

## Resumen

El presente libro expone un estudio realizado en una empresa de comida rápida en la ciudad de Guayaquil. Su objetivo general fue proponer la implementación de un sistema de recomendación de un filtro colaborativo basado en el algoritmo ALS, que permitiera analizar las predicciones de los siguientes productos que el cliente desee adquirir, lo que ayuda a comprender al cliente y a personalizar al máximo sus opciones de consumo. Se utilizó, además, la herramienta RapidMiner que, a través de su metodología, posibilitó la identificación de la puntuación de las particularidades de los productos dentro de la empresa. La investigación desarrollada es descriptiva y se asumió el enfoque cualitativo, a partir del cual se interpretaron eficientemente los resultados obtenidos durante todas las etapas investigativas, hasta lograr todos los objetivos planteados.



**Sergio Israel Peña Guano.** Magíster en Sistemas de información, mención en Inteligencia de Negocios. Ingeniero en Networking y Telecomunicaciones. Docente de la Facultad de Ciencias Matemáticas y Físicas de la Universidad de Guayaquil, Ecuador. Autor de varios artículos científicos e investigaciones en el área de las Telecomunicaciones.

Email: [pe.sergioi@gmail.com](mailto:pe.sergioi@gmail.com)

<http://orcid.org/0000-0003-4021-1892>



**William Rafael Raymondi Lomas.** Magíster en Sistemas de información, mención en Inteligencia de Negocios. Ingeniero en Networking y Telecomunicaciones. Docente de la Facultad de Ciencias Matemáticas y Físicas de la Universidad de Guayaquil, Ecuador. Autor de varios artículos científicos e investigaciones en el área de las Telecomunicaciones.

Email: [william.raymondil@ug.edu.ec](mailto:william.raymondil@ug.edu.ec)

<http://orcid.org/0000-0003-1641-6193>



9 780311 000425



9 780311 000425

Análisis de las predicciones en un filtro colaborativo basado en el algoritmo ALS  
para una empresa de comida rápida en la ciudad de Guayaquil



Análisis de las predicciones en un filtro colaborativo  
basado en el algoritmo ALS para una  
empresa de comida rápida en la ciudad de Guayaquil

**Sergio Israel Peña Guano**  
**William Rafael Raymondi Lomas**

## Análisis de las predicciones en un filtro colaborativo basado en el algoritmo ALS para una empresa de comida rápida en la ciudad de Guayaquil

**Diseño:** Ing. Erik Marino Santos Pérez.

**Traducción:** Prof. Dr. C. Ernan Santiesteban Naranjo.

**Corrección de estilo:** Prof. Dra. C. Kenia María Velázquez Avila.

**Diagramación:** Prof. Dr. C. Ernan Santiesteban Naranjo.

**Director de Colección Tecnología:** Prof. Dr. C. Wilber Ortiz Aguilar.

**Jefe de edición:** Prof. Dra. C. Kenia María Velázquez Avila.

**Dirección general:** Prof. Dr. C. Ernan Santiesteban Naranjo.

© Sergio Israel Peña Guano

William Rafael Raymondi Lomas

### Sobre la presente edición:

#### Primera edición

Esta obra ha sido evaluada por pares académicos a doble ciegos

**Lectores/Pares académicos/Revisores:** 0001 & 0006

#### Editorial Tecnocientífica Americana

**Domicilio legal:** calle 613nw 15th, en Amarillo, Texas. ZIP: 79104

Estados Unidos de América, 16 de enero de 2023

**Teléfono:** 7867769991

**Código BIC:** UMB

**Código EAN:** 9780311000425

**Código UPC:** 978031100042

**ISBN:** 978-0-3110-0042-5

La Editorial Tecnocientífica Americana se encuentra indizada en, referenciada en o tiene convenios con, entre otras, las siguientes bases de datos:





## Contenido

<b>Prólogo</b> .....	1
<b>Capítulo 1. Introducción a los sistemas de recomendación</b> .....	2
<b>1.1. Descripción del problema</b> .....	11
<b>1.1.1. Pregunta global</b> .....	11
<b>1.1.2. Preguntas específicas</b> .....	11
<b>1.2. Objetivos</b> .....	12
<b>1.2.1. Objetivo general</b> .....	12
<b>1.2.2. Objetivos específicos</b> .....	12
<b>1.3. Justificación</b> .....	13
<b>1.3.1. Justificación teórica</b> .....	13
<b>1.3.2. Justificación metodológica</b> .....	13
<b>1.3.3. Justificación práctica</b> .....	14
<b>Capítulo 2. Fundamentación teórica</b> .....	16
<b>2.1. Técnicas de filtro colaborativo</b> .....	16
<b>2.2. Filtrado colaborativo basado en memoria</b> .....	17
<b>2.3. Evolución de los filtrados colaborativos</b> .....	19
<b>2.3.1. Desafíos del filtrado colaborativo</b> .....	22
<b>2.4. Filtrado colaborativo y Web adaptable</b> .....	26
<b>2.5. Usos del filtrado colaborativo</b> .....	27
<b>2.5.1. Tareas del usuario</b> .....	27
<b>2.6. Funcionalidad del sistema de filtrado colaborativo</b> .....	29
<b>2.7. Propiedades de los dominios adecuados para el filtrado colaborativo</b> .....	31
<b>2.8. Persistencia de datos</b> .....	34
<b>2.9. Algoritmos de filtrado colaborativo: teoría y práctica</b> .....	36
<b>2.10. Algoritmos no probabilísticos</b> .....	39
<b>2.11. Algoritmo ALS</b> .....	40
<b>2.12. Filtrado colaborativo basado en modelos</b> .....	46
<b>2.13. Sistemas de recomendación basados en filtrado colaborativo</b> .....	47
<b>Capítulo 3. Marco teórico</b> .....	52





<b>3.1. Enfoque de investigación.....</b>	<b>52</b>
<b>3.2. Tipo de investigación .....</b>	<b>52</b>
<b>3.3. Diseño narrativo .....</b>	<b>53</b>
<b>3.4. Sistemas de recomendación .....</b>	<b>53</b>
<b>3.5. Recolección de datos.....</b>	<b>54</b>
<b>3.6. Sistemas de recomendación basado en filtro colaborativo .....</b>	<b>55</b>
<b>3.6.1. Enfoque del algoritmo de filtro colaborativo .....</b>	<b>56</b>
<b>3.6.2. Enfoque de los algoritmos basados en memoria .....</b>	<b>56</b>
<b>3.7. Métrica de similaridad .....</b>	<b>58</b>
<b>3.7.1. Coeficiente de correlación de Pearson .....</b>	<b>59</b>
<b>3.7.2. Distancia euclidiana.....</b>	<b>61</b>
<b>3.8. Metodología.....</b>	<b>62</b>
<b>3.8.1. Metodología KDD.....</b>	<b>62</b>
<b>3.8.2. Desarrollo de las fases KDD.....</b>	<b>64</b>
<b>3.9. Dataset.....</b>	<b>69</b>
<b>3.10. Algoritmo <i>Recommender Documentation</i> .....</b>	<b>70</b>
<b>3.11. Plan de general del proyecto.....</b>	<b>71</b>
<b>Capítulo 4. Análisis de los resultados .....</b>	<b>74</b>
<b>4.1. Población y muestra .....</b>	<b>74</b>
<b>4.2. Resultados.....</b>	<b>74</b>
<b>4.3. Análisis de predicciones .....</b>	<b>78</b>
<b>4.4. Conjunto de datos .....</b>	<b>80</b>
<b>4.5. Métricas de evaluación.....</b>	<b>81</b>
<b>4.6. Porcentaje de Prueba y Teste .....</b>	<b>81</b>
<b>4.7. Métrica de similaridad .....</b>	<b>82</b>
<b>4.8. Tamaño de la vecindad .....</b>	<b>82</b>
<b>4.9. Pruebas y resultados.....</b>	<b>84</b>
<b>4.9.1. Ejecución de las pruebas.....</b>	<b>84</b>
<b>4.9.2. Análisis de los resultados.....</b>	<b>85</b>
<b>Capítulo 5. Propuesta.....</b>	<b>93</b>



<b>5.1. Filtrado colaborativo</b> .....	93
<b>5.2. Recursos financieros, costos</b> .....	93
<b>5.3. Análisis de la solución</b> .....	94
<b>5.4. Primera etapa: Formación</b> .....	95
<b>5.5. Segunda etapa: Agregar opiniones</b> .....	96
<b>5.6. Tercera etapa: Recomendación</b> .....	96
<b>Epílogo</b> .....	100
<b>Bibliografía</b> .....	103



## Prólogo

En el presente libro se expone una investigación que se desarrolló en una empresa de comida rápida en la ciudad de Guayaquil. Este establecimiento tiene algunos años en el mercado, pero a pesar de que posee un adecuado manejo de redes sociales todavía no es conocido, por lo que no acoge a mucho público.

El negocio de comida rápida es uno de los más rentables hoy en día. Se encargan de ofrecer alimentos que, en muchos de los casos, no son tan saludables, pero que siempre están a disposición de las personas que necesitan ingerir estos alimentos o que no tienen tiempo suficiente para buscar otras opciones. De hecho, el tiempo es algo muy importante en estos negocios.

Se propone la implementación de un sistema de recomendación de un filtro colaborativo basado en el algoritmo ALS, que permita analizar las predicciones de los siguientes productos que el cliente desee adquirir, lo que ayudará a comprender al cliente y a personalizar al máximo sus opciones de consumo.

Los autores



# CAPÍTULO 1. INTRODUCCIÓN A LOS SISTEMAS DE RECOMENDACIÓN



## Capítulo 1. Introducción a los sistemas de recomendación

Un sistema de recomendación es una base de datos que analiza y procesa información histórica de los usuarios (edad, calificaciones, compras previas, etc.) de los productos o contenidos como marcas, precios, modelos, entre otras; y los transforma en conocimiento accionable; es decir, determina cuáles productos puede ser potencialmente interesante para el usuario (Walid-Ghobar, 2016-2017).

De ahí que, los sistemas de recomendación constituyen herramientas que posibilitan que los usuarios encuentren elementos y contenidos de su interés. Se trata de soluciones que partiendo de datos y una serie de criterios o valoraciones realizan recomendaciones mediante un ranking de los elementos que más se aproximarían a los intereses detectados en el usuario, o a través de una predicción de la valoración que el usuario daría a dicho elemento para presentarle otros que puedan gustarle (García, 2021).

El sistema de recomendación tiene algunos tipos como basados en contenidos (analiza el contenido de los ítems), filtrado colaborativo (utiliza datos de interacción con los ítems y encontrar patrones), basados en redes sociales (utiliza información de una red social) y los híbridos (incorpora lo mejor de los sistemas anteriores y los combina) (Castillo-Saco, 2014).



Sobre los modelos basados en redes sociales se debe añadir que desempeñan un rol fundamental en la recomendación a grupos de usuarios cuando se incluye información social en conjuntos de datos. Proponen un marco de recomendación grupal con conciencia social que utiliza en conjunto las relaciones sociales y los comportamientos sociales para inferir las preferencias de un grupo, y también para modelar la personalidad de los miembros del grupo como: tolerancia (voluntad de recibir contenido no preferido) y altruismo (disposición para recibir contenido preferido por amigos).

Desde otra perspectiva, Criado (2018) plantea que los tipos de sistemas de recomendación también son cuatro, pero sus nombres varían. Esta autora plantea que los dos principales sistemas son el filtrado colaborativo (FC) y el basado en contenido (BC), pero, además, existen los sistemas basados en conocimiento y los híbridos. A continuación, se aludirán parcialmente cada uno de ellos.

El filtrado colaborativo se sustenta en que, si dos usuarios son similares, entonces tendrán preferencias igualmente similares. Un punto de especial importancia en este algoritmo son las puntuaciones de los usuarios hacia los distintos productos.

La función del filtrado colaborativo es determinar que dos usuarios similares tendrán preferencias igualmente similares. Para establecer dicha similitud se

basa en que un usuario 1 y un usuario 2 tengan un historial de recomendaciones muy parecido hacia los distintos productos. De esta manera, se puede predecir que cuando el usuario 1 adquiere un producto nuevo es de suponer que al usuario 2 también le interese y es altamente probable que lo compre. Por tanto, se recomendará al usuario 2 la compra de ese artículo.

Los sistemas de recomendación basados en contenido (BC) consisten en emparejar los atributos de un perfil de usuario en el que están guardados sus preferencias e intereses, con los atributos de un producto, para así poder recomendar al usuario, los nuevos artículos. La idea básica es que un usuario elegirá productos iguales a los que ha elegido anteriormente.

Mientras que el filtrado colaborativo identifica usuarios con preferencias similares a las del usuario dado y recomienda los artículos que le han gustado, los sistemas basados en contenido recomiendan artículos similares a los que el usuario ha elegido con anterioridad.

Si comparamos los sistemas de recomendación basados en contenido con el filtrado colaborativo se puede apreciar que existen tanto ventajas como desventajas. A continuación, se abordarán algunas de ellas.



## Ventajas de los sistemas de recomendación basados en contenido

- La independencia de los usuarios constituye la primera ventaja, pues solo se necesita la información de las puntuaciones dadas por el usuario activo, mientras que en el FC necesitamos las puntuaciones de otros usuarios para poder utilizar el método de vecinos cercanos y poder hacer así una recomendación.
- La transparencia constituye la segunda ventaja, pues las explicaciones para poder demostrar cómo funciona el sistema de recomendación pueden darse mirando la lista de propiedades de los artículos. Esas propiedades son indicadores para saber si la recomendación ha funcionado bien o no. Por el contrario, en el filtrado colaborativo son todo “cajas negras” porque la única explicación para la recomendación de un producto es que usuarios desconocidos con gustos similares han valorado bien ese producto.
- Los nuevos productos son la tercera ventaja, pues los SR basados en contenido son capaces de recomendar artículos que todavía no han sido puntuados. En el filtrado colaborativo se necesita que un número de usuarios substancial haya puntuado el producto para poder hacer una recomendación fiable.

## Desventajas de los sistemas de recomendación basados en contenido

- El análisis de contenido limitado constituye la primera desventaja, pues las técnicas basadas en contenido tienen un número limitado de características y, en algunas ocasiones, este número de propiedades no son suficientes para poder distinguir aspectos de los productos que pueden ser necesarios para hacer una recomendación óptima.
- La serendipia constituye la segunda desventaja, que es la tendencia de subrayar de los sistemas basados en contenido de recomendar con un grado limitado de novedad. Una técnica perfecta basada en contenido va a encontrar muy pocas veces una novedad, limitando así el rango de aplicaciones para el que va a ser válido.
- Los nuevos usuarios constituyen la tercera desventaja, pues un sistema basado en contenido necesita un número elevado de puntuaciones para poder entender las preferencias de los usuarios y hacer una recomendación precisa. Por lo tanto, cuando hay pocas puntuaciones, para un usuario nuevo, no habrá buenos resultados.

Por último, se debe agregar que para este tipo de sistemas lo más importante es la manera en la que se representa cada producto; es decir, sus propiedades. Al tener mostradas las preferencias del usuario sobre la base de las propiedades de cada producto, se juntan las características del producto con las preferencias del



usuario. Para obtener esta información se utilizan dos técnicas: *inverse document frequency* (IDF) y *term-frequency* (TF). Este tipo de técnicas se usan en productos que tiene una información textual asociada, como puede ser el nombre o la descripción de un artículo, o en documentos.

Los sistemas de recomendación basados en conocimiento se adaptan a otro contexto, y es que en ocasiones las compras se realizan con muy poca frecuencia, o sea cada mucho tiempo, como es el caso de comprar una casa. En este caso un filtrado colaborativo puro no funcionaría correctamente debido al número tan pequeño de puntuaciones que habrá. Asimismo, sucede cuando un artículo de tecnología en el que las puntuaciones que se han dado hace un cierto tiempo quizás no sean útiles ahora por el avance de la tecnología. Es ahí cuando se utilizan los sistemas de recomendación basados en conocimiento.

Su mayor ventaja es que no necesitan puntuaciones, por lo tanto, no existe el problema del arranque en frío. En esta clasificación hay dos tipos de sistemas: basados en restricciones (*constraint-based*) y basados en casos (*case-based*). En ambas aproximaciones el usuario deberá especificar sus requerimientos y el sistema buscará una solución.

El funcionamiento de un SR *constraint-based* se basa en los filtros (preferencias) que quiera poner el usuario. Primero, el usuario especifica sus preferencias iniciales que pueden ser comunes para todos los usuarios o personalizadas. Esto



se suele hacer con una serie de preguntas o marcando opciones. Segundo, se introduce la información dentro del algoritmo y se presentan una serie de resultados. Finalmente, al revisar la información que el sistema de recomendación le ha dado, el usuario tiene la opción de buscar soluciones alternativas porque el resultado no ha sido satisfactorio, o hacer una búsqueda más selectiva con sus gustos para obtener unos resultados más personalizados.

El *case-based*, por su parte, resuelve problemas nuevos utilizando información de casos anteriores que han tenido un resultado satisfactorio. Primero, guarda casos anteriores en memoria y para resolver los nuevos supuestos vuelve a usar experiencias similares para usuarios similares. La reutilización puede ser parcial y hay que cambiar el algoritmo o en algunos casos se puede reutilizar por completo. Por último, se guarda este nuevo problema en la memoria para que el algoritmo aprenda.

La implementación práctica de este tipo de sistemas no es sencilla puesto que tiene que adaptarse a cada usuario y cada contexto en concreto. Los modelos de predicción que se suelen utilizar en este tipo de sistemas son los árboles de decisión, ya que mediante preguntas se puede ir guiando al usuario hacia un resultado final. Algunos de los problemas que se deben de evitar son los resultados por defecto, los conjuntos vacíos y la lentitud computacional.

Los tipos SR explicados anteriormente tienen que lidiar con algunas desventajas, con la intención de disminuirlas y, debido a la necesidad de un sistema “total” que combinara dos o más técnicas de recomendación, surgen los SR híbridos.

Para un sistema híbrido se necesitan las técnicas tanto de filtrado colaborativo como basado en contenido y basado en conocimiento. El filtrado colaborativo asume que hay una serie de clusters formados por usuarios que tienen preferencias similares. La función del FC entonces será buscar parejas similares y hacer recomendaciones a partir de los productos favoritos. Por otro lado, la técnica basada en contenido hace una función muy parecida al FC puesto que recomienda productos a usuarios que han tenido unos gustos similares en el pasado. Finalmente, los basados en conocimiento introducen la personalización del sistema.

Indiscutiblemente, son muchos los SR, así como sus ventajas y desventajas en correspondencia con el contexto en el que se quiera accionar. No obstante, en el presente libro se analizará las predicciones en un filtro colaborativo basado en el algoritmo Alternating Least Squares (ALS), para una empresa de comida rápida en la ciudad de Guayaquil.

Estos sistemas son comúnmente utilizados por plataformas de servicios online y de E-Commerce, para solucionar el problema de sobreinformación de una forma más eficaz. Aportan sugerencias personalizadas a cada usuario de plataforma,

todo esto en función de sus intereses y valoraciones, e información de usuarios similares (Manzano-Chicano, 2018).

El comercio ha evolucionado con el surgimiento del comercio electrónico. Cada vez son más las empresas que se suman a la tendencia de abrir una tienda en la web, en la que los consumidores pueden adquirir directamente un producto y recibirlo en su domicilio sin tener que moverse. Es por ello, que el uso de SR se ha vuelto imprescindible. Mediante ellos se puede fidelizar clientes, captar nuevos clientes, y hasta predecir las compras de un cliente, lo que implica un aumento en las ventas de la empresa.

El *Alternating Least Squares* (ALS) es un algoritmo, secuencia de instrucciones ordenadas y lógicas que siguen un procedimiento que contempla todos los escenarios posibles que se puedan encontrar en la resolución de un problema o situación, trabaja con filtro colaborativo y que es cotejado en cuanto a la optimización de función de costos respecto a la factorización de la matriz de interacción usuarios – ítems; es decir, es un algoritmo que trabaja con matrices de datos y, sobre esa base, arroja resultados que ayudan a la predicción de las compras de un cliente (Comon, Luciani, & De Almeida, 2009).

Según lo que aduce Takács & Tikk (2012) ALS es un enfoque computacional que garantiza la minimización directa de la función objetivo sin muestreo, con ello se posiciona como el mejor de los métodos usados para la clasificación.

Para la implementación de un sistema de recomendación basado en filtro colaborativo, se ha seleccionado un pequeño negocio de comida rápida ubicado en la ciudad de Guayaquil. Este establecimiento tiene algunos años en el mercado, pero a pesar de que posee un adecuado manejo de redes sociales todavía no es conocido, por lo que no acoge a mucho público.

El objetivo de esta investigación es analizar las predicciones de un filtro colaborativo basado en el algoritmo ALS, para una empresa de comida rápida en la ciudad de Guayaquil, además se elaborará una propuesta de implementación.

## **1.1. Descripción del problema**

### **1.1.1. Pregunta global**

¿De qué manera se debe analizar las predicciones de un filtro colaborativo basado en el algoritmo ALS para una empresa de comida rápida en la ciudad de Guayaquil?

### **1.1.2. Preguntas específicas**

- ¿Cómo se deben revisar fuentes bibliográficas acerca de los algoritmos de filtrado colaborativo?
- ¿De qué manera se deben analizar las técnicas que emplea el filtrado colaborativo que permitirá establecer sus características de funcionamiento?
- ¿Cómo elaborar cuadros comparativos que contemplen los resultados

obtenidos en la investigación?

- ¿Cómo crear una propuesta de los componentes de un sistema de recomendación basado en el algoritmo ALS de una empresa de comida rápida en la ciudad de Guayaquil?

## 1.2. Objetivos

### 1.2.1. Objetivo general

Analizar las predicciones de un filtro colaborativo basado en el algoritmo ALS para una empresa de comida rápida en la ciudad de Guayaquil.

### 1.2.2. Objetivos específicos

- Revisar fuentes bibliográficas acerca de los algoritmos de filtrado colaborativo.
- Analizar las técnicas que emplea el filtrado colaborativo que permitirá establecer sus características de funcionamiento.
- Elaborar cuadros comparativos que contemple los resultados obtenidos en la investigación
- Proponer la creación de los componentes de un sistema de recomendación basado en el algoritmo ALS de una empresa de comida rápida en la ciudad de Guayaquil.



## **1.3. Justificación**

### **1.3.1. Justificación teórica**

El presente trabajo se justifica desde la teoría debido a que se realizó una investigación profunda extraída desde fuentes bibliográficas digitales entre las que se consideraron artículos científicos, papers, tesis, proyectos de investigación, libros, diarios, artículos de revistas científicas publicados en plataformas como Dialnet, Scielo, Scopus, entre otras. Además, es importante recalcar la reunión de información acerca del tema de investigación que se encontró en idioma inglés, la cual sirve para aclarar la realidad de la problemática en países de habla inglesa. Todos estos aportes constituyen la base de la investigación y son importantes para conocer la situación investigada y de esa manera obtener resultados científicos.

### **1.3.2. Justificación metodológica**

Los aportes metodológicos que se usarán en esta investigación comprenden estudios de tipo exploratorio, descriptivo y explicativo los cuales se llevaran a cabo de una forma sistemática, con el fin de comprender la realidad de la problemática. Además, para la obtención de resultados se hará uso de una encuesta aplicada a la muestra.



### **1.3.3. Justificación práctica**

El negocio de comida rápida es uno de los más rentables hoy en día. Se encargan de ofrecer alimentos que, en muchos de los casos, no son tan saludables, pero que siempre están a disposición de las personas que necesitan ingerir estos alimentos o que no tienen tiempo suficiente para buscar otras opciones, de hecho, el tiempo es algo muy importante en estos negocios. Por ello se hace uso de un sistema de recomendaciones que es en realidad un algoritmo encargado de predecir los siguientes productos que el cliente desea adquirir, el análisis de estas predicciones ayudará a comprender al cliente y a personalizar al máximo sus opciones de consumo.



## CAPÍTULO 2. FUNDAMENTACIÓN TEÓRICA





## Capítulo 2. Fundamentación teórica

### 2.1. Técnicas de filtro colaborativo

El término de filtrado colaborativo fue acuñado por Goldberg en el sistema de recomendación Tapestry. Se ha convertido en uno de los enfoques más utilizados para brindar recomendaciones de servicios a los usuarios en muchos sistemas web en línea como Fab, Amazon, Netflix, Ringo y Jester (Olguín & De Jesús, 2019). La ventaja de los filtrados colaborativos sobre los enfoques existentes basados en contenido es que no necesita conocer el contenido del elemento o las características que son más complejas.

Normalmente, la intuición detrás del filtrado colaborativo es que, si los usuarios tienen preferencias similares en el pasado, tienden a tener el mismo interés en el futuro. Aunque se ha trabajado mucho en el sistema de recomendaciones en los últimos años, el filtrado colaborativo todavía enfrenta algunos problemas, y sus mecanismos aún tienen margen de mejora, lo que los convierte en un área de investigación rica para hacer frente al crecimiento de la información (Valdiviezo-Díaz, Ortega, & Mayor, 2020).

En el enfoque basado en contenido, la idea de que los usuarios tuvieran sus calificaciones en algunos elementos en el pasado tendrá una calificación similar en otros elementos similares en el futuro. Por tanto, la similitud calculada en el



espacio de elementos se basa en el contenido o las características de estos elementos. Así, la relación entre los elementos que el usuario ha calificado en el anterior y otros elementos de la base de datos utilizada para determinar cuáles son los elementos más adecuados para el usuario objetivo (Yauri Godoy, 2019).

Es decir, el sistema depende de la búsqueda de elementos que tengan una fuerte relación en comparación con su perfil. Estos sistemas se utilizan para recomendar páginas web y artículos de noticias, etc.

## 2.2. Filtrado colaborativo basado en memoria

La información en la web ha crecido de manera significativa en los últimos años. Los filtros de información surgieron para afrontar el desafío de la búsqueda de información en la WWW, un problema que puede compararse con “localizar agujas en un pajar que crece exponencialmente”. Los sistemas de recomendación son una clase de filtros de información que han demostrado su eficacia. Por ejemplo, los sistemas de recomendación de los sitios web de comercio electrónico ayudan a los usuarios a encontrar sus CD o libros favoritos (Hernández, 2019).

De manera similar, los sistemas de recomendación ayudan a localizar elementos como páginas web, noticias, bromas o películas, entre miles o incluso millones de elementos (Marín, Zapata, Rojas, & Mendez, 2016). El filtrado basado en contenido (CBF) y el filtrado colaborativo (CF) son dos tecnologías utilizadas en





los sistemas de recomendación. Los sistemas CBF analizan el contenido de un conjunto de elementos junto con las calificaciones proporcionadas por usuarios individuales para inferir qué elementos no calificados podrían ser de interés para un usuario específico.

En contraste, los métodos de filtrado colaborativo generalmente acumulan una base de datos de calificaciones de elementos emitidas por un gran conjunto de usuarios y luego usan esas calificaciones para predecir las preferencias de un usuario de la consulta para elementos no vistos (Collado, 2016). El filtrado colaborativo no se basa en las descripciones de contenido de los elementos, sino que depende puramente de las preferencias expresadas por un conjunto de usuarios.

El aprendizaje interactivo de perfiles de usuario es un sistema de recomendación que no puede brindar un servicio preciso a un nuevo usuario, cuyas preferencias inicialmente se desconocen. Esto se ha denominado el "problema del nuevo usuario" (Valdiviezo, 2019). Antes de poder hacer predicciones, un sistema CF normalmente requiere que el nuevo usuario califique una lista de elementos de consulta en una recopilación de información inicial etapa.

La heurística eficiente es esencial para seleccionar elementos de consulta informativos y así mantener la etapa de recopilación de información lo más corta posible, ya que los usuarios pueden perder la paciencia fácilmente cuando se



enfrentan a una lista larga de elementos de consulta. En cada paso de recopilación de información, se presentan al usuario los elementos de consulta que se espera que agudicen al máximo el perfil del usuario.

### 2.3. Evolución de los filtrados colaborativos

Como área formal de investigación, el filtrado colaborativo comenzó como un medio para manejar la naturaleza cambiante de los repositorios de texto. A medida que las bases de contenido crecieron de contenido principalmente "oficial", como bibliotecas y conjuntos de documentos corporativos, a contenido "informal" como listas de discusión y archivos de correo electrónico, el desafío de encontrar elementos de calidad cambió para bien (Olguín & De Jesús, 2019).

Las técnicas puramente basadas en contenido a menudo eran inadecuadas para ayudar a los usuarios a encontrar los documentos que buscaban. Las representaciones basadas en palabras clave podrían hacer un trabajo adecuado al describir el contenido de los documentos, pero podrían hacer poco para ayudar a los usuarios a comprender la aplicación de las palabras clave o la calidad de esos documentos.

Por lo tanto, una búsqueda de palabras clave para "*Chicago Rocks*" podría producir no solo artículos académicos de la *Chicago Rocks and Minerals Society*, sino también la publicación "superficial" en un tablero de anuncios de música con

respecto a la opinión de un visitante sobre la banda de rock de la década de 1970 (González, García, & Gil, 2020).

A principios de la década de 1990, parecía haber dos posibles soluciones a este nuevo desafío: 1. esperar mejoras en la inteligencia artificial que permitieran una mejor clasificación automatizada de documentos, o 2. incorporar el juicio humano al circuito. Si bien los desafíos de la clasificación automatizada aún no se han superado, el juicio humano ha demostrado ser valioso y relativamente fácil de incorporar en sistemas semiautomatizados.

El sistema Tapestry, desarrollado en Xerox PARC, dio el primer paso en esta dirección al incorporar las acciones y opiniones de los usuarios en una base de datos de mensajes y un sistema de búsqueda. Tapestry almacenaba el contenido de los mensajes, junto con metadatos sobre autores, lectores y respondedores. También permitía a cualquier usuario almacenar anotaciones sobre mensajes, como "encuesta útil" o "¡Phil debería ver esto!".

Los usuarios de Tapestry podían realizar consultas que combinaban información textual básica (por ejemplo, contiene la frase "sistemas de recomendación") con consultas de metadatos semánticos (por ejemplo, escritas por John O respondidas por Joe) y consultas de anotaciones (por ejemplo, marcadas como "excelentes" por Chris).



Este modelo se conoce como filtrado colaborativo pull-active, porque es responsabilidad del usuario que desea recomendaciones, extraer activamente las recomendaciones de la base de datos. Poco después de la aparición de Tapestry, otros investigadores comenzaron a reconocer el potencial para explotar los "centros de información" humanos que parecen ocurrir naturalmente dentro de las organizaciones.

Maltz y Ehrlich (1995) desarrollaron un sistema de recomendación de filtrado colaborativo push-active que facilitó a una persona que lee un documento enviarlo a otras personas de la organización que deberían verlo. Este tipo de rol de recomendación de empuje se ha vuelto popular, y muchas personas hoy sirven como "centros de bromas" que reciben bromas de todas partes y las reenvían a aquellos que creen que las apreciarían (aunque a menudo con un pensamiento mucho menos discriminatorio de lo que se imaginaba).

Una limitación de los sistemas de filtrado colaborativo activo es que requieren una comunidad de personas que se conozcan. Los sistemas pull-active requieren que el usuario sepa en qué opiniones confiar. Los sistemas push-active requieren que el usuario sepa a quién puede interesarle un contenido en particular. Los sistemas de filtrado colaborativo automatizado (ACF) alivian a los usuarios de esta carga mediante el uso de una base de datos de opiniones históricas de usuarios para hacer coincidir automáticamente a cada individuo con

otros con opiniones similares. Los primeros sistemas ACF incluían GroupLens en el ámbito de los artículos de grupos de noticias de Usenet, Ringo en el ámbito de la música y los artistas musicales, y el Recomendador de vídeo de Bellcore en el ámbito de las películas.

GroupLens, utilizando una interfaz muy explícita, en la que las calificaciones de los artículos de los grupos de noticias de Usenet se ingresaban manualmente presionando una tecla o un botón, y las calificaciones se mostraban numérica o gráficamente. Llevando esto un paso más allá, tanto Ringo como Video Recomendar fueron accesibles a través de la web y el correo electrónico, y proporcionaron funciones simples para la interacción de la comunidad.

### **2.3.1. Desafíos del filtrado colaborativo**

Independientemente del desarrollo que ha alcanzado esta herramienta, aún le quedan retos por cumplir. Entre estos desafíos se encuentran la escasez de datos, la escalabilidad, la sinonimia, la “oveja gris”, el shilling attacks, y la diversidad.

La escasez de datos se refiere a que, en la práctica, muchos sistemas de recomendación comerciales se basan en conjuntos de datos de gran tamaño. Por cuanto, la matriz usuario elemento empleada para el filtrado colaborativo podría ser muy grande, lo que implica dispersión en la realización de la recomendación.



Otro de los problemas causado por la escasez de datos es el arranque en frío, al que ya se hizo referencia anteriormente. Como los métodos de filtrado colaborativo recomiendan artículos basados en las preferencias anteriores de los usuarios, los nuevos usuarios necesitan evaluar el suficiente número de elementos para permitir que el sistema pueda recopilar sus preferencias con precisión y, por lo tanto, proporcionar recomendaciones fiables.

De igual manera, los productos nuevos también tienen el mismo problema. Cuando se agregan nuevos elementos al sistema, tienen que ser valorados por un gran número de usuarios antes de que se pueda recomendar a los usuarios que tienen gustos similares a los que calificaron. El problema de un nuevo elemento no afecta a la recomendación basada en contenido, ya que la recomendación de un elemento se basa en su conjunto discreto de cualidades descriptivas más que de evaluaciones.

La escalabilidad se refiere al aumento del número de usuarios y elementos en los algoritmos tradicionales FC, por lo que sufren serios problemas de escalabilidad. Por ejemplo, con decenas de millones de clientes y millones de elementos, un algoritmo de FC con la complejidad de  $O(N)$  ya es demasiado grande. Además, muchos sistemas tienen que reaccionar inmediatamente a los requerimientos en línea y hacer recomendaciones para todos los usuarios, independientemente de

sus compras y la historia de las evaluaciones, lo que exige una mayor escalabilidad de un sistema de FC.

La sinonimia se refiere a que un número de elementos iguales o muy similares tienen nombres o entradas diferentes. La mayoría de los sistemas de recomendación son incapaces de descubrir esta asociación latente y, por ende, se asume como productos de manera diferente. Por ejemplo, los términos "películas infantiles" y "filmes infantiles" hacen referencia a un mismo elemento. De ahí que, la prevalencia de sinónimos limita el rendimiento de los sistemas de recomendación de FC. Modelar tópicos con técnicas como *Latent Dirichlet Allocation* que agrupan palabras diferentes pertenecientes a un mismo tema podría ayudar a resolver esto.

La oveja gris (*Grey sheep*) se refiere a los usuarios cuyas opiniones no están de acuerdo o en desacuerdo con algún grupo de personas y, por lo tanto, no se benefician del filtrado colaborativo. Las ovejas negras son el grupo con gustos a los que hacer recomendaciones es casi imposible. Aunque se trata de un fallo en el sistema de recomendación, los recomendadores no-electrónicos también tienen grandes problemas en estos casos, las ovejas negras constituyen un fallo aceptable.

El shilling attacks se refiere a un sistema de recomendación donde todo el mundo puede evaluar. Las personas pueden emitir varias opiniones positivas para sus propios artículos y clasificaciones negativas para sus competidores. Por eso es necesario que los sistemas de filtrado colaborativo introduzcan precauciones para impedir este tipo de manipulaciones.

Es por esa razón que nuevos algoritmos se han desarrollado para la FC como resultado del premio Netflix. Sistemas de filtrado colaborativo cruzados donde los perfiles de usuario son combinados a través de múltiples sistemas de recomendación preservando la privacidad. Filtrados colaborativos robustos, donde la recomendación es estable frente a esfuerzos de manipulación.

Como indicador del interés e importancia que tienen estos sistemas a la hora de aumentar la precisión de la recomendación, destacamos el hecho de que la plataforma Netflix organiza todos los años un concurso, abierto a toda la comunidad informática, para competir con el mejor algoritmo de recomendación posible. La filosofía que subyace en estos sistemas no se restringe a las plataformas audiovisuales que acabamos de citar, sino que puede aplicarse al sector del comercio electrónico (Pajuelo, 2021).

Por último, está la diversidad que, sin lugar a dudas, aumenta con los filtros colaborativos, pues permiten descubrir nuevos productos entre varias opciones múltiples. Sin embargo, algunos de los algoritmos, sin proponérselo pueden



hacer lo contrario. Como los filtros colaborativos recomiendan productos basados en las ventas o calificaciones, por lo general, no pueden recomendar productos con escasos datos históricos.

## 2.4. Filtrado colaborativo y Web adaptable

Estos primeros sistemas de filtrado colaborativo se diseñaron para proporcionar explícitamente a los usuarios información sobre los elementos. Es decir, los usuarios visitaron un sitio web con el fin de recibir recomendaciones del sistema CF. Más tarde, los sitios web comenzaron a utilizar sistemas CF entre bastidores para adaptar su contenido a los usuarios, como elegir qué artículos de noticias debería presentar un sitio web de forma destacada a un usuario (López & Montes).

Los proveedores de información en la web deben lidiar con la atención limitada del usuario y el espacio de pantalla limitado. El filtrado colaborativo puede predecir qué información es probable que los usuarios deseen ver, lo que permite a los proveedores seleccionar subconjuntos de información para mostrar en el espacio limitado de la pantalla. Al colocar esa información en un lugar destacado, permite al usuario maximizar su atención limitada. De esta forma, el filtrado colaborativo permite que la web se adapte a las necesidades de cada usuario individual.

## 2.5. Usos del filtrado colaborativo

### 2.5.1. Tareas del usuario

Los diseñadores de servicios web deben identificar cuidadosamente las posibles tareas que los usuarios pueden desear realizar con su sitio, ya que diferentes tareas pueden requerir diferentes decisiones de diseño. Desde una perspectiva de marketing, este es el valor agregado por el sistema de FC. En esta sección, consideramos las tareas del usuario para las que el filtrado colaborativo es útil.

Las tareas para las que las personas utilizan el filtrado colaborativo que se han estudiado incluyen (Rios, Godoy, & Schiaffino, 2016):

1. “Ayúdame a encontrar nuevos elementos que me puedan gustar.” En un mundo de sobrecarga de información, no puedo evaluar todas las cosas. Presente algunos para que pueda elegir. Esto se ha aplicado con mayor frecuencia a artículos de consumo (música, libros, películas), pero también se puede aplicar a artículos de investigación, páginas web u otros artículos calificables.
2. “Avísame sobre un tema en particular.” Tengo un artículo en particular en mente; ¿Sabe la comunidad si es bueno o malo?
3. “Ayúdame a encontrar un usuario (o algunos usuarios) que me puedan gustar.” A veces, saber en quién concentrarse es tan importante como saber

en qué concentrarse. Esto podría ayudar a formar grupos de discusión, establecer contactos o conectar a los usuarios para que puedan intercambiar recomendaciones socialmente.

4. "Ayuda a nuestro grupo a encontrar algo nuevo que nos pueda gustar." La FQ puede ayudar a grupos de personas a encontrar elementos que maximicen el valor del grupo en su conjunto. Por ejemplo, una pareja que desea ver una película juntos o un grupo de investigación que desea leer un artículo apropiado.
5. "Ayúdame a encontrar una combinación de elementos "nuevos" y "antiguos"." Podría desear una "dieta equilibrada" de restaurantes, incluidos los que he comido anteriormente; o quizás desee ir a un restaurante con un grupo de personas, incluso si algunos ya han estado allí; o quizás desee comprar algunos alimentos que sean apropiados para mi carrito de compras, incluso si ya los he comprado antes.
6. "Ayúdame con las tareas específicas de este dominio". Por ejemplo, un recomendador de artículos de investigación también podría desear respaldar tareas como "recomendar artículos que mi artículo debería citar" y "recomendar artículos que deberían citar mi artículo". De manera similar, un recomendador para una película y un restaurante podría diseñarse para distinguir entre recomendaciones para una primera cita y una noche de chicos. Se han explorado recomendaciones para algunas tareas específicas

de dominio; muchos no lo han hecho. Hasta la fecha, muchas investigaciones se han centrado en tareas más abstractas (como "encontrar nuevos elementos") sin profundizar en los objetivos subyacentes del usuario (como "encontrar una película para una primera cita").

## 2.6. Funcionalidad del sistema de filtrado colaborativo

También existen amplios conjuntos abstractos de tareas que admiten los sistemas CF. No es casualidad que esta funcionalidad del sistema esté relacionada con las tareas de usuario de la sección anterior. Idealmente, el sistema admitiría todas las tareas del usuario, aunque el mapeo de una aplicación real a la funcionalidad de un sistema CF real puede ser un desafío.

1. Recomendar elementos. Muestre una lista de elementos a un usuario, en orden de su utilidad. A menudo, esto se describe como predecir lo que el usuario calificaría el artículo y, luego clasificar los artículos por esta calificación prevista. Sin embargo, algunos algoritmos de recomendación exitosos no calculan los valores de calificación pronosticados en absoluto. Por ejemplo, el algoritmo de recomendación de Amazon agrega elementos similares a las compras y calificaciones de un usuario sin siquiera calcular una calificación prevista. En lugar de mostrar una calificación prevista personalizada, su interfaz de usuario muestra la calificación promedio de los clientes. Como resultado, la lista de recomendaciones puede aparecer



desordenada con respecto al valor de calificación promedio mostrado. En muchas aplicaciones, elegir bien los primeros elementos es crucial; producir valores predichos es secundario.

2. Predecir para un artículo determinado. Dado un artículo en particular, calcule su calificación prevista. Tenga en cuenta que la predicción puede ser más exigente que la recomendación. Para recomendar elementos, un sistema solo necesita estar preparado para ofrecer algunas alternativas, pero no todas. Algunos algoritmos aprovechan esto para ser más escalables ahorrando memoria y tiempo de cálculo. Para proporcionar predicciones para un artículo en particular, un sistema debe estar preparado para decir algo sobre cualquier artículo solicitado, incluso los que rara vez se califican. ¿Cómo decide un sistema cómo un usuario en particular calificaría un artículo solicitado si muy pocos usuarios, y mucho menos usuarios similares al usuario en particular, han calificado el artículo? Las predicciones personalizadas pueden ser desafiantes, si no imposibles.

3. Recomendaciones restringidas: recomendar de un conjunto de elementos. Dado un conjunto particular o una restricción que da un conjunto de elementos, recomiende desde dentro de ese conjunto. Por ejemplo: "Considere el siguiente escenario. El sobrino de Mary, de 8 años, está de visita el fin de semana y le gustaría llevarlo al cine. A ella le gustaría una comedia o una película familiar con una calificación no "superior" a PG-

13. Preferiría que la película no contenga sexo, violencia o lenguaje ofensivo, dure menos de dos horas y, si es posible, se muestre en un teatro de su vecindario. Finalmente, le gustaría seleccionar una película que ella misma podría disfrutar ". Schafer y col (2007) proponen un "sistema de meta-recomendación" que genera recomendaciones a partir de una combinación de múltiples fuentes de recomendación. Los usuarios definen preferencias y requisitos a través de un formulario web que restringe el conjunto de posibles elementos candidatos.

## 2.7. Propiedades de los dominios adecuados para el filtrado colaborativo

Se puede simplemente tomar una aplicación de usuario, implementarla con un sistema CF y esperar que funcione. Sin embargo, es más conocido que la CF es eficaz en dominios con determinadas propiedades. Parece útil familiarizarnos con ellos y considerar si la aplicación de usuario es una buena opción. Agrupamos estas propiedades a continuación en distribución de datos, significado subyacente y persistencia de datos (Echevarría, 2017).

Es importante mencionar que, con especial consideración, CF se puede aplicar con éxito en dominios que no tienen algunas de las propiedades siguientes. Simplemente los enumeramos para provocar la reflexión y la discusión sobre qué dominios son fáciles o difíciles con el filtrado colaborativo. Distribución de datos. Estas propiedades se refieren a los números



y la forma de los datos: 1. Hay muchos elementos. Si hay pocos elementos para elegir, el usuario puede conocerlos todos sin necesidad de soporte informático. 2. Hay muchas calificaciones por artículo. Si hay pocas calificaciones por elemento, es posible que no haya suficiente información para proporcionar predicciones o recomendaciones útiles. 3. Hay más calificaciones de usuarios que elementos recomendados. Un corolario del párrafo anterior es que a menudo necesitará más usuarios que la cantidad de elementos que desea poder recomendar de manera competente.

Precisamente, si hay pocas calificaciones por usuario, necesitará muchos usuarios. Muchos sistemas son así. Por ejemplo, esto convierte a las páginas web en un dominio desafiante, especialmente si el sistema requiere calificaciones explícitas. Google, un motor de búsqueda popular, afirma indexar 8 mil millones de páginas web en la actualidad, que es más que la cantidad de personas en el mundo, sin mencionar la cantidad que tiene acceso a computadoras.

Como otro ejemplo, con un millón de usuarios, un sistema de CF podría hacer recomendaciones para cien mil elementos, pero solo puede hacer predicciones confiables para diez mil o menos, dependiendo de la distribución de calificaciones entre los elementos. La distribución de calificaciones es casi siempre muy sesgada: algunos elementos obtienen la mayoría de las

calificaciones, una larga cola de elementos que obtienen pocas calificaciones. Los elementos de esta cola larga no serán predecibles con seguridad.

Los usuarios califican varios elementos. Si un usuario califica solo un elemento, esto proporciona cierta información para las estadísticas de resumen, pero no información para relacionar los elementos entre sí. Significado subyacente. Estas propiedades son el significado subyacente de los datos: 1. Para cada usuario de la comunidad, hay otros usuarios con necesidades o gustos comunes. La FQ funciona porque las personas tienen necesidades o gustos en común. Si una persona tiene gustos tan únicos que nadie más los comparte, entonces la FQ no puede proporcionar ningún valor. De manera más general, la CF funciona mejor cuando cada usuario puede encontrar muchos otros usuarios que comparten sus gustos de alguna manera. 2. La evaluación de un artículo requiere un gusto personal.

En los casos en que existen criterios objetivos de bondad que pueden calcularse automáticamente, esos criterios pueden aplicarse mejor por medios distintos del filtrado colaborativo, por ejemplo, algoritmos de búsqueda. El filtrado colaborativo permite a los usuarios con gustos similares informarse entre sí. La FC agrega un valor sustancial cuando la evaluación de elementos es en gran medida subjetiva

(p. Ej., Música), o cuando esos elementos tienen muchos criterios objetivos diferentes que deben sopesarse subjetivamente entre sí (p. Ej., Automóviles).

A veces hay criterios objetivos que pueden ayudar (por ejemplo, recomendar solo libros escritos en inglés), pero si la recomendación se puede realizar utilizando solo criterios objetivos, entonces la FQ no es útil. 3. Los artículos son homogéneos. Es decir, por todos los criterios objetivos de consumo son similares, y difieren solo en criterios subjetivos.

Los álbumes de música son así, la mayoría tienen un precio similar cuando se compra, de una longitud similar. Los libros o trabajos de investigación también son así. Los artículos que se venden en una tienda departamental no son así: algunos son baratos, otros muy caros. Por ejemplo, si compra un martillo, tal vez no se le debería recomendar un refrigerador.

## 2.8. Persistencia de datos

Estas son propiedades de cuánto tiempo son relevantes los datos: 1. Los elementos persisten. Un sistema de FQ no solo necesita que un solo elemento sea calificado por muchas personas, sino que también requiere que las personas compartan varios elementos calificados, que haya una superposición en los elementos que califican. Considere el dominio de las noticias. Muchos aparecen por día, y muchos probablemente solo sean interesantes por unos días (Agüero Dejo, Chumacero Delgado, & Delgado Soto, 2019).



Para que un sistema de CF me genere una predicción con respecto a una noticia de reciente aparición, un algoritmo típico de CF requiere que a) uno o más usuarios hayan calificado la historia y b) estos usuarios también hayan calificado algunas otras historias que yo también he clasificado. En un dominio como las noticias, las historias son más interesantes cuando son nuevas, frescas y, desafortunadamente, no es tan probable que hayan sido calificadas por un gran número de personas. Todo esto significa que, si los elementos solo son importantes por poco tiempo, estos requisitos son difíciles de cumplir.

La CF ha tenido más éxito en dominios donde los gustos de los usuarios no cambian rápidamente: por ejemplo, películas, libros y electrónica de consumo. Si los gustos cambian con frecuencia o rápidamente, las clasificaciones más antiguas pueden ser menos útiles. Un ejemplo podría ser la ropa, donde el gusto de alguien de hace cinco años puede no ser relevante (Varela, Aguilar, Monsalve-Pulido, & Montoya, 2020).

Las propiedades de las secciones anteriores representan simplificaciones del mundo donde la CF se aplica con mayor facilidad. De hecho, la aplicación de CF en dominios donde estas propiedades no se mantienen puede proporcionar aplicaciones interesantes y áreas de investigación interesantes. Por ejemplo, se podría intentar aplicar CF a elementos no homogéneos utilizando recomendaciones restringidas o aplicando restricciones externas (llamadas

reglas comerciales en el mundo empresarial). Del mismo modo, para realizar tareas del sistema para elementos no persistentes, se puede intentar aplicar el filtrado de contenido, que se explora en la siguiente sección.

## 2.9. Algoritmos de filtrado colaborativo: teoría y práctica

Durante la última década, los algoritmos de filtrado colaborativo han evolucionado desde algoritmos de investigación que capturan intuitivamente las preferencias de los usuarios hasta algoritmos que satisfacen las demandas de rendimiento de las grandes aplicaciones comerciales.

En esta sección se mencionan algunos de los algoritmos de filtrado colaborativo más conocidos. Aunque una gran cantidad de literatura teórica describe los algoritmos de FQ, hay poca información disponible para ayudar a los profesionales en la construcción de sistemas de FQ. Se destaca no solo la definición teórica de estos algoritmos, sino también sus desafíos prácticos y, en su caso, sugerimos técnicas para abordar estos desafíos.

Breese y otros describen los algoritmos CF como separables en dos clases: algoritmos basados en memoria que requieren que todas las calificaciones, elementos y usuarios se almacenen en la memoria y algoritmos basados en modelos que periódicamente crean un resumen de los patrones de calificaciones fuera de línea.



Para autores como Ortega y otros (2016) & Hernando, Bobadilla y Ortega (2016) los sistemas de recomendación de filtrado colaborativo pueden ser clasificados en tres grupos.

- Métodos basados en memoria: actúan directamente sobre la matriz de votos. La implementación más popular es KNN, que realiza las recomendaciones usando los k vecinos más similares del usuario dado.
- Métodos basados en modelos: usan un modelo para generar recomendaciones. La implementación más popular es Matrix Factorization.
- Métodos híbridos: realizan la combinación de las técnicas de los dos métodos anteriores.

Los modelos basados en memoria pura no se escalan bien para aplicaciones del mundo real. Por lo tanto, casi todos los algoritmos prácticos utilizan alguna forma de cálculo previo para reducir la complejidad del tiempo de ejecución. Como resultado, los algoritmos prácticos actuales son algoritmos basados en modelos puros o un híbrido de algunos cálculos previos combinados con algunos datos de clasificación en la memoria.

Aquí, se explora una organización diferente de algoritmos de filtrado colaborativo: algoritmos no probabilísticos y algoritmos probabilísticos. Se considera que los algoritmos son probabilísticos si se basan en un modelo probabilístico subyacente. Es decir, representan distribuciones de probabilidad

cuando se calculan calificaciones pronosticadas o listas de recomendaciones clasificadas. En general, los profesionales utilizan mucho los modelos no probabilísticos. Sin embargo, los modelos probabilísticos han ido ganando popularidad, especialmente en la comunidad de aprendizaje automático.

Los modelos probabilísticos han sido desarrollados tanto para ser usados en enfoques de filtrado colaborativo (ejemplo: PMF, BNMF), como en aquellos basados en contenidos. Los modelos probabilísticos son una alternativa para modelos de recomendación que permitan justificar correctamente sus predicciones.

Los modelos probabilísticos son aplicables cuando el proceso de recomendación debe seguir modelos de comportamiento del usuario. El objetivo del aprendizaje en los modelos probabilísticos es estimar la función de densidad de probabilidad de los datos o la distribución de probabilidad, para lo cual es necesario hacer ciertas suposiciones sobre lo siguiente.

- El modelo de distribución que describe los atributos: es muy importante conocer con qué atributos cuenta nuestro conjunto de entrenamiento para asumir el tipo de distribución que estos atributos representan. Por ejemplo, una distribución multinomial es comúnmente utilizada cuando los atributos

son discretos, y cuando los atributos son continuos, estos son distribuidos mediante una distribución normal.

- El modelo de distribución que describe las clases: presupone conocer cuál es la salida esperada del algoritmo, si esta va ser binaria o categórica. En caso de que la clase sea discreta se utiliza una distribución multinomial. Además, ahí se puede considerar como realizar la estimación de la clase, la cual puede ser con una distribución posterior, es decir considerar o utilizar el método de estimación de máxima verosimilitud.
- La dependencia entre las variables: aquí puede darse el caso de que todas las variables sean independientes, que haya variables que tengan independencia y otras no, que las variables sean dependientes, etc. De esto depende el método que se utilice para el modelado. Por ejemplo, si todas las variables son independientes conocido el valor de la clase, entonces se puede utilizar el método *naives bayes*.

## 2.10. Algoritmos no probabilísticos

Los algoritmos de CF más conocidos son los algoritmos de clientes más cercanos. Introducimos las dos clases diferentes de algoritmos de CF de cliente más cercano: cliente más cercano basado en el usuario y cliente más cercano basado en elementos. También exploramos más brevemente algoritmos no



probabilísticos que transforman o agrupan el espacio de calificaciones para reducir la dimensionalidad del espacio de calificaciones.

## 2.11. Algoritmo ALS

Entre todas las múltiples técnicas utilizadas para implementar sistemas de recomendación, el filtrado colaborativo, que se basa en comparar el perfil de preferencias de los usuarios, es una técnica muy popular en las aplicaciones de comercio electrónico, debido a sus buenos resultados.

Los enfoques basados en vecindarios presentan problemas de escalabilidad, dado que el algoritmo tiene que procesar todos los datos para calcular una sola predicción. Por lo tanto, si hay una gran cantidad de usuarios y elementos, estos enfoques pueden no ser apropiados para los sistemas en línea que recomiendan en tiempo real.

Además, estos algoritmos son más sensibles que los basados en modelos a algunos problemas comunes de los sistemas de recomendación. Un problema común es la escasez de la matriz que almacena las calificaciones que representan las preferencias de los usuarios sobre los elementos disponibles.

Se refiere a una situación en la que los datos transaccionales o de retroalimentación son escasos e insuficientes para identificar similitudes en los

intereses de los usuarios, lo que hace difícil y poco fiable predecir qué consumidores son similares.

Otro problema recurrente en la generación de recomendaciones ocurre cuando deseamos recomendar elementos que nadie en la comunidad ha calificado o interactuado todavía. Esto como se mencionó anteriormente, se conoce como el problema del arranque en frío, y el filtrado colaborativo puro no puede ayudar en un entorno de arranque en frío, ya que no se dispone de información sobre las preferencias del usuario que sirva de base para las recomendaciones.

No obstante, existen modelos que pueden ayudar a cerrar la brecha entre los elementos existentes y los nuevos, al inferir similitudes entre ellos. Los enfoques basados en modelos, en lugar de utilizar directamente las calificaciones almacenadas, como los sistemas basados en el vecindario, utilizan calificaciones para aprender un modelo predictivo.

El proceso de construcción del modelo se realiza mediante diferentes algoritmos de aprendizaje automático como las redes bayesianas, las redes neuronales y la descomposición de valores singulares. Estos enfoques tienden a ser más rápidos en el tiempo de predicción que los enfoques basados en el vecindario. Sin embargo, la construcción del modelo es una tarea compleja que exige la estimación de una multitud de parámetros y, por lo general, requiere una cantidad de tiempo considerable.



Estos problemas se hacen más evidentes cuando se trata de construir sistemas de recomendación asociados a sitios web que tienen una gran cantidad de usuarios y elementos y, por lo tanto, asociados a grandes bases de datos. Los sistemas en línea exigen una alta disponibilidad y un tiempo de respuesta corto, ya que deben integrar y procesar rápidamente los flujos entrantes de datos de las actividades de todos los usuarios para generar las recomendaciones.

Todos estos procesos deben ocurrir con una latencia de segundos, ya que los elementos más prometedores seleccionados por los algoritmos de recomendación deben mostrarse a los usuarios mientras aún están navegando por el sitio web. Cuanto mayor sea la cantidad de usuarios a servir y los elementos de los que recomendar, mayor será la cantidad de procesamiento requerido, lo que aumenta el tiempo que lleva generar cada recomendación.

La plataforma de música digital Spotify es un ejemplo práctico de un sistema de recomendación en línea con alta demanda: su servicio de personalización de música tiene más de 50 millones de usuarios activos, 30 millones de canciones catalogadas y alrededor de 20 mil nuevas canciones agregadas por día.

Amazon genera recomendaciones a partir de una base de datos con 253 millones de productos para usuarios de 270 millones de cuentas de clientes activas. Un enfoque eficiente es esencial en todos esos casos. Hoy en día, para abordar estos desafíos de desempeño, los sistemas de recomendación en línea

han combinado dos estrategias: (i) algoritmos eficientes, que evitan la complejidad computacional de calcular cada una de las entradas de la matriz de alta dimensión y dispersión; y (ii) almacenamiento y procesamiento de datos optimizados. Esto significa procesar información en tiempo real para construir un modelo predictivo y presentar su salida en segundos.

Para resolver este problema, algunos autores han desarrollado una clase de algoritmos de filtrado colaborativo basados en modelos que son rápidos y fáciles de calcular, llamados modelos de factores latentes. Intentan identificar características relevantes (factores latentes) que explican las calificaciones observadas. Estas características se pueden interpretar como la preferencia de los usuarios y las características de los artículos que se recomiendan.

Usando estos factores latentes, es posible inferir la preferencia del usuario y hacer una recomendación de los mejores artículos para él/ella. Las técnicas más exitosas para realizar el modelado de factores latentes se basan en la factorización matricial. Se han popularizado recientemente porque combinan escalabilidad y precisión predictiva y, además, ofrecen flexibilidad para modelar diferentes situaciones reales, siendo superiores a los métodos basados en vecindarios para producir recomendaciones porque permiten la incorporación de información adicional como la retroalimentación implícita, efectos temporales y niveles de confianza.





Trabajos recientes sugieren modelar solo las calificaciones observadas, evitando el sobreajuste, a través de un modelo regularizado adecuado. Se han diseñado algunos algoritmos paralelos para modelos de factores latentes con regularización con el objetivo de mejorar el rendimiento del modelado. Entre ellos, se pueden destacar dos: (i) la factorización matricial de rango bajo con mínimos cuadrados alternos (ALS), que utiliza una serie de broadcast-joins, construido sobre la implementación de MapReduce de código abierto Hadoop y su ecosistema, al que llamamos HadoopMRMahout; y (ii) los mínimos cuadrados alternos con regularización  $\lambda$  ponderada (ALS-WR) que se ha implementado en la biblioteca de aprendizaje automático de Apache Spark, MLLib, que llamamos Spark-MLLib.

La escalabilidad y el rendimiento son cuestiones clave para los sistemas de recomendación, ya que la complejidad computacional aumenta con el número de usuarios y elementos, pero la ganancia de rendimiento para estas implementaciones aún no se ha evaluado sistemáticamente en ningún estudio comparativo.

Aunque los algoritmos de factorización de matrices de ALS no son nuevos, algunos trabajos recientes muestran que evaluar soluciones que pueden ser más rápidas en situaciones específicas, como restricciones de memoria y algunas otras situaciones de alto procesamiento que pueden ocurrir, aún requiere



atención. Los autores Cheng, Peng y Liu (2016) proponen algunas técnicas para encontrar factorización matricial ALS portátil y eficiente para sistemas de recomendación. Aplican la técnica de procesamiento por lotes de subprocesos y tres optimizaciones específicas de la arquitectura para una nueva solución, e implementan un solucionador de ALS en Op enCL para que pueda ejecutarse en varias plataformas (CPU, GPU y MIC).

Además, mencionan una nueva solución de software para mejorar el rendimiento de los sistemas de recomendación, basándose en gran medida en la tecnología Apache Spark para acelerar el cálculo de los algoritmos de recomendación. Este trabajo tiene como objetivo realizar un análisis experimental para comparar dos implementaciones escalables diferentes de los algoritmos de mínimos cuadrados alternos (Spark-MLLib y HadoopMRMahout) para la recomendación de filtrado colaborativo.

Realizamos experimentos para evaluar la precisión de las recomendaciones generadas y el tiempo de ejecución de ambos algoritmos, utilizando conjuntos de datos disponibles públicamente con diferentes tamaños y de diferentes dominios de recomendación.

## 2.12. Filtrado colaborativo basado en modelos

El supuesto fundamental de filtrado colaborativo es que, si los usuarios X e Y califican  $n$  elementos de manera similar, o tienen comportamientos similares (por ejemplo, comprar, mirar, escuchar), por lo tanto, calificarán o actuarán sobre otros elementos de manera similar. Las técnicas de filtrado colaborativo utilizan una base de datos de preferencias de artículos por parte de los usuarios para predecir temas o productos adicionales que podrían gustarle a un nuevo usuario.

El espacio del problema se puede formular como una matriz de usuarios frente a elementos, y cada celda representa la calificación de un usuario en un elemento específico. Esta matriz se denominará matriz de calificaciones a partir de ahora. Bajo esta formulación, el problema es predecir los valores para celdas vacías específicas. En el filtrado colaborativo, esta matriz suele ser muy escasa, ya que cada usuario solo califica un pequeño porcentaje del total de elementos disponibles.

Para completar las entradas que faltan en la matriz de calificaciones, los modelos se aprenden ajustando las calificaciones observadas anteriormente. Una vez que el objetivo es generalizar estas calificaciones observadas de una manera que nos permita predecir calificaciones futuras desconocidas, se debe tener cuidado para evitar sobre ajustar los datos observados.

Esto se puede lograr modelando los factores latentes de la matriz de calificaciones; es decir, encontrando un pequeño conjunto de características latentes que expliquen las calificaciones observadas y describan las características generales de los usuarios y los elementos. Las técnicas más exitosas para modelar factores latentes se basan en la factorización matricial, porque combinan escalabilidad y precisión predictiva.

La factorización matricial se ha convertido en una de las técnicas principales para dar solución a los problemas de escalabilidad. La factorización matricial es un método basado en modelos, donde los votos del usuario a ítems se modelan con un conjunto de factores latentes, que representan las características de los usuarios e ítems. Algunas implementaciones de este método son: factorización matricial con *bias* (BiasedMF), factorización matricial probabilística (PMF), factorización matricial no negativa (NMF) y factorización matricial no negativa bayesiana (BNMF).

### 2.13. Sistemas de recomendación basados en filtrado colaborativo

Los sistemas de recomendación se han convertido en una herramienta fundamental para la sociedad de la información. Su incorporación a la vida cotidiana ha permitido a los prestadores de servicios paliar el problema de la sobrecarga de información al que están expuestos los ciudadanos. Cada minuto, se publican cientos de horas de video en YouTube, se compran miles de

productos en Amazon, se publican decenas de miles de tweets y se envían millones de mensajes en servicios como WhatsApp o Telegram.

Los sistemas de recomendación, además, permiten a sus usuarios hacer una aplicación inteligente de esta información filtrando aquellos contenidos que no les son relevantes y promocionando aquellos que puedan ser de su interés. Una de las principales líneas de investigación en sistemas de recomendación es el desarrollo de modelos novedosos capaces de capturar información más profunda y sutil de los datos, y explotarla para ofrecer recomendaciones más precisas y relevantes.

Muchas de estas propuestas están orientadas a mejorar los sistemas, el llamado método de filtrado colaborativo. Hoy en día, el estándar de facto en los métodos de sistema de recomendación basados en filtrado colaborativo es la factorización matricial.

Esta se basa en el supuesto de que las preferencias de los usuarios por elementos particulares se caracterizan por una colección de características ocultas, denominadas factores latentes. El método, se encarga de desvelar estos factores latentes, de medirlos y, lo más importante, de aprovecharlos para brindar predicciones precisas y relevantes. Por estas razones, el proceso de extracción y refinamiento de factores latentes es un área de investigación muy activa en los sistemas de recomendación.



Gracias a ello, los autores son capaces de seguir la evolución de las preferencias de los usuarios a lo largo del tiempo, ajustando consecuentemente los factores latentes para brindar recomendaciones precisas y actualizadas. Esto es particularmente relevante en el dominio de la recomendación musical, donde las preferencias de los usuarios son muy cambiantes y las afinidades evolucionan rápidamente con las tendencias musicales predominantes.

Por tanto, los sitios web de comercio electrónico han florecido rápidamente y permiten la venta de millones de artículos. La elección de un elemento de este gran número de elementos hace necesario el uso de la herramienta complementaria sistema de recomendación. Además, proporciona una alternativa para descubrir elementos que los usuarios podrían no haber encontrado por sí mismos. Recopila información del usuario sobre los elementos que prefiere y luego sugiere esos elementos.

Ejemplos famosos de los sistemas de recomendación más utilizados basados en el enfoque de filtrado colaborativo, son aquellos que utilizan varias empresas de comercio electrónico, incluidas Yelp, Netflix, eBay y Amazon. La corriente principal de las técnicas se basa en los elementos comunes entre los usuarios. Los usuarios o elementos análogos se descubren calculando las similitudes de las calificaciones comunes de los usuarios. Sin embargo, su eficacia se ve afectada cuando se produce el problema de la escasez de



calificaciones, por la razón de que con frecuencia hay un número de calificaciones comunes restringido entre los usuarios.

Otra limitación es que los enfoques de filtrado colaborativo no captan la razón de las calificaciones del usuario y, por lo tanto, no pueden captar con precisión la preferencia de un usuario objetivo. Para hacer frente a estos problemas, se han desarrollado varios métodos basados en el contenido para representar a los usuarios y elementos mediante diversos tipos de datos, incluidas etiquetas, descripciones de elementos y factores sociales.

Después de todo, estas técnicas todavía son deficientes, particularmente cuando el grado de escasez de calificación es mayor, o el usuario objetivo no tiene muchas calificaciones históricas. Con el escenario actual de la Web, los usuarios se sienten cada vez más cómodos expresándose y compartiendo sus puntos de vista sobre los elementos de las plataformas electrónicas utilizando revisiones textuales.





## CAPÍTULO 3. MARCO TEÓRICO



## Capítulo 3. Marco teórico

### 3.1. Enfoque de investigación

Para el presente estudio, se tomó como base un enfoque cualitativo, ya que, lo que permitió interpretar eficientemente los resultados obtenidos del análisis de predicciones basados en algoritmo ALS en una empresa de comida rápida, y la utilización de este enfoque permitió generar una mejor visión sobre, cómo desarrollar el estudio, facilitando el logro de los objetivos planteados.

Por ende, el enfoque cualitativo, permitió recopilar toda la información que se usó como base para este estudio, así como revisar fuentes bibliográficas acerca de los Algoritmos de filtrado colaborativo, basadas con la revisión documental.

### 3.2. Tipo de investigación

En cuanto al tipo de investigación, se ha tomado en cuenta la descriptiva, debido a al tipo de análisis que se realizó sobre la base de la predicción del filtro colaborativo en la empresa de comida rápida, diagnosticando e interpretando la situación actual de los procesos que mantiene la empresa, es decir, se detallaron todos los factores que influyen en el proceso de Algoritmo ALS, con la realidad de analizar la técnica que se emplea frente a resultados integrales.



Este tipo de investigación, de igual manera, aportó con procesos prácticos y lógicos, mediante la identificación de las características sobre la base de los Algoritmos ALS, enfocados a lo social, económico y ambiental de la empresa relacionada al servicio de comida rápida, ya que, se encargó de puntualizar dichas particularidades dentro del estudio.

Por otro lado, se tomó en cuenta el tipo de investigación correlacional, ya que se centró en los procedimientos en los cuales se determinó la relación que existe entre el objeto de estudio, llegando a conocer las situaciones que predomina en la empresa, a través de la descripción exacta de procesos, además se utilizó para descubrir nuevos acontecimientos sobre las compras y calificaciones que se dieron dentro de la empresa que ofrece el servicio de comida rápida.

### **3.3. Diseño narrativo**

Mediante esta investigación se especifica la funcionalidad que muestra el algoritmo colaborativo teniendo como fundamentación teórica, artículos y referencias bibliográficas, que consiente en establecer cómo se puede emplear este algoritmo para una empresa de comida rápida.

### **3.4. Sistemas de recomendación**

Los sistemas de recomendación emplean métodos de descubrimiento de conocimiento para solucionar el problema de originar recomendaciones

personalizadas para la información, relacionados a productos, servicios, etc., que afectan el interés del usuario. Estos sistemas utilizan las preferencias del usuario para recomendarles ítems o artículo que aún no conocen.

Un artículo es cualquier entidad empleada como objeto de recomendación, definida en el dominio del sistema, estos pueden ser platos de comida, libros, películas, documentos, incluso personas.

Los sistemas comúnmente recomendados se encuentra Amazon, compañía estadounidense que ofrece sugerencia de forma personalizada en la compra de productos varios a través del internet; Facebook, la red social de mayor preferencia en el mundo con más 850 millones de usuarios registrados, tiene la capacidad de convocar a un conjunto de personas con intereses comunes, así como la función de búsqueda y sugerencias de amigos.

### **3.5. Recolección de datos**

La selección de datos es uno de los trabajos fundamentales dentro del sistema de recomendación, para crear aquello existen dos formas de recopilar información.

Explícita, donde el usuario formula de forma clara y concisa su interés por uno o un grupo de ítems o artículos, por ejemplo:

- Pedir al usuario que examine a partir de una escala proporcionada, algún tema en particular.
- Solicitar al usuario que pondere un conjunto de temas de una lista de temas preferidos.
- Pedir al usuario de que seleccione de una lista de temas, los temas que él cree preferido.

Implícita, donde el sistema selecciona información, sin contar, con la intervención continúa del usuario como, por ejemplo:

- Guardar el historial de ítems visitados por un usuario.
- Analizar el número de visitas que recibe un ítem o artículo.
- Analizar el comportamiento del usuario en las redes sociales para así conocer sus gustos y preferencias.

### **3.6. Sistemas de recomendación basado en filtro colaborativo**

Existen un sinnúmero de técnicas de procesamiento de recomendaciones, implementada por estos sistemas, entre los cuales cabe indicar a los basado en contenido, utilidad, conocimiento y filtro colaborativo.

Los sistemas establecidos en filtro colaborativo tienen como idea principal proporcionar ítems recomendados en base a la opinión o preferencia que tienen

otros usuarios, para eso mapea usuarios con intereses comunes para luego establecer las recomendaciones basada en las relaciones creadas.

### 3.6.1. Enfoque del algoritmo de filtro colaborativo

Se muestran dos tipos de enfoque para el filtro colaborativo, basados en memoria y basados en modelo. Los algoritmos basados en memoria manejan la base de datos que contiene una relación entre usuarios, ítems y valoración para formar las predicciones. Emplea métricas como correlaciones, coeficientes para establecer la similitud que existe con el usuario, es decir un historial de valoraciones sobre los elementos, comunes al del usuario actual y así poder deducir con un o una lista de artículos recomendados para el usuario activo.

Los algoritmos basados en modelo inicialmente emplean un modelo de valoraciones del usuario, presentan la problemática como una de predicción estadística, calculando el valor deseado para cada ítem en función de las valoraciones anteriores.

### 3.6.2. Enfoque de los algoritmos basados en memoria

Existen dos enfoques, uno de ellos, conocido como “basado en el usuario”, recomienda al usuario los ítems que aún no ha sido evaluados por él, pero que han sido evaluados por otros usuarios con gustos similares.

Este enfoque evalúa el interés de un usuario objetivo por un ítem utilizando los votos efectuados por otros usuarios de ese mismo ítem. El proceso de este enfoque es el siguiente.

1. Cálculo de similitudes: con una medida se calcula la similitud entre el usuario objetivo y cada uno de los demás usuarios de la base de datos.

2. Selección de K-vecinos: se seleccionan los usuarios más similares al usuario objetivo. Los usuarios más similares toman el nombre de K-vecinos, K representa la cantidad determinada de vecinos necesarios para realizar la predicción.

3. Predicción: mediante una función de agregación, normalmente el promedio de los votos de los K-vecinos, se predice la valoración que el usuario objetivo le daría a cada ítem no votado previamente.

4. Recomendación: se recomiendan los N-ítems con valores más altos de predicción, N representa la cantidad requerida de recomendaciones.

Otro de los enfoques, es aquel “basado en el ítem”, se basa en el hecho que a las personas les suele gustar cosas similares a las que ya les gusta, por lo tanto, recomienda a un usuario los ítems similares a los que ha preferido antes. Este enfoque evalúa el voto de un usuario por un ítem basado en los votos de los usuarios en ítems similares.

El proceso de este enfoque es el siguiente.

1. Cálculo de similitudes: con una medida se calcula la similitud entre el ítem objetivo y cada uno de los demás ítems de la base de datos.
2. Selección de K-vecinos: se seleccionan los ítems más similares al ítem objetivo.
3. Predicción: mediante una función de agregación, se predice la valoración del ítem objetivo para un usuario específico.
4. Recomendación: se determina si el ítem es o no es del gusto del usuario.

### 3.7. Métrica de similitud

La métrica de similitud es aquella medida que establece el grado de similitud entre dos objetos o artículos. Se pueden hallar como correlaciones, coeficientes, distancia o matriz de similitud. Es un sistema de recomendación que se emplea para identificar a los usuarios e ítems similares entre sí o que tengan características comunes.

La idea básica es escoger dos objetos  $i$  y  $j$ , aislar a los usuarios que los han calificado en común y luego aplicar una técnica para determinar la similitud  $s_{i,j}$ . Para que pueda calcularse la similitud entre dos usuarios  $u_i$  y  $u_j$ , deben cumplirse dos condiciones: los usuarios deben tener ítems calificados y, al menos, un elemento debe haber sido evaluado en común. Si alguna de estas



condiciones no se cumple, entonces debe trabajarse con otras técnicas que permitan realizar recomendaciones, por ejemplo, utilizar una base de conocimientos, o crear un perfil del usuario.

Las métricas que se describen a continuación son el coeficiente de correlación de Pearson y la distancia euclidiana que son las empleadas en las pruebas y análisis de algoritmos en este proyecto porque toman en cuenta las valoraciones elaboradas por los usuarios.

### 3.7.1. Coeficiente de correlación de Pearson

Es una métrica con dependencia lineal entre dos variables que poseen características aleatoria que se diferencia a la covarianza, la correlación de Pearson es autónoma de la escala de medida de las variables.

La métrica informal consigue identificar al coeficiente de correlación de Pearson como un índice que puede ser manejado para comprobar el grado de relación de dos variables, siempre y cuando, ambas sean cuantitativas y continuas.

Si  $u$  y  $v$  son dos usuarios, la igualdad entre  $u$  y  $v$  se obtiene a través de la expresión:

$$sim(u, v) = \frac{\sum_{i=1}^m (r_{u,i} - r_u^-)(r_{v,i} - r_v^-)}{\sigma_u \sigma_v}$$

La literatura presenta entre otras, dos variaciones a la correlación de Pearson: la versión ponderada y la variación basada en la función sigmoidea. La versión ponderada que se refiere a la utilización de pesos asignados a los sujetos en el cálculo de un coeficiente de correlación entre dos variables X e Y. Los pesos pueden estar disponibles de forma natural de antemano o pueden ser elegidos por el usuario para cumplir un propósito específico.

La variación basada en la función sigmoidea, por su parte, se propuso para evitar favorecer el tamaño de los usuarios comunes. Esta considera el tamaño del conjunto de los usuarios comunes en la métrica de similitud del elemento para reducir el error.

Una vez se tienen los pesos de correlación de cada algoritmo hay que saber la confiabilidad de estos pesos. Es posible tener un alto grado de correlación con vecinos con los que se comparten pocos elementos valorados por el usuario actual, pero con igual valoración.

El uso de estos pesos proporciona unas estimaciones malas, puesto que para tener una idea real de la correlación es necesario varios votos compartidos. En los casos de pocas muestras es recomendable disminuir el factor de correlación en función del número de votos compartidos. Para intentar mejorar más los pesos de la correlación entre usuarios se puede entrar a trabajar con la varianza

de los elementos que cada usuario ha votado. Si un elemento es votado positivamente por un gran porcentaje del conjunto de usuarios, el que dos usuarios compartan esa votación dice poca información acerca de la correlación entre usuarios. Lo contrario pasa en el caso de elementos que sean votados positiva o negativamente por pocos usuarios. Para tener en cuenta este hecho se añade a la fórmula de la correlación de Pearson un término con la varianza del elemento.

### 3.7.2. Distancia euclidiana

Esta métrica de similaridad calcula la distancia entre dos usuarios  $X$  y  $Y$ . Pensando en los ítems como las dimensiones y las preferencias como puntos a lo largo de esas dimensiones, la distancia se mide utilizando todos los ítems donde ambos usuarios han expresado su preferencia por ese tema. Es absolutamente la raíz cuadrada de la suma de los cuadrados de las diferencias en la posición (de preferencia) a lo extenso de cada dimensión.

La similaridad puede ser calculada como  $1 / (1 + a \text{ distancia})$ , por lo que los valores resultantes están en el rango  $(0,1]$ .

## 3.8. Metodología

### 3.8.1. Metodología KDD

El Descubrimiento de Conocimiento en Bases de Datos (KDD Knowledge Discovery in Databases) establece el primer modelo que precisa el descubrimiento de conocimiento en bases de datos como un “proceso”, combinado por distintas etapas y fases que van desde la selección de los datos hasta la interpretación y evaluación de los resultados.

Fayyad define a KDD como el “proceso no trivial de identificar patrones válidos, novedosos, potencialmente ventajosos y, en última instancia, entendibles en los datos” (Fayyad,1996).

El KDD es básicamente un proceso automático en el que se combinan descubrimiento y análisis. El proceso consiste en extraer patrones en forma de reglas o funciones, a partir de los datos, para que el usuario los analice. Esta tarea implica generalmente preprocesar los datos, hacer minería de datos (data mining) y presentar resultados.

KDD se puede aplicar en diferentes dominios, por ejemplo, para determinar perfiles de clientes fraudulentos (evasión de impuestos), para descubrir relaciones implícitas existentes entre síntomas y enfermedades, entre características técnicas y diagnóstico del estado de equipos y máquinas, para

determinar perfiles de estudiantes “académicamente exitosos” en términos de sus características socioeconómicas y para determinar patrones de compra de los clientes en sus canastas de mercado.

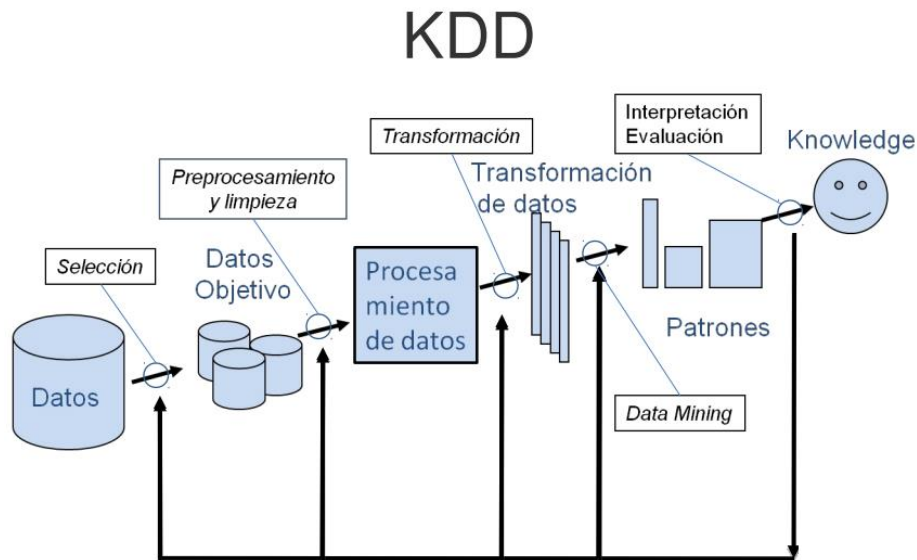
El término proceso cuenta con una secuencia iterativa de etapas o fases que lo componen. Los patrones corresponderían ser válidos para nuevos datos, novedosos en el sentido que deberían aportar nuevo conocimiento al dominio de aplicación y potencialmente útiles para el usuario final o tomador de decisiones.

KDD es un proceso interactivo, ya que la salida de alguna de las fases puede retroceder a pasos anteriores y porque a menudo pueden ser necesarias para varias iteraciones para extirpar conocimiento de alta calidad. El usuario es muy participativo, o más generalmente un experto en el dominio del problema, debe validar la preparación de los datos y verificar el del conocimiento extraído. Las cinco fases que constituyen esta metodología son las siguientes.

- Selección de los datos sobre los que se trabajará.
- Preprocesamiento de los datos, donde se realiza un tratamiento de los datos incorrectos y ausentes.
- Transformación de los datos y reducción de la dimensionalidad.
- Minería de datos, donde se obtienen los patrones de interés según la tarea de minería que llevemos a cabo (descriptiva o predictiva).

- Interpretación y evaluación del nuevo conocimiento en el dominio de aplicación.

Figura 1. Metodología KDD



Fuente: Datos de la investigación

### 3.8.2. Desarrollo de las fases KDD

Abstracción del escenario

No todo valor numérico es matemática o estadística, sino comprender la problemática que toca enfrentar y tener contexto para plantear soluciones viables y reales. Es primordial tener conocimiento sobre las propiedades, limitaciones y reglas del escenario en estudio, para posteriormente precisar las metas a alcanzar.

## Selección de los datos

Del conjunto de datos recolectados y ya definidos los objetivos por lograr, se deben seleccionar los datos disponibles para efectuar el estudio e integrarlos en uno solo que puedan beneficiar a alcanzar a los objetivos del análisis. Muchas veces esta información puede localizarse en una misma fuente (centralizado) o pueden estar distribuidos.

## Limpieza y preprocesamiento

Esta fase se determina la confiabilidad de la información; es decir, ejecutar las tareas que certifiquen la utilidad de los datos. Para esto se hace la duración de datos (tratamiento de datos extraviados o remover valores atípicos). Esto implica separar variables o atributos con datos faltantes o quitar información no útil para este tipo de tareas como el texto (aunque puede utilizarse para hacer Minería de Texto, que es otro asunto).

Lo anterior presupone analizar la calidad de los datos, para ello se aplican operaciones básicas como la remoción de datos ruidosos, se seleccionan estrategias para el manejo de datos desconocidos, datos nulos, datos duplicados y técnicas estadísticas para su reemplazo. En esta etapa, es de suma importancia la interacción con el usuario o analista. Los datos ruidosos son valores que están significativamente fuera del rango de valores esperados; se

deben principalmente a errores humanos, a cambios en el sistema, a información no disponible a tiempo y a fuentes heterogéneas de datos. Los datos desconocidos *empty* son aquellos a los cuales no les corresponde un valor en el mundo real y los *missing* son aquellos que tienen un valor que no fue capturado. Los datos nulos son datos desconocidos que son permitidos por los sistemas gestores de bases de datos relacionales (*sgbdr*). En el proceso de limpieza todos estos valores se ignoran, se reemplazan por un valor por omisión, o por el valor más cercano; es decir, se usan métricas de tipo estadístico como media, moda, mínimo y máximo para reemplazarlos.

## Transformación de los datos

En esta fase se optimiza la calidad de los datos con transformaciones que implican ya sea reducción de dimensionalidad (reducir la cantidad de variables del conjunto de datos) o bien transformaciones, como por ejemplo, transformar los valores que son números a categóricos (discretización).

La reducción horizontal implica la eliminación de tuplas idénticas como producto de la sustitución del valor de un atributo por otro de alto nivel, en una jerarquía definida de valores categóricos o por la discretización de valores continuos (por ejemplo, edad por un rango de edades). La reducción vertical implica la eliminación de atributos que son insignificantes o redundantes con respecto al problema, como la eliminación de llaves, la eliminación de columnas que



dependen funcionalmente (por ejemplo, edad y fecha de nacimiento). Se utilizan técnicas de reducción como agregaciones, compresión de datos, histogramas, segmentación, discretización basada en entropía, muestreo, entre otras (Han y Kamber, 2001).

### Selección de la apropiada tarea de Minería de Datos

Fase en la que se refiere escoger la tarea apropiada de la Minería de Datos, puede ser la clasificación, regresión o agrupación, según los objetivos que se haya planteado para la investigación (predicción o descripción), la primera ocupada para localizar el modelo que va ser utilizado para casos futuros y desconocidos; tanto que la segunda solo va observar su comportamiento.

### Elección del algoritmo de Minería de Datos

Posteriormente, se procede a seleccionar la técnica o algoritmo, o incluso más de uno para la búsqueda del patrón y obtener conocimiento. El meta-aprendizaje se enfoca en explicar la razón por la que un algoritmo funciona mejor en determinadas problemáticas, y para cada técnica existen diferentes posibilidades de cómo seleccionarlas. Cada algoritmo tiene su propia esencia, su propia manera de trabajar y obtener los resultados, por lo que es recomendable conocer las propiedades de aquellos candidatos a utilizar y ver cual se ajusta mejor a los datos. En 2015, se publicó un artículo que intenta abordar justamente este

problema, realizando una comparación entre diferentes clasificadores en distintas problemáticas.

### Aplicación del algoritmo

Por fin, una vez seleccionado las técnicas, el paso siguiente es aplicarlo a los datos ya seleccionados, limpiados y procesados. Es posible que la ejecución de los algoritmos sean varias, intentando ajustar los parámetros que optimicen los resultados. Estos parámetros varían de acuerdo al método seleccionado.

### Evaluación

Una vez aplicado los algoritmos al conjunto de datos, procedemos a evaluar los patrones que se generaron y el rendimiento que se obtuvo para verificar que cumpla con las metas planteadas en las primeras fases. Para realizar esta evaluación existe una técnica que se llama Validación Cruzada, la cual realiza una partición de los datos dividiéndose en entrenamiento (que servirán para crear el modelo) y prueba (que serán utilizados para ver que en verdad funciona el algoritmo y realiza su trabajo bien).

### Aplicación

Si todos los pasos se siguen correctamente y los resultados de la evaluación se satisfacen, la última etapa es simplemente aplicar el conocimiento encontrado al contexto y comenzar a resolver sus problemáticas. Si de lo contrario, los

resultados no son satisfactorios entonces es necesario regresar a las anteriores etapas a realizar algún ajuste, analizando desde la selección de los datos hasta en la etapa de evaluación.

### 3.9. Dataset

El término *dataset* en sí es un término extranjero, un anglicismo, que hemos incorporado a nuestra lengua como un término más en los países hispanohablantes. Su traducción a nuestra lengua sería conjunto de datos y es una colección de datos habitualmente tabulada.

Un conjunto de datos o *dataset* corresponde a los contenidos de una única tabla de base de datos o una única matriz de datos de estadística, donde cada columna de la tabla representa una variable en particular, y cada fila representa a un miembro determinado del conjunto de datos que estamos tratando. En un conjunto de datos o *dataset* tenemos todos los valores que puede tener cada una de las variables, como por ejemplo la altura y el peso de un objeto, que corresponden a cada miembro del conjunto de datos. Cada uno de estos valores se conoce con el nombre de dato. El conjunto de datos puede incluir datos para uno o más miembros en función de su número de filas.

El *dataset* incluye también las relaciones entre las tablas que contienen los datos.



Si nos movemos en el contexto de Big Data, entendemos por *dataset* aquellos conjuntos de datos tan grandes que las aplicaciones de procesamiento de datos tradicionales no los pueden procesar debido a la gran cantidad de datos contenidos en la tabla o matriz.

Podríamos definir un *dataset* como una colección o representación de datos residentes en memoria con un modelo de programación relacional coherente e independientemente sea cual sea el origen de los datos que contiene.

Una de las principales características de los *datasets* es que ya tienen una estructura, a diferencia de los RDD, conocidos como conjuntos de datos desestructurados y definidos como una colección de elementos tolerante a fallos y son capaces de operar en paralelo.

### **3.10. Algoritmo *Recommender Documentation***

Este motor de recomendación no está basado en Hadoop, anteriormente era un proyecto conocido como “Taste”, pero al ser incorporado en uno de los proyectos de Hadoop este adquiere mayor madurez, escalabilidad y flexibilidad. Se convierte así en un algoritmo ideal para trabajar con recomendadores distribuidos. Los algoritmos de recomendación de Mahout no son solo para la plataforma Java, sino que también puede funcionar como un servidor externo



que expone la lógica de la recomendación de su aplicación a través de servicios web y HTTP.

Los paquetes que conforman la arquitectura del algoritmo son los siguientes.

- DataModel (modelo de la base de datos)
- UserSimilarity (Similaridad entre usuarios)
- ItemSimilarity (Similaridad entre ítems)
- UserNeighborhood (vecindario de usuarios)
- Recommender (algoritmo de recomendación)

### 3.11. Plan de general del proyecto

El proyecto se dividirá en las siguientes etapas para facilitar su organización y estimar el tiempo de realización.

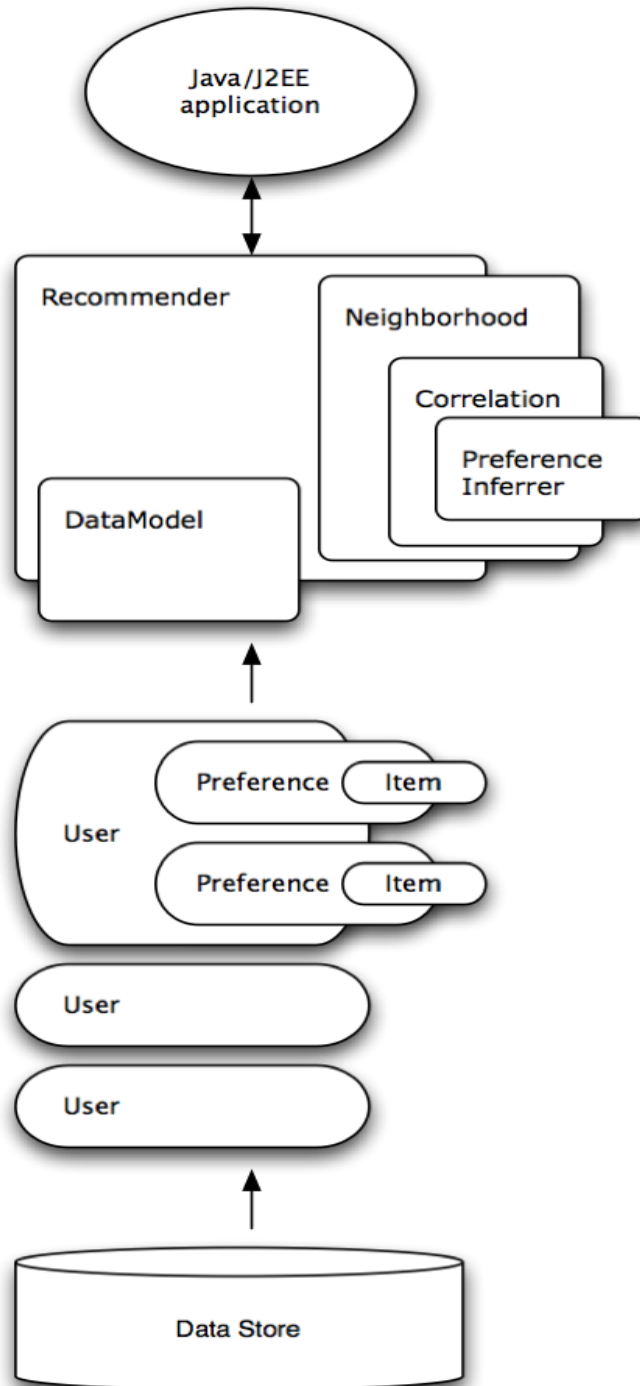
Tabla 1. Plan de general del proyecto

ACTIVIDAD	INICIO	DURACION	FIN
Análisis de la estructura	01/07/2021	14	20/07/2021
Ejecución de consultas	22/07/2021	14	10/08/2021
Preparación de los datos	13/08/2021	30	03/09/2021
Elección de las técnicas de modelado y ejecución	06/09/2021	14	13/09/2021
Análisis de resultados	14/09/2021	5	20/09/2021
Producción de Informes	21/09/2021	5	07/10/2021
Presentación de resultados	10/10/2021	5	24/10/2021

Fuente: Datos de la investigación



Figura 2. Algoritmo *Remmender Document*



Fuente: Datos de la investigación



## CAPÍTULO 4. ANÁLISIS DE LOS RESULTADOS



## Capítulo 4. Análisis de los resultados

### 4.1. Población y muestra

Debido a que esta investigación se enfoca en el análisis de un algoritmo ALS para mejorar los servicios que ofrece una empresa de comida rápida, además de revisión bibliográfica sobre el tema, no fue necesario la definición de la población y muestra.

### 4.2. Resultados

El trabajo al ser un estudio de predicción, donde se sugieren productos a los clientes del local de comida rápida, se presenta un estudio exhaustivo, a través de recomendaciones predictivos mediante una línea de tiempo en diferentes contextos que ofertan las empresas de comida rápida.

Es así que, en base a los productos que se ha ido consumiendo en tiempos anteriores, los mismos clientes fueron quienes han calificado dichos productos ofertados por el local, mediante lineamientos como el uso, orden, presentación y muchos casos la degustación, llegando a darles la puntuación referente a la calidad de los productos.



Para esto, los métodos utilizados en el análisis se dedican a buscar patrones en los datos para estimar o aprender un modelo que sea capaz de realizar predicciones.

Por ende, se ha utilizado la herramienta, denominada RapidMiner, la cual para MicroSystem (2005) es “una herramienta de Minería de Datos ampliamente usada y probada a nivel internacional en aplicaciones empresariales, de gobierno y academia, la cual, implementa más de 500 técnicas de pre - procesamiento de datos, modelación predictiva y descriptiva, métodos de prueba de modelos, visualización de datos, etc.”

Figura 3. RapidMiner

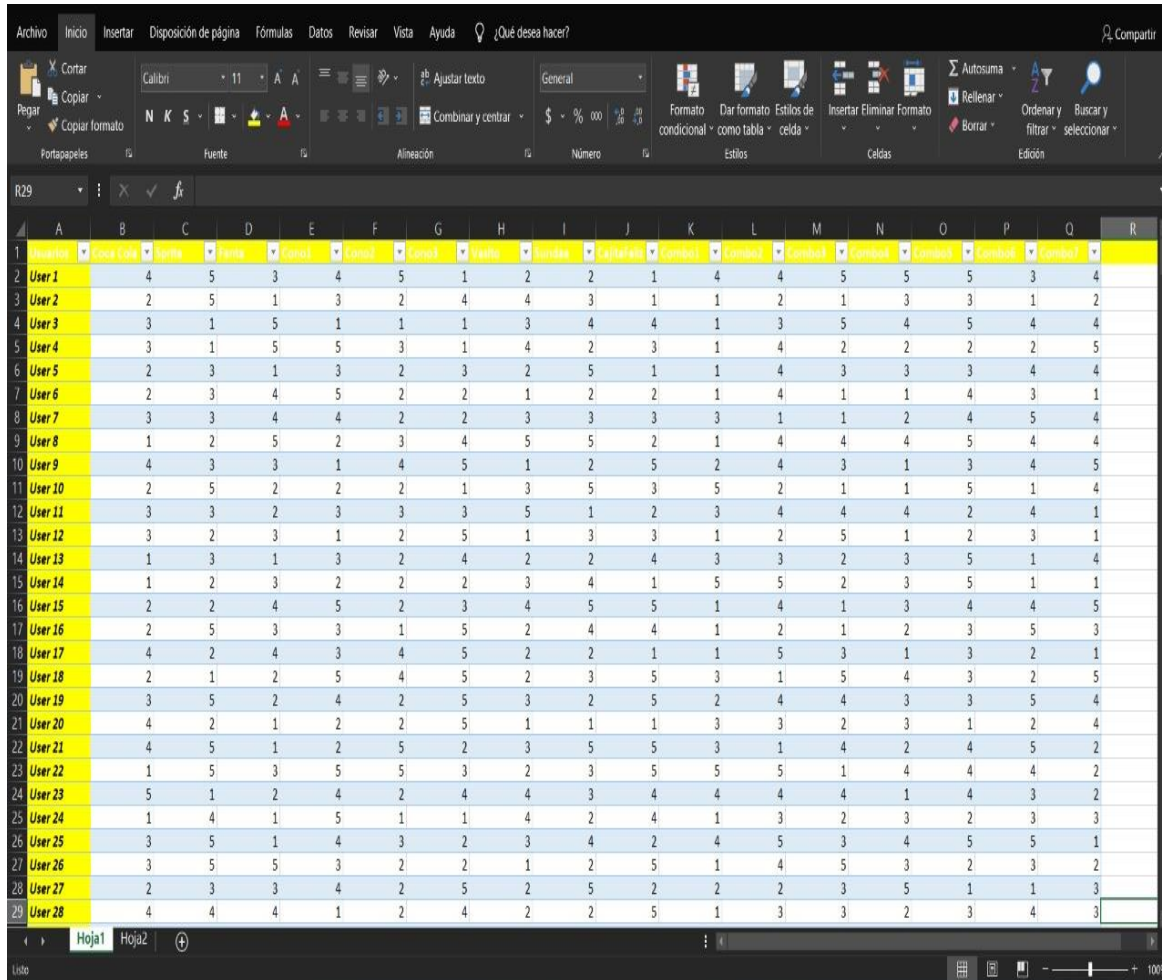


**Fuente:** MicroSystem (2005)

Esta herramienta cuenta con modelos predictivos para todas las empresas, y en el caso del presente estudio se complementó con la base de datos expuestos en Excel misma que se observa en la figura 2, estas herramientas cuentan con los datos de los clientes que visitan el local de comida rápida, al igual que los productos y puntuaciones que se los ha otorgado.



Figura 4. Base de Datos

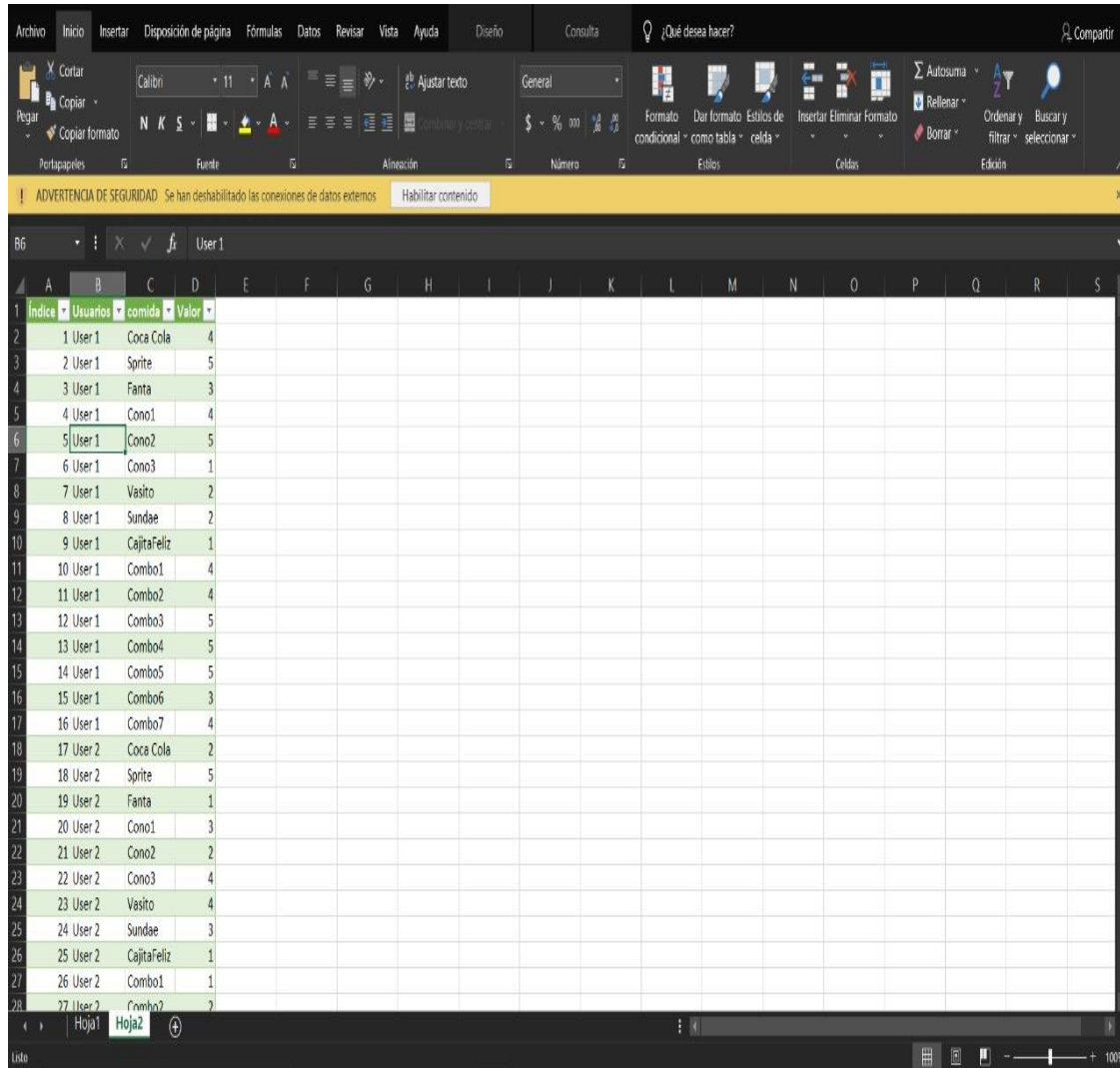


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	Usuario	Comida Rápida	Lechuga	Tomate	Queso	Carne	Carne	Carne	Vegeta	Alitas	Carne	Carne	Carne	Carne	Carne	Carne	Carne	
2	User 1	4	5	3	4	5	1	2	2	1	4	4	5	5	5	3	4	
3	User 2	2	5	1	3	2	4	4	3	1	1	2	1	3	3	1	2	
4	User 3	3	1	5	1	1	1	3	4	4	1	3	5	4	5	4	4	
5	User 4	3	1	5	5	3	1	4	2	3	1	4	2	2	2	2	5	
6	User 5	2	3	1	3	2	3	2	5	1	1	4	3	3	3	4	4	
7	User 6	2	3	4	5	2	2	1	2	2	1	4	1	1	4	3	1	
8	User 7	3	3	4	4	2	2	3	3	3	3	1	1	2	4	5	4	
9	User 8	1	2	5	2	3	4	5	5	2	1	4	4	4	5	4	4	
10	User 9	4	3	3	1	4	5	1	2	5	2	4	3	1	3	4	5	
11	User 10	2	5	2	2	2	1	3	5	3	5	2	1	1	5	1	4	
12	User 11	3	3	2	3	3	3	5	1	2	3	4	4	4	2	4	1	
13	User 12	3	2	3	1	2	5	1	3	3	1	2	5	1	2	3	1	
14	User 13	1	3	1	3	2	4	2	2	4	3	3	2	3	5	1	4	
15	User 14	1	2	3	2	2	2	3	4	1	5	5	2	3	5	1	1	
16	User 15	2	2	4	5	2	3	4	5	5	1	4	1	3	4	4	5	
17	User 16	2	5	3	3	1	5	2	4	4	1	2	1	2	3	5	3	
18	User 17	4	2	4	3	4	5	2	2	1	1	5	3	1	3	2	1	
19	User 18	2	1	2	5	4	5	2	3	5	3	1	5	4	3	2	5	
20	User 19	3	5	2	4	2	5	3	2	5	2	4	4	3	3	5	4	
21	User 20	4	2	1	2	2	5	1	1	1	3	3	2	3	1	2	4	
22	User 21	4	5	1	2	5	2	3	5	5	3	1	4	2	4	5	2	
23	User 22	1	5	3	5	5	3	2	3	5	5	5	1	4	4	4	2	
24	User 23	5	1	2	4	2	4	4	3	4	4	4	4	1	4	3	2	
25	User 24	1	4	1	5	1	1	4	2	4	1	3	2	3	2	3	3	
26	User 25	3	5	1	4	3	2	3	4	2	4	5	3	4	5	5	1	
27	User 26	3	5	5	3	2	2	1	2	5	1	4	5	3	2	3	2	
28	User 27	2	3	3	4	2	5	2	5	2	2	2	3	5	1	1	3	
29	User 28	4	4	4	1	2	4	2	2	5	1	3	3	2	3	4	3	

Fuente: Datos de la investigación

En base a estos datos, se ha dado las predicciones y sugerencias de cada producto ofertado por el local de comida rápida, mismos que fueron otorgados por de cada cliente que visitó el lugar, dando lugar al Algoritmo ALS, que generó ciertas recomendaciones a los clientes que requieren un producto, los cuales fueron en base a sus compras anteriores y dichas las calificaciones se le realizó por dichas compras variadas.

Figura 5. Predicciones



Índice	Usuarios	comida	Valor
1	User 1	Coca Cola	4
2	User 1	Sprite	5
3	User 1	Fanta	3
4	User 1	Cono1	4
5	User 1	Cono2	5
6	User 1	Cono3	1
7	User 1	Vasito	2
8	User 1	Sundae	2
9	User 1	CajitaFeliz	1
10	User 1	Combo1	4
11	User 1	Combo2	4
12	User 1	Combo3	5
13	User 1	Combo4	5
14	User 1	Combo5	5
15	User 1	Combo6	3
16	User 1	Combo7	4
17	User 2	Coca Cola	2
18	User 2	Sprite	5
19	User 2	Fanta	1
20	User 2	Cono1	3
21	User 2	Cono2	2
22	User 2	Cono3	4
23	User 2	Vasito	4
24	User 2	Sundae	3
25	User 2	CajitaFeliz	1
26	User 2	Combo1	1
27	User 2	Combo2	2
28	User 2	Combo3	2

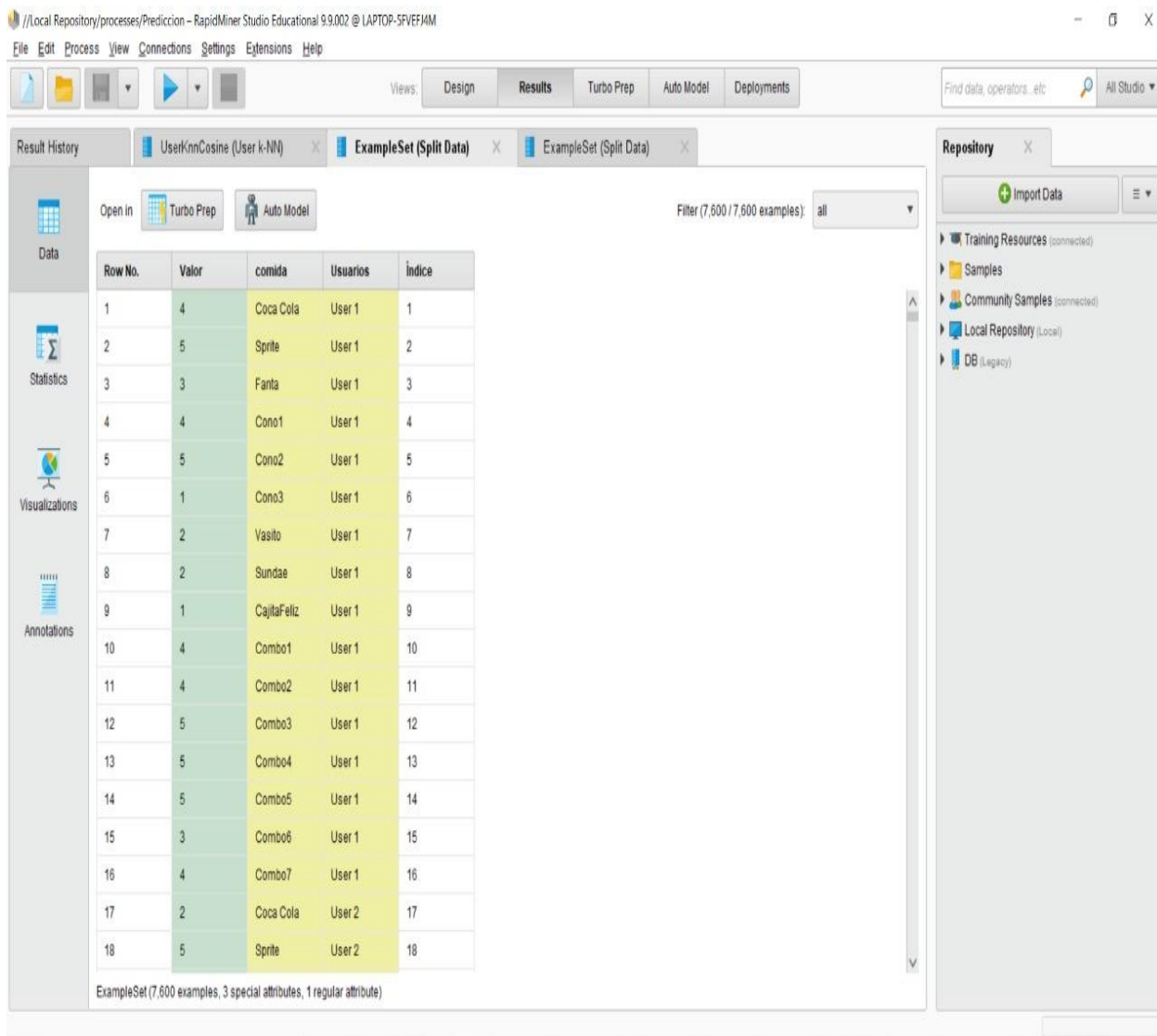
Fuente: Datos de la Investigación

Sin embargo, los algoritmos utilizados, están basados en la memoria que se utiliza en toda la base de datos de usuarios y productos para generar predicciones, es así que, cada usuario es parte de un grupo de personas con intereses similares, al momento de adquirir el producto ofertado por el local de comida rápida.

### 4.3. Análisis de predicciones

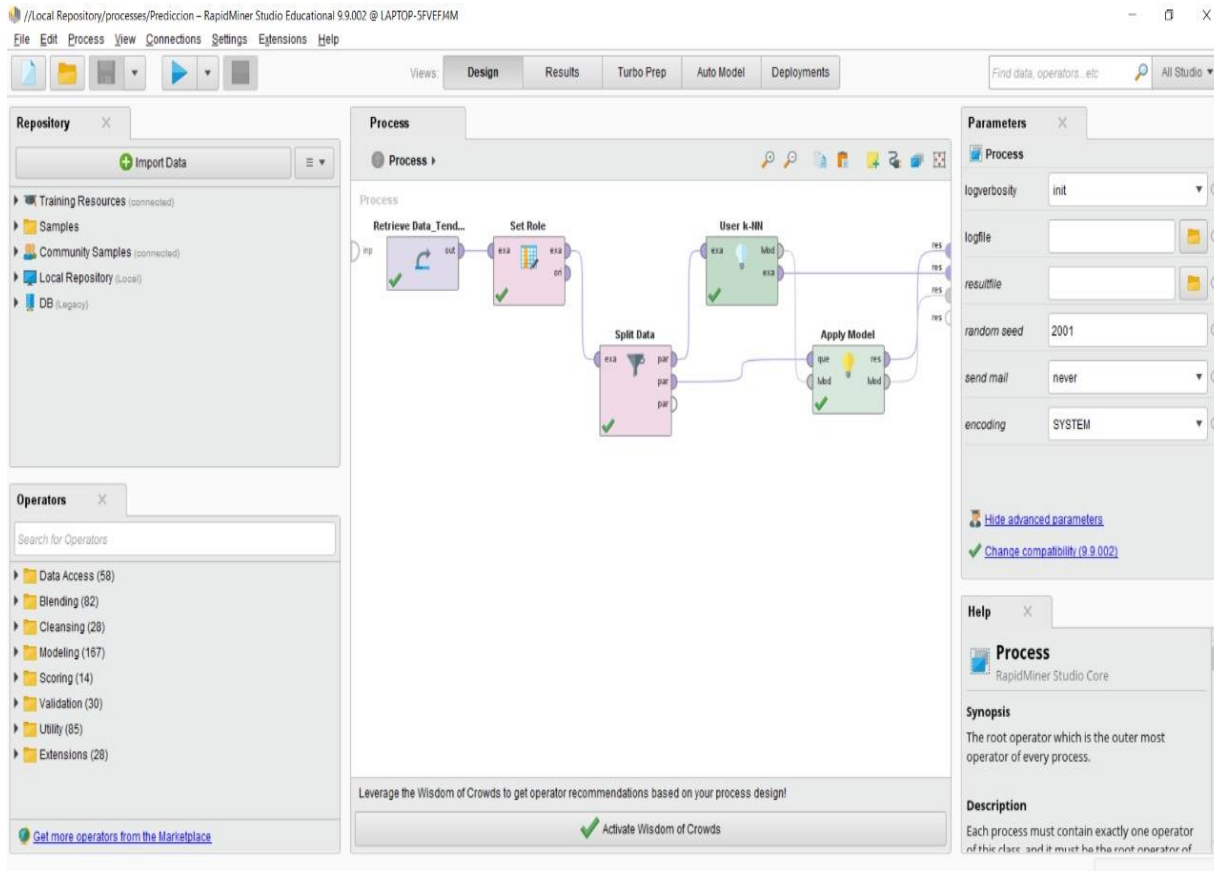
En las siguientes figuras se muestran los resultados obtenidos, mediante dichas predicciones:

Figura 6. Predicciones mediante la Herramienta RapidMiner



Fuente: Datos de la investigación

Figura 7. Predicciones mediante la Herramienta RapidMiner



Fuente: Datos de la investigación

Los componentes de un sistema de recomendación basados en el Algoritmo ALS, para la empresa de comida rápida en la ciudad de Guayaquil, se basa mediante las características necesarias para extender el servicio, utilizando la implementación de datos frente a nuevos productos de consumo rápido y eficiente, la cual será totalmente optimizada para ahorrar el tiempo de servicio dentro de la empresa.



De igual manera, el número de recomendaciones que se generan, deben ser registrados en la base de datos, como un parámetro al momento de ejecutar el programa, al igual que el número de factores que se utilizan para realizar el servicio. Por consiguiente, para poder procesar y utilizar la información en el proceso de predicción, es necesario crear una nueva interface, donde se registren todas recomendaciones, generadas por cada grupo de servicio y sobre todo de productos.

#### 4.4. Conjunto de datos

El conjunto de datos consta de tres archivos importantes:

u.data

Consiste en 1.000 calificaciones de 843 usuarios para 50 diferentes comidas rápidas. Cada usuario calificó por lo menos 10 comidas. Los datos están ordenados de forma aleatoria y los campos están clasificados de la siguiente manera:

Id\_usuario | id\_comida | calificación | fecha\_hora.

u.user

Contiene la información de los usuarios clasificados y ordenados de la siguiente manera:

Id\_usuario | edad | género | ocupación | código\_postal.





u.comida

Contiene la información de los comida clasificados y ordenados de la siguiente manera:

Id\_comida | descripcion | fecha\_lanzamiento | costo | valor | tiempo\_preparacion | foto |

Para realizar las pruebas solo se necesitará del archivo “u.data” que contiene las calificaciones.

#### 4.5. Métricas de evaluación

Entre las métricas de indicadas se manejó MAE (error medio absoluto) debido a que es la medida más sencilla de manejar y de demostrar. El método en el lenguaje de programación java, se denomina AverageAbsoluteDifferenceRecommenderEvaluator().

#### 4.6. Porcentaje de Prueba y Teste

Tal y como es recomendado en la página deGroupLense libro Mahout in Action se decidió separar los 1.000 registros de calificaciones, que se encuentran en el archivo u.data, en un 80% para training y 20% para test. Se usaron los mismos porcentajes para ambos algoritmos.



#### 4.7. Métrica de similitud

Se ha empleado dos métricas de similitud: Correlación de Pearson, cuyo método correspondiente en java es `PearsonCorrelationSimilarity(model)` y Distancia Euclidiana con su método `javaEuclideanDistanceSimilarity(model)` que se probaron con el conjunto de datos para seleccionar la mejor entre las dos.

#### 4.8. Tamaño de la vecindad

Este es un parámetro propio de los algoritmos basados en usuario (memoria) que tiene mucho impacto en los resultados obtenidos. Por lo tanto, el valor que se defina para este es muy importante. Se ha elegido el algoritmo NearestN para definir la vecindad. El número de clientes fue definido, de acuerdo a las pruebas que presentamos en el siguiente capítulo.

Con estos datos construimos la estructura del algoritmo basado en usuario y el algoritmo basado en ítem con filtro colaborativo.

Se realizaron combinaciones con los diferentes parámetros y se muestran a continuación.



Figura 8. Algoritmo basado en usuario con filtro colaborativo Correlación de Pearson

```

8 import org.apache.mahout.cf.taste.recommender.*;
9 import org.apache.mahout.cf.taste.similarity.*;
10 import java.io.*;
11 import java.util.*;
12
13 import org.apache.mahout.cf.taste.eval.*;
14 import org.apache.mahout.cf.taste.common.TasteException;
15 //import org.apache.mahout.cf.taste.impl.eval.GenericRecommenderIRStatsEvaluator;
16 import org.apache.mahout.cf.taste.impl.eval.AverageAbsoluteDifferenceRecommenderEvaluator;
17
18
19 class RecommenderIntro {
20 public static void main(String[] args) throws Exception {
21
22     DataModel model = new FileDataModel(new File("/home/veridu/hadoop/mahout-distribution-0.4/u2.csv"));
23     /*-----*/
24
25     RecommenderBuilder recommenderBuilder = new RecommenderBuilder() {
26     public Recommender buildRecommender(DataModel model) throws TasteException {
27         /*algoritmos de similaridad basado en el usuario*/
28         UserSimilarity similarity = new PearsonCorrelationSimilarity(model); // similarity es un valor entre 0 y 1
29         /*-----*/
30         /* algoritmos de vecindad basado en el usuario */
31         UserNeighborhood neighborhood = new NearestUserNeighborhood(7, similarity, model); //N vecinos(aquí debe ir el mejor v
32         /*-----*/
33         return new GenericUserBasedRecommender(model, neighborhood, similarity); //usuario
34     }
35 };

```

Fuente: Datos de la investigación

Figura 9. Algoritmo basado en usuario con filtro colaborativo Distancia Euclidiana

```

8 import org.apache.mahout.cf.taste.recommender.*;
9 import org.apache.mahout.cf.taste.similarity.*;
10 import java.io.*;
11 import java.util.*;
12 import org.apache.mahout.cf.taste.eval.*;
13 import org.apache.mahout.cf.taste.common.TasteException;
14 //import org.apache.mahout.cf.taste.impl.eval.GenericRecommenderIRStatsEvaluator;
15 import org.apache.mahout.cf.taste.impl.eval.AverageAbsoluteDifferenceRecommenderEvaluator;
16
17
18 class RecommenderIntroUser {
19 public static void main(String[] args) throws Exception {
20
21     DataModel model = new FileDataModel(new File("/home/veridu/hadoop/mahout-distribution-0.4/u2.csv"));
22     /*-----*/
23
24     RecommenderBuilder recommenderBuilder = new RecommenderBuilder() {
25     public Recommender buildRecommender(DataModel model) throws TasteException {
26         /*algoritmos de similaridad basado en el usuario*/
27         UserSimilarity similarity = new EuclideanDistanceSimilarity(model);
28         /*-----*/
29         /* algoritmos de vecindad basado en el usuario */
30         UserNeighborhood neighborhood = new NearestUserNeighborhood(7, similarity, model); //N vecinos(aquí debe ir el mejor v
31         /*-----*/
32         //return new GenericUserBasedRecommender(model, neighborhood, similarity); //usuario
33     }
34 };

```

Fuente: Datos de la investigación

Figura 10. Algoritmo basado en ítems con filtro colaborativo Correlación de Pearson

```
5 import org.apache.mahout.cf.taste.impl.similarity.*;
6 import org.apache.mahout.cf.taste.model.*;
7 import org.apache.mahout.cf.taste.neighborhood.*;
8 import org.apache.mahout.cf.taste.recommender.*;
9 import org.apache.mahout.cf.taste.similarity.*;
10 import java.io.*;
11 import java.util.*;
12 import org.apache.mahout.cf.taste.eval.*;
13 import org.apache.mahout.cf.taste.common.TasteException;
14 //import org.apache.mahout.cf.taste.impl.eval.GenericRecommenderIRStatsEvaluator;
15 import org.apache.mahout.cf.taste.impl.eval.AverageAbsoluteDifferenceRecommenderEvaluator;
16
17
18 class RecommenderIntroItem {
19     public static void main(String[] args) throws Exception {
20
21         DataModel model = new FileDataModel(new File("/home/veridu/hadoop/mahout-distribution-0.4/u2.csv"));
22         /*-----*/
23
24         RecommenderBuilder recommenderBuilder = new RecommenderBuilder() {
25             public Recommender buildRecommender(DataModel model) throws TasteException {
26                 /*algoritmos de similitud basados en el ítem*/
27                 ItemSimilarity similarity = new PearsonCorrelationSimilarity(model); // similarity es un valor entre 0 y 1
28                 /*-----*/
29                 return new GenericItemBasedRecommender(model, similarity); // ítem
30             }
31         };
32     }
33 }
```

Fuente: Datos de la investigación

## 4.9. Pruebas y resultados

### 4.9.1. Ejecución de las pruebas

Las pruebas se realizaron en un equipo cuyas características son las siguientes:

- Procesador: Intel Core I5 1.8Ghz 8tava Generación
- Disco duro: 500Gb
- Memoria RAM: 8 Gb
- Sistema Operativo: Windows 10

La realización de esta etapa tiene dos propósitos. El primero, realizar una comparación entre los distintos valores de número de clientes (múltiplos de 100),

para definir el tamaño de la vecindad respecto al usuario, para hacer aquello, los valores seleccionados se encuentran en un rango del 100 al 1000, con cada variación, se realiza pruebas con cada una de las métricas de similaridad el coeficiente de correlación de Pearson y la distancia euclidiana.

El segundo propósito para justificar la realización de pruebas, es efectuar una comparación entre los dos tipos de enfoque, para los sistemas de recomendación basada en memoria: usuario e ítem. Con el fin de llevar a cabo esta prueba se eligió como parámetro de entrada para el enfoque basado en usuario, los dos mejores valores de clientes obtenidos en la prueba anterior, cada una representa a las métricas de similaridad mencionadas en la prueba anterior.

En ambas pruebas se consideró como métrica de evaluación, siendo esta la que tiene mayor relevancia en las pruebas, esto se debe a que dicha métrica dice que tan bueno es, según los datos de entrada, los algoritmos de recomendación evaluados, otro indicador considerado es el tiempo de ejecución del algoritmo de recomendación.

#### **4.9.2. Análisis de los resultados**

En las tablas 2 y 3 se muestran las mediciones obtenidas, con respecto al indicadores de evaluación MAE y Tiempo de ejecución, para los distintos valores

de clientes, comparados tanto con la métrica de similaridad coeficiente de correlación de Pearson como la distancia euclidiana.

Tabla 2. Mediciones con respecto al MAE

NÚMERO DE CLIENTES	MAE	
	PEARSONCORRELATION	EUCLIDEANDISTANCE
100	0,95235338	0,83654373
200	0,91921116	0,80228761
300	0,9001536	0,84204603
400	0,89697539	0,84416329
500	0,87755667	0,86021461
600	0,90938966	0,85154518
700	0,93324102	0,8433514
800	0,90042604	0,8279683
900	0,89312209	0,82955307
1000	0,92635875	0,80015912

Fuente: Datos de la investigación

Tabla 3. Mediciones con respecto al tiempo de ejecución

NÚMERO DE CLIENTES	TIEMPO DE EJECUCIÓN	
	PEARSONCORRELATION	EUCLIDEANDISTANCE
100	14,29	22,91
200	16,89	26,18
300	18,67	33,93
400	16,61	25,18
500	25,17	34,39
600	18,44	26,12
700	17,22	18,55
800	19,52	27,74
900	21,48	13,43
1000	12,75	19,93

Fuente: Datos de la investigación

En la figura 11, el eje vertical representa el valor MAE, mientras que el eje horizontal representa el número de clientes. Como se observa en el gráfico, los valores de MAE oscilan entre 0.8 y 1, indicando que los valores de MAE obtenidos son muy bajos, es decir los resultados obtenidos no son buenos, esta tendencia se da a que el número de clientes seleccionados no es lo suficiente grande para agrupar usuarios con características similares.

Los valores obtenidos de MAE, más bajos, resultan provenir de las pruebas efectuadas con la métrica de similaridad “Distancia Euclidiana”, donde los valores oscilan entre 0,80 y 0,86.

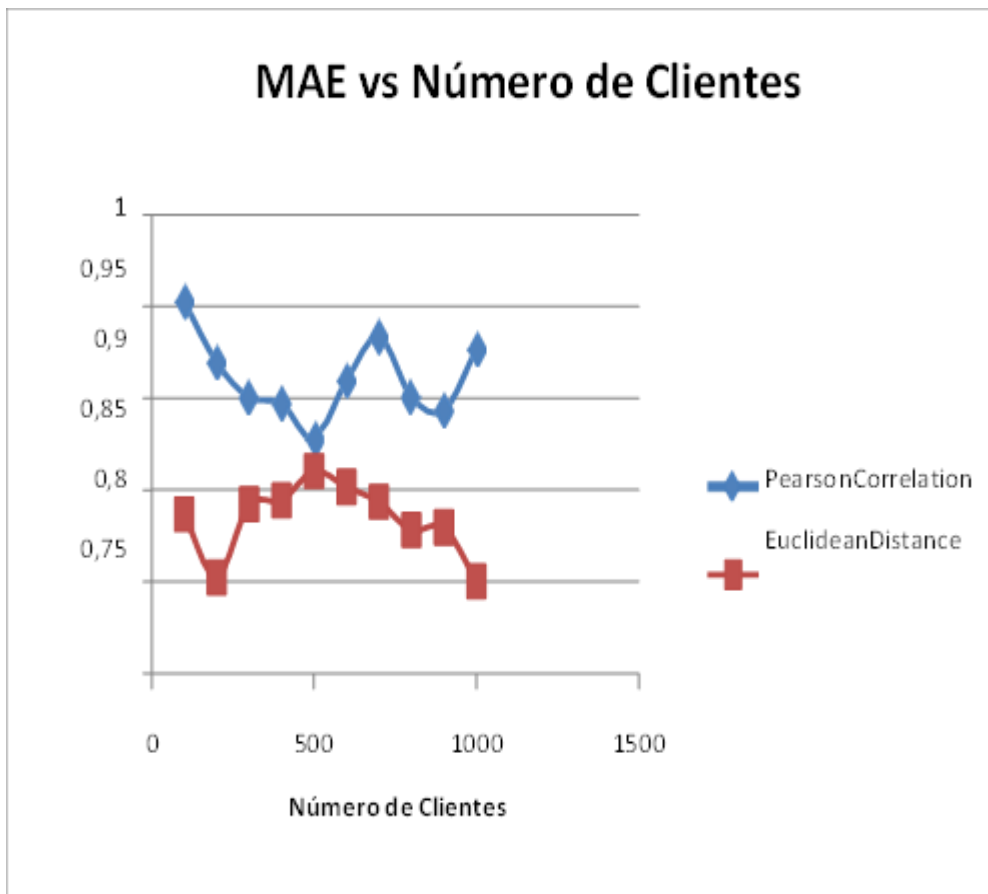
Los resultados de esta prueba indican que el error absoluto medio (MAE en inglés), predicen que el promedio de los errores absolutos, por cada tamaño de vecindad, donde el error absoluto se determina a partir de la diferencia entre la valoración predicha y la valoración real por un ítem, es menor y por consecuencia mejor, cuando se emplea la métrica de similaridad “Distancia Euclidiana” en vez de su contraparte, la métrica “coeficiente de correlación de Pearson”.

Los mejores valores MAE obtenidos para el coeficiente de correlación de Pearson y distancia Euclidiana son 0,8776 y 0,8002 respectivamente, en donde los números de clientes fueron 500 y 1000 respectivamente.



En la figura 12, el eje vertical representa el tiempo de ejecución del algoritmo basado en usuario, mientras que el eje horizontal representa el número de clientes. Con respecto a la prueba realizada, empleando como indicador de evaluación, el tiempo de ejecución, la métrica distancia Euclidiana, es la que tiene menor variación de tiempo, así como el menor de los tiempos.

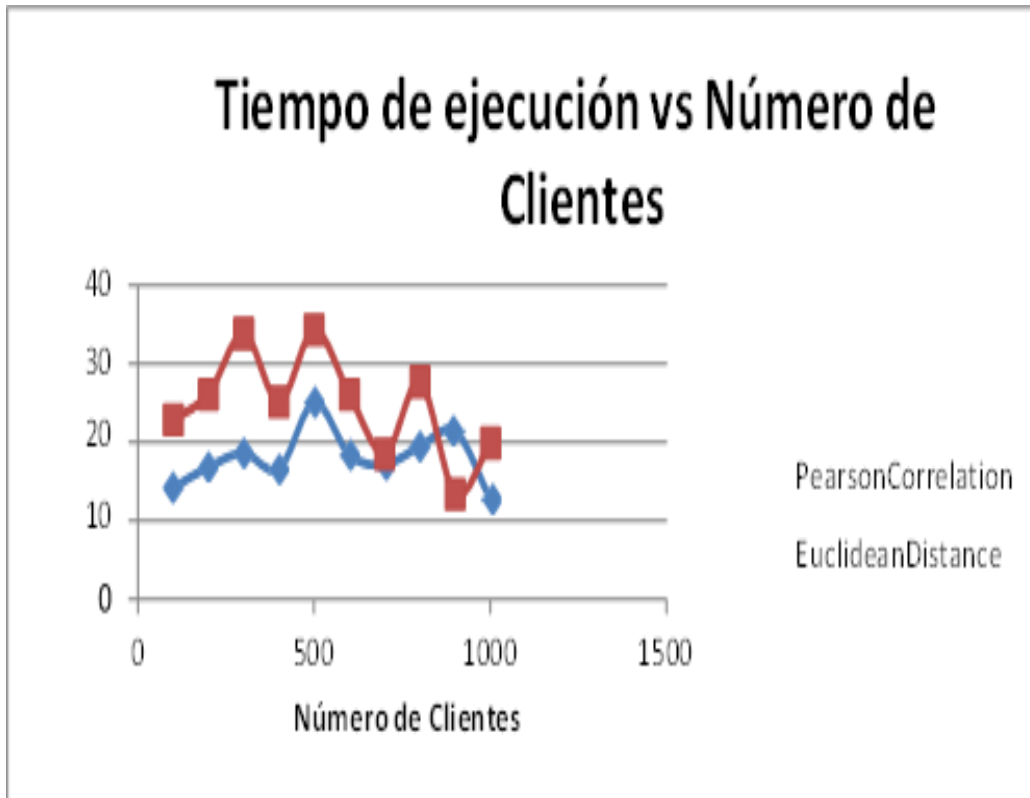
Figura 11. Algoritmo basado en ítems con filtro colaborativo Correlación de Pearson



Fuente: Datos de la investigación



Figura 12. Valor obtenido de los distintos números de clientes



Fuente: Datos de la investigación

En las tablas 2 y 3 se muestran las mediciones obtenidas, con respecto a los indicadores de evaluación: MAE y Tiempo de ejecución respectivamente; Evaluando los dos enfoques de sistemas de recomendación basado en memoria: Basado en usuario y basado en ítem, se procede a comparar tanto, con la métrica de similaridad coeficiente de correlación de Pearson, como con la distancia euclidiana.





Tabla 4. Mediciones con respecto al MAE

MÉTRICA DE SIMILARIDAD	MAE	
	BASADO EN USUARIO	BASADO EN ÍTEM
PearsonCorrelation	0,87755667	0,93203789
EuclideanDistance	0,80015912	0,83846361

Fuente: Datos de la investigación

En la figura 13, el eje vertical representa los distintos valores MAE, mientras que el eje horizontal se encuentra representado por los dos enfoques de los sistemas de recomendación basados en memoria. Como se aprecia en la figura los valores MAE obtenidos del enfoque basado en usuario, son menores con respecto al basado en ítem, con esto el algoritmo basado en usuario, resulta ser mejor, tanto, usando la métrica de similitud de coeficiente de correlación de Pearson como la distancia euclidiana.

Esto indica que resultó mejor obtener los ítems preferidos, de acuerdo, a la selección que hacen otros usuarios, que poseen características similares al usuario activo, frente a, los ítems seleccionados anteriormente por el usuario activo.

El tiempo de ejecución, de acuerdo a la figura 14, da como resultado al enfoque basado en ítem, como los tiempos de ejecución más corto, esto se debe a que no se debe realizar ninguna tarea adicional como por ejemplo normalizar las calificaciones de los ítems, para tener un mejor resultado en las pruebas.



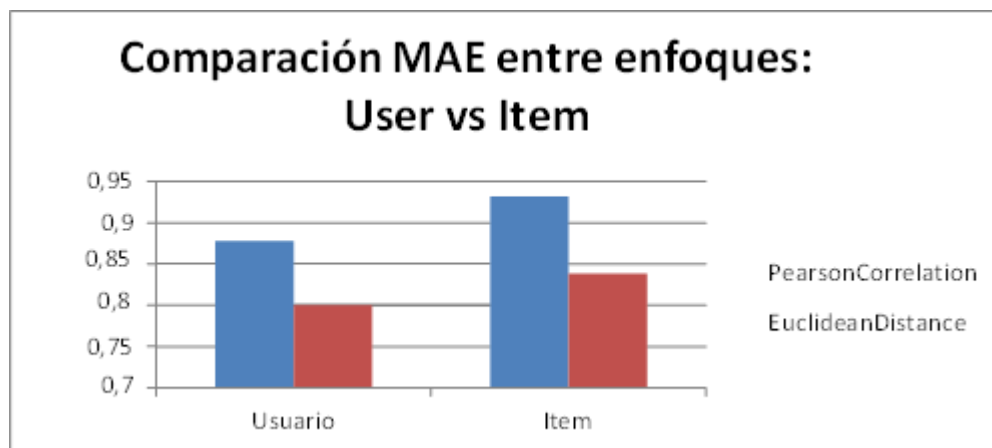


Tabla 5. Mediciones con respecto al tiempo de ejecución

MÉTRICA DE SIMILARIDAD	TIEMPO DE EJECUCIÓN	
	USUARIO	ITEM
PearsonCorrelation	12,75	7,06
EuclideanDistance	13,43	7,62

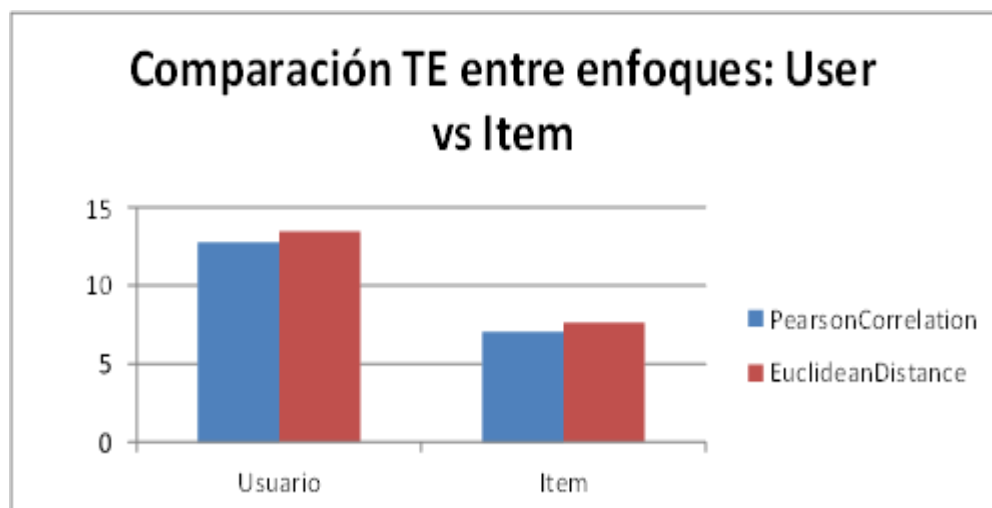
Fuente: Datos de la investigación

Figura 13. Valor MAE obtenido



Fuente: Datos de la investigación

Figura 14. Tiempo de Ejecución Obtenido



Fuente: Datos de la Investigación



## CAPÍTULO 5. PROPUESTA





## Capítulo 5. Propuesta

### 5.1. Filtrado colaborativo

Dentro de la empresa de comida rápida se observa algoritmos de filtrado colaborativo, los cuales se basan en buscar relaciones de similitud entre usuarios o productos y la idea principal es que, si dos usuarios muestran un patrón de puntuaciones similar, probablemente puedan coincidir en las puntuaciones restantes.

Por otro lado, existen clientes que se acostumbran a dar puntuaciones positivas, para productos que generalmente son malos y viceversa. Otros clientes, sin embargo, se reservan sus puntuaciones altas para los mejores productos, y en muchos de los casos suelen dar puntuaciones negativas.

### 5.2. Recursos financieros, costos

En este apartado del estudio, se establecen todos los costos generados para el análisis de las predicciones del filtro colaborativo, donde se especifica el monto que se debe invertir en las diversas actividades propias de empresa colaboradora.

Es por tal razón que, para el análisis de las predicciones en un filtro colaborativo basado en el Algoritmo ALS, de una empresa de comida rápida en la ciudad de



Guayaquil, fue necesario implementar componentes como los beneficios del sistema basado en algoritmo, mismo que no se disipó ninguna cantidad de dinero, ya que esta herramienta se lo puede descargar de forma gratuita desde la página web <https://rapidminer.com/get-started/>, misma que se puede utilizar introduciendo un correo electrónico, sea esta personal o institucional.

### 5.3. Análisis de la solución

En este trabajo se ha realizado los análisis necesarios para poder encontrar el mejor algoritmo con Filtro Colaborativo basado en Memoria. Para esto se realizó una serie de pruebas y análisis que se mostrarán posteriormente. Existen tres etapas en las que se basan los Sistemas de Recomendación con Filtro Colaborativo basado en Memoria.

Figura 14. Etapas del algoritmo colaborativo basado en memoria



Fuente: Datos de la investigación

## 5.4. Primera etapa: Formación

En los enfoques colaborativos basados en usuario, durante la primera fase se selecciona a los usuarios cuyas preferencias son similares a las del usuario activo. Por el contrario, los enfoques basados en ítem extraen las preferencias del usuario que son más similares al producto objetivo.

Para el objetivo de este trabajo solo se implementó el uso de vecindario en los algoritmos de enfoque colaborativo basado en usuario para generar los resultados porque solo en estos casos se necesita la información de otros usuarios para obtener las recomendaciones.

Para definir el vecindario lo ideal sería escoger todos los datos disponibles para seleccionar los mejores perfiles de usuarios y así lograr recomendaciones con menor margen de error. Sin embargo, esta opción no es la mejor si se cuenta con una abundante cantidad de datos y recursos limitados, en cuyo caso lo mejor es utilizar un muestreo aleatorio.

En cuanto al tamaño, para definir el adecuado, se realizaron pruebas con respecto a los N clientes más cercanos en un rango amplio para ir comprobando las variaciones de los resultados en cada caso. Es necesario tener en cuenta que si se elige un tamaño muy elevado se reduce la calidad de las recomendaciones obtenidas, mientras que si el tamaño es muy reducido se

limitará la capacidad predictiva del sistema. La principal ventaja de esta técnica es que permite recomendaciones más precisas.

Una vez que se definió el tamaño de la vecindad, se procedió a establecer quiénes pueden ser considerados clientes para generar las recomendaciones en base a sus ratings. Esto es lo que se denomina Similitud. Las métricas para definir esta similitud son variadas.

Para evaluar este parámetro se utilizaron los métodos de Distancia Euclidiana y Correlación de Pearson.

### **5.5. Segunda etapa: Agregar opiniones**

En esta etapa se busca recolectar las calificaciones de otros usuarios con respecto a él o los productos que se van recomendar, cuando se refiere al enfoque colaborativo basado en usuario. En el caso del enfoque basado en ítem, se busca recolectar los ratings que el usuario (para el cual son las recomendaciones) ha dado a productos similares.

### **5.6. Tercera etapa: Recomendación**

En esta etapa se genera la recomendación utilizando la estructura formada en las etapas anteriores. El algoritmo debe predecir el nivel de interés de un usuario en base a los datos que se entregados.



Para definir qué tan buenas son las recomendaciones generadas se utilizan las métricas de evaluación, que miden la calidad de los resultados, con el objetivo de analizar los puntos fuertes y débiles, visualizar las variables involucradas y ver cómo afectan los diferentes parámetros y sus variaciones.

Una métrica para la exactitud de un sistema de recomendación mide, empíricamente, que tan cerca está el ordenamiento de ítems predichos para un usuario por el sistema con respecto al ordenamiento verdadero que el usuario haría, según su preferencia, de los mismos ítems.

A pesar de que el tema de los sistemas de recomendación es muy extenso, todavía no existen estándares necesarios para su evaluación. Es por eso que las métricas que se utilizan pueden variar dependiendo del caso, de la función del SR, del tipo de algoritmo, etc.

Esta diversidad de métricas conlleva a tres grandes problemas:

1. Si dos investigadores distintos, evalúan sus sistemas con distintas métricas, los resultados no son comparables.
2. Si la métrica usada no está estandarizada, se puede pensar que los investigadores han elegido la métrica más adecuada de cara a obtener los resultados deseados.



3. Sin una métrica estandarizada, cada investigador debe realizar un esfuerzo extra en identificar o desarrollar una métrica apropiada.

Teniendo en cuenta estas consideraciones, se procedió a elegir la métrica de evaluación denominada Error absoluto Medio (MAE). Esta es la más usada en cuanto a métricas estadísticas, ya que es fácil de entender y de representar al momento de querer mostrar la calidad de los resultados en forma gráfica. Mientras menor MAE se obtenga mejor será el resultado generado por el algoritmo de recomendación.





## EPÍLOGO





## Epílogo

Se ha revisado todas las fuentes bibliográficas relacionadas teóricamente a las predicciones en un filtro colaborativo, la cual es basado en el algoritmo ALS, por ende, se ha realizado la indagación sobre las variables expuestas en el tema de estudio.

Es posible contar con diversas técnicas precisas y eficaces de recomendación, sin tener la necesidad de recurrir a modelos complejos que tienen factores y parámetros a estimar.

La ventaja de utilizar algoritmos se basa en la mejor comprensión del funcionamiento de los mismos incluyendo sus características, las cuales se adaptan a condiciones y requisitos particulares.

Los sistemas de recomendación colaborativos existentes, son los más utilizados en la actualidad, debido a que, cuenta las calificaciones de todos los clientes y se pueden ofrecer recomendaciones más variadas sobre productos y servicios, siempre y cuando manteniendo la apreciación del usuario.

El número de recomendaciones que se generan en la empresa, se deben registrar en la base de datos, generando cuantificaciones al momento de ejecutar el programa, al igual que el número de factores que se utilizan para realizar el servicio.



Para procesar y utilizar la información en el proceso de predicción, es necesario crear una nueva interface, donde se puedan registrar todas aquellas recomendaciones generadas por los clientes, mismos que son muy oportunas para mejorar el servicio y sobre todo de productos que se realizan dentro de la empresa de comida rápida.

Se debe profundizar la aplicación de la herramienta del RapidMiner, en las empresas de cualquier sector, acompañado del filtro colaborativo basado en algoritmos, ya que se analiza la funcionalidad del local mediante el tipo de producto y servicio ofertado, al igual que la mejor interacción entre usuario y sistema.



## BIBLIOGRAFÍA





## Bibliografía

- Agüero Dejo, S. P., Chumacero Delgado, I. T., & Delgado Soto, R. M. (2019).  
Uso de los sistemas de recomendación en la gestión de contenido para apuestas.
- Castillo-Saco, A. (2014). *Sistemas de recomendación distribuidos*. Universidad Autónoma de Madrid-Escuela Politécnica Superior.
- Cheng, D., Peng, R., & Liu, Y. (2016). SPALS: Fast alternating least squares via implicit leverage scores sampling. *Advances in neural information processing systems*, 29, 721-729.
- Collado, A. (2016). Sistema de recomendación de recursos basado en filtrado colaborativo para la plataforma edX (Bachelor's thesis).
- Comon, P., Luciani, X., & De Almeida, A. (2009). *Tensor decompositions, alternating least squares and other tales*. *Journal of Chemometrics: A Journal of the Chemometrics Society*.
- Echevarría, K. (2017). Diseño de un sistema de recomendación de recursos educativos basado en las emociones reflejadas por los estudiantes (Doctoral dissertation, Universidad Central "Marta Abreu" de Las Villas).
- González, C. J., García, M. N., & Gil, A. B. (2020). Sistema de Recomendación de Canciones Basado en Aspectos Emocionales. *Avances en Informática y Automática*.

Hernández, N. M. (2019). Recomendación de cursos basada en filtrado colaborativo e información de perfiles sociales y estudiantiles (Bachelor's thesis, Uniandes).

López, M. B., & Montes, A. J. (s.f.). EmoRemSys: Sistema de recomendación de recursos educativos basado en detección de emociones. . *RISTI-Revista Ibérica de Sistemas e Tecnologías* .

Maltz, D., & Ehrlich, K. (1995). *“Pointing the way: Active collaborative filtering”* *Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co.

Manzano-Chicano, A. (22 de 05 de 2018). *Introducción a los sistema de recomendación basados en filtrado colaborativo con PySaprk*. Obtenido de Ciencias y Datos: <https://medium.com/datos-y-ciencia/intro-als-pyspark-7de7f3ba3b0a>

Marín, P. A., Zapata, Á. M., Rojas, L. F., & Mendez, N. D. (2016). Sistema de recomendación de objetos de aprendizaje a través de filtrado colaborativo. *Teknos revista científica*, 85-94.

MicroSystem. (2005). *RapidMiner*. Obtenido de MicroSystem: <https://www.microsystem.cl/plataforma/rapidminer/>

Olguín, G. E., & De Jesús, Y. L. (2019). Métricas de similaridad y evaluación para sistemas de recomendación de filtrado colaborativo. *Revista de Investigación en Tecnologías de la Información: RITI*, 7(14), 224-240.

- Olguín, G. E., & De Jesús, Y. L. (2019). Métricas de similaridad y evaluación para sistemas de recomendación de filtrado colaborativo. *Revista de Investigación en Tecnologías de la Información: RITI*, 7(14), 224-240.
- Rios, C., Godoy, D. L., & Schiaffino, S. (2016). Análisis de estrategias de selección de clientes para recomendación en LBSN. In Simposio Argentino de Inteligencia Artificial (ASAI 2016)-JAIIO 45 (Tres de Febrero, 2016).
- Schafer, J., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. *Lecture Notes in Computer Science*, 291.
- Valdiviezo, P. M. (2019). Sistema recomendador híbrido basado en modelos probabilísticos (Doctoral dissertation, Universidad Politécnica de Madrid).
- Valdiviezo-Díaz, P., Ortega, F., & Mayor, J. (2020). Optimización del filtrado colaborativo basado en factorización matricial mediante la relevancia de las preferencias de los usuarios. *Revista Ibérica de Sistemas e Tecnologías*.
- Varela, D., Aguilar, J., Monsalve-Pulido, J., & Montoya, E. (2020). Propuesta Arquitectónica de un Sistema de Recomendación Híbrido Adaptativo. *Revista Ibérica de Sistemas e Tecnologías de Informação*, 171-184.
- Walid-Ghobar, E. (2016-2017 ). *Un sistema de recomendación basado en perfiles generados por agrupamiento y asociaciones*. Univeritat Politècnica de Valencia .



Yauri Godoy, R. A. (2019). Plataforma de compra online basado en técnicas de filtrado colaborativo para la predicción y recomendación de productos.



## Análisis de las predicciones en un filtro colaborativo basado en el algoritmo ALS para una empresa de comida rápida en la ciudad de Guayaquil



### Editorial Tecnocientífica Americana

**Domicilio legal:** calle 613nw 15th, en Amarillo, Texas. **ZIP:** 79104

Estados Unidos de América, 16 de enero de 2023

**Teléfono:** 7867769991

La Editorial Tecnocientífica Americana se encuentra indizada en, referenciada en o tiene convenios con, entre otras, las siguientes bases de datos:

