UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR

POSGRADOS

MAESTRÍA EN

**SOFTWARE CON MENCIÓN EN DISEÑO DE ARQUITECTURA DE SISTEMAS**

RPC-SO-34-NO.778-2021

OPCIÓN DE TITULACIÓN:

ARTÍCULOS PROFESIONALES DE ALTO NIVEL

TEMA:

# DESARROLLO DE UN SISTEMA DE RECOMENDACIÓN BASADO EN FILTRADO COLABORATIVO PARA BIG DATA MEDIANTE SPARK Y AWS

AUTOR:

BRYAM DAVID VEGA MORENO

DIRECTOR:

REMIGIO ISMAEL HURTADO ORTIZ

CUENCA – ECUADOR

2023

**Autor:**

**Bryam David Vega Moreno**
Ingeniero en Ciencias de la Computación.
Candidato a Magíster en Software con Mención en Diseño de Arquitectura de Sistemas por la Universidad Politécnica Salesiana – Sede Cuenca.
Vegabryam40@gmail.com

**Dirigido por:**

**Remigio Ismael Hurtado Ortiz**
Ingeniero en Sistemas.
Maestría en Software y Tecnologías de la Información. Máster en Ciencias y Tecnologías de la Computación. Doctorado en Ciencia y Tecnologías de la Computación para Smart Cities.
rhurtadoo@ups.edu.ec

***DEDICATORIA***

*Dedico este trabajo a mis padres, quienes han sido el motor que me ha impulsado siempre a seguir adelante en mi vida tanto profesional como personal. Gracias por su apoyo incondicional y cariño. Sin ellos, nada de lo que he logrado hasta el momento hubiese sido posible, este trabajo es logro mío y suyo.*

*A mi hermano y cuñada por su apoyo en cada momento de mi trayectoria como profesional. Han sido un gran ejemplo para seguir como persona, amigos y sobre todo como profesionales de calidad. De igual forma a mi gran familia, sobre todo a mis tíos Carlos, Yolanda, Wilson y Sonia que han estado siempre presentes en cada momento importante de mi vida y dando ánimo para seguir adelante.*

*A mis primos y sobrinos Ismael, Josue, Maria, Nicolas, Daniel, Melisa, David, Matias y Paula quienes siempre me sacaban una sonrisa en cada momento que lo necesitaba y apoyándome siempre en lo que necesitaba. Más que mis primos siempre los he considerado como mis hermanos por lo que este trabajo es para ellos.*

*De igual forma, dedico esto a mis amigos, de forma especial a Santiago Pardo, Ismael Peñafiel, Ruben Baculima, Jason Heras, Gabriel Reinoso, Paula Cherrez y Andrea Reyes por siempre haber estado a mi lado en cada momento de mi vida tanto como profesional y como persona, gracias por siempre estar presentes y darme todo su cariño desde el colegio hasta este momento, los llevo siempre en mi corazón.*

*Por último y no menos importante, a mi pareja María del Carmen Estrella, este trabajo es dedicado a la persona que me ha hecho feliz todo este tiempo y me ha apoyado en cada momento de todo este proceso.*

*Bryam David Vega Moreno*

***AGRADECIMIENTO***

*Primero quiero agradecer a Dios por siempre darme la Fortaleza y el conocimiento para afrontar cada meta que me he propuesto a lo largo de mi vida. Agradezco a la Universidad por todos los conocimientos compartidos a lo largo de este año en donde me he formado como un profesional de alto impacto y un buen ser humano.*

*Agradezco a mis maestros por toda su dedicación y esfuerzo realizado a lo largo de este camino. De manera especial agradezco a mi tutor, maestro y amigo Remigio Hurtado, por toda su dedicación y compromiso a lo largo de este trabajo, así como también como consejero en mi vida profesional. Gracias por siempre estar guiándome en cada momento a lo largo de mi carrera profesional dándome ánimos para seguir adelante con todas mis metas.*

*Agradezco a mis padres por todo su cariño y dedicación hacia mí, su amor y responsabilidad como padres me ha permitido llegar al final de esta etapa de la mejor manera, todo este trabajo es gracias a su gran trabajo como padres y seres humanos.*

*Agradezco a mis amigos y ex compañeros de Software Social Consultores, por todo el apoyo que recibí de ellos cuando empecé este camino de maestría, su apoyo como profesionales y amigos me impulso siempre ha mejorar cada día.*

*Por último, gradezco a toda mi familia, ellos han sido pieza fundamental para mi desarrollo como profesional y personal, cada miembro de mi familia ha aportado con su grano de área para ser la persona que son ahora.*

*Bryam David Vega Moreno*

# Tabla de Contenido

# Indice de Figuras

# INDICE DE TABLAS

# DESARROLLO DE UN SISTEMA DE RECOMENDACIÓN BASADO EN FILTRADO COLABORATIVO PARA BIG DATA MEDIANTE SPARK Y AWS

AUTOR(ES):

BRYAM DAVID VEGA MORENO

# Resumen

Los sistemas de recomendación hoy en día permiten solucionar los problemas de sobrecarga de información o infoxicación. De estos sistemas, el enfoque basado en filtrado colaborativo es de los sistemas de recomendación más utilizados en la actualidad. Sin embargo, el crecimiento exponencial de información produce que dichos sistemas no puedan procesar toda la data proveniente de diferentes fuentes, por lo que es necesario crear además de un sistema de recomendación, una infraestructura que soporte el constante crecimiento de información. Adicional a ello, el problema de crear una infraestructura que soporte Big Data incurre en grandes costes de mantenimiento, construcción de espacio, entre otros; lo que hace extremadamente costoso tratar de implementar esta infraestructura de forma física. Por ello, en este proyecto se propone el desarrollo de un sistema de recomendación basado en filtrado colaborativo para un entorno Big Data utilizando una infraestructura en la nube con Amazon Web Services. En dicho trabajo se propone una infraestructura que permita desarrollar e implementar un sistema de recomendación el cual soporte una cantidad masiva de información. Así mismo, se realiza una evaluación de dicho sistema de recomendación utilizando métricas de calidad, como a su vez medir los tiempos de cada uno de los procesos que se necesitaron para desarrollar el sistema comparándolo en un entorno Big Data frente a un entorno tradicional, que no trabaje bajo un sistema distribuido.

**Palabras clave:** Sistema de recomendación, Amazon Web Services, filtrado colaborativo, Filtrado Colaborativo, Arquitecturas Big Data, Sistemas Distribuidos, Big Data, Rendimiento

# ABSTRACT

Nowadays, recommender systems help to solve the problems of information overload or infoxication. Of these systems, the collaborative filtering approach is one of the most widely used recommender systems in use today. However, the exponential growth of information means that these systems cannot process all the data coming from different sources, so it is necessary to create, in addition to a recommendation system, an infrastructure that supports the constant growth of information. In addition to this, the problem of creating an infrastructure that supports Big Data incurs large maintenance costs, space construction, among others, which makes it extremely expensive to try to implement this infrastructure physically. Therefore, this project proposes the development of a recommendation system based on collaborative filtering for a Big Data environment using a cloud infrastructure with Amazon Web Services. This work proposes an infrastructure that allows the development and implementation of a recommendation system that supports a massive amount of information. Likewise, an evaluation of this recommendation system is performed using quality metrics, as well as measuring the time of each of the processes that were needed to develop the system, comparing it in a Big Data environment versus a traditional environment that does not work under a distributed system.

**Keyboards:** Recommender System, Amazon Web Services, collaborative filtering, Collaborative Filtering, Big Data Architectures, Distributed Systems, Big Data, Performance

# 1. Introducción

Con el acceso a internet y la creación de distintas aplicaciones en constante crecimiento, adquirir información por parte de las empresas se ha vuelto cada vez más fácil, provocando un problema denominado infoxicación o también llamado sobrecarga de información. En la actualidad, los usuarios pueden registrar información de cualquier forma y desde cualquier lugar, provocando así un incremento exponencial y veloz de datos. Todo este gran conjunto de registros obtenidos por distintos medios permite a los **Sistemas de Recomendación** (SR) sugerir productos al gusto del usuario con la finalidad de llamar su atención y en consecuencia generar un mayor crecimiento al negocio.

Los SR en la actualidad se han convertido en pieza fundamental en la mayoría de las aplicaciones que se utilizan hoy en día. Empresas como Netflix, TikTok, Meta, Google, entre otras, utilizan dichos sistemas para presentar y ofrecer productos o servicios que al usuario le gustaría o preferiría consumir. Para (Hurtado et al., 2019) un SR es parte de un "conjunto de herramientas de inteligencia artificial para abordar problemas de sobrecarga de información", en otras palabras, los SR buscan mitigar los problemas de infoxicación. Por dicho motivo, los SR se han dividido en distintos enfoques para resolver distintos problemas, dicha clasificación según lo indica (Zhu & Hurtado, 2018) consta de los siguientes enfoques: Basado en filtrado colaborativo, basado en contenidos, basado en información demográfica, basada en base social y conscientes del contexto. De toda esta clasificación, el enfoque más utilizado por los SR es el filtrado colaborativo.

El ***filtrado colaborativo*** (FC) es una técnica de recomendación que se basa en el análisis de las evaluaciones y preferencias de un conjunto de usuarios para de esta forma, generar recomendaciones personalizadas. La idea principal de dicho enfoque es que los usuarios que tienen gustos similares en el pasado probablemente tendrán gustos similares en el futuro. Para lograr estas recomendaciones (Zhu & Hurtado, 2018) indican que "la información se estructura como una matriz que almacena las preferencias (explícitas o implícitas) del usuario", en otras palabras, el sistema recomienda productos con base a los ratings compartidos por la comunidad de usuarios, es decir, no solamente tomando en cuenta los ratings individuales, si no, los ratings de una gran cantidad de usuarios, lo cual

ha generado que el filtrado colaborativo genere mejores resultados en comparación a otros enfoques como el basado en contenidos, haciendo que dicho enfoque sea uno de los temas más relevantes de índole científico en revistas de alto impacto. Dentro del filtrado colaborativo existen dos enfoques principales: El enfoque basado en memoria y el enfoque basado en modelos.

## A. Sistemas de recomendación basados en memoria

Los ***enfoques basados en memoria*** toman en consideración la información de interacción histórica entre usuarios y elementos para generar recomendaciones. Dicho enfoque se basa en la idea de que aquellos usuarios que tuvieron gustos o preferencias similares en el pasado seguirán teniendo preferencias similares en el futuro. Dentro de las técnicas más populares cuyo enfoque es basado en memoria es la técnica de *k-Nearest Neighbor* **(KNN)**. Según el autor (Subramaniyaswamy & Logesh, 2017) "KNN utiliza técnicas estadísticas para tratar a los usuarios y elementos y determinar usuarios similares con preferencias similares como vecinos. Las recomendaciones se predicen basándose en las características del vecindario del usuario objetivo activo". Para lograr estas recomendaciones, dicho algoritmo utiliza medidas de similaridad tales como Correlación de Pearson, Cosenos y Jacard.

Una de las ventajas más importantes de estos SR basados en memoria es que son fáciles de implementar y usar, sin embargo, tal y como explica (Sarwar et al., 2002) estos sistemas de recomendación sufren de problemas de escalabilidad cuando la cantidad de ítems y usuarios son grandes. Por lo cual, los SR basados en modelos resuelven del problema dicho problema y están adaptados para trabajar con una gran cantidad de datos.

## B. Sistemas de recomendación basados en modelos

Los ***enfoques basados en modelos*** hacen referencia sobre aquellos métodos del FC que utilizan técnicas de aprendizaje automático para construir un modelo que pueda

predecir las evaluaciones de los usuarios hacia ítems que aún no los ha evaluado. Adicional a ello, según explican (McNee et al., 2006) y (Said & Bellogín, 2014) "el proceso de predicción y recomendación mejora la calidad, a través de enfoques basados en modelos".

Dentro de este enfoque encontramos distintas técnicas de recomendación como la técnica de factorización matricial, el cual según explica (Guan et al., 2017) "Las matrices de clasificaciones dispersas se comprimen en dos matrices de factores densos: 1. Una matriz que contiene la información de los usuarios cuyo tamaño esta dado por (usuarios-factores) y 2. Otra que contiene la información de los elementos cuyo tamaño es (elementos-factores)".

Por otro lado tenemos la técnica *Alternating Least Squares* **(ALS)** el cual según indica (Said & Bellogín, 2014) "es un algoritmo utilizado en el filtrado colaborativo para la recomendación de elementos en un conjunto de datos con gran cantidad de usuarios y elementos". Dicho algoritmo se utiliza para aproximar la matriz original de calificaciones de usuarios y elementos en dos matrices de menor rango, lo que permite una mayor eficiencia computacional y escalabilidad en comparación con otros enfoques de factorización de matrices en el filtrado colaborativo. Para este trabajo, utilizaremos esta técnica como modelo principal de recomendación.

Adicional a ello, tal y como lo indica (Hurtado, 2020) "el filtrado colaborativo tiene dos ventajas muy importantes: La capacidad de realizar recomendaciones novedosas y la ventaja de no necesitar información personal de los usuarios registrados. Por lo tanto, no resulta muy costoso recolectar información de los usuarios, ni mucho menos se requiere ir en contra de su privacidad".

Tal como se menciona anteriormente referente a técnicas relacionadas a enfoques basados en modelos, ALS funciona con una gran cantidad de información, por lo que SR tradicionales no son capaces de manejar dicho modelo por la gran cantidad de información que se requiere. Para ello, para desarrollar dicho modelo se propone utilizar la metodología CRISP-DM el cual, según los autores (Huber et al., 2019) no es más que "un modelo de proceso estándar utilizado en la minería de datos y en el análisis de datos. Fue desarrollado por un consorcio de expertos de la industria en minería de datos y se utiliza comúnmente para ayudar a las empresas a planificar y ejecutar proyectos de minería de datos de manera efectiva". Adicional a ello, se propone trabajar en un entorno

**Big Data** el cual ofrece escalabilidad y eficiencia al momento de realizar recomendaciones.

## C. Fundamentos de Big Data

Big data es un término que se utiliza para describir grandes cantidades de datos que son demasiado complejos y variados para ser procesados por métodos tradicionales tal como lo explica (Schell, 2013). La característica principal de Big Data es su tamaño, que se mide en términos de volumen, velocidad y variedad. Hoy en día existe una variedad de herramientas y técnicas especializadas como Hadoop, Spark, NoSQL y Machine Learning. Siendo Spark, uno de los más utilizados en la actualidad.

***Apache Spark*** es un motor de procesamiento de código abierto que permite procesar grandes volúmenes de datos de manera rápida y eficiente. La ventaja de usar Spark es su procesamiento basado en memoria y no en disco como lo indica (Gu & Li, 2014). Adicional a ello Spark utiliza una abstracción de datos distribuida llamada Resilient Distributed Datasets (RDD) para procesar datos en paralelo a través de clústeres de computadoras, lo que lo hace ideal como herramienta para este trabajo.

Si bien Apache Spark y los entornos Big Data son muy potentes para procesar gran cantidad de información, el problema radica principalmente en que para tener éxito en estos entornos se requiere de una infraestructura que soporte los mismos. Por lo que crear infraestructuras físicas resulta muy costoso y altamente complejo de mantener. Por tanto, se propone utilizar proveedores en Cloud para levantar estos entornos de una forma más eficiente y sin la necesidad de realizar complejas configuraciones.

## D. Fundamentos de Cloud Computing

En la actualidad, las tecnologías en la nube han ganado gran popularidad a nivel de muchas empresas debido a que hoy en día mantener una infraestructura física incurre en varios temas a tomar en consideración, tales como el lugar donde se implementara la infraestructura, costos de mantenimiento y seguridad, costos en brindar los servicios necesarios para mantener disponible la infraestructura… entre otras. Por ello, las tecnologías en la nube toman en cuenta dichas consideraciones por un costo mucho menos elevado que tener una infraestructura física. Adicional a ello,

se logra tener una personalización en cuanto a recursos de hardware y software se requiere. De igual forma tal como indica (Sharma & Shamkuwar, 2019) “hoy en día, el cloud computing proporciona soluciones más rentables y de menor riesgo”.

Dentro del Cloud Computing, encontramos la nube como proveedora de servicios para las distintas empresas. Con ello, actualmente la nube cuenta con distintos modelos disponibles los cuales son: Infraestructura As A Service (IaaS), Platorm As A Service (PaaS) y Software As A Service (SaaS).

Existe una gran variedad de proveedores que ofrecen diferentes servicios orientados a distintas necesidades del negocio, sin embargo, los proveedores más populares hoy en día son: Azure de Microsoft, Cloud Platform de Google y Web Services de Amazon, tomando la posta como los mayores proveedores de servicios en la nube para la mayoría de las empresas.

De dicho Top, ***Amazon Web Services*** (AWS) actualmente es uno de los líderes de Infraestructura As A Service (IaaS por sus siglas en inglés) según el cuadrante mágico de Gartner. Adicional a ello, tal como indica (Raj Bala, 2022) “AWS es líder debido a su amplia gama de servicio TI que van desde nativos en la nube y edge hasta cargas de trabajo críticas y de ERP”, lo cual hace que AWS sea un proveedor de nube ideal para trabajar con servicios Big Data y procesos de Inteligencia Artificial.

Dentro de los servicios de AWS para procesos de ciencia de datos y sistemas de recomendación, tenemos a *Amazon Elastic Map Reduce* (EMR), el cual según (Web Services, 2015) es “un servicio de análisis de Big Data que permite analizar y procesar grandes cantidades de datos de manera eficiente en la nube de Amazon Web Services (AWS)”. Este servicio utiliza la tecnología de Hadoop y su marco de trabajo MapReduce para procesar los datos de manera distribuida en un cluster de servidores virtuales, lo que permite una mayor velocidad y capacidad de procesamiento.

De igual forma EMR permite trabajar con Apache Spark para procesar grandes cantidades de datos a través de sus nodos maestros y sus cluster. A continuación, en la figura 1 se puede apreciar cómo funciona Amazon EMR.


**Figura 1.** Arquitectura provisionada por Amazon Web Services para manejo de Big Data con EMR

Como se puede apreciar en la figura 1, Amazon EMR ofrece una infraestructura ya preestablecida que permite trabajar con diferentes cluster (instancias) administrador por un nodo maestro, lo que posibilita distribuir el trabajo de procesamiento de información en cada uno de estos nodos esclavos que a la final son instancias EC2 en AWS. Con ello Amazon EMR provee una arquitectura muy potente para trabajar en un entorno de Big Data, en dónde adicional a ello, administra la gestión y configuración de cada uno de los cluster del entorno. Por dicha razón, se escoge el servicio de Amazon EMR como base principal para nuestra infraestructura Big Data.

Teniendo en consideración los problemas mencionados anteriormente y sus posibles soluciones, en este proyecto se plantea el desarrollo de un SR basado en FC para un entorno de Big Data utilizando como herramienta Apache Spark y una infraestructura en la nube con AWS como proveedor principal de servicios. Este SR busca correr bajo una infraestructura en la nube soportada en un entorno de Big Data, lo cual permitirá mejorar las recomendaciones debido a la cantidad masiva de datos con las que se entrenará el modelo. Adicional a ello, se utilizarán servicios de AWS para obtener una infraestructura robusta que permita no solamente desarrollar un SR, sino contar con una infraestructura que permita realizar todo un proceso de Ingeniería de Datos Big Data. Por último se medirán los tiempos de ejecución de esta propuesta y se realizara una comparación frente a un SR que no está adaptado a un entorno de Big Data, para apreciar la mejora existente en este nuevo entorno.

# 2. Trabajos relacionados

Dentro del mundo de los SR enfocados en el FC encontramos algunos proyectos ya realizados anteriormente, tales como en (Yang et al., 2016) en el cual se plantea un SR que presenta recomendaciones sobre los intereses de los usuarios móviles tomando en consideración datos de los usuarios que son su comportamiento y valoraciones. De igual forma por otro lado tenemos a (Moreno et al., 2019) el cual plantea un SR con un enfoque mezclando dos enfoques tales como el Filtrado Colaborativo y Filtrado Basado en Contenido para lograr mejores recomendaciones orientadas a grupos de usuarios y no usuarios individuales.

Tal y como se había explicado anteriormente, la técnica de KNN es una de las utilizadas dentro de los enfoques basados en memoria, sin embargo dicha técnica presenta un problema escazes de ratings el cual puede provocar que las medidas de similitud que se utilizan en dicha técnica no sean efectivas. Por ello en el artículo presentado por los autores (Bobadilla et al., 2010) propone una nueva medida de similitud denominada JMSD el cual según lo explican los autores (Bobadilla et al., 2010) "combina la información numérica de votos con información independiente de dichos valores, basada en proporciones de votos comunes y no comunes entre cada par de usuarios".

De igual forma, dentro de los enfoques basados en modelos como una de las técnicas más populares encontramos la técnica de factorización matricial, en donde los autores presentan un enfoque denominado *non-negative matrix factorization* (NMF) el cual busca descomponer una matriz en dos matrices negativas, cuyo objetivo es encontrar una representación de bajo rango de la matriz original, donde las partes básicas y sus combinaciones formen una aproximación de los datos originales. De igual forma los autores (Salakhutdinov & Mnih, 2009) proponen un enfoque llamado *Probabilistic Matrix Factorization* (PMF), el cual consiste en descomponer una matriz de interacciones usuario-elemento en dos matrices de factores latentes, una para los usuarios y otra para los elementos.

Por otro lado, también tenemos a (Yehuda et al., 2009) en donde se trata sobre técnicas de factorización de matrices aplicadas a sistemas de recomendación. En el artículo, los

autores presentan un enfoque basado en la factorización de matrices para modelar los datos de las recomendaciones de usuarios y proporcionar recomendaciones precisas y personalizadas.

Dentro del artículo mencionado, se toma en cuenta el algoritmo ALS que de igual forma se ha usado en algunos temas de investigación relacionados a los SR. Por ejemplo, en (J. Chen et al., 2017) se presenta un enfoque eficiente y portátil para ALS que permite una mejor escalabilidad y capacidad de procesamiento en sistemas de recomendación masivos. Los autores también discuten los desafíos y limitaciones existentes en la implementación de ALS para sistemas de recomendación y proponen soluciones para abordar estos desafíos. Por otro lado, en el artículo de (Subasish et al., 2021) se presenta una solución práctica y aplicada a la creación de un sistema de recomendación personalizado para e-commerce, que utiliza la técnica de factorización de matrices mediante el algoritmo ALS en Apache Spark. Los autores proporcionan detalles sobre el proceso de implementación y los resultados obtenidos, así como las ventajas y limitaciones de esta solución.

Como se menciona anteriormente, ya existen proyectos relacionados a SR con Apache Spark, entre algunos de dichos proyectos los autores (Aljunid & Manjaiah, 2019) presentan una solución para la creación de un sistema de recomendación de películas utilizando técnicas de filtrado colaborativo, que se basa en la colaboración entre usuarios para generar recomendaciones precisas y personalizadas. Los autores también explican cómo Apache Spark se puede utilizar para implementar el algoritmo de filtrado colaborativo, y cómo esta solución ofrece una mayor escalabilidad y capacidad de procesamiento que las soluciones tradicionales. De igual forma, en el artículo escrito por (Panigrahi et al., 2016) toma la implementación de un motor de recomendación híbrido de filtrado colaborativo distribuido utilizando Apache Spark como tema principal del proyecto. Los autores presentan una arquitectura híbrida que combina el filtrado colaborativo basado en usuario y el filtrado colaborativo basado en artículo para mejorar la precisión del motor de recomendación. Además, se utiliza Apache Spark como plataforma de procesamiento distribuido para mejorar el rendimiento y la escalabilidad del sistema. Por otro lado, en el artículo presentado por (L. Chen et al., 2017) en donde proponen una solución basada en el aprendizaje automático que utiliza Apache Spark como plataforma de procesamiento distribuido para manejar grandes conjuntos de datos

de usuario y producto. El enfoque propuesto utiliza el algoritmo de filtrado colaborativo basado en modelo, que se centra en la construcción de modelos de preferencia de usuario y la predicción de la preferencia del usuario para los elementos aún no vistos.

Tal y como se comentó anteriormente, los entornos Big Data deben estar soportados bajo una infraestructura que les permita procesar gran cantidad de información, por lo que en (Manakkadu et al., n.d.) se propone un nuevo método de filtrado colaborativo basado en usuarios utilizando el framework MapReduce para recomendar los mejores k elementos para cada usuario en un conjunto de datos dado usando Amazon Web Services como proveedor cloud. De igual forma en (Blake & Nowlan, 2007) se presenta un SR de servicios web que descubre y administra proactivamente servicios web. Señala que, aunque la computación orientada a servicios (SOC) permite a las organizaciones y usuarios individuales descubrir capacidades accesibles de forma abierta realizadas como servicios en Internet, los registros de servicios pueden ser muy grandes y esto puede evitar que las organizaciones descubran servicios en tiempo real, basándose en un entorno en Cloud y Big Data.

Como se puede tomar en cuenta, hoy en día existen algunas investigaciones enfocadas en los SR para entornos Big Data en Cloud, sin embargo, no se toma mucho en cuenta la creación de una infraestructura en nube que soporte todo un proceso de Ingeniería de Datos, para la creación del modelo de recomendación. Por lo cual, en este proyecto se aborda la creación de un SR que está bajo una infraestructura de Big Data en Cloud, el cual a su vez soporta un proceso de Ingeniería de Datos de forma continua, así como también su despliegue por medio de un API que soporte Big Data. Todo ello con el objetivo final de demostrar la potencialidad de contar una arquitectura Big Data en nube.

Por tanto, en la siguiente sección, se presenta el desarrollo de este sistema de recomendación en la cual se da a conocer la propuesta de la arquitectura desde un alto nivel hasta llegar a la explicación del modelo de recomendación y como se vincula dicho modelo con un entorno Big Data en Cloud.

# 3. METODOLOGÍA

En esta sección se abordará la metodología que se usó para desarrollar la infraestructura en AWS, así como también la arquitectura en donde se muestra el modelo de sistema de recomendación desplegado. De igual forma se detalla la metodología de la creación del SR desde su extracción de datos hasta su evaluación y recomendación.

## A. DISEÑO DE LA ARQUITECTURA

La arquitectura propuesta para el sistema de recomendación consta desde su implementación a través de un API que puede ser consumida por N. Esta primera arquitectura consta de un modelo de 3 capas compuesta por una capa de aplicación (Front-end), una capa de servicios (Back-end) y una capa de datos. En este proyecto solo se plantea la implementación de la API desde la capa de servicios y capa de datos, por lo que la capa de aplicación se la agrega como concepto práctico de una aplicación consumida por el usuario. Tal y como se aprecia en la figura 2, se presenta la primera arquitectura desde una vista de alto nivel, en el cual no se entra a detalle la arquitectura interna del API, sino se muestra la interacción de las capas de aplicación, servicio y datos.

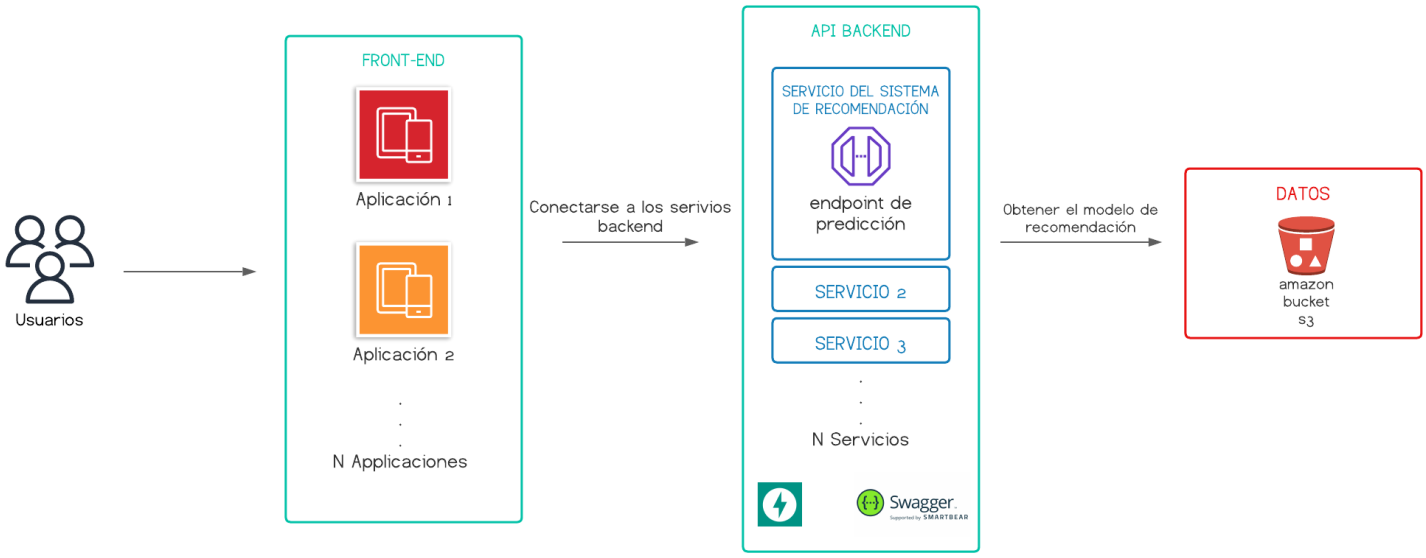


**Figura 2.** Arquitectura de 3 capas para la implementación del modelo de recomendación

Como se puede notar en la figura 2, se tienen 3 capas que se explican a continuación:

- ***Capa de aplicación (Front-end):*** Dicha capa, se encarga de la interacción directa con el usuario final que utilizara el producto, en este caso, puede ser cualquier aplicación que consuma el servicio del SR. Tal y como se mencionó anteriormente para este proyecto, dicha capa no se implementó ya que se dio importancia a la capa de servicio y su infraestructura interna. Por lo tanto, dicha capa representa simplemente el nivel tradicional de una infraestructura basada en n capas.
- ***Capa de servicios (Back-end):*** En esta capa, se implementa el servicio del SR el cual está levantando en un framework de Python llamado FAST API. Dicho servicio contiene los servicios web para realizar las recomendaciones correspondientes. El modelo de recomendación se obtiene a través de la capa de datos que es la encargada el almacenar el modelo del SR.
- ***Capa d de datos:*** En la presente capa, se tiene almacenado el modelo de recomendación generado. Esta capa está utilizando un servicio de AWS denominado **S3**. Dicho servicio es un sistema de almacenamiento distribuido que permite almacenar diferentes fuentes de información a través de contenedores de información llamados **buckets** Para este caso, el modelo del SR esta almacenado en un bucket del servicio S3 el cual será consumido por la capa de servicio para obtener el modelo de recomendación.

Tanto la capa de servicio que contiene el modelo de recomendación como el proceso para generar el SR requiere de un entorno de Big Data debido a la gran cantidad de procesamiento de información que se debe realizar. Por ello, en la figura 3, se presenta la infraestructura planteada en este proyecto para generar un proceso de ingeniería de datos que permite generar el SR como así también contener el API para llamar al modelo de recomendación.

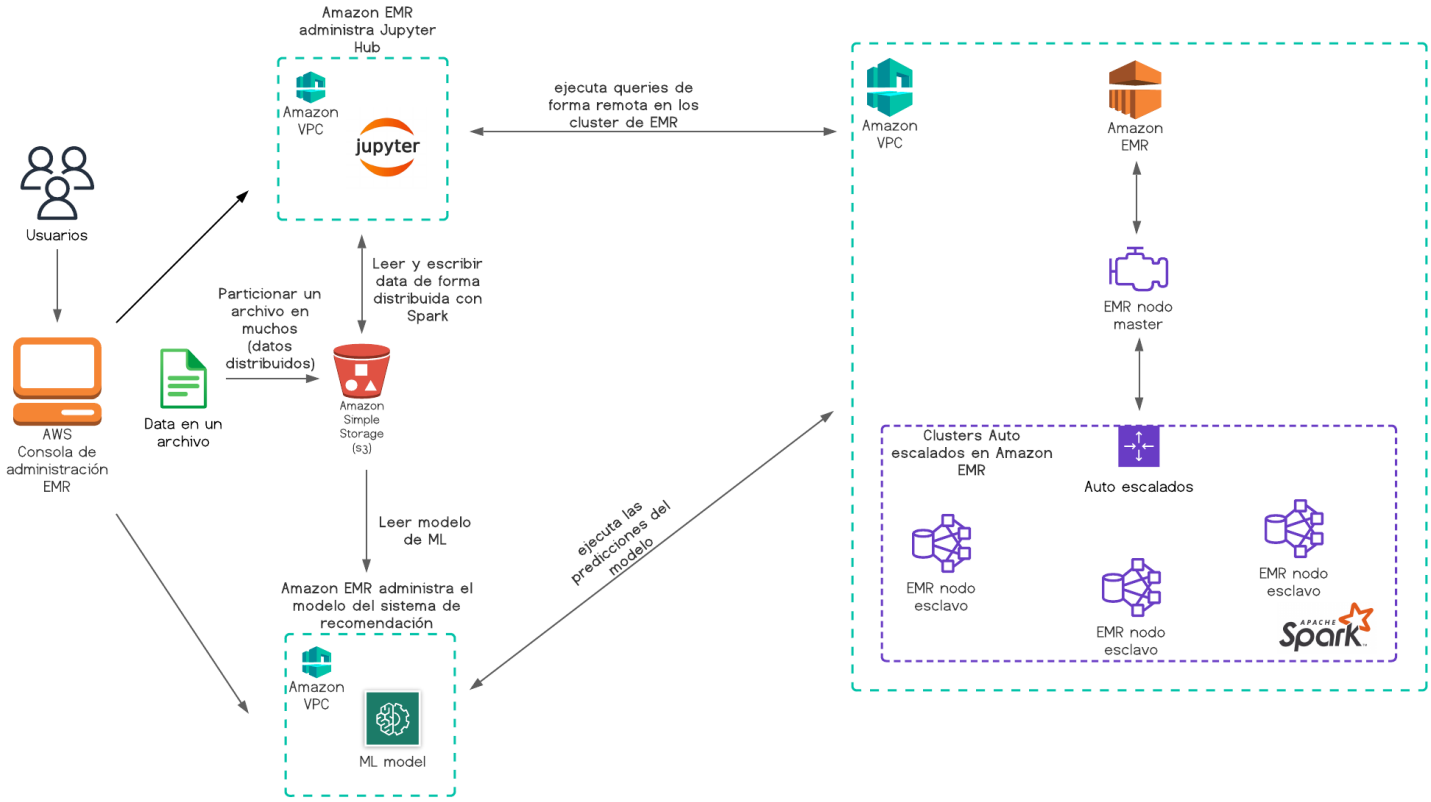


**Figura 3.** Infraestructura en AWS para la generación del SR utilizando Amazon EMR

Tal y como se puede apreciar en la figura 3, la arquitectura presenta a Amazon EMR como servicio central del resto de procesos. A continuación, se explicará cada una de las partes presentadas en la infraestructura.

- Es importante mencionar que todos los servicios que se encuentran en la arquitectura utilizan el servicio *Amazon Virtual Private Cloud* (VPC) el cual permite a los usuarios configurar una red virtual dentro de la arquitectura para todos los servicios puedan comunicarse. En este caso, se utilizó la VPC por defecto de AWS.
- Como primera parte de la infraestructura, se levanta el servicio de Amazon EMR mediante un nodo maestro y varios nodos esclavos que trabajan como cluster con una característica de auto-escalado, lo que permite que en caso de que se requiere más procesamiento, AWS automáticamente levante otro nodo para proveer mayor capacidad de procesamiento.
- A partir de ello, en la instancia principal de Amazon EMR se administra tanto el sistema del modelo de recomendación y la herramienta de desarrollo para

generar el modelo. Con EMR como gestor se asegura que todos los procesos que se ejecutan en esta infraestructura corren bajo un sistema distribuido, mejorando la capacidad de procesamiento de información.

- De igual forma, los servicios administrados por Amazon EMR se conectan a un bucket S3 que internamente juega un rol de sistema de datos distribuidos con la finalidad de procesar dicha información paralelamente. Dicho sistema de datos distribuidos se logra gracias a la partición exacta de un archivo de datos como se muestra en la infraestructura. Adicional a ello, dicho bucket también almacena el modelo de recomendación el cual será obtenido por el servicio que contiene la lógica para recomendar ítems a usuarios.
- Por último, toda esta administración se da por medio de una consola que Amazon AWS nos provee, mediante el cual se tiene la posibilidad de configurar mejor el servicio para desarrollar el modelo como el servicio por el cual se usa el modelo (API).

Con las arquitecturas explicadas, se procede a explicar la metodología y los puntos que se tomaron en consideración para desarrollar el SR.

## B. Diseño del modelo

El objetivo principal del marco de recomendación propuesto es abordar el problema de escalabilidad en los sistemas de recomendación actuales, por lo cual, se utilizan modelos de recomendación como ALS que evitan dicho problema. Para ello, en la figura 4, se explica de una forma más detallada como se realiza el proceso de recomendación.

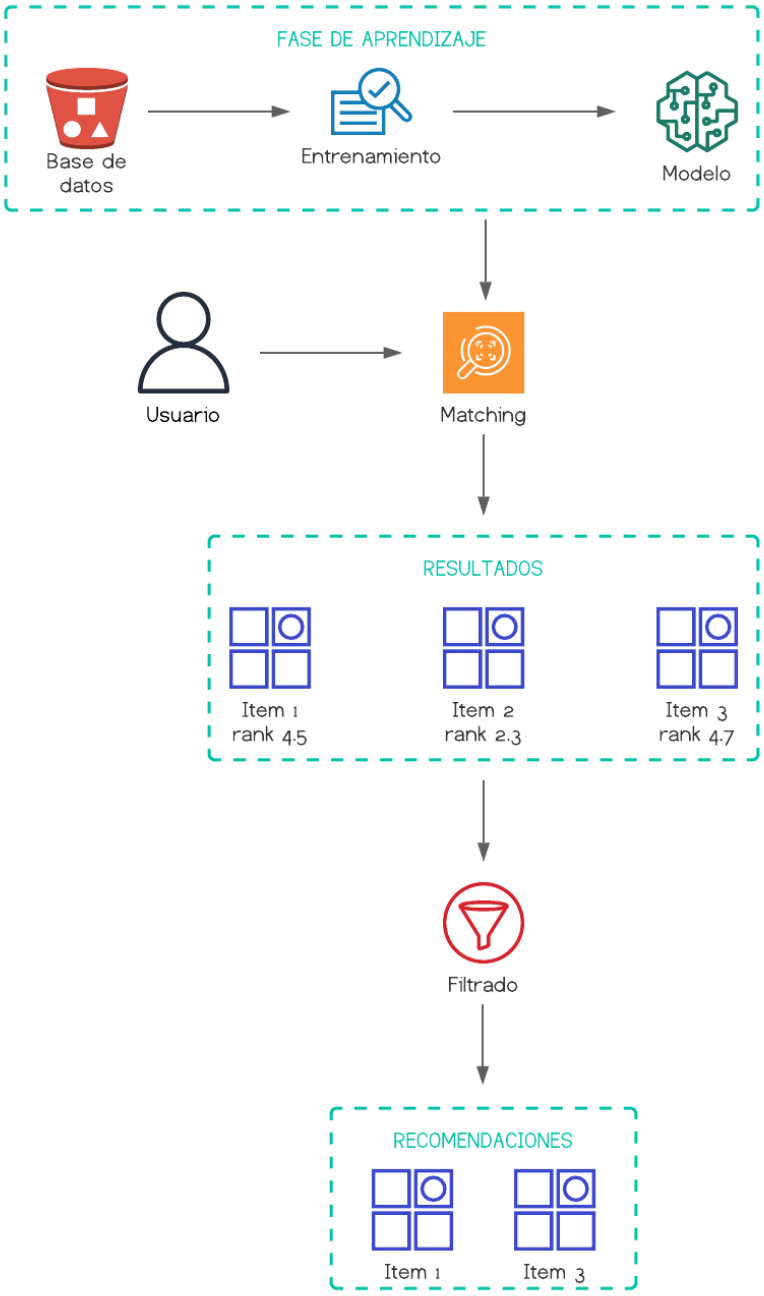


**Figura 4.** Representación del proceso de recomendación

Tal y como se presenta en la figura 4, en nuestro modelo de recomendación, dicho proceso se genera a partir de un modelo entrenado. A su vez, la recomendación comienza cuando el usuario solicita las recomendaciones. Una vez solicitado dicho proceso, se empieza a realizar la recomendación el modelo entrenado y mediante un filtrado, con el objetivo de obtener los ítems más relevantes para el usuario, generando así, recomendaciones significativas para el mismo.

Con el proceso de recomendación explicado, es importante mencionar el conjunto de pasos que siguieron para obtener el modelo de recomendación, tomando en cuenta la metodología CRISP-DM tal y como se había explicado anteriormente. El modelo propuesto consta de 4 pasos, los cuales se subdividen en otra seria de etapas, generando un total de 8 pasos para todo el desarrollo del modelo, empezando desde la extracción de información hasta la medición de tiempos de ejecución de los pasos anteriores. En la figura 5 se puede apreciar de mejor forma y de manera más detalla cada uno de los pasos que se tomaron en cuenta para desarrollar el modelo de recomendación.

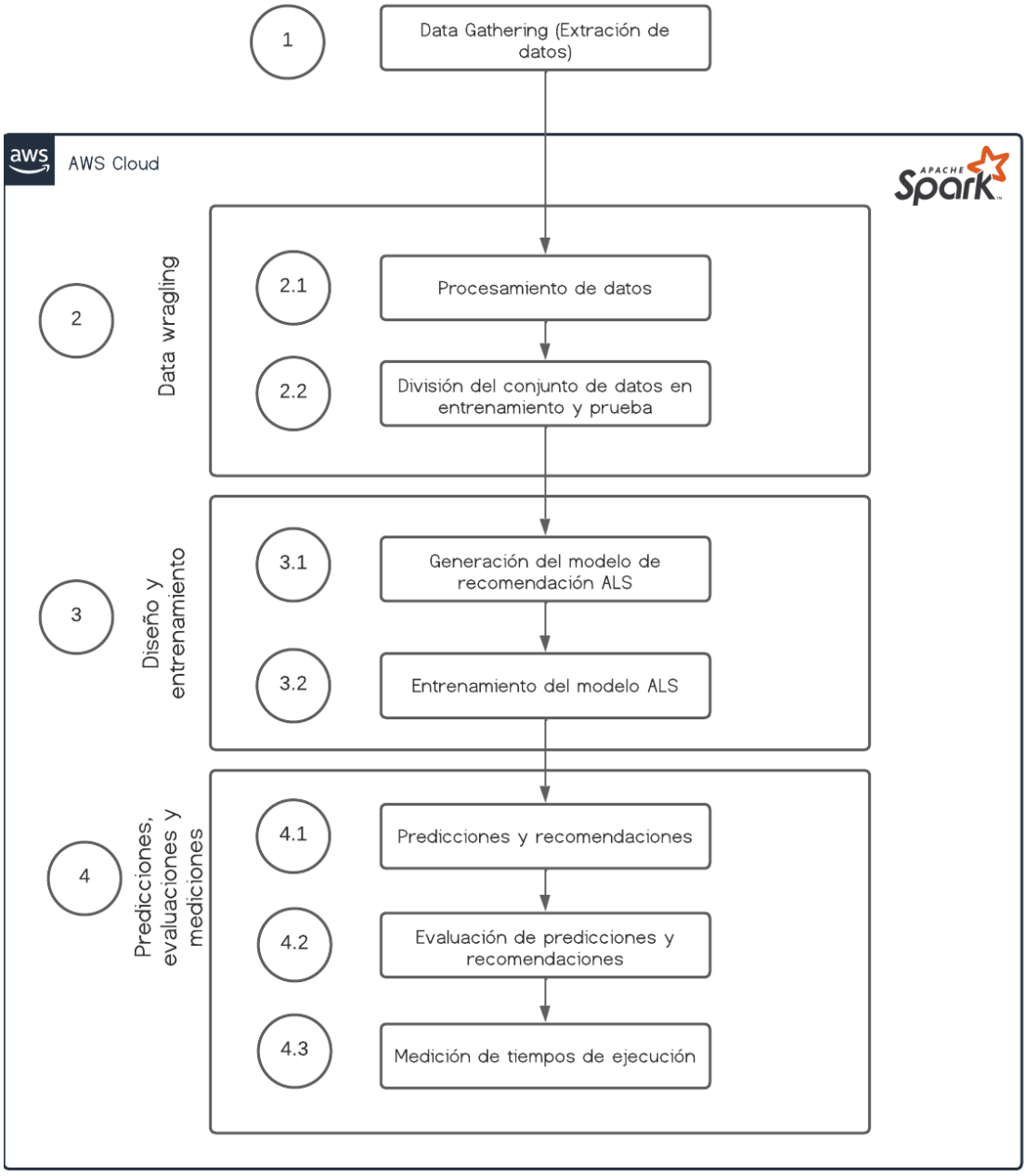


**Figura 5.** Arquitectura del modelo del SR basado en la metodología CRISP-DM

Tal y como se presenta en la figura 5, a partir de la fase 2 en adelante cada uno de los procesos esta desarrollado en la arquitectura de la figura 3, por lo que se procede a explicar cada uno de los pasos a continuación.

- ***Fase 1: Data Gathering (Extracción de datos)***

En dicha fase se realiza el proceso que se ilustra en el paso 1 de la figura 5 y es la extracción de datos. Para realizar dicho proceso, se extrae un conjunto de datos que nos permita trabajar con el modelo de recomendación ALS, en este caso se obtuvo un dataset que contenga información de usuarios, ítems y la valoración del usuario hacia dicho item. Con el conjunto de datos obtenido, se procede a realizar la siguiente fase del proceso de recomendación.

- ***Fase 2: Data Wrangling***

Dicha fase consta de 2 pases los cuales son: a) Procesamiento de datos y b) División del conjunto de datos en train y test. A continuación, se explica a detalle cada uno de estos pasos.

  - *Paso 2: Procesamiento de datos:* Como parte del desarrollo del modelo es importante asegurar que el conjunto de datos este estandarizado y limpio con el objetivo de que el modelo de recomendación no provoque malos resultados, por lo que se realizó un conjunto de pasos como, filtrar aquellos valores nulos, eliminar usuarios que no tengan un mínimo de votos realizados… entre otros pasos de limpieza más.
  - *Paso 3: División del conjunto de datos en entrenamiento y prueba:* Es importante recalcar que el modelo de recomendación no puede ser entrenado con todo el conjunto de datos ya que puede sufrir de overfitting (sobre entrenamiento) generando que el modelo solo se ajuste a los datos proporcionados. Por ello es necesario dividir el conjunto de datos en data de train y test y así obtener mejores resultados.

- ***Fase 3: Diseño y entrenamiento***

En esta fase, se debe tomar muy en cuenta ya que es la fase más importante de todas, puesto que se crea el modelo de recomendación y se lo entrena tomando en cuenta el conjunto de datos procesado. Dicha fase consta de dos pasos los cuales son: a) Generación del modelo de recomendación ALS y b) Entrenamiento del modelo ALS. Dichas secciones se explican a continuación.

  - *Paso 4: Generación del modelo de recomendación ALS:* En este paso se planea las características y parámetros que se utilizaran para generar el modelo ALS, tomando en consideración las columnas de usuario, item y rating.
  - *Paso 5: Entrenamiento del modelo ALS:* Con el modelo de recomendación creado y el conjunto de datos ya estandarizado y dividido, se procede a realizar el entrenamiento de datos correspondiente utilizando solamente los datos de entrenamiento para posteriormente en la siguiente fase realizar las predicciones y recomendaciones.

- ***Fase 4: Predicciones, evaluaciones y mediciones***

Una vez obtenido el modelo de recomendación y entrenado satisfactoriamente, se llega a esta última fase, la cual consta de realizar recomendaciones y evaluaciones a todo el modelo. Por tanto, en esta fase se realizan 3 pasos importantes que son: a) Predicciones y recomendaciones, b) Evaluación de predicciones y recomendaciones y c) Medición de tiempos de ejecución. Dichos pasos se explican a más detalle a continuación.

  - *Paso 6: Predicciones y recomendaciones:* Con el modelo de recomendación entrenado, se realizan las predicciones con el conjunto de datos de prueba y las recomendaciones a partir de nuevos conjuntos de datos. Con dichos resultados, se procede a evaluar el modelo de recomendación mediante métricas de calidad.
  - *Paso 7: Evaluación de predicciones y recomendaciones:* Con las recomendaciones y predicciones ya realizadas, en este paso se procede

a evaluar cada uno de dichos resultados. Para ello, se utiliza la métrica de calidad MAE que permite obtener el error de nuestro sistema en cuanto a recomendaciones se refiere.

- *Paso 8: Medición de tiempos de ejecución:* Una vez evaluado el modelo de recomendación y finalizado cada una de las fases y etapas anteriores, se procede a medir los tiempos de ejecución de cada etapa con el fin de comparar los tiempos de un modelo de recomendación enfocado a un entorno Big Data frente a un modelo de recomendación no enfocado a dicho entorno.

Una vez explicado cada una de las etapas del modelo de recomendación, así como también la arquitectura utilizada para implementar dicho modelo, se procede a explicar los experimentos y resultados obtenidos con la metodología propuesta.

# 4. Resultados y discusión

En la presente sección, se presentarán los experimentos planteados con sus respectivos resultados. Se tomará en cuenta el experimento realizado para el modelo de recomendación, sus métricas de calidad como así también el conjunto de datos que se utilizó para realizar las pruebas. Adicionalmente, se da a conocer los resultados de la infraestructura planteada en la nube, en cuanto a tiempos de ejecución se refiere.

## A. Experimentos establecidos

Para el desarrollo del método propuesto, se utilizó el dataset público de Netflix. Dicho conjunto de datos está compuesto por usuarios que realizan una valoración a un conjunto de ítems (películas) en un tiempo dado. El dataset de Netflix es el conjunto de datos adecuado debido a la gran cantidad de información que contiene, lo cual lo hace ideal para trabajar en un entorno Big Data y adicional a ello, se acopla muy bien con el modelo de recomendación a realizar. En la tabla 1 se encuentra a más detalla la información referente al dataset.

**Tabla 1.** Propiedades del dataset usado para el modelo de recomendación

| Dataset | Total de Datos | Número de ítems | Número de usuarios | Rango de valoración |
|---|---|---|---|---|
| Netflix | 100,000,000 | 17,770 | 480,189 | Escala de 5 valores |

Con la información de la tabla 1, se procede a realizar un procesamiento de datos, con la finalidad de limpiar la información del conjunto de datos y estandarizarlo para posteriormente enviarlo al modelo de recomendación. Con ello se dividió el conjunto de datos en entrenamiento y prueba. Tal y como se aprecia en la tabla 2, se indica la cantidad de datos que quedaron para cada partición.

**Tabla 2.** Principales parámetros del método propuesto

| Dataset | Datos de entrenamiento (80%) | Datos de prueba (20%) |
|---|---|---|
| Netflix | 3,710,871 | 926,829 |

Posterior a ello, se diseña el modelo de recomendación ALS tomando en cuenta la cantidad de valores obtenidos en la tabla 2. Para dicho modelo, es muy importante tomar en cuenta parámetros como la estrategia de recomendación, número de factores latentes… entre otros. Con dichas características en consideración se obtuvo un modelo que contiene los siguientes parámetros, tal y como se muestra en la tabla 3.

**Tabla 3**. Parámetros del modelo de recomendación ALS

| Modelo | Estrategia | Número de factores latentes | Valor de regularización |
|---|---|---|---|
| ALS | Drop | 8 | 0.1 |

Como se presenta en la tabla 3, el modelo ALS usa una estrategia denominada Drop que permite eliminar el problema de arranque frio en el modelo en caso de existir datos faltantes. Adicional a ello, se utiliza un parámetro de regularización para evitar el overfitting. Por último, se configura el número de factores latentes que permitirá crear las matrices de factorización.

A nivel de arquitectura, tal y como se mencionó anteriormente, se propuso usar AWS como proveedor de servicios principal, en donde se utilizó Amazon EMR como entorno de Big Data, mediante el cual, se configuro la cantidad de nodos maestros y esclavos. De igual forma, los datos con los que se desarrolló el modelo se encuentran distribuidos en distintos archivos que en el proceso de procesamiento se los juntan para trabajar con ellos. Tomando en cuenta dichas consideraciones, en la tabla 4 se presenta las características de la infraestructura.

**Tabla 4.** Características de la infraestructura en AWS

| Servicio | Características | Valor |
|---|---|---|
| **Instancia EC2** | Tipo | M5.xlarge (enfocadas a Big Data) |
| | RAM | 16 GB |
| | VCPU | 4 |
| **Amazon EMR** | Número de nodos maestros | 1 |
| | Número mínimo de clusters | 2 |
| | Número máximo de clusters | 4 |
| **Bucket S3** | Número de buckets | 1 |
| | Número de archivos distribuidos | 6 |

Por otro lado, para comparar la arquitectura propuesta en este trabajo, se implementó un entorno tradicional el cual es un computador personal de alto rendimiento. Adicional a ello dicho entorno no posee un sistema distribuido lo cual genera que los procesos que se ejecuten en este entorno sean procesados de forma estructural. A continuación, en la tabla 5 se puede apreciar de mejor forma las características de este entorno.

**Tabla 5.** Características del entorno tradicional

| Características | Valor |
|---|---|
| **Procesador** | Core i9-10850K |
| **RAM** | 32 GB |
| **Disco** | 1 TB SSD |

Con los experimentos explicados, es necesario medir el rendimiento del modelo de recomendación a través de métricas de calidad, las cuales nos permiten conocer que tan bueno es nuestro modelo de recomendación. Por otro lado, se medirán los tiempos de ejecución del desarrollo del SR en cuanto a lectura y procesamiento de datos se refiere y a su vez se los comparará contra un SR cuya infraestructura no está soportada en un ambiente de datos distribuido y tampoco trabaja bajo un entorno Big Data.

## B. Medidas de calidad

Con el objetivo de medir la calidad de recomendaciones de nuestro modelo de recomendación, utilizamos las métricas de calidad, las cuales nos permiten conocer el error de nuestro sistema y que tan efectivo es el mismo frente al conjunto de datos. Para este trabajo, se utilizó la métrica denominada Mean Absolute Error (MAE) en el cual tal y como indica (Hurtado Ortiz, 2020) permite medir y conocer el error de las recomendaciones obtenidas entre el conjunto de datos de prueba y predicción. La fórmula del MAE está dada de la siguiente forma.

$$MAE = \frac{\sum_{i=1}^{n} |y_i - y_{pred_i}|}{n}$$

Donde:

- $y_i$: Valoración real del usuario hacia el item
- $y_{pred_i}$: Predicción de la valoración del usuario hacia el item
- $n$: Conjunto de datos de prueba

Por otro lado, para medir los tiempos de ejecución de los procesos de lectura y procesamiento de información se utilizará la medida de tiempo en segundos, puesto que es una medida estándar para ejecución de procesos. Con ello se busca, medir dichos tiempos de ejecución para el proceso orientado a un entorno Big Data, frente a un entorno tradicional.

# C. Resultados

En este apartado, presentamos los resultados obtenidos por cada uno de los experimentos realizados. Empezando con el modelo de recomendación con cada uno de los parámetros configurados conforme a la tabla 3. Adicional a ello, se realizó una comparación de resultados del modelo de recomendación generado, frente a las técnicas tradicionales de los sistemas de recomendación. En la tabla 6, se puede apreciar dicha comparación de mejor forma.

**Tabla 6.** Resultado del modelo de recomendación

| Modelo | MAE |
|---|---|
| ALS | 0.8696 |
| KNN JMSD (Bobadilla et al., 2012) | 0.7421 |
| PMF (Salakhutdinov & Mnih, 2009) | 0.6842 |
| NMF (Lee & Seung, 1999) | 1.2412 |

Con ello, dicho resultado es apropiado acorde a la configuración realizada, tomando en cuenta que no se realizó ninguna optimización al modelo a comparación de dichos modelos tradicionales que se presentan en la tabla 6, los cuales cuentan con un proceso de optimización de parámetros. Adicional a ello, es importante recalcar que el uso de un entorno Big Data permite obtener resultados iniciales favorables para el modelo de recomendación. Por otro lado, para darle fuerza a dicho resultado, es importante tomar en cuenta el tiempo de procesamiento que se necesita para llegar a dicho resultado. Por ello, a continuación, en la figura 6, se muestra una comparación del tiempo total de lectura y procesamiento en el modelo de recomendación en un entorno Big Data y un entorno tradicional el cual fue presentado en la tabla 5. Todos estos resultados de comparación están bajo un entorno Big data con 2 cluster y un entorno tradicional trabajando con todos sus recursos disponibles y a un Core de procesamiento.

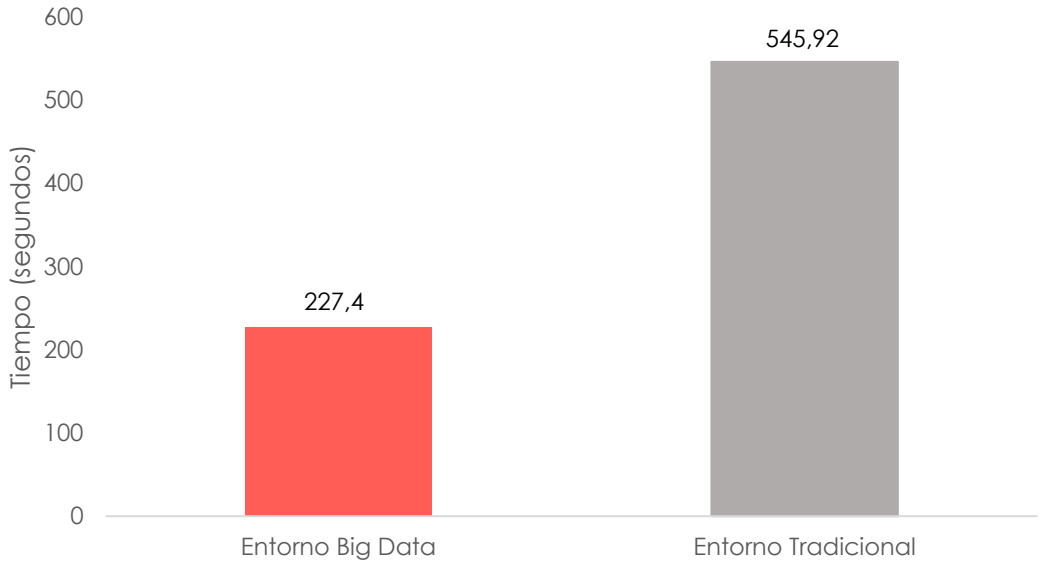


**Figura 6.** Tiempo total de lectura y procesamiento de datos en diferentes entornos

Tal y como podemos notar en la figura 6, la diferencia de tiempos entre un entorno Big Data y un entorno tradicional es muy grande, el tiempo de lectura y procesamiento en Big Data es muy superior debido a su sistema distribuido, es decir, puede ejecutar una misma tarea en varios cluster o nodos trabajadores, dividiendo el trabajo, logrando así que la tarea termine mucho más rápida que en un entorno tradicional, puesto que dicho entorno trabaja de forma estructural, generando que los procesos tarden mucho más. Para más detalle, en la figura 7, podemos ver los tiempos de lectura y procesamiento de forma separada.

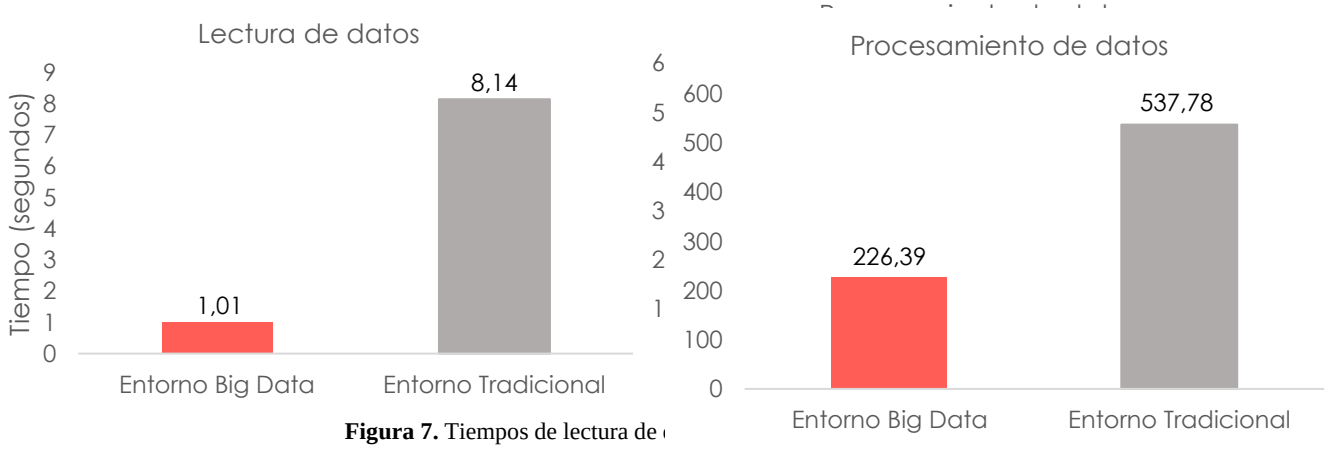


**Figura 7.** Tiempos de lectura de
diferentes

Como podemos notar en la figura 7, existe una gran diferencia en cuanto a lectura de datos se refiere, podríamos decir que la diferencia de lectura de un entorno Big Data frente a un entorno tradicional es **8x** veces más rápido ya que el entorno tradicional realiza una lectura secuencial, mientras que el entorno Big Data realiza una lectura de forma distribuida. De igual forma, en cuanto a procesamiento, de igual forma existe una diferencia muy grande, debido a lo explicado anteriormente en el proceso de lectura. De igual forma es muy importante tomar en cuenta el tiempo que toma entrenar el modelo de recomendación, ya que uno de los procesos más demorados, es el entrenamiento del modelo, por ello de igual forma en la figura 8, se muestra la diferencia de tiempo existente entre dichos entornos anteriormente mencionados y cómo se comportan al momento de entrenar el modelo de recomendación.

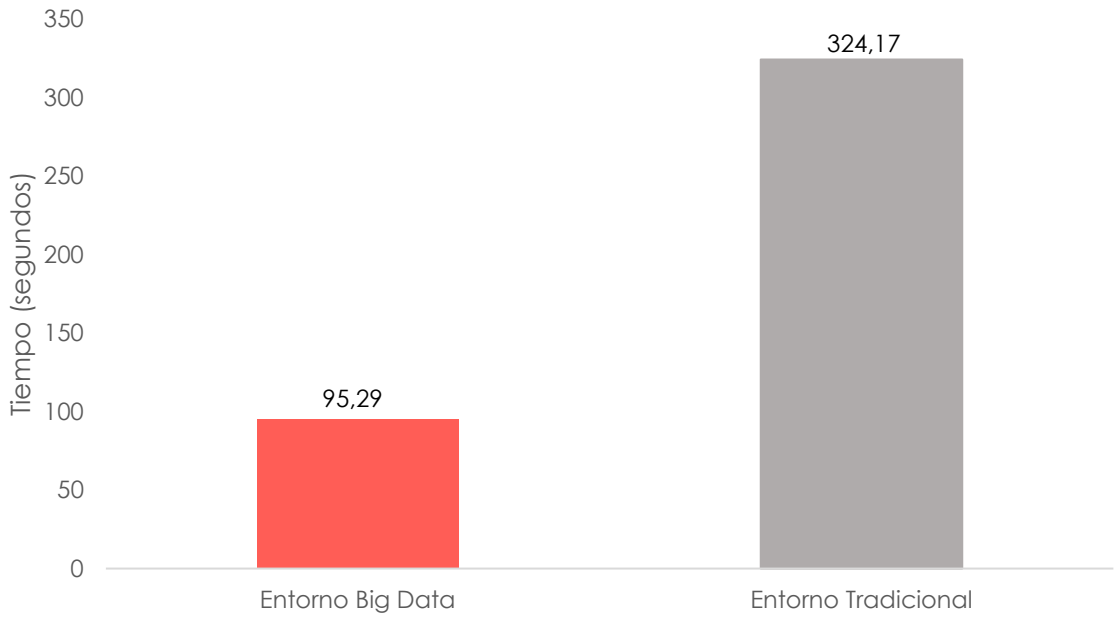


**Figura 8.** Tiempo total de entrenamiento del modelo de recomendación

Tal y como se aprecia en la figura 8, de igual forma, existe una gran diferencia en el tiempo que requiere el modelo de recomendación para ser entrenado, para este caso, el modelo ALS al ser un modelo que trabaja con un gran conjunto de datos, está más enfocado a un entorno Big Data, por lo que en un entorno tradicional, dicho modelo no entrena de forma eficiente.

Por otro lado, a parte de los resultados obtenidos en cuanto a tiempos usando diferentes entornos, es importante mostrar también los resultados obtenidos usando el entorno Big Data frente a diferentes cantidades de cluster, ya que como se mencionó en la tabla 4,

actualmente el entorno trabaja con un mínimo de 2 cluster y un máximo de 4, por lo cual es interesante conocer la diferencia existente entre usar más o menos clusters. Para ello, en la figura 9 se muestra el resultado de tiempos en cada proceso para las cantidades de clusters comentadas.

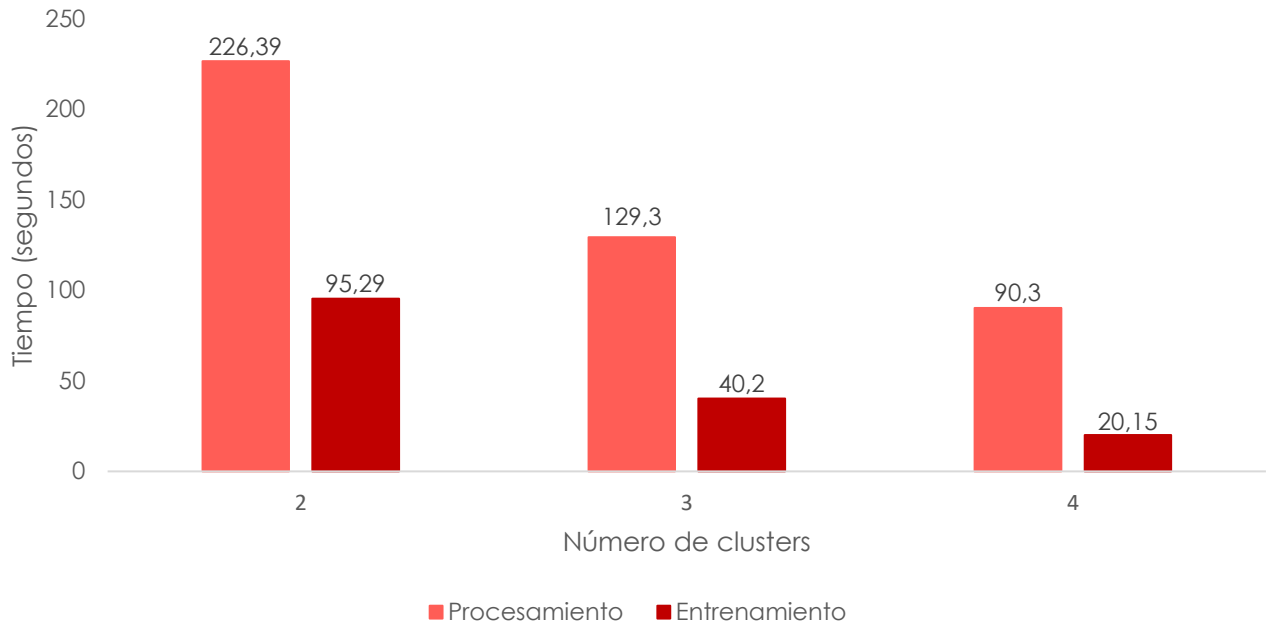


**Figura 9.** Tiempos en los diferentes procesos del modelo de recomendación usando diferentes cluster en la infraestructura

Tal y como se puede apreciar en la figura 9, a medida que se incrementa el número de clusters en la infraestructura del Amazon EMR, el tiempo de procesamiento de datos y el entrenamiento del modelo de recomendación baja considerablemente, por lo que es importante acoplarse a las necesidades de los datos.

Por último, es importante recalcar que los servicios de AWS requieren un costo para ser usados, en este caso es importante tomar en cuenta que cada servicio tiene un costo diferente por tiempo de uso. Por ello, en la tabla 7 se presenta los costos relacionados al uso de los servicios de AWS.

**Tabla 7.** Costos de los servicios utilizados en Amazon Web Services

| Servicios | Costo | Tipo de precio |
|---|---|---|
| Instancia EC2 | $ 0.192 | Tarifa por hora bajo demanda |
| Amazon EMR | $ 0.14 | Tarifa por hora bajo demanda |
| Bitbucket S3 | Gratuito | - |

Tal y como se puede apreciar en la tabla 7, podemos notar como el servicio de S3 para este proyecto fue gratuito ya que AWS nos provee un uso gratuito de hasta 5G, por lo que no fue necesario utilizar más para este caso. Por otro lado, podemos notar que los precios que se indican en la tabla 7 se basan en una tarifa base por hora.

# 5. Conclusiones

Los resultados y experimentos presentados en este proyecto confirman las hipótesis antes planteadas: Los entornos Big Data proporcionan mejoras significativas en términos de alto procesamiento, facilidad de implementación y tiempos de ejecución frente a entornos tradicionales. El presente trabajo dio a conocer los diferentes servicios en la nube que ofrece Amazon Web Services, así como también la ventaja de tener un alto nivel de procesamiento en cloud sin la necesidad de realizar una configuración compleja desde el inicio. Por otro lado, el modelo planteado mediante este proyecto nos da a entender las ventajas de usar un sistema de recomendación enfocado a una gran cantidad de información, en este caso, el modelo ALS se comportó de forma correcta frente al conjunto de datos proporcionado y el desarrollo planteado. En conclusión, los sistemas de recomendación son de gran interés científico, sin embargo, al momento de realizar una implementación y despliegue en producción, es importante tomar en consideración el uso de modelos escalables, que permitan trabajar con una cantidad masiva de datos. Por ello, Este trabajo sienta las bases para aquellos que deseen implementar modelos de sistemas de recomendación en un entorno de Big Data. Su relevancia se extiende a arquitectos de software e ingenieros de datos, ya que proporciona información valiosa sobre las ventajas y desafíos asociados con este tipo de implementaciones. A su vez, dicho trabajo se deja como una línea base para futuras investigaciones, en las que el modelo planteado actualmente puede ser mejorado introduciendo aprendizaje profundo, técnicas bayesianas, métodos combinados y otras técnicas de aprendizaje automático moderno. Por otro lado, a nivel de infraestructura se podría utilizar distintas bases de datos en nube relaciones o no relaciones con el objetivo de almacenar información en otras fuentes de almacenamiento de datos. De igual forma, utilizar otro tipo de instancias EC2 para levantar el entorno de Big Data permitiría obtener mejores resultados a nivel de tiempo, resultados del modelo, entre otros más.

# Referencias


Aljunid, M. F., & Manjaiah, D. H. (2019). Movie Recommender System Based on Collaborative Filtering Using Apache Spark. In *Advances in Intelligent Systems and Computing* (Vol. 839, pp. 283–295). Springer Verlag. https://doi.org/10.1007/978-981-13-1274-8_22

Blake, M. B., & Nowlan, M. F. (2007). *A Web Service Recommender System Using Enhanced Syntactical Matching*.

Bobadilla, J., Hernando, A., Ortega, F., & Gutiérrez, A. (2012). Collaborative filtering based on significances. *Information Sciences*, *185*(1), 1–17. https://doi.org/10.1016/j.ins.2011.09.014

Bobadilla, J., Serradilla, F., & Bernal, J. (2010). A new collaborative filtering metric that improves the behavior of recommender systems. *Knowledge-Based Systems*, *23*(6), 520–528. https://doi.org/10.1016/j.knosys.2010.03.009

Chen, J., Fang, J., Liu, W., Tang, T., Chen, X., & Yang, C. (2017). Efficient and portable ALS matrix factorization for recommender systems. *Proceedings - 2017 IEEE 31st International Parallel and Distributed Processing Symposium Workshops, IPDPSW 2017*, 409–418. https://doi.org/10.1109/IPDPSW.2017.91

Chen, L., Rui, L., Yige, L., Ruixuan, Z., & Myung-kyung, D. (2017). *Machine Learning-based Product Recommendation using Apache Spark*.

Gu, L., & Li, H. (2014). Memory or time: Performance evaluation for iterative operation on hadoop and spark. *Proceedings - 2013 IEEE International Conference on High Performance Computing and Communications, HPCC 2013 and 2013 IEEE International Conference on Embedded and Ubiquitous Computing, EUC 2013*, 721–727. https://doi.org/10.1109/HPCC.and.EUC.2013.106

Guan, X., Li, C. T., & Guan, Y. (2017). Matrix Factorization with Rating Completion: An Enhanced SVD Model for Collaborative Filtering Recommender Systems. *IEEE Access*, *5*, 27668–27678. https://doi.org/10.1109/ACCESS.2017.2772226

Huber, S., Wiemer, H., Schneider, D., & Ihlenfeldt, S. (2019). DMME: Data mining methodology for engineering applications - A holistic extension to the CRISP-DM model. *Procedia CIRP*, *79*, 403–408. https://doi.org/10.1016/j.procir.2019.02.106

Hurtado Ortiz, R. (2020). *Recomendación a grupos de usuarios usando el concepto de singularidades*. http://oa.upm.es/58148/1/REMIGIO_ISMAEL_HURTADO_ORTIZ.pdf

Hurtado, R. (2020). *Recomendación a grupos de usuarios usando el concepto de singularidades*.

Hurtado, R., Bobadilla, J., Bojorque, R., Ortega, F., & Li, X. (2019). A new recommendation approach based on probabilistic soft clustering methods: A scientific documentation case study. *IEEE Access*, *7*, 7522–7534. https://doi.org/10.1109/ACCESS.2018.2890079

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*(6755), 788–791. https://doi.org/10.1038/44565

Manakkadu, S., Joshi, S. P., Halverson, T., & Dutta, S. (n.d.). Top-k_User-Based_Collaborative_Recommendation_System_Using_MapReduce. *2021*.

McNee, S. M., Riedl, J., & Konstan, J. A. (2006). Being accurate is not enough: How accuracy metrics have hurt recommender systems. *Conference on Human Factors*

*in Computing Systems - Proceedings*, 1097–1101. https://doi.org/10.1145/1125451.1125659

Moreno, B. D. V., Ortiz, R. I. H., & Rivera, D. A. M. (2019). A new approach hybrid recommender system of item bundles for group of users. *2019 IEEE International Autumn Meeting on Power, Electronics and Computing, ROPEC 2019, Ropec*. https://doi.org/10.1109/ROPEC48299.2019.9057121

Panigrahi, S., Lenka, R. K., & Stitipragyan, A. (2016). A Hybrid Distributed Collaborative Filtering Recommender Engine Using Apache Spark. *Procedia Computer Science*, *83*, 1000–1006. https://doi.org/10.1016/j.procs.2016.04.214

Raj Bala, D. S. K. J. D. W. M. A. B. (2022, October 19). *Magic Quadrant for Cloud Infrastructure and Platform Services*.

Said, A., & Bellogín, A. (2014). Comparative recommender system evaluation: Benchmarking recommendation frameworks. *RecSys 2014 - Proceedings of the 8th ACM Conference on Recommender Systems*, 129–136. https://doi.org/10.1145/2645710.2645746

Salakhutdinov, R., & Mnih, A. (2009). Probabilistic matrix factorization. *Advances in Neural Information Processing Systems 20 - Proceedings of the 2007 Conference*, 1–8.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2002). *Incremental Singular Value Decomposition Algorithms for Highly Scalable Recommender Systems*. www.movielens.umn.edu

Schell, R. (2013). *Security – A Big Question for Big Data*. IEEE.

Sharma, N., & Shamkuwar, M. (2019). Big Data Analysis in Cloud and Machine Learning. In *Studies in Big Data* (Vol. 43, pp. 51–85). Springer Science and Business Media Deutschland GmbH. https://doi.org/10.1007/978-981-13-0550-4_3

Subasish, G., Nazmun, N., Mohammad, A. W., Munmun, B., Mohammad Shahadat, H., & Karl, A. (2021). *Recommendation System for E-commerce Using Alternating Least Squares (ALS) on Apache Spark*. https://doi.org/10.1007/978-3-030-68154-8_75

Subramaniyaswamy, V., & Logesh, R. (2017). Adaptive KNN based Recommender System through Mining of User Preferences. *Wireless Personal Communications*, *97*(2), 2229–2247. https://doi.org/10.1007/s11277-017-4605-5

Web Services, A. (2015). *Amazon Elastic MapReduce Developer Guide*.

Yang, Z., Wu, B., Zheng, K., Wang, X., & Lei, L. (2016). A survey of collaborative filtering-based recommender systems for mobile internet applications. *IEEE Access*, *4*(c), 3273–3287. https://doi.org/10.1109/ACCESS.2016.2573314

Yehuda, K., Robert, B., & Chris, V. (2009). *MATRIX FACTORIZATION TECHNIQUES FOR RECOMMENDER SYSTEMS*. https://doi.org/10.1109/mc.2009.263

Zhu, B. O., & Hurtado, R. (2018). *An Efficient Recommender System Method Based on the Numerical Relevances and the Non-Numerical Structures of the Ratings*. 49935–49954. https://doi.org/10.1109/ACCESS.2018.2868464