



**UNIVERSIDAD DEL BÍO-BÍO**

# Estudio de estrategias de transformación de datos implícitos a datos explícitos para Sistemas de Recomendación Consciente del Contexto

Gabriela Katherine Aguayo Diaz

Universidad del Bío-Bío  
Facultad de Ciencias Empresariales, Departamento de Sistemas de Información  
Concepción, Chile  
2016

# Estudio de estrategias de transformación de datos implícitos a datos explícitos para Sistemas de Recomendación consciente del Contexto

Gabriela Katherine Aguayo Diaz

Proyecto de Título presentado en conformidad a los requisitos para obtener el título de Ingeniero Civil en Informática

Profesor Guía:  
Pedro Gerónimo Campos Soto

Universidad del Bío-Bío  
Facultad de Ciencias Empresariales, Departamento de Sistemas de Información  
Concepción, Chile  
2016



## Agradecimientos

*Agradezco principalmente a mi familia, que sin su apoyo nada de esto sería posible. Mi madre siempre preocupada, mi hermano dándome ánimos, mi hermana y cuñado como unos segundos padres, mi pareja Jaime Vásquez Camaño, mis compañeros y amigos que estuvieron enriqueciendo mis conocimientos y alegrando los días de estudio. También a mi profesor guía Don Pedro Campos, quien me dio la oportunidad de trabajar en este proyecto. Por último le agradezco a la personita más importante, mi hija, quien estuvo conmigo durante todo el desarrollo de este proyecto, espero que algún día leas estas palabras y sientas orgullo de mí, te ama mamá.*

Este proyecto fue financiado en parte por el proyecto DIUBB 151115 4/R.

## Resumen

Un sistema de recomendación es un software que entrega a los usuarios sugerencias personalizadas, en base a preferencias que los usuarios hayan manifestado sobre determinados grupos de productos o servicios, también llamados ítems.

En este trabajo se estudian estrategias de agregación de datos que permitan resumir o condensar de manera eficiente grandes cantidades de registros implícitos, con el objetivo de transformar datos implícitos en datos explícitos, minimizando la pérdida de información de contexto asociado a ellos. Se comienza presentando los sistemas de recomendación, los procesos de obtención de retroalimentación explícita e implícita, posteriormente se describe que se entiende por contexto y los Sistemas de Recomendación consientes del contexto. También se presentan algoritmos encontrados en la literatura que pueden servir de base para la estrategia que se busca desarrollar. Uno de ellos permite la transformación de datos implícitos en explícitos y otro que estudia diferencias existentes en información de contexto.

En base a estos algoritmos se diseña e implementa una estrategia compuesta por cuatro métodos que transforman datos implícitos en datos explícitos utilizando la Distribución Acumulada Complementaria. Además se considera la información de contexto como un parámetro adicional para agrupar u organizar estos registros al momento de transformarlos.

Los métodos que conforman la estrategia son probados por separados para determinar el que mejor representa los datos originales. De estas pruebas unitarias se elige el método al cual incorporar la variable contexto.

Finalmente la implementación es sometida a experimentos en un sistema de recomendación con datos obtenidos en tiempo real, para medir las recomendaciones generadas por el sistema de recomendación aplicando cada método de transformación desarrollado.

**Palabras clave: Sistemas de recomendación, Retroalimentación Implícita, Contexto, Transformación de datos.**

## Contenido

<b>Resumen</b> .....	IV
<b>Lista de figuras</b> .....	VII
<b>Lista de tablas</b> .....	VIII
<b>1. Introducción</b> .....	1
<b>1.1. Origen del Tema</b> .....	2
<b>1.2. Objetivos</b> .....	4
<b>1.2.2. Objetivos Específicos</b> .....	4
<b>1.3. Presentación de Capítulos</b> .....	5
<b>2. Marco Teórico</b> .....	6
<b>2.1. Sistemas de recomendación</b> .....	6
<b>2.2. Historia de los sistemas de recomendación</b> .....	7
<b>2.3. Retroalimentación</b> .....	10
<b>2.3.1. Retroalimentación Explícita</b> .....	10
<b>2.3.2. Retroalimentación Implícita</b> .....	11
<b>2.4. Contexto</b> .....	12
<b>2.4.1. Descripción</b> .....	12
<b>2.4.2. Modelando información Contextual en los Sistemas de Recomendación.</b> .....	13
<b>2.5. Distribución de Frecuencias</b> .....	15
<b>2.6. Algoritmos estudiados</b> .....	16
<b>2.6.1. Retroalimentación Explícita consciente del Contexto</b> .....	17
<b>2.6.2. Retroalimentación Implícita</b> .....	20
<b>3. Transformación de datos implícitos a datos explícitos</b> .....	24
<b>3.2. Estrategia de transformación de datos ideada</b> .....	30
<b>3.3. Diseño e Implementación de la Estrategia de agregación de datos</b> .....	30
<b>3.3.1. Diseño</b> .....	31
<b>3.3.2. Implementación</b> .....	32
<b>3.3.3. Pruebas unitarias</b> .....	38
<b>4. Experimentos</b> .....	48
<b>5. Conclusiones del proyecto</b> .....	51

Contenido

---

**Referencias** ..... 52

## Lista de figuras

FIGURA 1: VALORES QUE PODRÍA TOMAR EL CONTEXTO MOMENTO DEL DÍA. ....	14
FIGURA 2: VALORES QUE PODRÍA TOMAR EL CONTEXTO ESTACIÓN DEL AÑO. ....	14
FIGURA 3: DIVISIÓN DE ÍTEMS [22]. ....	18
FIGURA 4: REPRODUCCIONES EFECTUADAS POR USUARIO REPRESENTADAS CON DISTRIBUCIÓN DE FRECUENCIA, LA HORIZONTAL SON LOS ARTISTAS TOP-50 Y LA VERTICAL EL NÚMERO DE VECES QUE SE REPRODUJO ESE ARTISTA[23]. ....	21
FIGURA 5: REPRODUCCIONES EFECTUADAS POR USUARIO DE LA FIGURA 4 REPRESENTADAS CON LA DISTRIBUCIÓN ACUMULADA COMPLEMENTARIA[23]. ....	22
FIGURA 6: MATRIZ DE DISEÑO .....	32
FIGURA 7: ESQUEMA GLOBAL DEL SISTEMA .....	33
FIGURA 8: DIAGRAMA DE CLASES IMPLICITFEEDBACKANALYZER, CLASE IMPLICITFEEDBACKANALYZERINDIVIDUAL Y CLASE IMPLICITFEEDBACKANALYZERALL .....	35
FIGURA 9: DIAGRAMA DE CLASES IMPLICITTOEXPLICITFEEDBACK, CLASE IMPLICITTOEXPLICITFEEDBACKADJUSTMENT, CLASE IMPLICITTOEXPLICITFEEDBACKBASIC, CLASE IMPLICITTOEXPLICITFEEDBACKCELMA Y CLASE IMPLICITTOEXPLICITFEEDBACKFINAL ...	36
FIGURA 10: DIAGRAMA GENERAL .....	37
FIGURA 11: MATRIZ ORIGINAL TRANSFORMADA A RATINGS AL APLICAR ESQUEMA DE TRANSFORMACIÓN IMPLICITTOEXPLICITFEEDBACKBASIC. ....	40
FIGURA 12: MATRIZ C1 TRANSFORMADA A RATINGS AL APLICAR ESQUEMA DE TRANSFORMACIÓN IMPLICITTOEXPLICITFEEDBACKBASIC. ....	40
FIGURA 13: MATRIZ C2 TRANSFORMADA A RATINGS AL APLICAR ESQUEMA DE TRANSFORMACIÓN IMPLICITTOEXPLICITFEEDBACKBASIC. ....	40
FIGURA 14: MATRIZ ORIGINAL TRANSFORMADA A RATINGS AL APLICAR ESQUEMA DE TRANSFORMACIÓN IMPLICITTOEXPLICITFEEDBACKCELMA. ....	42
FIGURA 15: MATRIZ C1 TRANSFORMADA A RATINGS AL APLICAR ESQUEMA DE TRANSFORMACIÓN IMPLICITTOEXPLICITFEEDBACKCELMA. ....	42
FIGURA 16: MATRIZ C2 TRANSFORMADA A RATINGS AL APLICAR ESQUEMA DE TRANSFORMACIÓN IMPLICITTOEXPLICITFEEDBACKCELMA. ....	43
FIGURA 17: MATRIZ ORIGINAL TRANSFORMADA A RATINGS AL APLICAR ESQUEMA DE TRANSFORMACIÓN IMPLICITTOEXPLICITFEEDBACKADJUSTMENT. ....	44
FIGURA 18: MATRIZ C1 TRANSFORMADA A RATINGS AL APLICAR ESQUEMA DE TRANSFORMACIÓN IMPLICITTOEXPLICITFEEDBACKADJUSTMENT. ....	44
FIGURA 19: MATRIZ C2 TRANSFORMADA A RATINGS AL APLICAR ESQUEMA DE TRANSFORMACIÓN IMPLICITTOEXPLICITFEEDBACKADJUSTMENT. ....	44
FIGURA 20: MATRIZ ORIGINAL TRANSFORMADA A RATINGS AL APLICAR ESQUEMA DE TRANSFORMACIÓN IMPLICITTOEXPLICITFEEDBACKFINAL. ....	46
FIGURA 21: MATRIZ C1 TRANSFORMADA A RATINGS AL APLICAR ESQUEMA DE TRANSFORMACIÓN IMPLICITTOEXPLICITFEEDBACKFINAL .....	46
FIGURA 22: MATRIZ C2 TRANSFORMADA A RATINGS AL APLICAR ESQUEMA DE TRANSFORMACIÓN IMPLICITTOEXPLICITFEEDBACKFINAL .....	46



## Lista de tablas

TABLA 1: TABLA DE FRECUENCIAS. ....	15
TABLA 2: CONJUNTO DE REGISTROS DE USUARIO U1. ....	25
TABLA 3: CONTEO DE CANCIONES. ....	26
TABLA 4: ASIGNACIÓN DE RATINGS SIN CONSIDERAR CONTEXTOS. ....	26
TABLA 5: REGISTROS CON CONTEXTOS ESTABLECIDOS. ....	27
TABLA 6: CONTEO DE CANCIONES CON VALOR DE CONTEXTO "NOCHE" Y "SEMANA". ....	27
TABLA 7: CONTEO DE CANCIONES CON VALOR DE CONTEXTO "NOCHE" Y "FIN DE SEMANA" . ....	28
TABLA 8. CONTEO DE CANCIONES CON VALOR DE CONTEXTO "DÍA" Y "FIN DE SEMANA" . ....	28
TABLA 9. RATING PARA VALOR DE CONTEXTO "NOCHE" Y "SEMANA" . ....	28
TABLA 10: RATINGS PARA VALOR DE CONTEXTO "NOCHE" Y "FIN DE SEMANA" . ....	29
TABLA 11: RATINGS PARA VALOR DE CONTEXTO "DÍA" Y "FIN DE SEMANA" . ....	29
TABLA 12: MATRIZ ORIGINAL, DE PREFERENCIAS GENERALES POR CADA USUARIO E ÍTEM, SIN CONSIDERAR CONTEXTO. ....	38
TABLA 13: MATRIZ C1, CONTIENE PREFERENCIAS SÓLO EN EL CONTEXTO "NOCHE". ....	39
TABLA 14: MATRIZ C2, CONTIENE PREFERENCIAS SÓLO EN EL CONTEXTO "Día". ....	39
TABLA 15: MATRIZ C2, CONTIENE PREFERENCIAS SÓLO EN EL CONTEXTO C2 "Día". ....	39
TABLA 16: COEFICIENTE DE VARIACIÓN DE CADA USUARIO. ....	41
TABLA 17: COEFICIENTE DE VARIACIÓN DE CADA USUARIO. ....	41
TABLA 18: COEFICIENTE DE VARIACIÓN DE CADA USUARIO EN LA MATRIZ C1 . ....	41
TABLA 19: COEFICIENTE DE VARIACIÓN DE CADA USUARIO EN EL CONTEXTO C1 . ....	41
TABLA 20: COEFICIENTE DE VARIACIÓN DE CADA USUARIO EN EL CONTEXTO C2 . ....	42
TABLA 21: COEFICIENTE DE VARIACIÓN DE CADA USUARIO EN EL CONTEXTO C2 . ....	42
TABLA 22: DESCRIPCIÓN DE CONJUNTO DE DATOS UTILIZADOS PARA EXPERIMENTOS. ....	48
TABLA 23: RESULTADOS EXPERIMENTALES. ....	50

## 1. Introducción

Los sistemas de recomendación son aplicaciones relativamente nuevas, pero quizás es uno de los tipos de sistemas con los que todos tenemos contacto actualmente, van desde sistemas sofisticados y predictivos, hasta otros menos rebuscados. El uso de estos sistemas se está poniendo cada vez más de moda en Internet debido a que son muy útiles para evaluar y filtrar la gran cantidad de información disponible en la Web, con objeto de asistir a los usuarios en sus procesos de búsqueda y recuperación de información.<sup>1</sup>

Para lograr que dichas recomendaciones sean lo más acertadas es necesario capturar información respecto de las preferencias de los usuarios, lo que se conoce como retroalimentación. Dependiendo de la forma en que se capturan estos datos, se pueden clasificar en dos grandes grupos. Retroalimentación explícita y retroalimentación implícita. La primera es aquella en la cual el usuario expresa su preferencia de forma explícita en el sistema de recomendación, por ejemplo la respuesta a una encuesta donde se le pregunte si un ítem le gusta o no. La segunda es más compleja pero no por esto menos importante, ya que aquí el sistema de recomendación infiere la preferencia del usuario a partir de su interacción normal con el sistema. Por ejemplo, estableciendo si una canción es o no de preferencia de un usuario, a partir del registro de las veces que el usuario escuchó dicha canción[1].

Otro tema a considerar para obtener recomendaciones de buena calidad es el contexto, el cual puede corresponder a uno o varios factores que pueden hacer que los intereses de los usuarios sean diferentes a medida que este o estos factores cambien. Por ejemplo en una página de turismo, las preferencias acerca de una playa pueden ser muy diferentes en verano que en invierno.

En este trabajo se estudian estrategias de agregación de datos de retroalimentación implícita que incluyen además el contexto. Donde se busca de manera eficiente lograr sintetizar registros implícitos sin perder información de contexto.

---

<sup>1</sup> <https://www.upf.edu/hipertextnet/numero-2/recomendacion.html>

## 1.1. Origen del Tema

La importancia de los sistemas de recomendación es lograr filtrar una gran cantidad de información disponible en la Web con objeto de asistir a los usuarios en sus procesos de búsqueda y recuperación de información. Esta interacción relaciona dos conceptos principalmente: Ítems: Son los objetos que serán recomendados, Usuarios: De quien se necesita obtener sus gustos o preferencias.[2]

Para generar una buena recomendación se necesita recolectar información de los usuarios, es decir, retroalimentación, que de acuerdo a la forma en que se recolecta, se clasifica en dos tipos: retroalimentación explícita y retroalimentación implícita. La retroalimentación explícita es aquella donde el usuario evalúa un artículo, ya sea mediante una única indicación de agrado (por ejemplo “like” de Facebook) o una escala de Likert indicando su nivel agrado (por ejemplo un rating o valoración entre 1 y 5). Por otro lado la retroalimentación implícita es aquella donde no se pide al usuario que manifieste su nivel de preferencia, sino que el mismo sistema de recomendación observa las acciones que el usuario ha efectuado, y a partir de esta información infiere el nivel de agrado del usuario con respecto a los ítems con que ha interactuado. Por ejemplo, en un sistema de venta de películas online como Netflix<sup>2</sup>, lo que el sistema observe del usuario puede ser la cantidad de reproducciones de cada película que ha efectuado. Luego, podría inferir que aquellas películas que han sido completamente reproducidas han sido del agrado del usuario, mientras que aquellas que fueron reproducidas parcialmente no han sido de su agrado.

Para estos dos tipos de retroalimentaciones existen pro y contras. La retroalimentación explícita por un lado es precisa y absoluta, sin embargo es difícil obtener suficiente información de este tipo por parte de los usuarios. La reticencia a proporcionar información explícita puede deberse en parte por el esfuerzo cognitivo que requiere del usuario[3]. Por otro lado, la retroalimentación implícita es abundante y busca evitar este cuello de botella infiriendo algo similar a las valoraciones que un usuario podría asignar a partir de observaciones que están disponibles para el sistema, pero no representa

---

<sup>2</sup> Sitio de entretenimiento que proporciona bajo tarifa plana mensual streaming multimedia (películas o series de TV), Netflix, disponible en <https://www.netflix.com>

## Introducción

---

necesariamente el real nivel de agrado del usuario[4]. Por ejemplo, una película podría haber sido reproducida mientras el usuario estaba prestando atención a otra tarea, y por tanto interpretarse erróneamente que es del agrado del usuario.

Otro parámetro importante a considerar en la calidad de las recomendaciones es el contexto, el cual podría influir en las variaciones de comportamiento del usuario que el sistema ha observado. Por ejemplo, en una página web de turismo, el usuario tendría preferencias distintas por un paquete de viaje a la nieve, dependiendo si es verano o invierno. El sistema de recomendación en este caso debe generar recomendaciones distintas en cada estación del año para el usuario.

Variedades de servicios online están dispuestos a invertir en trabajar, desarrollar o mejorar sus sistemas de recomendación, para lograr distintos objetivos, que pueden ser: satisfacer al usuario, entregar un servicio personalizado o mejorar ventas. Estos servicios son por ejemplo páginas de comercio electrónico como Amazon<sup>3</sup>, de entretenimiento como Netflix, Last.fm<sup>4</sup> y YouTube<sup>5</sup>, de redes sociales como Facebook<sup>6</sup>. En el New York Times podemos encontrar un artículo de Clive Thompson, el cual habla sobre los intentos de Netflix por mejorar su motor de recomendación y que su sistema de recomendación Cinematch actualmente representa un sorprendente 60 por ciento de los alquileres de Netflix[5].

El origen de este trabajo nace principalmente de investigaciones realizadas por el profesor guía y la necesidad de seguir investigando la retroalimentación implícita, que como ya se ha dicho, ayuda y da ventajas a los sistemas de recomendación[6][7]. El agregar información de contexto a los estudios e investigaciones puede llevar a obtener recomendaciones más eficientes para el usuario, es por ello que este trabajo abordará específicamente la agregación de datos en retroalimentación implícita teniendo en cuenta el contexto.

---

<sup>3</sup> Portal de ventas por internet, Amazon, disponible en <https://www.amazon.com>

<sup>4</sup> Red social y radio vía internet, Last.fm, disponible en <https://www.last.fm>

<sup>5</sup> Sitio web en el cual los usuarios pueden compartir y subir videos, aloja videos musicales, clips de películas, YouTube, disponible en <https://www.youtube.com>

<sup>6</sup> Red social donde se puede compartir contenidos de forma sencilla en internet, Facebook, disponible en <https://www.facebook.com>

## **1.2. Objetivos**

### **1.2.1. Objetivo General**

Desarrollar una estrategia de transformación de retroalimentación implícita en retroalimentación explícita, minimizando la pérdida de información de contexto, basada en estrategias existentes de transformación de retroalimentación sin contexto. Diseñando e implementando esquemas de transformación para este fin.

### **1.2.2. Objetivos Específicos**

- Estudiar estrategias de transformación basadas en datos implícitos sin tomar en cuenta información de contexto, y estrategias basadas en datos explícitos conscientes del contexto, y ver las diferencias entre ellas.
- Desarrollar una estrategia que permita transformar datos implícitos con información de contexto en datos explícitos, que pierdan la menor cantidad de información de contexto.
- Diseñar, implementar y probar la estrategia conformada por esquemas de transformación desarrollados en java.

### 1.3. Presentación de Capítulos

Este informe se presenta organizado en cinco capítulos, cuyo contenido se describe a continuación:

En el Capítulo 1 (**Introducción**) se presenta una descripción general de este trabajo, comienza con el origen del tema, seguido por los objetivos generales y específicos.

En el Capítulo 2 (**Marco teórico**) se describe que son los sistemas de recomendación, se explica lo es el contexto y la retroalimentación explícita e implícita y se presenta algoritmos encontrados en literatura, que manipulan información de contexto y retroalimentación Implícita.

En el Capítulo 3 (**Transformación de datos**) se explica que es y el por qué hacer transformación de datos. Se muestra el diseño de la estrategia a implementar complementados con pseudocódigos y esquemas. Además se describe su implementación en java y sus correspondientes diagramas de clases. Finalmente a lo implementado se le efectúan pruebas unitarias para verificar su desempeño.

En el Capítulo 4 (**Experimentos**) se aplica un sistema de recomendación con datos reales sobre cada método de transformación desarrollado y se efectúan mediciones a las recomendaciones generadas por el sistema de recomendación.

Finalmente en el Capítulo 5, **Conclusiones del proyecto**, se plantean los aportes y posibles trabajos futuros en la misma línea temática propuesta.

## 2. Marco Teórico

### 2.1. Sistemas de recomendación

En la actualidad existe gran cantidad de información en Internet, por lo que lograr encontrar lo que buscamos se vuelve difícil, además de requerir de bastante tiempo. Para dar solución a esto surgen los Sistemas de Recomendación (S.R.).

Básicamente, los Sistemas de Recomendación reciben información del usuario acerca de productos o servicios en los que el usuario se encuentra interesado y le recomienda aquéllos que estén más cercanos a sus necesidades[8].

Los Sistemas de Recomendación son herramientas de software y técnicas que proporcionan sugerencias de los artículos que sean útiles para el usuario[2].

R. Burke[9] define los Sistemas de Recomendación como “sistemas que producen recomendaciones personalizadas como salida o tienen el efecto de guiar al usuario de una forma personalizada a productos interesantes o útiles entre una gran cantidad de productos disponibles”.

Los sistemas de recomendación tienen como objetivo asistir a cada usuario en su proceso de búsqueda de información, realizando recomendaciones de forma que se simplifique al máximo la búsqueda que el usuario debe efectuar, estos sistemas utilizan las preferencias de los usuarios y mediante el procesamiento de estos datos son capaces de brindar a cada uno recomendaciones personalizadas. De acuerdo al tipo de información que utilizan para realizar las recomendaciones se pueden clasificar en diferentes tipos, siendo tres los principales: Basados en Contenido, de Filtrado Colaborativo y Enfoques Híbridos. Los primeros tratan de recomendar productos similares a los que le ha gustado a un usuario determinado en el pasado. Los segundos identifican a los usuarios cuyos gustos son similares a los de un usuario determinado y recomiendan a este usuario ítems que agradan a dichos usuarios “similares”. Los

terceros son sistemas construidos como una combinación entre uno o más sistemas de basados en contenidos y de filtrado colaborativos[2].

En la actualidad existen una amplia gama de sistemas de recomendación que son usados en diferentes áreas, ya sea con fines comerciales, científicos o experimentales. A continuación se mencionan algunos de los más conocidos o usados[10].

- Sistema de recomendación Fab (Sistema híbrido, basado en contenido y filtrado colaborativo, que recomienda al usuario páginas de Web)[11],
- Sistema de recomendación PHOAKS (sistema de recomendación que reconoce y redistribuye recomendaciones de recursos de Web buscando en mensajes electrónico)[12],
- Sistema de recomendación FilmAffinity (sistema de recomendación de películas, documentales, cortometrajes, medimetrajes y series de televisión)<sup>7</sup>
- Sistema de recomendación Movilens (sistema recomendación de películas que se basa en el filtrado colaborativo para generar recomendación de películas)<sup>8</sup>
- Sistema de recomendación Amazon (sistema de recomendación utilizado por la tienda virtual Amazon)<sup>9</sup>
- LastFm (Radio online que recomienda música, eventos y festivales en función de los gustos musicales del usuario)<sup>10</sup>

## 2.2. Historia de los sistemas de recomendación

---

<sup>7</sup> <http://www.filmaffinity.com>

<sup>8</sup> <http://www.movielens.org>

<sup>9</sup> <http://www.amazon.com/>

<sup>10</sup> <http://www.lastfm.es/>



Los primeros trabajos proceden de principios de los años 90 y en un comienzo los sistemas de recomendación eran conocidos tan sólo como filtros colaborativos [13]. Empezaron a surgir dentro de los servicios de noticias, los cuales filtraban noticias que permitían a su comunidad de usuarios acceder exclusivamente a aquellas noticias que potencialmente podían ser de su interés[14]. Uno de los primeros sistemas de recomendación que aparecen en la literatura es “Tapestry”[15], desarrollado por XeroX Palo Alto Research Center<sup>11</sup>. Tapestry es un sistema que permite almacenar las preferencias de los usuarios sobre los artículos o noticias que éstos han leído y posteriormente ser utilizado por otros usuarios que aún no han leído el artículo o noticia. Otra de las primeras formas de filtrado de información electrónica apareció con el trabajo de Housman y Kaskela[16] en el que se diseñó un método que de forma automática se pudiera mantener a los científicos informados sobre nuevos documentos publicados en sus áreas de trabajo o especialización.

Ya el año 1997, Resnick y Varian proponen llamar a estos sistemas con el nombre de “sistemas de recomendación”[17], dado que por esa fecha ya no se limitaban sólo al filtrado de información.

Según Ricci, Rokach, Shapira et al. [2] en los últimos años, el interés en los sistemas de recomendación ha aumentado de manera espectacular, como los siguientes hechos indican:

- Los sistemas de recomendación juegan un papel importante en sitios de Internet de alta connotación como Amazon.com (Página de compra por internet), YouTube o Last.fm (Sistemas de música y videos), Netflix (Servicio de alquiler de series y películas) e IMDb (Base de datos de películas en internet).
- En las instituciones de educación superior de todo el mundo, imparten cursos de pre y post grado dedicados por completo a los sistemas de Recomendación.

---

<sup>11</sup> <https://www.parc.com/>

Marco teórico

---

- Existen variedades de revistas académicas que cubren la investigación y desarrollo del campo. Entre las revistas que se han dedicado a hacer publicaciones sobre los Sistemas de Recomendación están: AI Communications (2008); IEEE Intelligent Systems (2007); International Journal of Electronic Commerce (2006); International Journal of Computer Science and Applications (2006); ACM Transactions on Computer-Human Interaction (2005); and ACM Transactions on Information Systems (2004).

## 2.3. Retroalimentación

Los sistemas de recomendación deben recopilar la mayor cantidad de información de los usuarios con la finalidad de descubrir sus gustos e intereses por determinados objetos y posteriormente realizar mejores recomendaciones. El mismo sistema proporciona un mecanismo para obtener esta información, el cual se llama proceso de retroalimentación y es clasificado en dos tipos: retroalimentación explícita y retroalimentación implícita [18].

### 2.3.1. Retroalimentación Explícita

En la retroalimentación explícita, el usuario valora conscientemente los contenidos mediante un sistema de calificación específico, expresando su interés en el artículo. Puede ser a través de un proceso de encuesta, donde el usuario valora los contenidos mediante la asignación de una puntuación[1]. Por ejemplo, la tienda online Amazon, Filmaffinity entre otros, utilizan el sistema de puntuación de “cinco estrellas” que permite a los usuarios valorar los productos que son de su interés. Por otro lado, las redes sociales como Facebook<sup>12</sup>, YouTube<sup>13</sup> y otros usan el sistema valoración “Like” para calificar los contenidos. Todos estos datos permiten al sistema incorporar nuevas preferencias que enriquezcan el proceso de recomendación.

---

<sup>12</sup> Red social donde se puede compartir contenidos de forma sencilla en internet, Facebook, disponible en <https://www.facebook.com>

<sup>13</sup> Sitio web en el cual los usuarios pueden compartir y subir videos, aloja videos musicales, clips de películas, YouTube, disponible en <https://www.youtube.com>.

### **2.3.2. Retroalimentación Implícita**

Este proceso consiste en evaluar los objetos, sin intervenciones adicionales de los usuarios. Es decir, a través de la captura de la información obtenida de las acciones realizadas por los usuarios en la aplicación en su uso normal. Por ejemplo, el conteo de las reproducciones de música o páginas web visitadas. A partir de esta información, el sistema puede deducir si el contenido es de su interés.

La retroalimentación implícita se han utilizado para recuperar, filtrar y recomendar una variedad de artículos: películas, artículos de revistas, documentos de Web, artículos de noticias en línea, libros, programas de televisión, y otros. Estas técnicas aprovechan del comportamiento del usuario para comprender los intereses y preferencias del usuario[19].

## **2.4. Contexto**

### **2.4.1. Descripción**

La inclusión del contexto en los sistemas de recomendación ha ido en aumento, debido a que la información capturada que tenga únicamente la relación ítem-usuario, lleva a generar recomendaciones menos precisas. Complementar esta información capturada con el contexto hace que el sistema posea más detalle acerca del usuario.

El contexto es un concepto multifacético que se ha estudiado a través de diferentes disciplinas de investigación, incluyendo la informática, la ciencia cognitiva, la lingüística, la filosofía, la psicología y ciencias de la organización [20]. En adelante se utilizará la siguiente definición de contexto, ampliamente utilizada en el área[21]:

“El contexto es cualquier información que se puede utilizar para caracterizar la situación de una entidad. Una entidad es una persona, un lugar o un objeto que se considera relevante para la interacción entre un usuario y una aplicación, incluyendo el usuario y aplicaciones en sí”.

## 2.4.2. Modelado información Contextual en los Sistemas de Recomendación.

Adomavicius y Tuzhilin et al.[20] presentan un esquema de modelado donde los sistemas de recomendación para lograr su objetivo de recomendar un artículo a un usuario, necesitan un grupo inicial de valoraciones o rating para establecer un perfil del usuario. En sus comienzos utilizaban un sistema de dos dimensiones ya que relacionaban únicamente al usuario y al ítem, determinando una función R que establece el rating asociado al par (usuario, ítem).

$$R: \text{Usuario} \times \text{Ítem} \rightarrow \text{Rating}$$

A partir de este modelo, es posible incorporar información sobre el contexto asociado a las interacciones entre usuarios e ítems, en cuyo caso la función de rating queda definida de la siguiente manera.

$$R: \text{Usuario} \times \text{Ítem} \times \text{Contexto} \rightarrow \text{Rating}$$

Como ya se ha mencionado el contexto puede ser cualquier información que intervenga o afecte la preferencia del usuario por un ítem, por ejemplo: estación del año, momento del día, compañía. Cada uno de estos contextos puede tender variedades de valores (Figura 1, 2), como por ejemplo:

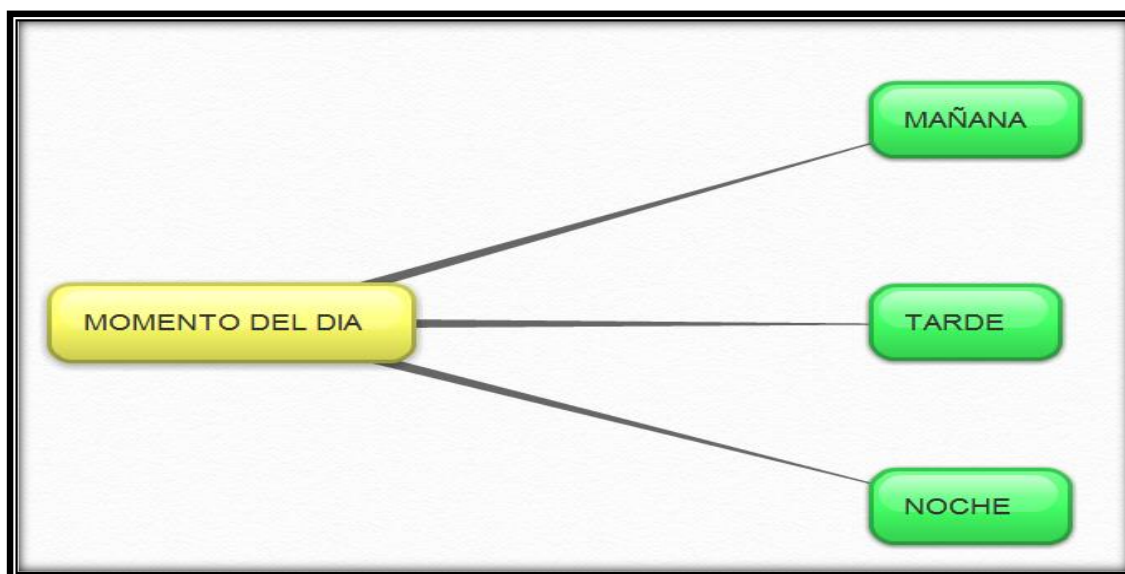


Figura 1: Valores que podría tomar el contexto momento del día.

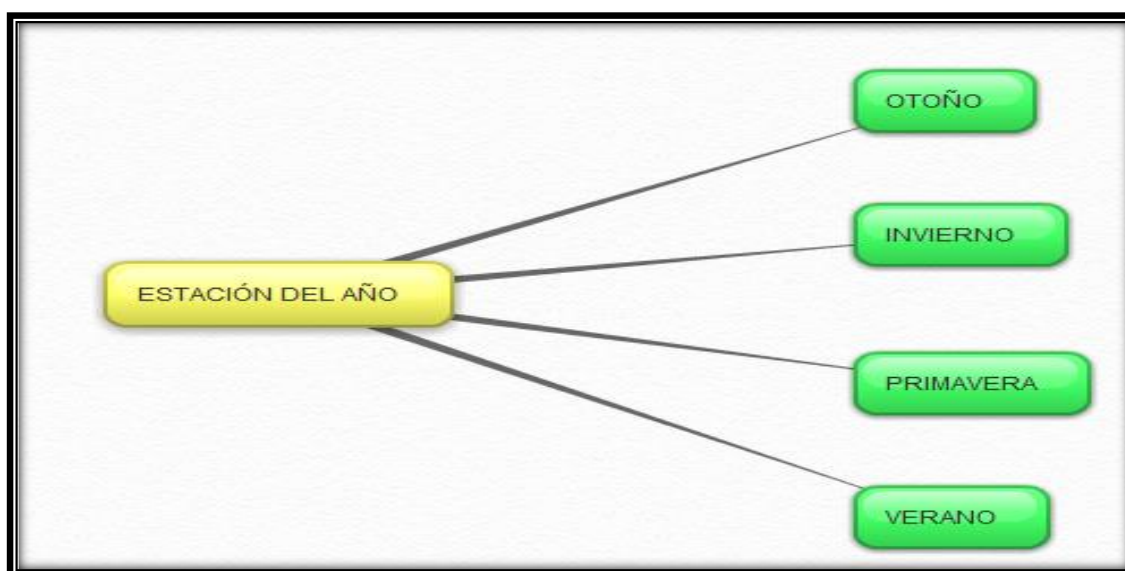


Figura 2: Valores que podría tomar el contexto Estación del año.

## 2.5. Distribución de Frecuencias

Una manera de agrupar datos es usar una tabla de frecuencia o una distribución de frecuencia. La distribución de frecuencia es una tabla que organiza los datos en base a la cantidad de observaciones de datos en los diferentes grupos, poniendo al principio los grupos o categorías con mayor número de observaciones.

Para ejemplificar mejor el uso de la Distribución de Frecuencia en este trabajo, en la Tabla 1 se muestran observaciones correspondientes a un usuario y cuatro ítems. Se usa como dato la Frecuencia absoluta, ordenada de mayor a menor y con ella se llega a determinar la Frecuencia acumulada complementaria relativa.

*Tabla 1: Tabla de Frecuencias.*

Posición	Ítem	Frecuencia Absoluta	Frecuencia Relativa	Frecuencia Acumulada	Frecuencia Acumulada Complementaria	Frecuencia Acumulada Complementaria Relativa
1°	1	200	0,641	200	312	1
2°	2	60	0,192	260	112	0,358
3°	4	50	0,160	310	52	0,166
4°	3	2	0,006	312	2	0,006
		<b>312</b>			<b>0</b>	

En adelante se explica cada columna de la tabla 1, para entender cómo se obtiene la Frecuencia Acumulada Complementaria Relativa, la cual se utiliza en este trabajo para realizar la transformación de los datos implícitos.

En este ejemplo, 213 es el número total de preferencias de un usuario.



**Frecuencia Absoluta:** Es el número de observaciones. En este ejemplo corresponde al número de veces que un usuario prefiere ese ítem.

**Frecuencia Relativa:** Esta columna indica el porcentaje que representa la frecuencia absoluta (de cada ítem) con respecto a la distribución total. La frecuencia relativa se obtiene dividiendo la frecuencia absoluta de cada posición, entre el número total de preferencias.

**Frecuencia Acumulada:** La frecuencia acumulada que aparece en la primera fila, siempre será igual a la frecuencia absoluta de la misma. Para las posiciones siguientes se suma a la Frecuencia acumulada anterior la Frecuencia absoluta de esa posición.

**Frecuencia Acumulada Complementaria:** La Frecuencia Complementaria es la inversa de la Acumulada; es decir, partiendo del total de preferencias, nos va indicando cómo van disminuyendo las observaciones (en este caso preferencias) a medida que se desciende en la tabla.

Se calcula de la siguiente forma: la Frecuencia Complementaria de la primera fila, será siempre igual al número total de observaciones, en este ejemplo al número total de preferencias de un usuario.

Para las siguientes posiciones, se deberá restar a la frecuencia complementaria anterior, la frecuencia absoluta de la posición anterior.

**Frecuencia Acumulada Complementaria Relativa:** Esta columna indica el porcentaje que representa la frecuencia complementaria de cada posición con respecto a la distribución total. La Frecuencia Acumulada Complementaria Relativa se encuentra dividiendo la Frecuencia Acumulada Complementaria de cada posición, entre el número total de preferencias.

## 2.6. Algoritmos estudiados

### 2.6.1. Retroalimentación Explícita consciente del Contexto

Baltrunas y Ricci et al.[22] proponen un algoritmo que toma en consideración, que en ciertos dominios el consumo de un mismo artículo puede provocar diferentes experiencias cuando cambia el contexto. Por ejemplo en una aplicación de turismo, la experiencia de visitar la playa en verano es notablemente diferente a visitarla en invierno. Los sistemas de recomendación tradicionales no distinguen entre estas dos experiencias, otorgando una mala recomendación en estas situaciones.

Este algoritmo hace frente al contexto, presentando una técnica llamada ítem dividido. Este enfoque utiliza una matriz bidimensional de  $m \times n$  donde  $m$  son los usuarios y  $n$  los ítems y cada celda representa el rating otorgado por un usuario a un ítem, además en la celda se almacena información contextual que describe, bajo que condición el usuario asigno la preferencia.

La finalidad de esta técnica es que el conjunto de calificaciones que entrega cada usuario de un artículo, se dividen en dos subconjuntos de calificaciones, de acuerdo a los ratings que existan en cada valor del contexto. Por ejemplo separar los ratings con contexto “estación del año”, lo que implica separar por un lado los ratings que fueron percibidos en invierno y por otro lado aquellos que fueron percibidos en verano. Estos dos grupos de ratings son asignados a dos nuevos ítems ficticios que se desprenden del ítem original, (por ejemplo el grupo de ratings de invierno asignado a un ítem<sub>A</sub>, y el grupo de ratings de verano asignado a un ítem<sub>B</sub>, pero tanto con ítem<sub>A</sub> como con ítem<sub>B</sub> se refieren al mismo ítem original).

Esta división se realiza únicamente si existe evidencia estadística, que demuestre que bajo estos dos valores de contextos los usuarios entregan diferentes ratings del ítem en cuestión.

Marco teórico

Para ello este método busca la variable contextual que será usada para dividir e identifica si los ítems tienen diferencias significativas en sus ratings. Si es el caso de que exista evidencia y la división es beneficiosa, esta se hace y en la matriz de ratings el ítem original es reemplazado por dos nuevos ítems generados. Para determinar si la división es beneficiosa se define un criterio de impuridad, la estrategia presenta cinco tipos de criterios, aquí se muestra uno ellos:

Uno criterio de impuridad puede ser la cantidad de valores mayor a 4 que existe en cada contexto, si al comparar estas cantidades, estas presentan una diferencia notoria, esto significa, que existe evidencia para efectuar la división y la división resultaría beneficiosa.

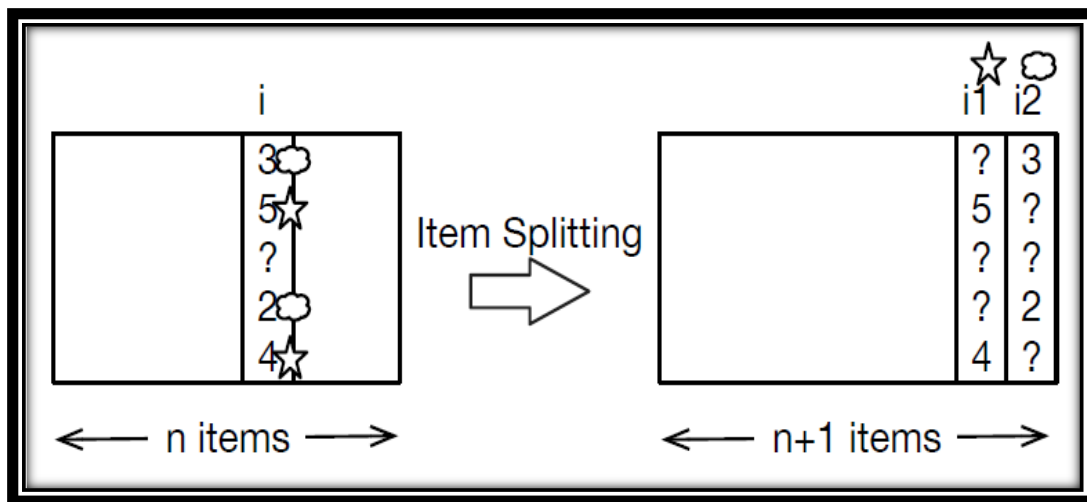


Figura 3: División de ítems [22].

La Figura 3 ilustra la división de un ítem *i* con contexto “estación del año”, donde los valores del contexto son invierno que es la nube y verano la estrella, el signo interrogación indica que no hay información del usuario. Existe evidencia significativa para realizar la división, ya que los valores menores de ratings están en invierno y los valores altos de ratings están en verano. Se realiza la división, quedando así el ítem separado en *i1* e *i2* (reemplazando al ítem original *i*), en la columna *i1*, sólo los valores

de ratings de invierno y en la columna **i2**, sólo los valores de ratings de verano. Los signos de interrogación continúan indicando que no hay información del usuario para ese valor del contexto.

Como entrada, una matriz de ratings de  $m \times n$ ,  $m$  usuarios y  $n$  ítems. Como salida una matriz  $m \times n + 1$  ítems. El número total de ratings en la matriz no cambia.

### **Seudocódigo del Algoritmo**

Input matriz  $m \times n$  ( $m$  usuarios,  $n$  ítems)

1. Recorre cada ítem de la matriz, obteniendo el puntaje asignado por el usuario y el valor de contexto asociado.
2. Verifica si existe valores de contexto diferentes para cada puntaje entregado.
3. Verifica si existe evidencia significativa entre los puntajes entregados bajos los diferentes contextos, con el criterio de impuridad.
4. Si existe evidencia y la separación resulta beneficiosa, el ítem queda dividido en dos subgrupos de puntajes para cada valor de contexto encontrado.
5. Output matriz  $m \times n+1$  ( $m$  usuarios,  $n$  ítems)

### 2.6.2. Retroalimentación Implícita

Celma et al. [23] propone un algoritmo que transforma datos implícitos en datos explícitos, ya que las primeras investigaciones de los sistemas de recomendación en el dominio de la música utilizan retroalimentación explícita, pero la captura de forma implícita de las reproducciones de los usuarios, se ha convertido en lo más común.

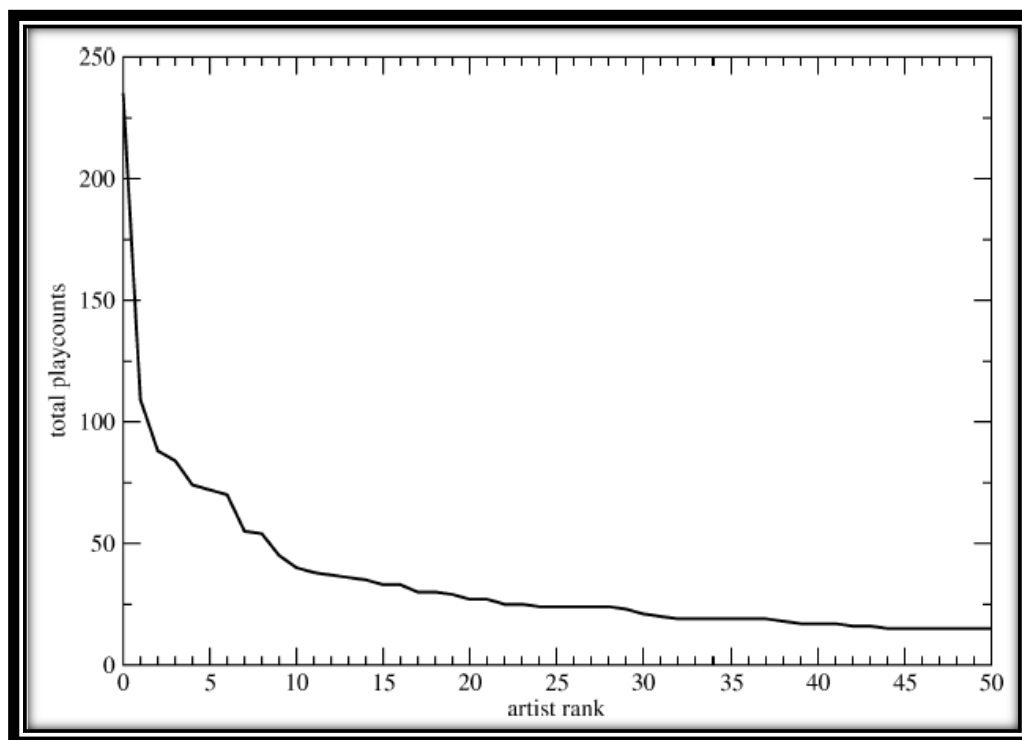
Para utilizar la retroalimentación implícita se sugiere:

- Convertir los datos implícitos en una matriz sencilla binaria, donde las celdas con 1 significan que el usuario ha escuchado ese artista al menos una vez.
- Transformar los datos implícitos en una matriz normalizada. En lugar de asignar 0 o 1 en las celdas, colocar un valor que demuestre que el usuario escucha ese artista, por ejemplo un 5 si escucha mucho a ese artista y 1 si lo ha escuchado al menos una vez.
- Normalizar cada fila (usuarios), de modo que la suma de las filas sea igual a 1. Esta opción describe la distribución de probabilidad de un artista con respecto a un usuario.
- Crear una matriz usuario-artista con la cantidades de veces que se escucha ese artista en sus celdas. Aquí no hay una normalización como la matriz contiene valores absolutos.

Para el desarrollo del experimento propuesto en Celma, se utilizan datos obtenidos de last.fm<sup>14</sup>, transformando el conjunto de datos implícitos en una matriz M de usuario-artista. Donde cada celda de la matriz representa el número de veces que el usuario escuchó al artista. Para normalizar la matriz con valores entre 1..5, se utiliza la distribución acumulada complementaria por usuario, para evitar que los artistas queden distribuidos de manera, que pocos obtendrían ratings altos y muchos artistas obtengan ratings bajos.

---

<sup>14</sup> Red social y radio vía Internet y además un sistema de recomendación de música, disponible en <http://www.last.fm/>



*Figura 4: Reproducciones efectuadas por usuario representadas con distribución de frecuencia, la horizontal son los artistas top-50 y la vertical el número de veces que se reprodujo ese artista[23].*

En la Figura 4 se muestra la distribución de las reproducciones de los artistas sin aplicar la distribución acumulada complementaria. Se puede observar que la mayoría de los artistas escuchados por el usuario están entre 0 y 50 reproducciones y que sólo uno tiene sobre las 200 reproducciones. Esto hace que la distribución sea muy sesgada a la derecha y poco equitativa, para ello se le aplica la distribución acumulada complementaria.

Luego de aplicar la distribución acumulada complementaria, los artistas se localizan en segmentos de la distribución y según su ubicación se le asigna un rating. Entre el 80 – 100% de la distribución obtienen un 5, los que estén entre un 60 – 80% obtendrán un 4 y así hasta que los que estén entre 0 – 20% de la distribución se les asigne un 1.

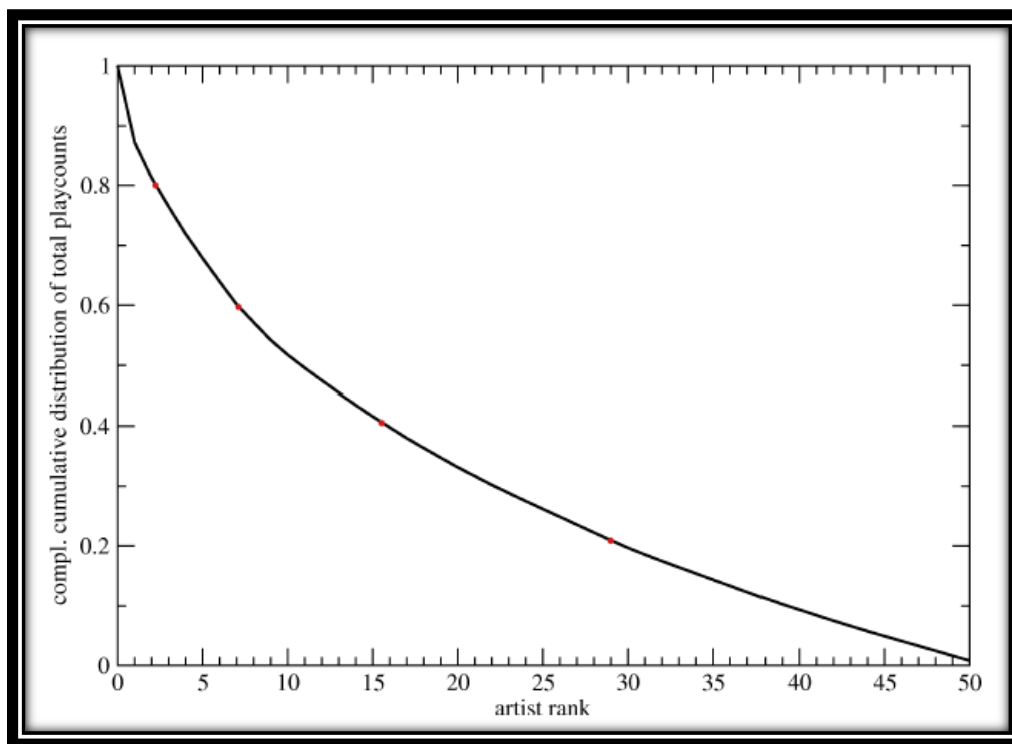


Figura 5: Reproducciones efectuadas por usuario de la Figura 4 representadas con la distribución acumulada complementaria[23].

En la Figura 5 se muestra la aplicación de la distribución acumulada complementaria en la Figura 4. Acá se observa que reproducciones de los artistas las que realizó el usuario ya no están con sesgo, sino con una distribución más equitativa para aplicar la transformación de cantidad de reproducciones por artista a ratings por artista.

El algoritmo además propone que en casos donde las reproducciones por usuario no tienen sesgo, sino más bien son homogéneas, es decir, las reproducciones de cada artista son muy parecidas entre ellas. En este caso se usa el coeficiente de variación CV, el cual divide la desviación estándar de cada usuario por el promedio de cada usuario. Si el CV es menor o igual a 0,5 no se aplica la distribución acumulada complementaria si no que se asigna el valor de rating igual a 3, a todos esos artistas. Esto quiere decir que esos artistas son escuchados la misma cantidad de veces por el usuario.

### **Seudocódigo algoritmo**

Input matriz  $m \times n$  ( $m$  usuarios,  $n$  artistas)

1. Se toma cada fila de la matriz, correspondiente a cada usuario.
2. Se ordena el número de reproducciones de la fila de mayor a menor (que transforma en la tabla de distribución de frecuencias).
3. Se calcula el número total de reproducciones para cada usuario.
4. Para cada celda, se calcula frecuencia relativa (valor de celda dividido por total de reproducciones).
5. Para cada celda, se calcula frecuencia relativa acumulada.
6. Para cada celda, de calcula frecuencia acumulada complementaria porcentual.
7. Para cada celda, se calcula un valor de rating de acuerdo a lo siguiente:
  - Si la frecuencia acumulada se encuentra entre 80 – 100%, se asigna un rating de 5.
  - Si la frecuencia acumulada se encuentra entre 60 – 80%, se asigna un rating de 4.
  - Si la frecuencia acumulada se encuentra entre 40 – 60%, se asigna un rating de 3.
  - Si la frecuencia acumulada se encuentra entre 20 – 40%, se asigna un rating de 2.
  - Si la frecuencia acumulada se encuentra entre 10 – 20%, se asigna un rating de 1.



### **3. Transformación de datos implícitos a datos explícitos**

Los sistemas de recomendación necesitan las preferencias de los usuarios para generar las recomendaciones y la forma más fácil y abundante de obtenerlas, es a través de la retroalimentación implícita [1]. Estos datos implícitos como se ha descrito en el capítulo 2(Marco teórico), sección Retroalimentación, son el resultado de almacenar las acciones que realice cada usuario. Pero en sitios webs donde la cantidad de usuarios es sobre el millón y las acciones que realice cada uno de ellos son numerosa, lleva a la obtención de un gran volumen de información. Por ejemplo en Facebook, el número de usuarios es más de 1.700 millones, en Twitter es de 313 millones<sup>15</sup>. Si multiplicamos cada usuario por cada actividad que realice en el sitio, la cantidad de registros se eleva considerablemente. Al tener este gran volumen de registros para ingresar a los Sistemas de Recomendación y que además los algoritmos de los Sistemas de Recomendación suelen hacer cálculos complejos, el lograr reducir la cantidad de información que se le entrega al sistema, hace que al momento de calcular la recomendación, el procesamiento de datos sea menor. Esta es una de las finalidades de realizar esta transformación de datos.

Además, es importante destacar que muchos algoritmos de recomendación se basan en la noción de rating (retroalimentación explícita) para calcular recomendaciones, por lo que no son compatibles con datos de retroalimentación implícita. Por ello, el realizar la transformación y obtener datos explícitos, permitirá aplicar tales algoritmos para la obtención de recomendaciones en casos donde sólo se cuenta con información de retroalimentación implícita.

En este proyecto utilizamos el contexto para realizar una clasificación de los datos implícitos. Además se utiliza la Distribución Acumulada Complementaria para

---

<sup>15</sup> Estadísticas de redes sociales: <http://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/>

### Transformación de datos implícitos a datos explícitos

---

determinar valores de rating correspondientes a la información implícita, realizando de esta forma la transformación de datos implícitos a explícitos.

En las siguientes tablas se muestra una transformación de un conjunto de registros de datos implícitos, que corresponden a un usuario. Como ítems se utiliza un grupo de canciones, además la fecha y hora en que fueron escuchadas las canciones. Con esta última información se determinan dos contextos, con los cuales se trabajan clasificaciones de acuerdo a cada combinación de valores que de cada contexto.

En la Tabla 2 se muestra un grupo de registros de datos implícitos a transformar en datos explícitos, este conjunto de datos consta de diez lecturas de un usuario que escuchó variedades de canciones.

*Tabla 2: Conjunto de registros de usuario U1.*

#	Usuario	Ítem	Fecha y hora
1	U <sub>1</sub>	Kelly Clarkson - Stronger	02/02/2015 22:35
2	U <sub>1</sub>	Radiohead - Creep	07/05/2015 20:23
3	U <sub>1</sub>	Kelly Clarkson - Stronger	24/12/2015 21:08
4	U <sub>1</sub>	Fifth Harmony - Work from Home	28/07/2014 16:35
5	U <sub>1</sub>	Rihanna - Kiss It Better	09/03/2014 23:34
6	U <sub>1</sub>	Kelly Clarkson - Stronger	09/04/2015 09:34
7	U <sub>1</sub>	Radiohead - Creep	09/10/2014 22:34
8	U <sub>1</sub>	Radiohead - Creep	11/10/2014 21:59
9	U <sub>1</sub>	Fifth Harmony - Work from Home	29/07/2014 16:35
10	U <sub>1</sub>	Kelly Clarkson - Stronger	28/07/2014 16:35

En la Tabla 3 se muestra la cantidad de veces que cada canción fue escuchada por el usuario. Se puede observar que la canción más escuchada es “Kelly Clarkson - Stronger”, la cual la escuchó 4 veces.

## Transformación de datos implícitos a datos explícitos

Tabla 3: Conteo de canciones.

#	Usuario	Canción	Conteo
1	U <sub>1</sub>	Kelly Clarkson - Stronger	4
2	U <sub>1</sub>	Radiohead - Creep	3
3	U <sub>1</sub>	Fifth Harmony - Work from Home	2
4	U <sub>1</sub>	Rihanna - Kiss It Better	1

La Tabla 4 corresponde a la transformación de la Tabla 3 en ratings, con valores entre [1...5]. El conteo con valor más alto obtiene el rating de 5 y así sucesivamente decreciendo hasta llegar al conteo con valor 1, que obtiene rating 1.

Tabla 4: Asignación de ratings sin considerar contextos.

#	Usuario	Canción	Ratings
1	U <sub>1</sub>	Kelly Clarkson - Stronger	5
2	U <sub>1</sub>	Radiohead - Creep	2
3	U <sub>1</sub>	Fifth Harmony - Work from Home	2
4	U <sub>1</sub>	Rihanna - Kiss It Better	1

La Tabla 5 es la determinación de los contextos en base a la información de la Tabla 2, información de la fecha y hora en que el usuario escuchó cada canción. Con ella se determinaron dos contextos, el primero corresponde a “momento del día”, con valores “día” y “noche”, el segundo corresponde a “momento de la semana”, con valores “semana” y “fin de semana”.

Transformación de datos implícitos a datos explícitos

Tabla 5: Registros con contextos establecidos

#	Usuario	Canción	(contexto 1) Momento de día	(contexto 2) Momento de la semana
1	U <sub>1</sub>	Kelly Clarkson - Stronger	noche	semana
2	U <sub>1</sub>	Radiohead - Creep	noche	semana
3	U <sub>1</sub>	Kelly Clarkson - Stronger	día	fin de semana
4	U <sub>1</sub>	Fifth Harmony-Work from Home	día	fin de semana
5	U <sub>1</sub>	Kelly Clarkson - Stronger	noche	semana
6	U <sub>1</sub>	Kelly Clarkson - Stronger	día	fin de semana
7	U <sub>1</sub>	Radiohead - Creep	noche	semana
8	U <sub>1</sub>	Radiohead - Creep	noche	fin de semana
9	U <sub>1</sub>	Fifth Harmony - Work from Home	día	fin de semana
10	U <sub>1</sub>	Kelly Clarkson - Stronger	noche	semana

La Tabla 6 es la selección de canciones escuchadas de “noche” y de “semana”, con sus respectivas cantidades de veces escuchadas.

Tabla 6: Conteo de canciones con valor de contexto “noche” y “semana”.

#	Usuario	Canción	Contexto 1	Contexto 2	Conteo
1	U <sub>1</sub>	Kelly Clarkson - Stronger	Noche	Semana	3
2	U <sub>1</sub>	Radiohead - Creep	Noche	Semana	2

La Tabla 7 es la selección de canciones escuchadas de “noche” y de “fin de semana”, con sus respectivas cantidades de veces escuchadas.

Transformación de datos implícitos a datos explícitos

Tabla 7: Conteo de canciones con valor de contexto “noche” y “fin de semana”

#	Usuario	Canción	Contexto 1	Contexto 2	Conteo
1	U <sub>1</sub>	Radiohead - Creep	Noche	Fin de semana	1

La Tabla 8 es la selección de canciones escuchadas de “día” y de “fin de semana”, con sus respectivas cantidades de veces escuchadas.

Tabla 8. Conteo de canciones con valor de contexto “día” y “fin de semana”.

#	Usuario	Canción	Contexto 1	Contexto 2	Conteo
1	U <sub>1</sub>	Fifth Harmony - Work from Home	Día	Fin de semana	2
2	U <sub>1</sub>	Kelly Clarkson - Stronger	Día	Fin de semana	2

La Tabla 9 es la transformación a ratings de la Tabla 6, donde a las canciones escuchadas de “noche” y de “semana” se les asignó un rating de acuerdo a los conteos de la Tabla 6. La canción “Kelly Clarkson - Stronger” es la que obtiene rating 5, ya que en la Tabla 6 tiene el conteo más alto.

Tabla 9. Rating para valor de contexto “noche” y “semana”.

#	Usuario	Canción	Contexto 1	Contexto 2	Ratings
1	U <sub>1</sub>	Kelly Clarkson - Stronger	Noche	Semana	5
2	U <sub>1</sub>	Radiohead - Creep	Noche	Semana	4

Transformación de datos implícitos a datos explícitos

La Tabla 10 es la transformación a ratings de la Tabla 7, donde a las canciones escuchadas de “noche” y de “fin de semana” se les asignó un rating de acuerdo a los conteos de la Tabla 6.

*Tabla 10: ratings para valor de contexto "noche" y "fin de semana".*

#	Usuario	Canción	Contexto 1	Contexto 2	Ratings
1	U <sub>1</sub>	Radiohead - Creep	Noche	Fin de semana	5

La Tabla 11 es la transformación a ratings de la Tabla 8, donde a las canciones escuchadas de “día” y de “fin de semana” se les asignó un rating de acuerdo a los conteos de la Tabla 6. En esta tabla las canciones “Kelly Clarkson - Stronger” y “Fifth Harmony - Work from Home” obtienen ratings iguales, ya que en la Tabla 8 ambas canciones tienen un conteo igual.

*Tabla 11: Ratings para valor de contexto "día" y "fin de semana".*

#	Usuario	Canción	Contexto 1	Contexto 2	Ratings
1	U <sub>1</sub>	Fifth Harmony - Work from Home	Día	Fin de semana	3
2	U <sub>1</sub>	Kelly Clarkson - Stronger	Día	Fin de semana	3

Los criterios para asignar ratings pueden ser diversos. Para este ejemplo el criterio fue que a mayor conteo, se asignara el rating mayor (rating 5) y así sucesivamente, a

## Transformación de datos implícitos a datos explícitos

---

medida que disminuye el conteo disminuye el rating asignado, fuese separado por contexto o bien sin hacer división por contexto.

En la sección siguiente se presenta el criterio para asignar rating que se ideó en la estrategia desarrollada para este proyecto.

### **3.2. Estrategia de transformación de datos ideada**

La estrategia presentada en este informe se basa en el algoritmo descrito por Celma et al. [23], ya que utiliza la Distribución Acumulada Complementaria como base para transformar datos de Implícitos a Explícitos y también utiliza la idea de Baltrunas y Ricci et al.[22], que busca mostrar las diferencias en la generación de ratings bajo contextos diferentes.

En primera instancia la estrategia clasifica en grupos las preferencias, de acuerdo a los valores de los contextos que las acompañen, posteriormente se cuentan los registros que quedan en cada clasificación, es decir, la cantidad de veces que cada usuario prefirió un ítem, bajo ciertos valores de contexto distintos entre ellos. A las cantidades obtenidas después del conteo se le aplica la Distribución Acumulada Complementaria por cada usuario, la que segmenta los datos en cinco partes y a cada uno le asigna un rating distinto de [1-5]. La aplicación de la Distribución Acumulada Complementaria es por usuario en cada grupo obtenido y por usuario de todos los grupos obtenidos, así la comparación se efectúa en todos los grupos y en cada uno de ellos.

### **3.3. Diseño e Implementación de la Estrategia de agregación de datos**

### 3.3.1. Diseño

La estrategia de transformación que se desarrolla en este proyecto, transforma registros de preferencias de usuarios (datos implícitos) a ratings (datos explícitos), tomando en consideración los valores de contextos que tienen los registros. En el diseño se describe como se relacionó cada dato del grupo de registros capturados.

Para esquematizar y organizar los datos implícitos, lo hacemos en una matriz de dos dimensiones (Figura 6), donde la primera dimensión (filas) hace referencia a los usuarios, la segunda (columnas) a los ítems y cada celda contiene una frecuencia, es decir, la cantidad de veces que un usuario prefiere ese ítem y además los contextos para ese usuario y ese ítem.

Ítems





Transformación de datos implícitos a datos explícitos

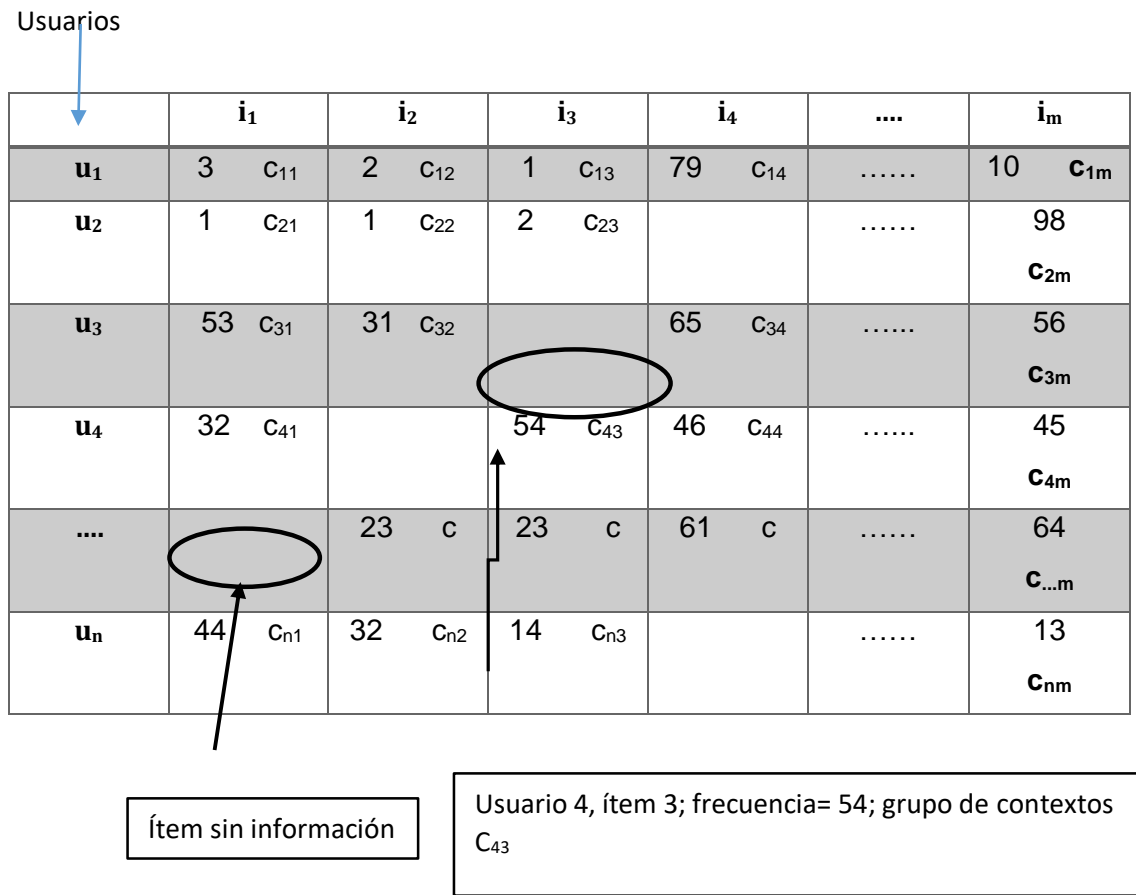


Figura 6: Matriz de diseño

### 3.3.2. Implementación

Para implementar la agregación de datos se decidió utilizar el entorno de desarrollo integrado Netbeans, el cual funciona bajo el lenguaje Java, lenguaje que en el transcurso de la mis estudios es muy utilizado y solicitado por los profesores. Además este entorno de desarrollo posee características como el control de versiones y refactoring, lo cual lo hacen ideal para el desarrollo de un proyecto donde existen varias personas que necesitan seguir ramas distintas de este. También cabe destacar que la

Transformación de datos implícitos a datos explícitos

estrategia diseñada se adiciona a un proyecto mayor que igualmente está desarrollado en lenguaje Java. El proyecto fue realizado por el profesor guía con la finalidad de estudiar los Sistemas de Recomendación (Figura 7). En él se encuentran cuatro bloques principales: Interfaz de datos: que se refiere a la parte donde se realiza la carga de los datos implícitos, Sistema de Recomendación: es un Sistema de Recomendación que simula recomendaciones en base a los registros que se le ingresen, Evaluación de resultados: son las pruebas que se efectúen el Sistema de Recomendación y por ultimo Transformación de datos: son donde se ubica la estrategia de agregación de datos desarrollada en este proyecto, para el sistema.

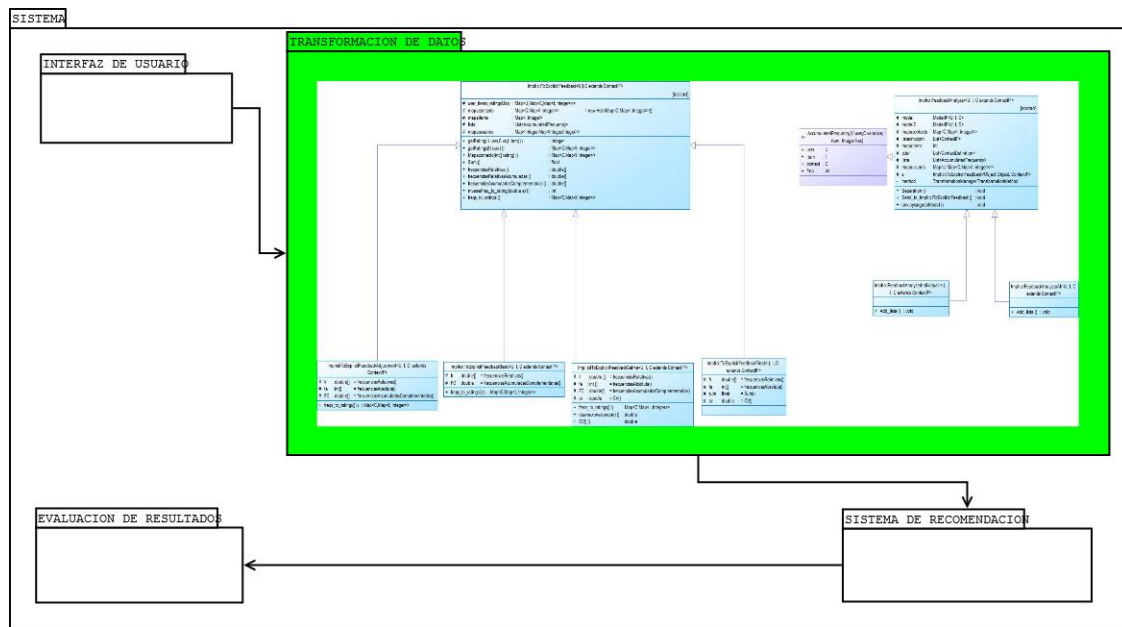


Figura 7: Esquema Global del Sistema

A continuación se presenta en detalle los esquemas de transformación que conforman la Estrategia de Transformación de datos implícitos a datos explícitos para Sistemas de Recomendación consciente del contexto implementada.

## Transformación de datos implícitos a datos explícitos

---

Antes de comenzar con la clasificación y posterior transformación, los datos que vienen de forma implícita se cargan en un modelo, del cual se puede obtener variedad de información de la carga efectuada, como por ejemplo la cantidad de usuarios, la cantidad de ítem y un listado de preferencias por usuario (por ejemplo los ítems que prefiere cada usuario y bajo que contextos eligió cada uno de ellos), este modelo es el que utilizaremos principalmente para el desarrollo de los esquemas de transformación.

Para utilizar la información contextual en el proceso de separación de las preferencias por valor de contexto se creó una clase llamada “ImplicitFeedbackAnalyzer”, la cual es una clase padre encargada de hacer la separación y posteriormente contar las preferencias en las cuales sus contextos coincidan con los que se les entregue, para ello en el constructor se recibe bajo que contextos se efectúe dicha separación. Esta clase hereda a dos clases llamadas “ImplicitFeedbackAnalyzerAll” e “ImplicitFeedbackAnalyzerIndividual”. La primera es la encargada de enviar una lista de preferencias separadas por usuario, donde se incluyen todos los contextos. La segunda envía listas por contextos encontrado de cada usuario. Por ejemplo si un grupo de preferencias se puede clasificar en las que fueron efectuadas de día y las que se efectuaron de noche, la clase “ImplicitFeedbackAnalyzerAll” generaría una lista por cada usuario con las preferencias tanto de día como de noche, para la clase “ImplicitFeedbackAnalyzerIndividual” generaría dos listas, por cada usuario, una donde están sólo las preferencias de noche y otra sólo las preferencias de día. Estas listas contienen instancias de una interfaz llamada “AccumulatedFrequency”, la cual guarda: usuario, ítem, frecuencia e información de contexto.

La transformación a datos explícitos de estas listas la efectúa la clase “ImplicitToExplicitFeedback”, la cual posee métodos que calculan frecuencias absolutas, frecuencias relativas, frecuencias relativas acumuladas, frecuencias acumuladas complementarias y un método para convertir a ratings las preferencias con apoyo de los métodos mencionados. Esta clase hereda a otras cuatro clases llamadas “ImplicitToExplicitFeedbackBasic”, “ImplicitToExplicitFeedbackAdjustment”, “ImplicitToExplicitFeedbackCelma” e “ImplicitToExplicitFeedbackFinal”. Las primeras tres aportan diferentes métodos para llegar a la mejor representación de los ratings, la última clase usa todos los métodos de las clases anteriores, haciéndola más completa.

### 3.3.3. Diagrama de Clases

Se presentan tres diagramas de clases, el primero representa las clases encargadas de clasificar por contexto, el segundo diagrama hace referencias a las clases encargadas de hacer la transformación a ratings y un tercero es la representación general de todas las clases e interfaz que interactúan.

En la Figura 8, se puede observar una clase padre `ImplicitFeedbackAnalyzer`, la cual utiliza una interfaz y hereda a dos clases, `ImplicitFeedbackAnalyzerAll` y `ImplicitFeedbackAnalyzerIndividual`.

#### 3.3.3.1. Diagrama de clases `ImplicitFeedbackAnalyzer`

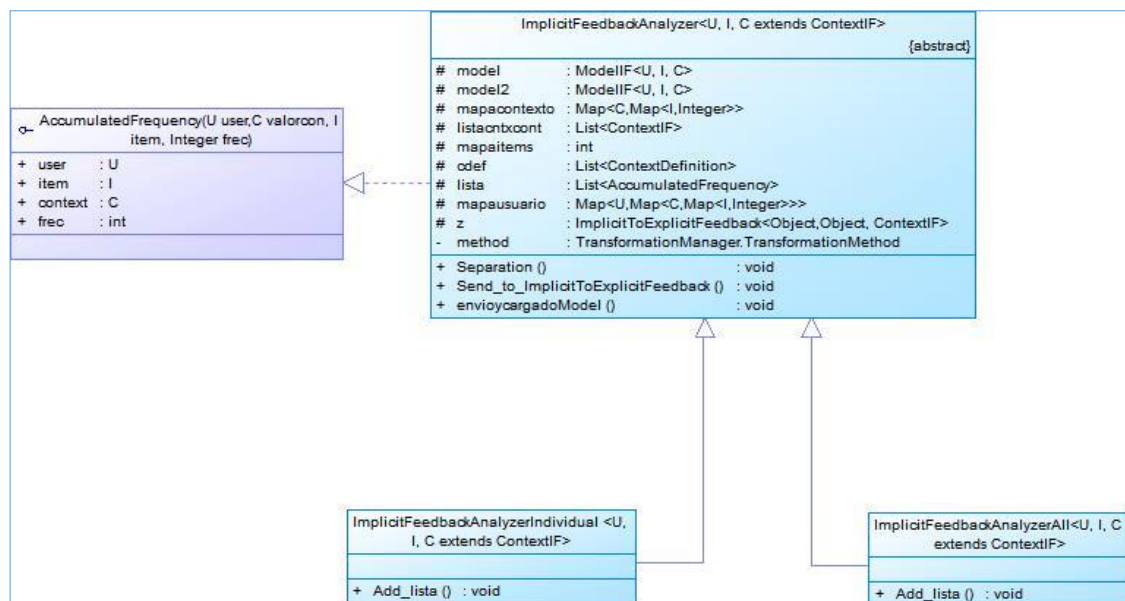


Figura 8: Diagrama de clases `ImplicitFeedbackAnalyzer`, clase `ImplicitFeedbackAnalyzerIndividual` y clase `ImplicitFeedbackAnalyzerAll`

Transformación de datos implícitos a datos explícitos

El diagrama de la Figura 9, lo componen las clases encargadas de realizar la transformación de los datos. Una clase padre `ImplicitToExplicitFeedback` que hereda cuatro clases, `ImplicitToExplicitFeedbackBasic`, `ImplicitToExplicitFeedbackAdjustment`, `ImplicitToExplicitFeedbackCelma` y `ImplicitToExplicitFeedbackFinal`.

3.3.3.2. Diagrama de clases “ImplicitToExplicitFeedback

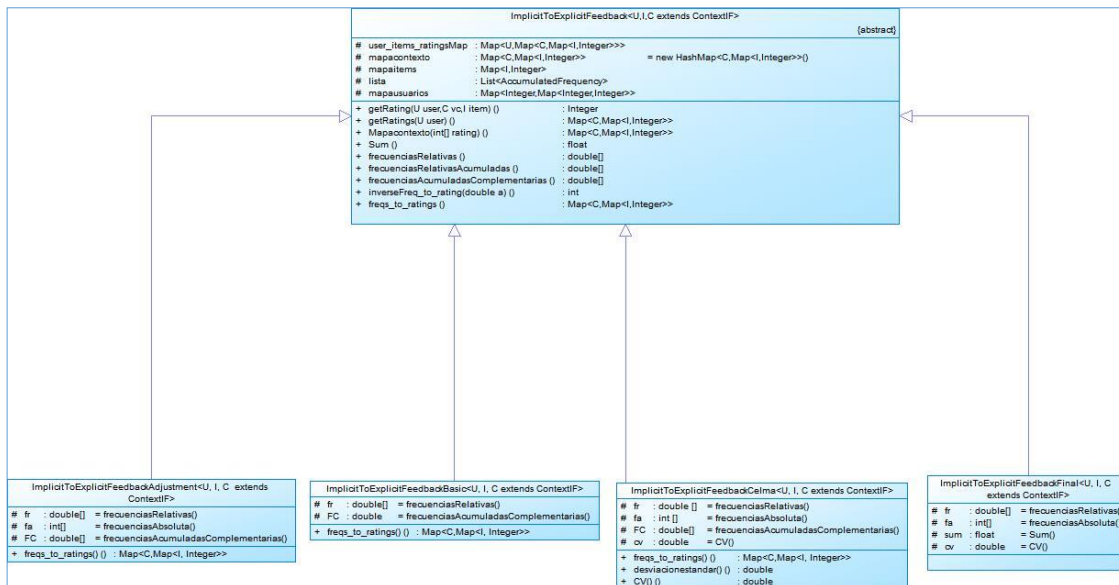


Figura 9: Diagrama de clases `ImplicitToExplicitFeedback`, clase `ImplicitToExplicitFeedbackAdjustment`, clase `ImplicitToExplicitFeedbackBasic`, clase `ImplicitToExplicitFeedbackCelma` y clase `ImplicitToExplicitFeedbackFinal`.

## Transformación de datos implícitos a datos explícitos

En la Figura 10 se muestran la estrategia conformada por las clases que corresponden a cada esquema de transformación y además las clases que realizan las separaciones por contextos.

### 3.3.3.3. Diagrama General

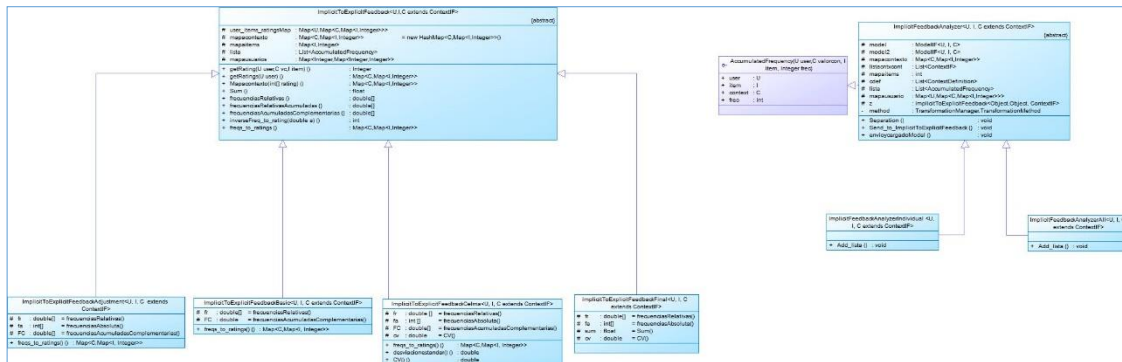


Figura 10: Diagrama General

### 3.3.3. Pruebas unitarias

A continuación se muestran pruebas unitarias, realizadas a cada esquema de transformación que se implementó. Para ello se utiliza una matriz original y dos matrices que se desprenden de ella en base a dos contextos diferentes.

Como entrada se presenta una Matriz original (Tabla 12) donde cada celda es la frecuencia que el usuario ( $u_i$ ) ha preferido un ítem ( $i_j$ ). Las matrices C1 (Tabla 13) y C2 (Tabla 14) representan la frecuencia del mismo usuario ( $u_i$ ) ha preferido del mismo ítem ( $i_j$ ), en dos contextos. Por ello que cada frecuencia de la matriz original queda dividida en dos frecuencias menores en sus correspondientes celdas en las matrices C1 y C2. Para ejemplificar mejor se consideró como contexto “Momento del día” y C1 representa las preferencias que fueron realizadas en la “Noche” y C2 las preferencias que fueron realizadas de “Día”.

#### Entrada:

*Tabla 12: Matriz Original, de preferencias generales por cada usuario e ítem, sin considerar contexto.*

	i1	i2	i3	i4
u1	20	10	2	5
u2	300	3	2	4
u3	7	3	3	7
u4	2	1	2	6

Transformación de datos implícitos a datos explícitos

---

Tabla 13: Matriz C1, contiene preferencias sólo en el contexto "Noche".

	i1	i2	i3	i4
u1	10	4	1	3
u2	100	2	1	2
u3	5	1	1	3
u4	1	1	1	3

Tabla 14: Matriz C2, contiene preferencias sólo en el contexto "Día".

	i1	i2	i3	i4
u1	10	6	1	2
u2	200	1	1	2
u3	2	2	2	4
u4	1		1	3



Transformación de datos implícitos a datos explícitos

---

**Esquema de transformación ImplicitToExplicitFeedbackBasic:** En este método se calculan los ratings en base al algoritmo de Celma, utilizando las frecuencias absolutas, frecuencias relativas, frecuencias relativas acumuladas y por ultimo las frecuencias complementarias. Con las últimas frecuencias se determina el valor del rating de acuerdo a estos rangos: 80 – 100% obtiene un puntaje de 5, entre 60 – 80% obtiene de puntaje 4, entre 40 – 60% obtiene un puntaje de 3 y así sucesivamente hasta llegar al rango de 0 – 20% que se le asigna un 1.

En las siguientes figuras (Figura 11, 12, 13) se muestra la salida al aplicar este esquema de transformación.

**Salida:**

U:1	5.0	3.0	1.0	1.0
U:2	5.0	1.0	1.0	1.0
U:3	5.0	2.0	1.0	4.0
U:4	3.0	1.0	2.0	5.0

Figura 11: Matriz original transformada a ratings al aplicar Esquema de transformación ImplicitToExplicitFeedbackBasic.

CONTEXTO: MomentOfDay:night				
U:1	5.0	3.0	1.0	2.0
U:2	5.0	1.0	1.0	1.0
U:3	5.0	2.0	1.0	3.0
U:4	3.0	2.0	1.0	5.0

Figura 12: Matriz C1 transformada a ratings al aplicar Esquema de transformación ImplicitToExplicitFeedbackBasic.

CONTEXTO: MomentOfDay:day				
U:1	5.0	3.0	1.0	1.0
U:2	5.0	1.0	1.0	1.0
U:3	4.0	3.0	2.0	5.0
U:4	3.0	null	2.0	5.0

Figura 13: Matriz C2 transformada a ratings al aplicar Esquema de transformación ImplicitToExplicitFeedbackBasic.

Transformación de datos implícitos a datos explícitos

---

**Esquema de transformación ImplicitToExplicitFeedbackCelma:** Este método replica el algoritmo mencionado por Óscar Celma Herrada en su tesis doctoral, al igual que el esquema de transformación ImplicitToExplicitFeedbackBasic también utiliza la Distribución Acumulada Complementaria, pero además agrega el cálculo del Coeficiente de Variación, el cual se utiliza para determinar si existen valores homogéneos por cada fila de usuario. Si el Coeficiente de Variación es menor o igual a 0,5 toda esa fila de la matriz se le asigna un puntaje igual a 3, esto se refiere a que el usuario tiene la misma preferencia por ese grupo de ítems. Por ejemplo u3 de la matriz original y u3 de la matriz C2.

**Coeficiente de Variación de cada usuario (fila).**

*Tabla 16: Coeficiente de Variación de cada usuario.*

Matriz original	CV
u1	0,7
u2	1,65
u3	0,4
u4	0,68

*Tabla 18: Coeficiente de variación de cada usuario en la matriz C1*

Matriz C1	CV
u1	0,74
u2	1,61
u3	0,66
u4	0,57

Transformación de datos implícitos a datos explícitos

---

Tabla 20: Coeficiente de variación de cada usuario en el contexto C2

Matriz C2	CV
u1	0,7
u2	1,6
u3	0,3
u4	0,6

En las siguientes figuras (Figura 14, 15, 16) se muestra la salida al aplicar este esquema de transformación.

**Salida:**

U:1	5.0	3.0	1.0	1.0
U:2	5.0	1.0	1.0	1.0
U:3	3.0	3.0	3.0	3.0
U:4	3.0	1.0	2.0	5.0

Figura 14: Matriz original transformada a ratings al aplicar Esquema de transformación *ImplicitToExplicitFeedbackCelma*.

CONTEXTO: MomentOfDay:night				
U:1	5.0	3.0	1.0	2.0
U:2	5.0	1.0	1.0	1.0
U:3	5.0	2.0	1.0	3.0
U:4	3.0	2.0	1.0	5.0

Figura 15: Matriz C1 transformada a ratings al aplicar Esquema de transformación *ImplicitToExplicitFeedbackCelma*.

Transformación de datos implícitos a datos explícitos

---

CONTEXTO: MomentOfDay:day				
U:1	5.0	3.0	1.0	1.0
U:2	5.0	1.0	1.0	1.0
U:3	3.0	3.0	3.0	3.0
U:4	3.0	null	2.0	5.0

Figura 16: Matriz C2 transformada a ratings al aplicar Esquema de transformación *ImplicitToExplicitFeedbackCelma*.

Como resultado al aplicar este esquema de transformación, se acerca más a la representación deseada de las frecuencias, ya que asigna el mismo rating para aquellas frecuencias que tienen valores parecidos (u3 de matriz original y u3 de matriz C2).

Transformación de datos implícitos a datos explícitos

---

**Esquema de transformación ImplicitToExplicitFeedbackAdjustment:** Este método busca solucionar que cuando en una fila se presente un grupo de frecuencias iguales estas tengan un mismo rating y las frecuencias restantes de la fila otro rating. Utiliza como base el esquema de transformación ImplicitToExplicitFeedbackBasic y hace un ajuste a los ratings que se obtienen. Por ejemplo el u3 de la matriz original.

En las siguientes figuras (Figura 17, 18, 19) se muestra la salida al aplicar este esquema de transformación.

**Salida:**

U:1	3.0	1.0	1.0	1.0
U:2	1.0	1.0	1.0	1.0
U:3	2.0	1.0	1.0	2.0
U:4	1.0	1.0	1.0	3.0

Figura 17: Matriz original transformada a ratings al aplicar Esquema de transformación ImplicitToExplicitFeedbackAdjustment.

CONTEXTO: MomentOfDay:night				
U:1	3.0	2.0	1.0	1.0
U:2	1.0	1.0	1.0	1.0
U:3	3.0	1.0	1.0	2.0
U:4	1.0	1.0	1.0	3.0

Figura 18: Matriz C1 transformada a ratings al aplicar Esquema de transformación ImplicitToExplicitFeedbackAdjustment.

CONTEXTO: MomentOfDay:day				
U:1	3.0	1.0	1.0	1.0
U:2	1.0	1.0	1.0	1.0
U:3	1.0	1.0	1.0	4.0
U:4	1.0	null	1.0	3.0

Figura 19: Matriz C2 transformada a ratings al aplicar Esquema de transformación ImplicitToExplicitFeedbackAdjustment.

Transformación de datos implícitos a datos explícitos

---

La salida de este esquema de transformación muestra los casos donde existen grupos de frecuencias iguales dentro de una fila, como el caso del u3 de la matriz original donde existen dos grupos de valores igual y su representación en rating cumple la finalidad donde el grupo de frecuencias con valor 7 obtiene un puntaje de rating de 2 y el grupo de frecuencias con valor de 3 obtiene un rating de 1.

Transformación de datos implícitos a datos explícitos

**Esquema de transformación ImplicitToExplicitFeedbackFinal:** Este método utiliza las formas de cada esquema de transformación anterior, es una especie de híbrido, suma el uso de las frecuencias acumuladas complementaria, hace un ajuste y además utiliza el coeficiente variación.

En las siguientes figuras (Figura 20, 21, 22) se muestra la salida al aplicar este esquema de transformación.

**Salida:**

U:1	5.0	3.0	1.0	1.0
U:2	5.0	1.0	1.0	1.0
U:3	3.0	3.0	3.0	3.0
U:4	2.0	1.0	2.0	5.0

Figura 20: Matriz original transformada a ratings al aplicar Esquema de transformación ImplicitToExplicitFeedbackFinal.

CONTEXTO: MomentOfDay:night				
U:1	5.0	3.0	1.0	2.0
U:2	5.0	1.0	1.0	1.0
U:3	5.0	1.0	1.0	3.0
U:4	1.0	1.0	1.0	5.0

Figura 21: Matriz C1 transformada a ratings al aplicar Esquema de transformación ImplicitToExplicitFeedbackFinal

CONTEXTO: MomentOfDay:day				
U:1	5.0	3.0	1.0	1.0
U:2	5.0	1.0	1.0	1.0
U:3	3.0	3.0	3.0	3.0
U:4	2.0	null	2.0	5.0

Figura 22: Matriz C2 transformada a ratings al aplicar Esquema de transformación ImplicitToExplicitFeedbackFinal

Este esquema de transformación, a simple vista, es el que mejor representa en ratings los datos implícitos. Por ejemplo cuando un usuario tiene una alta preferencia por un ítem en particular, este ítem obtiene el mayor rating y no así aquellas preferencias

### Transformación de datos implícitos a datos explícitos

---

bajas. Ejemplo de esto es el u2 de la matriz original, donde la frecuencia de consumo por el i1 es mucho más alta que por i2, i3 e i4. Al separar la matriz por contextos se sigue evidenciando la preferencia por ese ítem. Por otro lado las preferencias que son homogéneas para un usuario tienen un mismo rating, esto se refiere a cuando a un usuario prefiere igualmente un ítem que otro.

El u4 de la matriz C2 de contexto "Día" no tiene preferencias para el ítem i2, por lo cual no se le puede asignar un rating. Principalmente, estos métodos son para trabajar en base a preferencias que existan, puesto que asignando un rating bajo a una preferencia inexistente sería alterar los datos obtenidos del usuario y por ende errar en una futura recomendación por sistema, también la finalidad de la agregación de datos es simplificar el registros de preferencias existentes del usuario y no agregar más.



## 4. Experimentos

En este capítulo se presentan experimentos que permiten validar los métodos de transformación de retroalimentación implícita-a-explicita implementados. Para esto, se utilizan algoritmos de recomendación disponibles en una librería de recomendación consciente del contexto desarrollada al interior de la Universidad del Bío-Bío[24].

Se utiliza un subconjunto de datos de visitas a lugares de interés (*venue check-in*) de usuarios del sistema de recomendación de lugares Foursquare<sup>16</sup>. Estos datos fueron recolectados por Sarwat et al. [25]. Estos datos están compuestos por un identificador de usuario, un identificador del lugar de interés, las coordenadas GPS del lugar de interés, y la marca de tiempo (timestamp) del momento en que el usuario visitó el lugar de interés. Adicionalmente, se cuenta con una “categorización” de las coordenadas GPS, la cual se calculó de la siguiente manera: se agruparon las diferentes coordenadas GPS en base a “cuadrantes” (división del espacio geográfico en 2, 4 u 8 segmentos cuadrados). Cada uno de estos segmentos es considerado el contexto de localización de la visita correspondiente. La Tabla 22 describe las principales características del conjunto de datos utilizado:

Tabla 22: Descripción de conjunto de datos utilizados para experimentos.

Usuarios	3.382
Ítems	3.634
Visitas	10.000
Promedio de Visitas de un usuario a un mismo lugar	2.0863
Fecha inicial	08 diciembre 2011
Fecha final	23 de abril de 2012

<sup>16</sup> Servicio de descubrimiento de lugares de interés Foursquare, disponible en: <http://es.foursquare.com>

## Experimentos

---

Para generar las recomendaciones, se utiliza un algoritmo de recomendación consciente del contexto de Pre-Filtrado, descrito por [18]. Este algoritmo consiste en seleccionar aquellas preferencias (en este caso, visitas) con el mismo contexto para el cual se desea recomendar. En este caso, se considerará como contexto el “cuadrante geográfico” asociado a la visita. Una vez seleccionadas las preferencias, entre todos los usuarios, asociados a dicho contexto, se utiliza un algoritmo de recomendación tradicional. En este caso, utilizamos el algoritmo de filtrado colaborativo “kNN” o vecinos más cercanos basado en usuario provisto por la librería de algoritmos de recomendación Mahout<sup>17</sup>. Este algoritmo se basa en seleccionar aquellos usuarios más similares al usuario objetivo (usuario al cual se le desea generar recomendaciones), y utilizar sus preferencias como base para generar las recomendaciones. Como caso base, aplicamos este algoritmo sobre los datos sin transformar. Posteriormente, aplicamos el algoritmo sobre los datos transformados mediante los diferentes esquemas de transformación propuestos.

Para medir la efectividad de este algoritmo, utilizamos algunas métricas comúnmente usadas en el área de los Sistemas de Recomendación. Las métricas más utilizadas en esta área son aquellas que buscan medir el error en las predicciones de ratings que generan estos algoritmos. Sin embargo, dado que para el caso base no es posible generar ratings (ya que sólo cuenta con información sobre si se visitó o no un lugar de interés), no usamos este tipo de métricas. El otro tipo de métricas que se ha utilizado con mayor frecuencia son las métricas de utilidad del ranking de recomendaciones. Estas métricas, provenientes del área de Recuperación de Información, en general miden cuantas de las recomendaciones generadas son efectivamente de interés para el usuario, de acuerdo al comportamiento de los usuarios, tomando en cuenta el orden en que se incorporan al listado. Para evaluar lo anterior, se utiliza un conjunto de datos de prueba que son “ocultados” al algoritmo de recomendación. Se utilizan en particular las métricas *normalized Discounted Cumulative Gain* (nDCG) y *Mean Average Precision* (MAP). Se obtuvo el valor para estas métricas sobre las listas de recomendación generadas, así como en los resultados obtenidos hasta la décima posición de la lista.

---

<sup>17</sup> Librería de algoritmos de Machine Learning Apache Mahout, disponible en: <http://mahout.apache.org>

## 4.1. Resultados de experimentos

En la Tabla 23 se presentan los resultados obtenidos al aplicar los diferentes algoritmos de transformación propuestos.

Tabla 23: Resultados experimentales.

Algoritmo	Análisis de Contexto	nDCG	nDCG 10	MAP	MAP 10
Caso base		0.3996	0.3668	0.2345	0.2239
ImplicitToExplicitFeedbackBasic	Individual	<b>0.4818</b>	<b>0.4594</b>	<b>0.3420</b>	<b>0.3349</b>
	Todos	0.4202	0.3947	0.2637	0.2556
ImplicitToExplicitFeedbackAdjustment	Individual	0.4150	0.3763	0.2552	0.2428
	Todos	0.4373	0.4070	0.2859	0.2762
ImplicitToExplicitFeedbackCelma	Individual	0.4741	0.4300	0.3349	0.3208
	Todos	0.4290	0.3966	0.2725	0.2621
ImplicitToExplicitFeedbackFinal	Individual	0.4338	0.4072	0.2823	0.2739
	Todos	0.4638	0.4430	0.3188	0.3123

En la Tabla 23 se puede observar que de acuerdo a las métricas nDCG y MAP, en todos los casos se obtienen mejores valores con las propuestas de transformación, respecto del caso base. Por otro lado, se puede observar que no es claro si resulta mejor realizar el análisis en cada contexto individual, o bien entre todos los contextos. Finalmente, se observa que los mejores resultados (resaltados en negrita) corresponden a la transformación básica, realizando el análisis en cada contexto. Aunque sería necesario realizar nuevos experimentos en otros conjuntos de datos para determinar si este es el mejor esquema de transformación en general.

## 5. Conclusiones del proyecto

Los sistemas de recomendación han evolucionado considerablemente en el último tiempo, sobre todo impulsado por grandes empresas, dueñas de sistemas web reconocidos por todo el mundo que buscan una mejor llegada de lo que ofrecen a sus usuarios y por ende incrementar sus beneficios económicos. También, esto ha significado el desarrollo en el ámbito educativo como Universidades o Instituciones educativas que desarrollan investigaciones sobre el tema.

En este proyecto se investigó en desarrollar e implementar una estrategia de transformación de datos implícitos, lo cual puede llegar a mejorar en cierta medida el desempeño de los Sistemas de Recomendación.

En primer término, se revisó el estado actual de los Sistemas de Recomendación, la Retroalimentación Implícita y el Contexto, además se estudió algoritmos que proponen el trabajo con datos Implícitos y con Contexto.

Una vez terminado el trabajo investigativo, se procedió a implementar algoritmos que realizaran transformación de datos implícitos a explícitos y que además en esta transformación considerara el contexto. De esta implementación surgieron cuatro esquemas de transformación, a los cuales se les efectuaron pruebas unitarias para verificar cuál de ellos se desempeña mejor, posteriormente a los esquemas se les aplicaron experimentos en algoritmos de recomendación disponibles por el profesor, quedando como preferido el esquema `ImplicitToExplicitFeedbackBasic`, el cual al aplicar métricas para determinar cuántas de las recomendaciones generadas son realmente efectivas, fue el que mejor se comportó.

Los métodos de transformación desarrollados en este trabajo pueden ser aplicados a cualquier sistema de recomendación que necesite brindar recomendaciones precisas.

Este proyecto puede servir como base para el desarrollo de trabajos futuros en esta misma universidad que tengan relación con los Sistemas de Recomendación consciente del contexto.

## Referencias

- [1] G. Jawaheer, M. Szomszor, and P. Kostkova, "Comparison of implicit and explicit feedback from an online music recommendation service," in Proceedings of the 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems - HetRec '10, 2010, pp. 47–51.
- [2] F. Ricci, L. Rokach, and B. Shapira, "Introduction to Recommender Systems Handbook," in Recommender Systems Handbook, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Springer, 2011, pp. 1–35.
- [3] G. Jawaheer, M. Szomszor, and P. Kostkova, "Characterisation of explicit feedback in an online music recommendation service," in Proceedings of the fourth ACM Conference on Recommender Systems, 2010, pp. 317–320.
- [4] D. W. Oard, D. W. Oard, and J. Kim, "Implicit Feedback for Recommender Systems Implicit Feedback for Recommender Systems," pp. 81–83, 1998.
- [5] CLIVE THOMPSON, "New York Times Magazine," 2008.
- [6] P. G. Campos, F. Díez, and I. Cantador, "Time-Aware Recommender Systems: A Comprehensive Survey and Analysis of Existing Evaluation Protocols," User Model. User-adapt. Interact., vol. 24, no. 1–2, pp. 67–119, 2014.
- [7] P. G. Campos, A. Bellogín, F. Díez, and J. E. Chavarriaga, "Simple Time-Biased KNN-based Recommendations," in Proceedings of the Workshop on Context-Aware Movie Recommendation, 2010, pp. 20–23.
- [8] D. M. Araque, "Especificación de un Sistema de Recomendación basado en Etiquetado Social," 2012.
- [9] R. Burke, "Hybrid recommender systems: Survey and experiments," User Model. User-adapt. Interact., vol. 12, no. 4, pp. 331–370, 2002.
- [10] Edward Rolando Núñez Valdéz, "Sistemas de Recomendación de contenidos para libros inteligentes," Univ. Oviedo, pp. 31–38, 2012.
- [11] M. Balabanović and Y. Shoham, "Fab: content-based, collaborative recommendation," Commun. ACM, vol. 40, no. 3, pp. 66–72, 1997.
- [12] L. Terveen, W. Hill, B. Amento, D. McDonald, and J. Creter, "Phoaks: a system for sharing recommendations," Commun. ACM, vol. 40, no. 3, pp. 59–62, 1997.
- [13] S. Manuel and G. Nieto, "Filtrado Colaborativo y Sistemas de Recomendación," 1994.
- [14] M. S. Font, "Sistemas de recomendación para webs de información sobre la salud," Univ. Técnica Cataluña, 2009.
- [15] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," Commun. ACM, vol. 35, no. 12, pp. 61–70, 1992.

Referencias

---

- [16] E. M. Housman and E. D. Kaskela, "State of the art in selective dissemination of information," *IEEE Trans. Eng. Writ. Speech*, vol. 13, pp. 78–83, 1970.
- [17] P. Resnick and H. R. Varian, "Recommender system," *Commun. ACM*, vol. ACM 40(3), pp. 56–58, 1997.
- [18] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin, "Incorporating Contextual Information in Recommender Systems Using a Multidimensional Approach," *ACM Trans. Inf. Syst.*, vol. 23, no. 1, pp. 103–145, Jan. 2005.
- [19] D. Kelly and J. Teevan, "Implicit feedback for inferring user preference: a bibliography," *SIGIR Forum*, vol. 37, no. 2, pp. 18–28, 2003.
- [20] G. Adomavicius and A. Tuzhilin, "Context-Aware Recommender System," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. 2011, pp. 217–253.
- [21] A. K. Dey, "Understanding and Using Context," *Pers. Ubiquitous Comput.*, vol. 5, no. 1, pp. 4–7, Feb. 2001.
- [22] L. Baltrunas and F. Ricci, "Context-based splitting of item ratings in collaborative filtering," in *Proceedings of the third ACM conference on Recommender systems*, 2009, pp. 245–248.
- [23] Ò. Celma, "Music Recommendation and Discovery in the Long Tail," *Universitat Pompeu Fabra*, 2009.
- [24] C. Palma and M. Valdés, "Desarrollo, validación e integración de una extensión a un framework para la evaluación de recomendaciones conscientes del contexto temporal.," *Universidad del Bío-Bío*, 2015.
- [25] M. Sarwat, J. J. Levandoski, A. Eldawy, and M. F. Mokbel, "LARS\*: An Efficient and Scalable Location-Aware Recommender System," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 6, pp. 1384–1399, Jun. 2014.
- [26] R. A. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999.