las valoraciones. La figura 6 representa la distribución de las valoraciones que los usuarios han realizado a determinado ítem, en el cálculo del valor medio y la mediana de las valoraciones, muestra que los usuarios tienden a valorar positivamente los ítems. En una distribución uniforme de 1 a 5 la media esperada es de 3, para esta distribución la mediana es de 4 y el valor medio (mean) es de 3.529982.

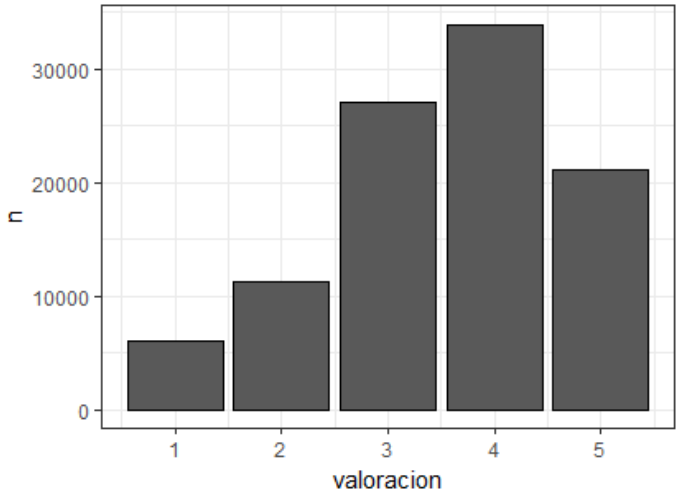

Figura 6. Distribución de las valoraciones. Adaptado por autor.

Tabla 7.

Resumen de las valoraciones ilustradas en la figura 6.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	NA's
1.0	3.0	4.0	3.5	4.0	5.0	1469760

La tabla 7 muestra el resumen de las valoraciones realizas por los usuarios, la mínima calificación es de 1 y la máxima calificación es de 5, el 1st Qu. muestra que el 25% de las valoraciones están por debajo de la cantidad, la media es de 3,5, la mediana es de 4 y el 3rd Qu. muestra que el 75% de las valoraciones están por debajo de las valoraciones, en este caso, la median.

6.1.2 Atributos de las películas. Entre los atributos descriptivos de cada película se encuentran el título, el año, una dirección web y 19 posibles temáticas. La figura 7 y 8 muestran que la mayoría de las películas disponibles en el conjunto de datos son de 1990 o posterior, y las temáticas más frecuentes son drama y comedia.

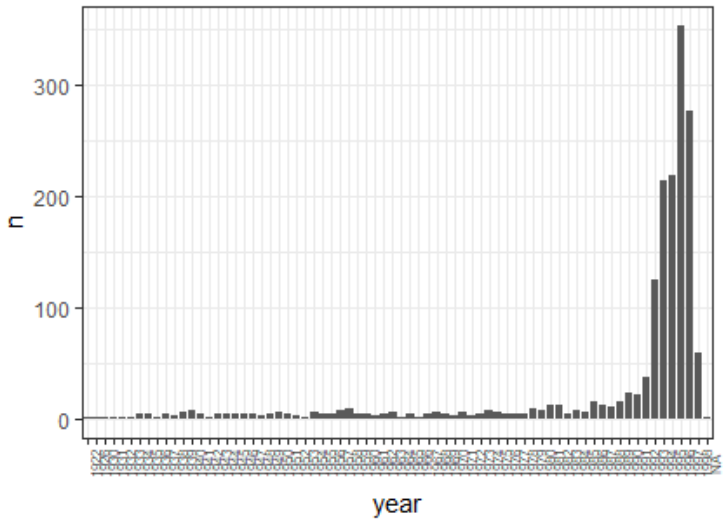

Figura 7. Películas disponibles por año. Adaptado por autor.

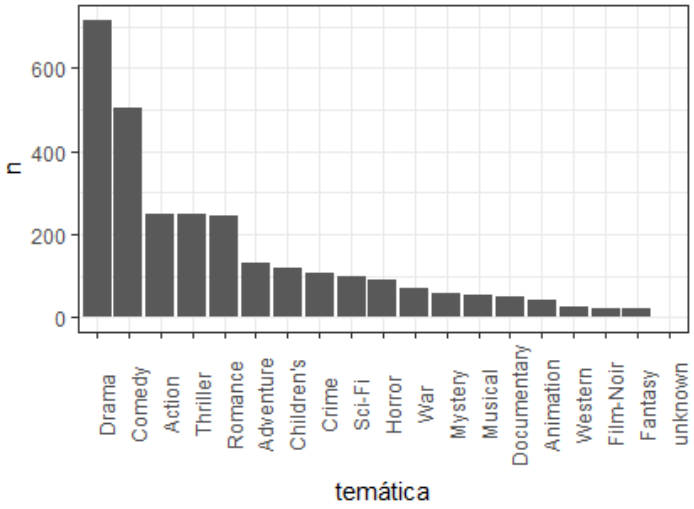

Figura 8. Temáticas más frecuentes. Adaptado por autor.

6.2 Sistema de recomendación basado en contenido.

En el sistema basado en contenido, se realizó un análisis de un top 10 a recomendar a un determinado usuario, se toma el usuario 329 donde se identifica las películas no vistas por él, asumiendo aquellas las que no le ha asignado un valor, seguido para cada *p* películas seleccionadas se calcula su similitud con las películas vistas, dado que los atributos son binarios, se emplea el índice de *Jaccard*, seleccionando las *n=15* películas más parecidas, calculando la media ponderada de las valoraciones que el usuario *329* ha dado las *n=15* películas más parecidas. Este valor se almacena como el valor predicho para las películas *p*.

En la figura 9 se muestran como recomendaciones las 10 películas con mayor valor predicho, y en la tabla 8 las películas no vistas y peso de la predicción, relacionado hacia los gustos del usuario.

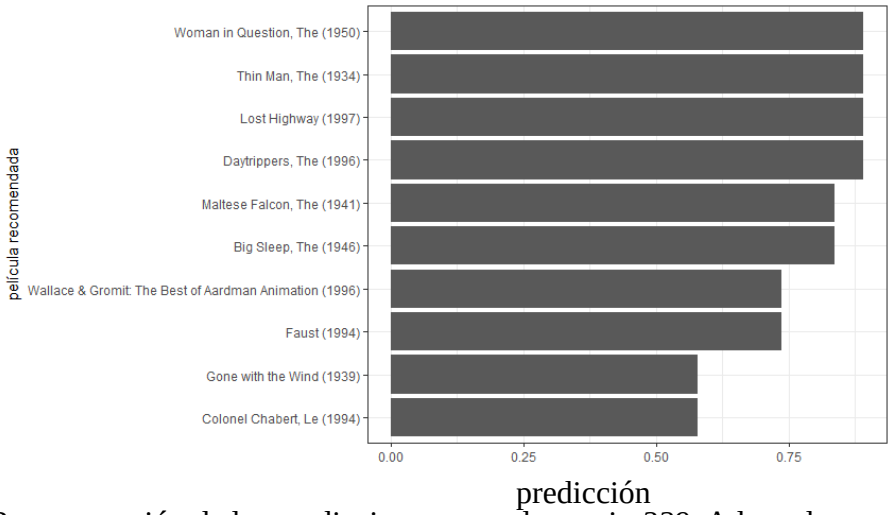

Figura 9. Representación de las predicciones para el usuario *329*. Adaptado por autor.

Tabla 8.

Predicción de las películas no vistas para el usuario 329, basado en contenido.

	Película no vista	***Predicción***
1	Daytrippers, The (1996)	0.8903585
2	Lost Highway (1997)	0.8903585
3	Thin Man, The (1934)	0.8903585
4	Woman in Question, The (1950)	0.8903585
5	Big Sleep, The (1946)	0.8361922
6	Maltese Falcon, The (1941)	0.8361922
7	Faust (1994)	0.7355976
8	Wallace & Gromit: The Best of Aardman Animation (1996)	0.7355976
9	Colonel Chabert, Le (1994)	0.5781453
10	Gone with the Wind (1939)	0.5781453

6.3 Sistema de recomendación basado en usuarios.

En el sistema basado en usuario, se calcula la similitud entre usuarios basados en sus perfiles de valoración, es decir, utilizando los vectores formados por sus valoraciones, empleando la correlación de Pearson como medida de similitud. En el algoritmo se calcula la similitud entre usuarios. Para este caso, se incluyen aquellos usuarios que hayan valorado un mínimo de películas. El valor limite se determinó en función a los datos disponibles. Se identificaron las películas que un determinado usuario no ha visto, asumiendo que son aquellas que no ha calificado.

Para cada película p seleccionada, se seleccionó un n=15 usuarios más parecidos entre sí, cuyo valor de similitud es positivo y que sí han visto la película *p*. Dado que se empleó la media ponderada como estimación final, no se incluyeron pesos negativos. Como la correlación de Pearson toma valores en el rango [-1, +1], se empleó solo aquellas observaciones con valores mayores o iguales a cero. Por lo tanto, no se tuvo en cuenta aquellas valoraciones de los usuarios que tienen el perfil opuesto. Se hizo la predicción basándose en un mínimo de usuarios, ya que para algunas películas no había suficientes usuarios.

En el sistema de recomendación basado en usuario, fue necesario calcular la media ponderada de las valoraciones que los *n=15* usuarios han dado de la película, y se muestra las recomendaciones de las 10 películas con mayor valor predicho.

6.3.1 Similitud entre usuarios. En la tabla 8 se observa el resumen de 10 similitudes entre usuarios, en más de 700 similitudes entre ellos. Para ellos se realizó el cálculo de correlación de Pearson, para identificar cuáles son los vecinos más cercanos y que puedan aportar una recomendación al usuario.

Para efectos de programación y el cálculo de la correlación de Pearson, se ordenaron los datos que aportaron más información sobre la similitud entre los usuarios, para dar una recomendación adecuada al usuario, en este caso 329.

Se identifica en la tabla 9, el top 10 de películas recomendadas para el usuario 329, con la predicción de que el usuario 329, pueda dar una valoración a este top de películas recomendadas.

Tabla 9.

Resumen de las similitudes entre el usuario 329 y los otros usuarios.

	usuario	similitud
1	329	1
2	861	1
3	544	0.9689628
4	122	0.9682458
5	338	0.9280237
6	306	0.9271726
7	370	0.8749156
8	563	0.8660254
9	731	0.8502909
10	45	0.8488382

Tabla 10.

Predicción de las películas no vistas por el usuario 329, filtrado colaborativo basado en usuario.

	Película	Predicción	n_obs_predicción
1	Raging Bull (1980)	0.8892095	15
2	Ran (1985)	0.8797687	15
3	Empire Strikes Back, The (1980)	0.8711033	15
4	One Flew Over the Cuckoo's Nest (1975)	0.8301386	15
5	This Is Spinal Tap (1984)	0.8228707	15
6	Safe (1995)	0.8122988	10
7	Third Man, The (1949)	0.7935308	15
8	Amadeus (1984)	0.7928368	15
9	Close Shave, A (1995)	0.7551640	15
10	Breaking the Waves (1996)	0.7512220	15

Nota: La columna *n_obs_prediccion* contiene el número de usuarios que se han empleado para estimar la valoración de la película. Es importante tenerlo, aunque por defecto son 15, puede ocurrir que para algunas películas no haya tantos usuarios que las hayan valorado.

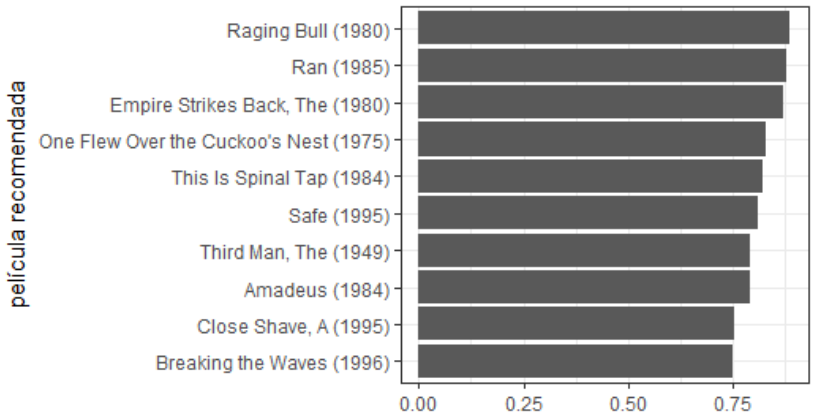

Figura 10. grafico del top 10 de película predichas. Adaptado por autor.

6.4 Filtrado colaborativo basado en ítems

En el filtrado colaborativo basado en ítems, se identificó las películas que, determinado usuario no ha visto, asumiendo son aquellas que no les ha dado valoración, donde en cada *p* películas seleccionadas, se calculó la similitud que un usuario ha visto basando en el perfil de la valoración que han recibido, es decir, se utilizó vectores formados por sus valoraciones y empleando la correlación de Pearson como medida de similitud.

Se calculó la similitud entre las películas, incluyendo únicamente aquellas películas que hayan sido valoradas por un mínimo de usuarios. Seleccionando los *n=15* películas más parecidas, también se emplea la media ponderada, al igual que basado en usuario no se tuvieron en cuenta los pesos negativos, sabiendo que en la correlación de Pearson toma valores en el rango [-1, +1], tomando aquellas observaciones con valores mayores o iguales a cero.

Se calculó la media ponderada de las valoraciones que un usuario ha hecho de las *n=15* películas más parecidas. Es conveniente saber que las recomendaciones se basaron en un mínimo de observaciones. Para que el cálculo de similitudes entre películas sea válido, se emplean únicamente películas que hayan recibido un mínimo de 10 valoraciones. En la tabla 11 se muestra, el top 10 de películas recomendadas para el usuario 329, con la predicción.

Tabla 11.

Predicción de las películas no vistas por el usuario 329, filtrado colaborativo basado en ítem.

	Película no vista	**predicción**
1	Postino, Il (1994)	0.9094037
2	Hurricane Streets (1998)	0.8242524
3	Mina Tannenbaum (1994)	0.7286342
4	Story of Xinghua, The (1993)	0.7256725
5	One Flew Over the Cuckoo's Nest (1975)	0.7099126
6	In the Company of Men (1997)	0.6634061
7	Patton (1970)	0.6486872
8	In the Bleak Midwinter (1995)	0.6274024
9	Live Nude Girls (1995)	0.6023929
10	Curdled (1996)	0.5906285

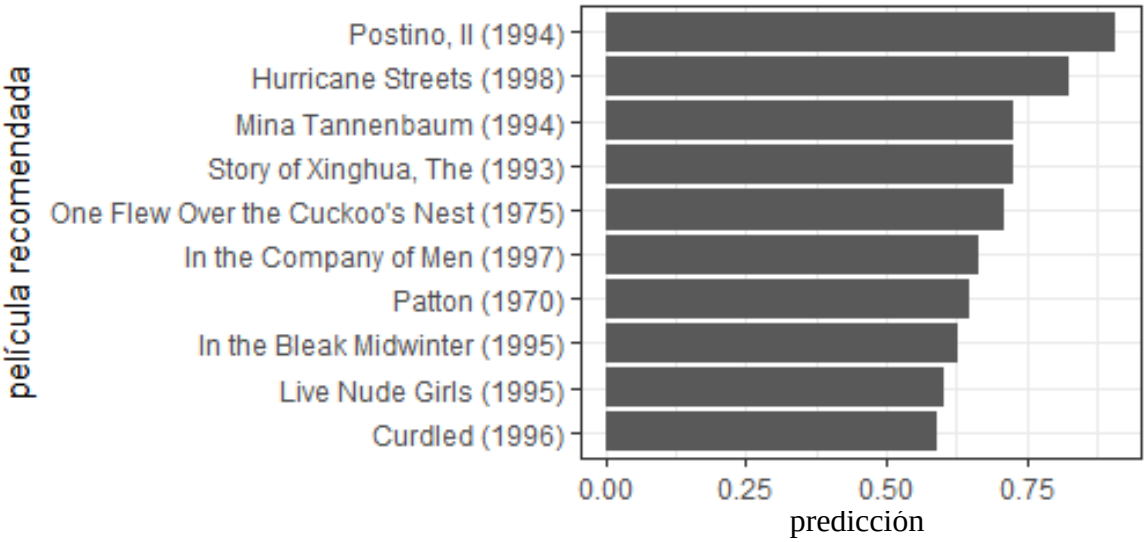

Figura 11 grafico del top 10 de película predichas. Adaptado por autor.

Según cada uno de los filtrados colaborativos muestra resultados diferentes para cada usuario con peso de predicción diferente, teniendo en cuenta que las predicciones hechas por cada sistema de recomendación son diferentes para el usuario, las películas recomendadas son diferentes en cada uno de los filtrados colaborativos mostrados anteriormente. En la tabla 12 se muestra el peso de las predicciones para cada sistema.

Tabla 12.

Comparación del peso de las predicciones según el filtrado colaborativo.

Basado en contenido.	**Basado en el usuario**	**Basado en ítems.**
Predicción	***Predicción***	***predicción***
0.8903585	0.8892095	0.9094037
0.8903585	0.8797687	0.8242524
0.8903585	0.8711033	0.7286342
0.8903585	0.8301386	0.7256725
0.8361922	0.8228707	0.7099126
0.8361922	0.8122988	0.6634061
0.7355976	0.7935308	0.6486872
0.7355976	0.7928368	0.6274024
0.5781453	0.7551640	0.6023929
0.5781453	0.7512220	0.5906285

Nota: Solo se muestran el peso de las predicciones, porque las películas recomendadas en cada sistema fueron diferentes.

Teniendo en cuenta los pesos en los sistemas de recomendación, tienen variación significativa, también se puede apreciar que las películas recomendadas son diferentes y que el usuario no ha visto y posiblemente le pueda interesar.

6.5 Sistema de recomendación utilizando la herramienta MapReduce con filtrado colaborativo.

En esta sección se realizó un sistema de recomendación aplicando MapReduce teniendo en cuenta el filtrado colaborativo, se hizo la comparación con la asociación de roles para poder identificar la precisión entre ellos, no se tiene en cuenta a que usuario se le va a realizar recomendaciones según la interacción con los ítems y similitud con otros usuarios según el ítem. Para este caso, tuvo lugar la frecuencia entre los ítems.

En la figura 12 se muestra el top 20 de los ítems más frecuentes, Es notable que la frecuencia del artista más habitual no es tan alta, en relación con el tamaño del conjunto de datos (n = 859, número de elementos = 285).

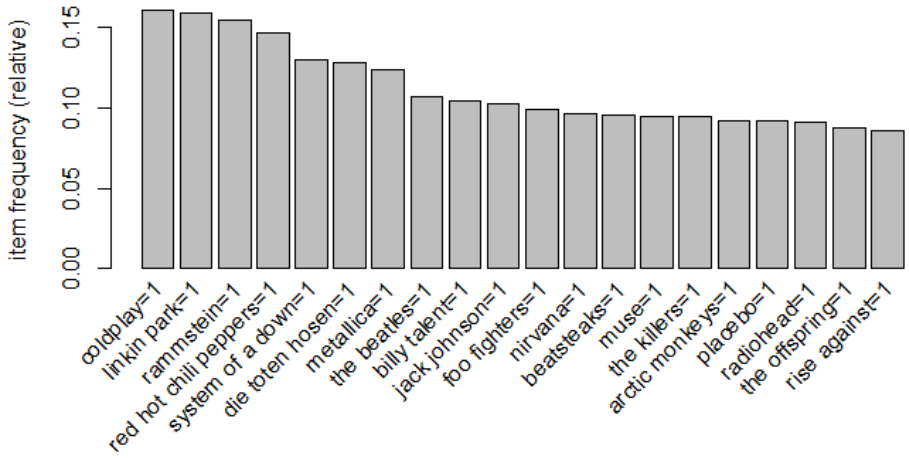

Figura 12. Frecuencia de artículo para los 20 ítems principales. Adaptado por autor.

En la tabla 13 se encuentra el resumen de los elementos de frecuencia, representado por el 1Qu., 3Qu., la mínima y máxima frecuencia, la media y la mediana según los datos.

Tabla 13.

Resumen de los elementos de frecuencia.

RESUMEN					
Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
12.00	22.00	28.00	37.17	43.00	138.00

En la figura 13 se muestra el número de ítems y la frecuencia de visitas, en este caso a los artistas musicales. La distribución de la frecuencia está sesgada correctamente con alrededor del 2% de los ítems que aparecen más de 100 veces en el conjunto de datos.

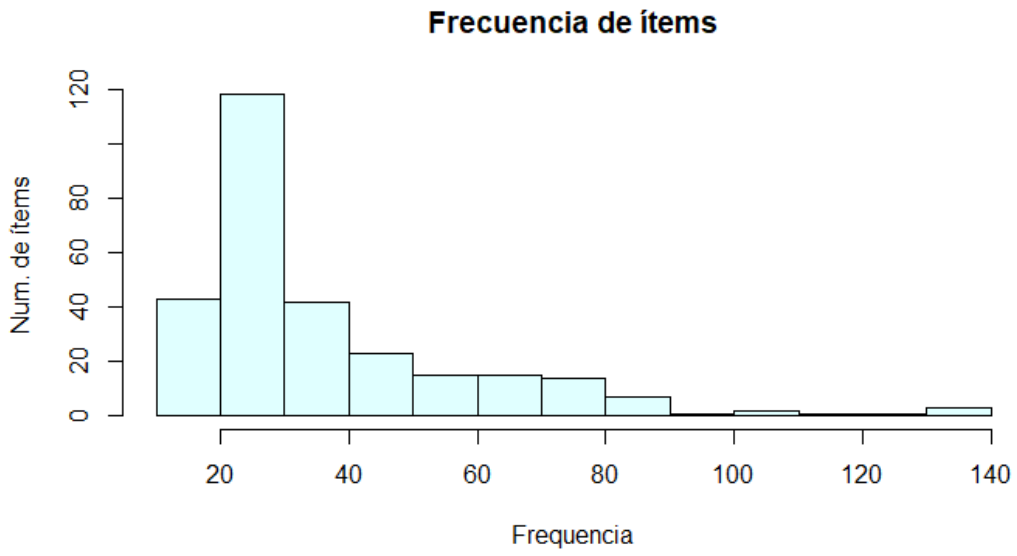

Figura 13. Frecuencia de los ítems vs el número de ítems. Adaptado por autor.

En las figuras 14 y 15 se muestra la distribución del top 5 de la asociación de roles para realizar la precisión según matriz, es decir, que para cada distribución hay una matriz similitud.

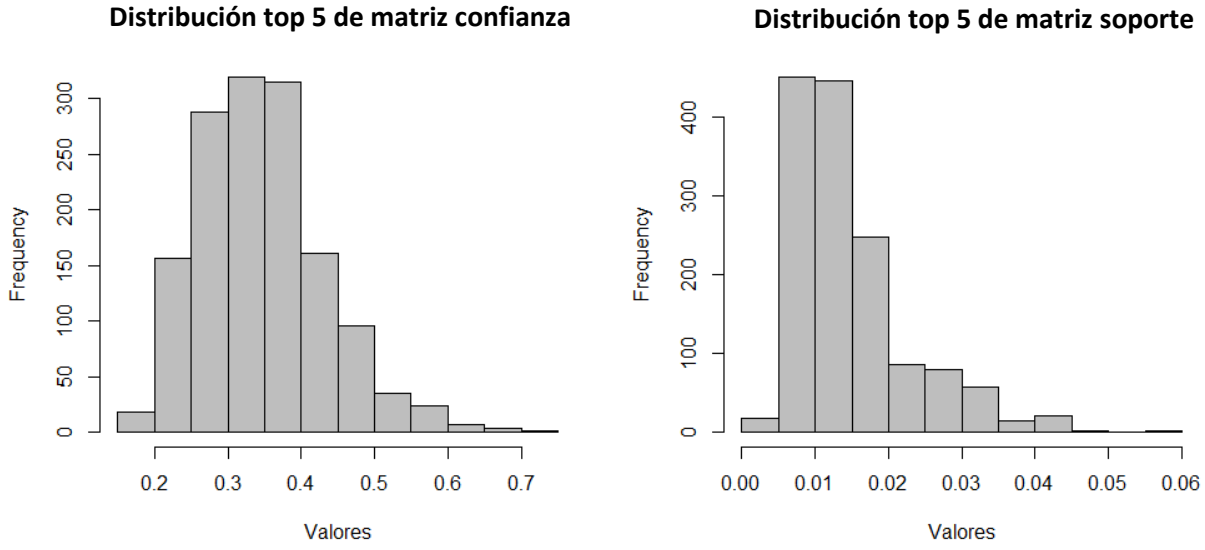

Figura 14. Distribución del top 5 de la matriz de confianza y soporte. Adaptada por autor.

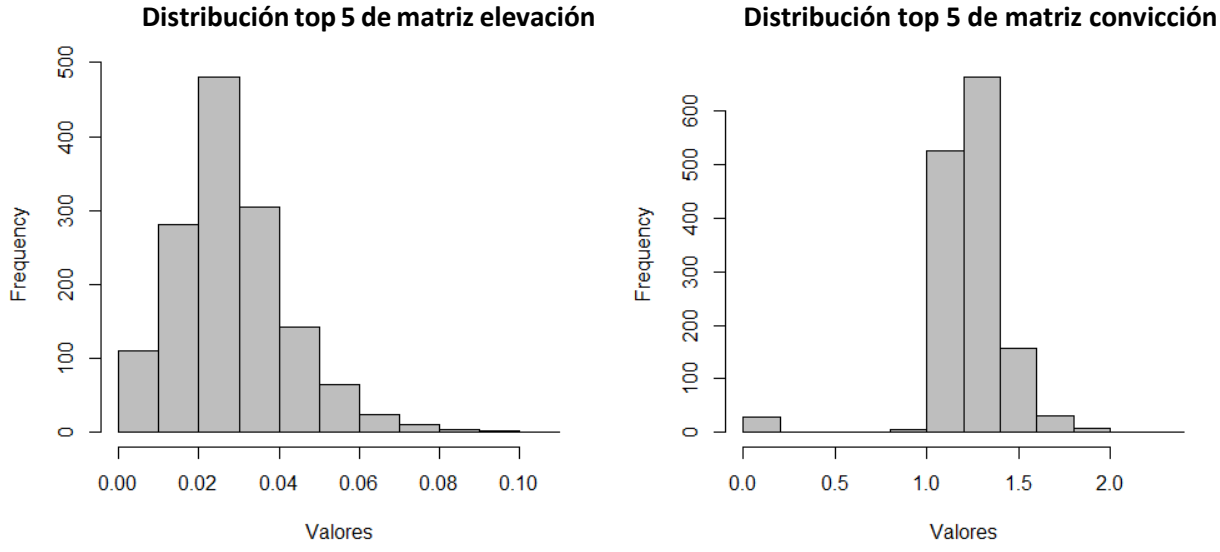

Figura 15. Distribución del top 5 de la matriz de elevación y convicción. Adaptada por autor.

Al observar la distribución de cada medida, se concluye que el tamaño del conjunto de datos no fue suficiente para extraer recomendaciones "fuertes". Además, es muy probable que una recomendación tenga un soporte de 0.009-0.017 y, en el mejor de los casos, un soporte máximo de 0.057. Este tipo de apoyo puede implicar que las recomendaciones pueden ocurrir simplemente por casualidad, por lo que debemos interpretar cada recomendación con precaución.

En el apéndice F se muestra la programación en el lenguaje natural en R apoyado de MapReduce; donde se observó las recomendaciones para cada una de los casos propuestos y vistos en cada figura. También se muestra en la tabla 14 la precisión para cada uno de los casos que se propusieron.

Tabla 14.

Precisión en el sistema de recomendación, Filtrado colaborativo, asociación de roles de soporte, de confianza, de elevación y de convicción

	Precisión
CF	0.2298021
AR soporte	0.2520256
AR confianza	0.2520256
AR elevación	0.2357509
AR convicción	0.1861234

7 Conclusiones

En el presente proyecto se realizó una revisión de los grandes volúmenes de datos, conocido como Big Data y sistemas de recomendación. Todo esto enmarcado al análisis de las técnicas utilizadas en los sistemas de recomendación, conocidas como el filtrado colaborativo basado en contenido, basado en ítems y basado en usuario. También se resalta las medidas de similitud que acompañan los sistemas de recomendación.

Es importante resaltar que la revisión de la literatura solo se basó en Big Data y sistemas de recomendación designando un intervalo de tiempo, iniciando en el año 2009 y terminando en el año 2017, observando un aumento en las investigaciones y aportes en sistemas de recomendación basados en filtrado colaborativo y aplicando la herramienta escalable de MapReduce.

Se buscó hacer una revisión de las bases de datos del benchmarking, que se ajustan para obtener y hacer una recomendación a partir de los datos, en este caso, los datos que se obtuvieron en realidad eran escasos, es decir, que la precisión para entregar las recomendaciones dependía de qué tan nutrida estaba la base de datos.

Se realizaron tres escenarios de sistemas de recomendación basados en filtrados colaborativos; filtrado colaborativo basado en usuario, basado en ítems y basado en contenido. Por lo tanto, se obtuvo que, en los tres escenarios las predicciones y las recomendaciones eran diferentes entre sí, todo obtenido como un top 10 de películas que le podrían interesar al usuario que se analizó a partir de la base de datos. También se obtuvo otro sistema de recomendación basado en filtrado colaborativo basado en ítems, aplicando la herramienta de MapReduce comparado con la

asociación de roles (basado en contenido) y se obtuvo que el filtrado colaborativo tuvo un mayor rendimiento en la precisión de las recomendaciones.

Por lo tanto, se da cumplimiento a los objetivos propuestos, aunque todavía hay mucho margen de mejora y la posibilidad de explorar sobre base de datos más grandes que permitan integrar las técnicas y herramientas que se utilizaron.

8 Recomendaciones

En la interacción usuario – ítems, se pudo ver que la predicción en las recomendaciones no era tan exacta, es decir, que los datos adquiridos no estaban tan nutridos para poder hacer recomendaciones. Es recomendable tener una base de datos con más información para realizar mejor las predicciones y así haya una mejor precisión en el análisis.

El número de NA (No Aplica o espacio vacío) puede cambiar por predicciones tomadas a partir de la información de los usuarios y las valoraciones realizadas que se asemejan a estos vacíos, es recomendable analizar los vecinos más cercanos y que se asemejen a los ítems ya valorados por los usuarios y así poder realizar un análisis más exacto.

En el marco de MapReduce en complicado complementarlo con los sistemas de recomendación en el lenguaje de programación R, es decir, que se debe realizar una serie de descargar de paquetes y después integrarlo con líneas de códigos para hacer la función de Mapeo y la función de Reduce.

El complemento de los sistemas de recomendación con la herramienta de MapReduce, puede mejorar a partir de sus necesidades y que simplifique el algoritmo, sin escribir tantas líneas de código.

Referencias bibliográficas

Aguilar, L. J. (2013). *Big Data: Analisis de Grandes Volúmenes de Datos en Organizaciones* (primera edición ed.). México: Alfaomega.

Adomavicius, G., & Tuzhilin, A. (2005, June). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*. https://doi.org/10.1109/TKDE.2005.99

Almazro, D., Shahatah, G., Albdulkarim, L., Kherees, M., Martinez, R., & Nzoukou, W. (2010). A Survey Paper on Recommender Systems. *Arxiv Preprint ArXiv*, *abs/1006.5*(5), 129–151. https://doi.org/abs/1006.5278

Bennett, J., & Lanning, S. (2007). The Netflix Prize. *KDD Cup and Workshop*, 3–6. https://doi.org/10.1145/1562764.1562769

Breese, J. S., Heckerman, D., & Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. *UAI'98 Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, 43–52. https://doi.org/10.1111/j.1553-2712.2011.01172.x

Chen, Q., Guo, M., Deng, Q., Zheng, L., Guo, S., Shen, Y., … Guo, S. (2013). HAT: history-based auto-tuning MapReduce in heterogeneous environments. *J Supercomput*, *64*, 1038–1054. https://doi.org/10.1007/s11227-011-0682-5

Cho, Y. H., Kim, J. K., & Kim, S. H. (2002). A personalized recommender system based on web usage mining and decision tree induction. *Expert Systems with Applications*, *23*(3), 329–342. https://doi.org/10.1016/S0957-4174(02)00052-0

Christensen, I., & Schiaffino, S. (2013). Matrix Factorization in Social Group Recommender Systems. In *2013 12th Mexican International Conference on Artificial Intelligence* (pp. 10–

16). IEEE. https://doi.org/10.1109/MICAI.2013.7

Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified Data Processing on Large Clusters. *Communications of the ACM*, *51*(1), 107. https://doi.org/10.1145/1327452.1327492

Deshpande, M., & Karypis, G. (2004). Item-based top- N recommendation algorithms. *ACM Transactions on Information Systems*, *22*(1), 143–177. https://doi.org/10.1145/963770.963776

Dooms, S., Audenaert, P., Fostier, J., De Pessemier, T., & Martens, L. (2014). In-memory, distributed content-based recommender system. *JOURNAL OF INTELLIGENT INFORMATION SYSTEMS*, *42*(3), 645–669. https://doi.org/10.1007/s10844-013-0276-1

Galán, S. M. (2007). Filtrado Colaborativo y Sistemas de Recomendación. *IRC 2007 , Universidad Carlos III de Madrid*, 1–8. Retrieved from http://www.it.uc3m.es/~jvillena/irc/practicas/06-07/31.pdf

García, S., Ra-Mírez-Gallego, S., Luengo, J., Herrera, F., & Ramírez-Gallego, S. (2016). Big Data monografía monografía Big Data: Preprocesamiento y calidad de datos. Retrieved from http://sci2s.ugr.es/sites/default/files/ficherosPublicaciones/2133_Nv237-Digital-sramirez.pdf

Ghemawat, S., Gobioff, H., & Leung, S.-T. (2003). The Google file system. In *Proceedings of the nineteenth ACM symposium on Operating systems principles - SOSP '03* (Vol. 37, p. 29). New York, New York, USA: ACM Press. https://doi.org/10.1145/945445.945450

GroupLens. (n.d.). Retrieved May 7, 2018, from https://grouplens.org/

Hahsler, M. (2011). recommenderlab: A Framework for Developing and Testing Recommendation Algorithms. *Nov*, 1–37. Retrieved from https://cran.r-project.org/web/packages/recommenderlab/vignettes/recommenderlab.pdf

Jasim Hadi, H., Hameed Shnain, A., Hadishaheed, S., & Haji Ahmad, A. (2014). BIG DATA AND FIVE V'S CHARACTERISTICS. *Institute of Research and Journals - IRAJ*, 8. Retrieved from http://www.iraj.in/up_proc/pdf/110-141576915829-36.pdf

Jones, M. T. (2013). Recommender systems , Part 1 : Introduction to approaches and algorithms Learn about the concepts that underlie web recommendation engines, 1–8. Retrieved from http://www.alvaroriascos.com/mineriadatos/sistemasRecomendacion/os-recommender1-pdf.pdf

Kavakiotis, I., Tsave, O., Salifoglou, A., Maglaveras, N., Vlahavas, I., & Chouvarda, I. (2017). Machine Learning and Data Mining Methods in Diabetes Research. *Computational and Structural Biotechnology Journal*, *15*, 104–116. https://doi.org/10.1016/j.csbj.2016.12.005

Kordon, A. K. (2010). Machine Learning: The Ghost in the Learning Machine. In *Applying Computational Intelligence* (pp. 73–113). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-69913-2_4

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. *Computer*, *42*(8), 42–49. https://doi.org/10.1109/MC.2009.263

Kumar, B. (2016). Cosine Based Latent Factor Model for Precision Oriented Recommendation. *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS*, *7*(1), 451–457.

Lai, C. F., Chang, J. H., Hu, C. C., Huang, Y. M., & Chao, H. C. (2011). CPRS: A cloud-based program recommendation system for digital TV platforms. In *Future Generation Computer Systems* (Vol. 27, pp. 823–835). https://doi.org/10.1016/j.future.2010.10.002

Lawrence, R. D., Almasi, G. S., Kotlyar, V., Viveros, M. S., & Duri, S. S. (2001). Personalization of supermarket product recommendations. *Data Mining and Knowledge Discovery*, *5*(1–2),

11–32. https://doi.org/10.1023/A:1009835726774

Li, X., & Chen, H. (2013). Recommendation as link prediction in bipartite graphs: A graphkernel-based machine learning approach. *DECISION SUPPORT SYSTEMS*, *54*(2), 880–890. https://doi.org/10.1016/j.dss.2012.09.019

Manzato, M. G., Domingues, M. A., Fortes, A. C., Sundermann, C. V, D'Addio, R. M., Conrado, M. S., … Pimentel, M. G. C. (2016). Mining unstructured content for recommender systems: an ensemble approach. *Information Retrieval Journal*, *19*(4), 378–415. https://doi.org/10.1007/s10791-016-9280-8

Meng, S., Dou, W., Zhang, X., & Chen, J. (2014). KASR: A keyword-aware service recommendation method on mapreduce for big data applications. *IEEE Transactions on Parallel and Distributed Systems*, *25*(12), 3221–3231. https://doi.org/10.1109/TPDS.2013.2297117

Netflix. (2009). Grand Prize awarded to team BellKor's Pragmatic Chaos. Retrieved September 24, 2017, from http://www.netflixprize.com/community/topic_1537.html

Nilashi, M., Bagherifard, K., Rahmani, M., & Rafe, V. (2017). A recommender system for tourism industry using cluster ensemble and prediction machine learning techniques. *Computers & Industrial Engineering*, *109*, 357–368. https://doi.org/10.1016/j.cie.2017.05.016

Oancea, B., & Dragoescu, R. M. (2014). Integrating R and Hadoop for Big Data Analysis. *Romanian Statistical Review*, (2), 83–94. Retrieved from http://arxiv.org/abs/1407.4908

Ortega, F., Hernando, A., Bobadilla, J., & Kang, J. H. (2016). Recommending items to group of users using Matrix Factorization based Collaborative Filtering. *Information Sciences*, *345*, 313–324. https://doi.org/10.1016/j.ins.2016.01.083

Papagelis, M., & Plexousakis, D. (2005). Qualitative analysis of user-based and item-based

prediction algorithms for recommendation agents. *Engineering Applications of Artificial Intelligence*, *18*(7), 781–789. https://doi.org/10.1016/j.engappai.2005.06.010

Pilares, F. L., M, E. De, Conacyt, D., Mar, H. T., Alberto, C., Garc, R., & Schaeffer, E. (2015). c Komputer Sapiens , A ˜ no V Volumen III , septiembre-diciembre 2013 , es una publicaci ´ on cuatrimes- tral de la Sociedad Mexicana de Inteligencia Artificial , A . C ., con domicilio en Ezequiel Montes 56 s / n , nos de M ´ exico S . A . de C . V ., ca. *KOMPUTER SAPIENS*, 1–40. Retrieved from http://komputersapiens.smia.mx/files_ALMoStuNrEaChaBLe/ks71_1.83MB_compacta.pdf#page=14

Ráez, A. M. (2014). Sistemas de recomendación. Retrieved from https://repositorio.uam.es/handle/10486/662240

Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). GroupLens : An Open Architecture for Collaborative Filtering of Netnews. *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, 175–186. https://doi.org/10.1145/192844.192905

Sarwar, B., Karypis, G., Konstan, J., & Reidl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the tenth international conference on World Wide Web - WWW '01* (pp. 285–295). New York, New York, USA: ACM Press. https://doi.org/10.1145/371920.372071

Shardanand, U., & Maes, P. (1995). Social information filtering: Algorithms for Automating "Word of Mouth." *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '95*, 210–217. https://doi.org/10.1145/223904.223931

Taffese, W. Z., & Sistonen, E. (2017). Machine learning for durability and service-life assessment

of reinforced concrete structures: Recent advances and future directions. *Automation in Construction*, *77*, 1–14. https://doi.org/10.1016/j.autcon.2017.01.016

Taylor, P., Bell, R. M., Koren, Y., & Volinsky, C. (2013). All Together Now : A Perspective on the. *CHANCE*, *23*(December 2014), 37–41. https://doi.org/10.1080/09332480.2010.10739787

Tran, T., & Cohen, R. (2000). Hybrid Recommender Systems for Electronic Commerce. *AAAI Technical Report*, *12*(4), 78–84. https://doi.org/10.1023/A:1021240730564

Tsai, C. F., & Hung, C. (2012, April). Cluster ensembles in collaborative filtering recommendation. *Applied Soft Computing Journal*. https://doi.org/10.1016/j.asoc.2011.11.016

Xu, R., Wang, S., Zheng, X., & Chen, Y. (2014). Distributed collaborative filtering with singular ratings for large scale recommendation. *Journal of Systems and Software*, *95*, 231–241. https://doi.org/10.1016/j.jss.2014.04.045

Yejas, O. D. L., Zhuang, W., & Pannu, A. (2014). Big R: Large-scale analytics on hadoop using R. In *Proceedings - 2014 IEEE International Congress on Big Data, BigData Congress 2014* (pp. 570–577). https://doi.org/10.1109/BigData.Congress.2014.88

Yu, H. F., Hsieh, C. J., Si, S., & Dhillon, I. S. (2014a). Parallel matrix factorization for recommender systems. *Knowledge and Information Systems*, *41*(3), 793–819. https://doi.org/10.1007/s10115-013-0682-2

Yu, H. F., Hsieh, C. J., Si, S., & Dhillon, I. S. (2014b). Parallel matrix factorization for recommender systems. *Knowledge and Information Systems*, *41*(3), 793–819. https://doi.org/10.1007/s10115-013-0682-2