

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

UNIDAD DE TITULACIÓN

**DESARROLLO DE UN MODELO ADAPTATIVO-DINÁMICO DE
RECOMENDACIÓN PARA GRUPOS BASADO EN ASPECTOS
TEMPORALES**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL GRADO DE
MAGISTER EN SISTEMAS DE INFORMACIÓN**

CHRISTIAN ALEXANDER VALDIVIESO PINZÓN

christian.valdivieso@epn.edu.ec

DIRECTOR: PhD. LORENA KATHERINE RECALDE CERDA

lorena.recalde@epn.edu.ec

CODIRECTOR: PhD. EDISON FERNANDO LOZA AGUIRRE

edison.loza@epn.edu.ec

Quito, junio 2022

APROBACIÓN DEL DIRECTOR

Como director del trabajo de titulación DISEÑO DE UN MODELO ADAPTATIVO-DINÁMICO DE RECOMENDACIÓN PARA GRUPOS BASADO EN ASPECTOS TEMPORALES desarrollado por Christian Alexander Valdivieso Pinzón, estudiante de la Maestría en Sistemas de Información Mención Inteligencia de Negocios y Analítica de Datos Masivos, habiendo supervisado la realización de este trabajo y realizado las correcciones correspondientes, doy por aprobada la redacción final del documento escrito para que prosiga con los trámites correspondientes a la sustentación de la Defensa oral.



PhD. Lorena Katherine Recalde Cerda
DIRECTORA DE PROYECTO



PhD. Edison Fernando Loza Aguirre
CODIRECTOR DE PROYECTO

DECLARACIÓN DE AUTORÍA

Yo, Christian Alexander Valdivieso Pinzón, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento. La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



Christian Alexander Valdivieso Pinzón

AGRADECIMIENTOS

Agradezco profundamente a la PhD. Lorena Katherine Recalde Cerda y al PhD. Edison Fernando Loza Aguirre por todo cuanto supieron enseñarme, por la confianza, el tiempo, el apoyo y los valiosos aportes para este trabajo que, además de oportunos, fueron faros que iluminaron el camino correcto a seguir, faros de una luz tan clara que solo puede nacer de profesionales llenos de sabiduría y ética.

A handwritten signature in blue ink, appearing to read 'Christian Alexander Valdivieso Pinzón', enclosed within a large, loopy blue oval scribble.

Christian Alexander Valdivieso Pinzón

CONTENIDO

Resumen	1
Abstract	2
1 INTRODUCCIÓN	3
1.1 Preguntas de investigación	6
1.2 Objetivo general	6
1.2.1 Objetivos específicos	6
1.3 Alcance	7
1.4 Marco teórico	7
1.4.1 Sistemas de recomendación y aspectos temporales	7
1.4.2 Factorización Matricial	9
1.4.3 Estrategias de agregación	11
1.4.4 Modelos de recomendación basados en aspectos secuenciales	13
2 METODOLOGÍA	18
2.1 Contexto y definición conceptual	20
2.2 Descripción modelo	23
2.2.1 Sistemas de recomendación adaptativo - dinámico considerando aspectos temporales	23
2.2.2 Conjuntos de datos 3.1	24
2.2.3 Análisis y transformación de datos	25
2.2.4 Predicción de datos con factorización matricial	26
2.2.5 Generación de grupos y aplicación de estrategias de agregación	28
2.2.6 Modelos de predicción de calificaciones para grupos tomando en cuenta aspectos temporales	30
2.3 Desarrollo del modelo	30
2.3.1 Esquema de desarrollo del modelo	31
2.4 Pruebas del modelo	31
2.4.1 Pruebas en los conjuntos de datos	32
2.5 Resultados	33
2.6 Comunicación	34

3	RESULTADOS Y DISCUSIÓN	35
3.1	RMSE	35
3.2	Evaluación de estrategias de agregación	35
3.3	NDCG	36
3.4	Evaluación de los modelos de recomendación	38
3.5	Preguntas de investigación	41
3.5.1	Primera pregunta	41
3.5.2	Segunda pregunta	43
3.5.3	Tercera pregunta	44
3.5.4	Cuarta pregunta	46
3.5.5	Quinta pregunta	46
4	CONCLUSIONES	49
5	REFERENCIAS BIBLIOGRÁFICAS	51

ÍNDICE DE FIGURAS

Figura 1.1 Matriz de calificaciones Usuario-Artículo usada para aplicar filtrado colaborativo y encontrar las predicciones de las calificaciones faltantes C_{jk}	8
Figura 1.2 Gráfico de barras donde se indican picos de popularidad de nueve artículos.	14
Figura 1.3 Canasta secuencial para determinar la predicción de B_t para cada usuario. Fuente: Factorizing Personalized Markov Chains for Next-Basket Recommendation [24].	15
Figura 1.4 Matrices de transición para cada usuario. Fuente: Factorizing Personalized Markov Chains for Next-Basket Recommendation [24].	16
Figura 1.5 Cubo de transición. Fuente: Factorizing Personalized Markov Chains for Next-Basket Recommendation [24].	16
Figura 1.6 Modelo para obtener las predicciones de un artículo para un usuario. Fuente: E-commerce in Your Inbox: Product Recommendations at Scale [25].	17
Figura 2.1 Metodología Desing Science Research (DSRP).	19
Figura 2.2 Matrices usuario-artículo para las tres temporalidades.	27
Figura 2.3 Matrices usuario-artículo para las tres temporalidades.	28
Figura 2.4 Esquema de desarrollo del modelo	31
Figura 2.5 Archivos de grupos generados.	32
Figura 2.6 Archivos de calificaciones grupales MPS y LMS.	33
Figura 2.7 Registros de secuencias formadas por cada grupo.	34
Figura 3.1 Métrica RMSE por número de miembros de grupo al aplicar cuatro estrategias de agregación - MovieLens.	36
Figura 3.2 Métrica RMSE por número de miembros de grupo al aplicar cuatro estrategias de agregación - Amazon.	37
Figura 3.3 Mapa de calor RMSE con Most Pleasure - MovieLens.	39
Figura 3.4 Mapa de calor RMSE con Least Misery - MovieLens.	39
Figura 3.5 Mapa de calor RMSE con Most Pleasure - Amazon.	39
Figura 3.6 Mapa de calor RMSE con Least Misery - Amazon.	40

Figura 3.7 Mapa de calor de la métrica nDCG de los cuatro métodos de recomendación con la estrategia Most Pleasure (MPS) para 2, 4 y 8 miembros por grupo - MovieLens.	40
Figura 3.8 Mapa de calor de la métrica nDCG de los cuatro métodos de recomendación con la estrategia Least Misery (LMS) para 2, 4 y 8 miembros por grupo - MovieLens.	41
Figura 3.9 Mapa de calor de la métrica nDCG de los cuatro métodos de recomendación con la estrategia Most Pleasure (MPS) para 2, 4 y 8 miembros por grupo - Amazon.	41
Figura 3.10 Mapa de calor de la métrica nDCG de los cuatro métodos de recomendación con la estrategia Least Misery (LMS) para 2, 4 y 8 miembros por grupo - Amazon.	42
Figura 3.11 Barras horizontales de los valores de Métrica RMSE al aplicar MPS con tamaño de grupo 2, 4 y 8 en los conjuntos de datos MovieLens (superior) y Amazon (inferior), para todos los modelos de recomendación basados en aspectos secuenciales, para números de recomendaciones entre 1 a 50.	43
Figura 3.12 Gráficos de barras verticales utilizados para comparar las métricas RMSE con cada cantidad de miembros de grupos 2(verde), 5(azul) y 8(amarillo), para cada uno de los modelos de recomendación basado en aspectos secuenciales con 1, 5 y 10 recomendaciones.	44
Figura 3.13 Comparación de las estrategias de agregación MPS y LMS, para grupos de usuarios de 2, 5 y 8 miembros con 1, 5 y 10 recomendaciones aplicando el modelo de recomendación Factorization Personalized Markov Chains Recommender.	45
Figura 3.14 Tendencia de la métrica RMSE para los modelos de recomendación Popularity Recommender y Factorization Personalized Markov Chains Recommender con varias cantidades de recomendaciones, aplicando la estrategia de agregación MPS con 2, 5 y 8 miembros por grupo.	46
Figura 3.15 Desviación estándar con las estrategias de agregación Most Pleasure (MPS) y Least Misery (LMS) para T1, T2 y T3 de 2 a 8 miembros de grupos en los conjuntos de datos de MovieLens (superior) y Amazon (inferior).	48

ÍNDICE DE TABLAS

Tabla 1.1	Estrategia de agregación Utilitaria Aditiva.	11
Tabla 1.2	Estrategia de agregación Utilitaria Multiplicativa.	12
Tabla 1.3	Cálculo de <i>Least Misery Strategy</i> (LMS).	12
Tabla 1.4	Cálculo de la estrategia de agregación Most Pleasure Strategy (MPS).	13
Tabla 2.1	Descripción de las columnas de los conjuntos de datos.	26
Tabla 2.2	Número de registros, artículos y usuarios, antes de aplicar factorización matricial para Movielens y Amazon.	26
Tabla 2.3	Matriz de calificaciones [17].	29
Tabla 2.4	Detalle de los conjuntos de datos después de la formación de grupos.	31
Tabla 2.5	Número de registros, usuarios y artículos de cada conjunto de datos T1, T2 y T3.	32

RESUMEN

Los sistemas de recomendación (SR) implementados en plataformas Web han alcanzado cierta madurez y hoy en día son capaces de encontrar los productos o servicios que mejor se adaptan a las potenciales necesidades e intereses de los usuarios o grupos de usuarios. En realidad, los sistemas de recomendación de grupos (SRG) permiten recomendar artículos a un grupo de usuarios que tienen intereses comunes. Los aspectos temporales de SRG se basan en la idea de generar una secuencia de calificaciones donde cada calificación pertenece a un espacio o período temporal, de modo que se captura el interés del grupo a lo largo del tiempo. Por lo tanto, un SRG sensible al tiempo busca predecir la calificación de elementos grupales para el próximo período de tiempo en el futuro. Como en otros contextos, esto también permite generar listas de recomendaciones para grupos. El enfoque para el diseño de un SRG que considere aspectos temporales que se propone en esta investigación. Esto se basa en la evaluación de algunos métodos. De hecho, utilizamos la factorización matricial para las predicciones de calificaciones individuales para luego detectar grupos y extraer los modelos grupales con estrategias de agregación. Finalmente, se aplicó un modelo SR con reconocimiento de secuencias de última generación en un escenario SRG. Se seleccionaron conjuntos de datos de MovieLens y Amazon que tenían el campo de timestamp para poder evaluar los modelos. Se informaron métricas como RMSE y NDCG.

PALABRAS CLAVE: Sistema de recomendación grupal, Estrategias de agregación, Aspectos-temporales, Aspectos-secuenciales, FPMC, MPS

ABSTRACT

Recommender systems (RS) implemented on Web platforms have reached a certain maturity and today, they are capable of finding the products or services that best suit the needs and interests of potential users or groups of users. Actually, group recommender systems (GRS) allow items to be recommended to a group of users who have common interests. The temporal aspects of GRS are based on the idea of generating a sequence of ratings where each rating belongs to a temporary space or period, so group interest drift is captured over time. Therefore, a temporal-sensitive GRS seeks to predict the group-item rating for the next time period in the future. As in other contexts, this also allows the generation of recommendation lists for groups. The approach for the design of a GRS that considers temporal aspects that is proposed in this research. This is based on the evaluation of some methods. Indeed, we used matrix factorization for individual rating predictions to later detect groups and extract the group models with strategies of aggregation. Finally, state-of-the-art sequence-aware RS model were applied in a GRS scenario. MovieLens and Amazon datasets having the timestamp field were selected to be able to evaluate the models. Metrics such as the RMSE and NDCG were reported.

Keywords: Group Recommender System, Aggregation Strategies, Time-Aspects, Sequential-Aspects, FPMC, MPS

1 INTRODUCCIÓN

En la actualidad, donde el comercio electrónico ha adquirido un gran auge, y en el que cada vez se encuentran más productos y servicios publicados en diversas plataformas, los sistemas de recomendación (SR) son muy importantes debido a que facilitan la búsqueda y permiten encontrar los artículos que más se adapten a las necesidades e intereses de los posibles compradores. Sin embargo, los gustos de los posibles clientes se modifican con el tiempo en base a diversas variables, por ejemplo, círculos de amigos cambiantes, necesidades satisfechas previamente, prioridad en las preferencias, entre otros [1]. En consecuencia, los SR deben ser suficientemente capaces de reconocer estos cambios de tendencias y necesidades para poder ofrecer de manera acertada opciones que se adapten a las nuevas exigencias.

Dado que las recomendaciones se presentan normalmente en forma de lista o grupo, la experiencia del usuario depende en gran medida de la calidad general de tales recomendaciones.

Los sistemas de recomendación grupal (SRG) permiten recomendar ítems a un grupo de usuarios cuyos miembros tienen un interés, preferencia o comportamiento común. Es necesario utilizar métodos de agregación para combinar la información sobre las distintas preferencias individuales de los miembros del grupo, para que se pueda satisfacer de la forma más justa a cada uno de ellos [2].

Muy a menudo, en el diseño del modelo de un SR, el perfil de usuario se considera como una instantánea estática sin tener en cuenta su *dimensión temporal*. En realidad, se ha demostrado que la importancia de analizar el aspecto temporal para el modelado de usuarios afecta los resultados de la recomendación final [3], [4] y [5]. La dinámica temporal podría permitir comprender mejor los intereses del usuario con respecto a las características de los artículos a ser recomendados y luego proporcionar una diversificación más precisa y consciente de la intención. Sin embargo, la consideración de los aspectos temporales no

es la estrategia más comúnmente aplicada en la actualidad, por lo que, en ocasiones las recomendaciones suministradas por el sistema pueden no ser las más acordes a las necesidades actuales o futuras del usuario.

Modelar la dinámica temporal debe ser clave al diseñar un SR o modelos generales de preferencias del cliente o grupo de clientes. Sin embargo, esto plantea desafíos únicos. Dentro del ecosistema que cruza múltiples productos y clientes, muchas características diferentes están cambiando simultáneamente, mientras que muchas de ellas se influyen entre sí y, a menudo, esos cambios son delicados y están asociados con algunas instancias de datos. Los cambios en las características de usuarios y productos se dan con el paso del tiempo, estos cambios se los modela a través de instancias de datos. Los enfoques clásicos de ventana de tiempo o decadencia de instancias podrían no funcionar, ya que pierden demasiada señal al descartar instancias de datos [5]. Se requiere un enfoque más sensible, que pueda hacer mejores distinciones entre efectos transitorios y patrones a largo plazo.

La importancia de tener en cuenta la dinámica temporal aquí en los SR se ha abordado recientemente en diferentes trabajos para diversos dominios de recomendación. En [4] se propone un método para modelar las sesiones de los usuarios en el dominio de la música. Consideran como sesión a cada conjunto de calificaciones consecutivas sin un intervalo de tiempo prolongado entre ellas. Teniendo en cuenta que existen diversos fenómenos psicológicos que llevan a agrupar un conjunto de valoraciones en una única sesión, el método capta estos efectos mediante sesgos de sesión de usuario.

Los autores en [5] proponen un algoritmo de filtrado colaborativo capaz de modelar la conducción temporal de las preferencias del usuario, y los resultados en un conjunto de datos de Netflix indicaron la importancia de revelar los efectos temporales para producir recomendaciones más precisas. Un método más reciente propuesto para aprovechar la información temporal en el comportamiento del usuario se llama Incrustación de Markov basada en el tiempo [3], que se utiliza para encontrar la mejor recomendación de la siguiente canción a través de la Incrustación de Markov latente.

La necesidad de modelar los cambios de tiempo a nivel de cada individuo reduce significativamente la cantidad de datos disponibles para detectar tales cambios [5]. Por tanto, se debe recurrir a técnicas más precisas que las que se consideran como suficientes para modelar los cambios globales, es decir, se requiere filtrar patrones a largo plazo y desechar el ruido

transitorio. Esto implica una metodología más sensible para abordar las preferencias de los grupos de clientes. No sería adecuado concentrarse en identificar y modelar solo lo que es relevante para el presente o el futuro cercano. En cambio, es necesario un modelo preciso de cada punto en el pasado, lo que permitirá distinguir entre la señal persistente que debe capturarse y el ruido que debe aislarse de las partes del modelo a más largo plazo.

Generalmente, es mucho más difícil usar la información temporal en las calificaciones para hacer recomendaciones; los modelos temporales existentes [5] utilizan la información temporal en las calificaciones de una manera limitada y cuidadosamente calibrada. Por otro lado, la literatura sobre retroalimentación implícita [6] y modelos discretos es bastante rica porque se ha explorado ampliamente en el contexto de registros y secuencias de clics web. El último problema está estrechamente relacionado con el de la previsión de datos secuenciales con atributos categóricos. En este caso, los métodos de minería de datos discretos, como los modelos de Markov y la minería de patrones secuenciales, son muy útiles [7]. Las fuentes de datos más directas para construir un SR son las secuencias pasadas de interacciones de los usuarios, que son abundantes y están bien estructuradas. Actualmente, los modelos basados en reglas todavía se utilizan ampliamente para hacer recomendaciones dinámicas [8].

En la actualidad se han realizado dos investigaciones referentes al factor temporal dentro de los SRG, pero no se ha hecho un estudio a profundidad de cómo estos afectan la eficiencia de los sistemas, y de momento no existe ningún modelo que tome en cuenta los aspectos temporales, por lo que es necesario realizar una investigación en esta área con un análisis más pertinente [9], [10].

Muchos trabajos han demostrado que los modelos basados en reglas, por ejemplo, recomendar los elementos con el mayor número de interacciones, es de hecho una línea de base muy sólida [11], [12]. Por otro lado, en los últimos años se ha visto el poder de los modelos de aprendizaje profundo basados en secuencias en la visión por computadora [13], en el procesamiento del lenguaje natural [14] y en los datos estructurados en gráficos. Además, trabajos recientes también han aplicado modelos de secuencia para sistemas de recomendación dinámica [15], [16].

1.1 PREGUNTAS DE INVESTIGACIÓN

En este proyecto se desea responder las siguientes preguntas:

1. ¿Hasta qué punto se puede emplear eficazmente el estado del arte respecto a los algoritmos basados en secuencia y temporalidad para modelar un SRG basado en aspectos temporales?
2. ¿Hasta qué punto la precisión del modelo basado en aspectos secuenciales depende del tamaño de los grupos?
3. ¿Cómo afectan las estrategias de agregación aplicadas para el modelado de grupos al rendimiento de los modelos de recomendación grupal basados en aspectos secuenciales?
4. Según el tiempo, ¿cómo afecta la calidad de las recomendaciones o preferencias sobre el consumo de los mismos artículos en un SRG?
5. ¿Cómo cambian los intereses de los grupos con el tiempo?

1.2 OBJETIVO GENERAL

Desarrollar un modelo adaptativo-dinámico para sistemas de recomendación para grupos considerando aspectos temporales.

1.2.1 Objetivos específicos

- Definir el problema y sus antecedentes en base a la revisión de literatura.
- Describir un modelo adaptativo dinámico en base a la formulación adecuada del problema de optimización de los sistemas de recomendación existentes.
- Describir los algoritmos necesarios para la construcción del modelo.
- Comparar los resultados obtenidos entre los diversos enfoques adoptados.

1.3 ALCANCE

El diseño del modelo es el resultado de la combinación de conocimiento sobre técnicas, modelos o algoritmos que han sido previamente usados en otras investigaciones sobre detección de grupos, aplicación de estrategias de agregación enfocadas en obtener calificaciones grupales [17] y modelos de SR basados en aspectos secuenciales [18]. Todos estos enfoques han sido adaptados para obtener un modelo de recomendación grupal que considere los aspectos temporales.

En consecuencia, no se propone un nuevo algoritmo o estrategia matemática destinada a la manipulación de cambios de "comportamiento" o "intereses" de los miembros de un grupo en el tiempo. Desarrollar este tipo de investigación llegaría a ser de mayor alcance, incluso, pudiendo ser desarrollado para una tesis doctoral, por lo que no se aplicará en un ambiente productivo. Las pruebas se realizarán con datos de compras de productos electrónicos de Amazon y consumo de películas de MovieLens, estos datos de calificaciones de usuarios reales son públicos en la Web.

1.4 MARCO TEÓRICO

1.4.1 Sistemas de recomendación y aspectos temporales

Los sistemas de información actuales tienen relojes que permiten almacenar registros y asignarles una fecha y hora de creación o modificación, pues es de vital importancia para las empresas saber cuándo un cliente realizó una compra y con ayuda de sus sistemas de contabilidad o ERP generar una factura con la fecha respectiva de esta operación.

Un sistema de recomendación basado en contexto tiene en cuenta varios tipos de información de contextualización para hacer una recomendación. Estas contextualizaciones pueden ser el tiempo, ubicación o motivo [19]. Estos sistemas son muy valiosos, ya que se pueden enfocar en una sola idea de contextualización.

Los sistemas de recomendación *sensibles a temporalidad* que en inglés se demoninan *Time-Aware Recommender System* (TARS) son un derivado de los sistemas de recomendación basados en aspectos contextuales dado el tiempo como un factor de contexto. El tiempo fecha-hora en que un usuario entrega una calificación a una película o artículo que

acaba de consumir aporta información valiosa que puede ser generada desde cualquier sistema que integre un reloj al almacenar información en una base de datos. Las recomendaciones de un artículo, en función de su comportamiento o seguimiento de sus compras previas, pueden ser determinadas con ayuda de la dimensión tiempo.

Investigadores [20], [21] presentan siete categorías de cómo el tiempo puede ser utilizado en SR:

1. Restricción: El tiempo es usado para conciliar con el momento en el cual se va a dar la recomendación. En este tipo, los SR hacen una comparación de los tiempos disponibles del usuario con el tiempo en el que se va a dar uso al artículo. Ejemplo: Si un usuario va a ir a cenar, el SR solo deberá dar recomendaciones de restaurantes que abren en la noche.

2. Micro-Perfil: el SR almacena un tipo de perfil por usuario, que es un comportamiento de compra o interés en ciertos artículos por parte del comprador para un periodo de tiempo en particular. Por ejemplo, si un usuario tiene dos perfiles, uno para fines de semana y otro para días entre semana; cuando el usuario esté haciendo compras el sábado en la mañana, el SR recomendará los artículos que son consumidos en el periodo de tiempo en cuestión.

3. BIAS: El tiempo es la tercera dimensión de la matriz Usuario-Artículo. Estos datos se usan como la matriz para el filtrado colaborativo para comparar usuarios, encontrar los k-vecinos y predecir la calificación del usuario para un artículo que no ha sido calificado dentro de la matriz, como se muestra en la Figura 1.1.

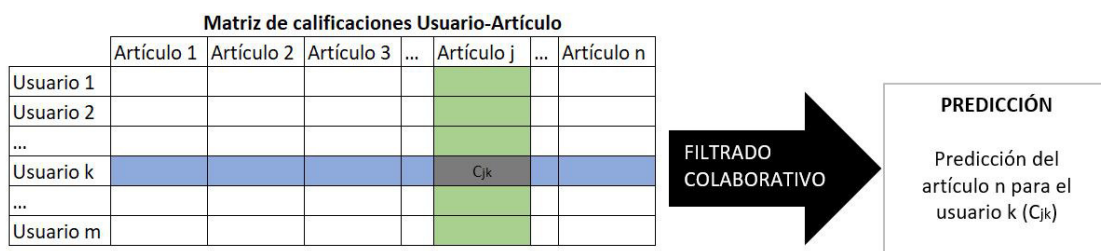


Figura 1.1: Matriz de calificaciones Usuario-Artículo usada para aplicar filtrado colaborativo y encontrar las predicciones de las calificaciones faltantes C_{jk} .

4. Decaimiento: el tiempo es usado como un factor de decaimiento, en el cual las interacciones más antiguas son menos importantes que las nuevas. Ejemplo: las películas de estreno son consideradas para la producción de la recomendación.

5. Calificación de tiempo: el tiempo es considerado como una retroalimentación implícita por los SR para inferir en las preferencias del usuario. Por ejemplo: mientras más tiempo,

un usuario permanece en un local se siente más cómodo o le gusta más en comparación a otros.

6. Novedad: Se define una fecha límite a la cual el SR solo toma en cuenta artículos posteriores a esa fecha para recomendar. Por ejemplo: noticias de la última semana.

7. Secuencia: los sistemas de recomendación determinan los artículos que usualmente son consumidos uno después de otro. Así que, si un artículo es consumido primero, probablemente el segundo también sea consumido. Ejemplo: Si escucho la primera canción del álbum de un artista, probablemente escuche varias canciones reconocidas del mismo artista a continuación.

En este artículo, con el objetivo de modelar aspectos temporales en un SRG, consideramos 2) *Micro-perfiles* dado que el grupo G tendrá diferentes perfiles. G tiene un modelo de preferencias en T_x y otro modelo en T_y . Además, 5) *Calificación de tiempo* establece diferentes rangos de tiempo o periodos definidos por las marcas de tiempo de las interacciones usuario-elemento. De hecho, en cada periodo de tiempo se extrae el modelo de preferencias del grupo.

1.4.2 Factorización Matricial

Los modelos de factorización matricial (FM) han sido relevantes por su alta calidad en las predicciones de calificaciones para los SR en general [22]. Estos modelos de FM tienen un problema, el cual una vez calculada la matriz de factorización, el modelo se vuelve estático, lo que significa que no es aplicable al futuro para la llegada de nuevos usuarios y nuevos artículos.

La matriz de calificaciones asignadas por usuarios a los artículos, tiene el problema de ser dispersa: existes muchas calificaciones faltantes en la matriz. R es la matriz a ser completada, las filas de la matriz corresponden a los usuarios U y las columnas a los artículos A. Entonces la matriz tiene dimensiones $|U| \times |A|$. C_{ua} es la calificación de una celda de la matriz R que contiene el valor de calificación de un usuario u para un artículo a. Entonces, las calificaciones generadas por el proceso de predicción son las que completan la matriz R y se va a denominar \hat{R} [22].

La factorización matricial es un proceso de aproximación de la matriz de calificaciones R, que viene dada por $\hat{R}:|U| \times |A|$. \hat{R} es el producto de las matrices de características $W:|U| \times k$

y $H:|A| \times k$, donde la u -ésima fila $[w_u]$ de W contiene la característica k que describe al u -ésimo usuario y la a -ésima fila $[h_a]$ de H contiene la característica k que describe al a -ésimo artículo.

$$\hat{R} = W.H^t \quad (1.1)$$

y su equivalencia es:

$$c_{u,a} = \langle w_u, h_a \rangle = \sum_{f=1}^k w_{u,f} \cdot h_{a,f} \quad (1.2)$$

1.4.2.1 Factorización matricial Kernel

Tanto la factorización matricial como la factorización matricial kernel tienen dos matrices características para usuarios y artículos. La interacción entre el vector característica w_u de un usuario y el vector característica h_a de un artículo es kernelizado y está dado por la Ecuación 1.3. Los kernels proveen un método extra para generar un nuevo método de factorización matricial.

$$c_{u,a} = x + c \cdot K(w_u, h_a) \quad (1.3)$$

Los términos x y c son introducidos para permitir re-escalar las predicciones. Para el kernel K se tiene:

$$K : \mathbb{R}^k \times \mathbb{R}^k \rightarrow \mathbb{R} \quad (1.4)$$

se pueden usar los kernels conocidos que se presentan a continuación:

$$K_l(w_u, h_a) = \langle w_u, h_a \rangle \rightarrow \text{Lineal} \quad (1.5)$$

$$K_p(w_u, h_a) = (1 + \langle w_u, h_a \rangle)^d \rightarrow \text{Polinomial} \quad (1.6)$$

$$K_s(w_u, h_a) = \phi_s(b_{u,i+\langle w_u, h_u \rangle}) \quad \text{donde} \quad \phi_s(x) = \frac{1}{1 + e^{-x}} \quad \rightarrow \text{Logística} \quad (1.7)$$

1.4.3 Estrategias de agregación

Un grupo está conformado por dos o más miembros que comparten características u objetivos en común. Para los sistemas de recomendación grupal (SRG), un modelo grupal parte de la síntesis de un modelo de usuario, que se constituye de la combinación de las preferencias de los usuarios en un grupo. Un modelo grupal en SRG consiente en determinar una calificación conjunta de un artículo que englobe el interés de los miembros del grupo, por lo que está estrechamente relacionado con una idea de selección colectiva.

La agregación de las preferencias individuales de los miembros del grupo consiste en determinar los artículos más afines a un grupo determinado, esto se calcula en función de las calificaciones individuales asignadas a los artículos por cada miembro que pertenece al grupo. Al final se obtiene una lista de las ponderaciones o calificaciones grupales que corresponden a cada artículo. A continuación se va a describir cuatro estrategias de agregación de preferencias comúnmente usadas en SRG [17].

1.4.3.1 Estrategia Utilitaria Aditiva

La estrategia utilitaria aditiva, que en inglés se denomina *Additive Utilitarian Strategy*, realiza una suma de las calificaciones individuales de cada artículo y genera una lista de las calificaciones grupales. Mientras más grande es el valor de la suma, el artículo aparecerá antes en la lista de recomendaciones. La lista producida por este método es exactamente la misma que se produce cuando se realiza un promedio de las calificaciones individuales [17].

Un ejemplo de cálculo con la estrategia utilitaria aditiva se muestra en la Tabla 1.1.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
u_1	5	4	5	3	4	3	5	2	4	5
u_2	4	2	4	2	5	4	2	5	4	4
u_3	3	5	3	5	4	3	4	3	5	3
Grupo	12	11	12	10	13	10	11	10	13	12

Tabla 1.1: Estrategia de agregación Utilitaria Aditiva.

1.4.3.2 Estrategia Utilitaria Multiplicativa

La estrategia utilitaria multiplicativa, que en inglés se denomina *Multiplicative Utilitarian Strategy*, realiza la multiplicación de las calificaciones otorgadas por los usuarios para cada uno de los artículos y se produce una lista con los resultados de los productos. Los artículos que aparecen primeros en la lista de recomendaciones son los que han obtenido un mayor valor del producto [17].

Un ejemplo del cálculo realizado se indica en la Tabla 1.2

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
u_1	5	4	5	3	4	3	5	2	4	5
u_2	4	2	4	2	5	4	2	5	4	4
u_3	3	5	3	5	4	3	4	3	5	3
Grupo	60	40	60	30	80	36	40	30	80	60

Tabla 1.2: Estrategia de agregación Utilitaria Multiplicativa.

1.4.3.3 Estrategia de Menor Misera

La estrategia de menor misera, que en inglés se denomina *Least Misery Strategy* (LMS), es un proceso donde se selecciona la calificación más baja de entre los miembros del grupo a un artículo y se la asigna como una calificación grupal. Esta estrategia es usualmente empleada para modelos de grupos pequeños, lo que asegura la satisfacción de todo los usuarios. Un inconveniente de esta estrategia es que si a un miembro del grupo no le gusta un artículo y a todos los demás sí les gusta, se asigna esta calificación baja al grupo, en consecuencia no se recomienda el artículo al grupo [17].

Un ejemplo de cálculo con la estrategia de menor miseria se muestra en la Tabla 1.3.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
u_1	5	4	5	3	4	3	5	2	4	5
u_2	4	2	4	2	5	4	2	5	4	4
u_3	3	5	3	5	4	3	4	3	5	3
Grupo	3	2	3	2	4	3	2	2	4	3

Tabla 1.3: Cálculo de *Least Misery Strategy* (LMS).

1.4.3.4 Estrategia de Mayor Placer

La estrategia de mayor placer, que en inglés se denomina *Most Pleasure Strategy* (MPS), asigna a un grupo la calificación más alta entre todas las calificaciones que los miembros del grupo han asignado a un artículo [17]. Los artículos con las calificaciones más altas encabezarán la lista de recomendaciones.

Un ejemplo de cálculo con la estrategia de mayor placer se muestra en la Tabla 1.4.

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}
u_1	5	4	5	3	4	3	5	2	4	5
u_2	4	2	4	2	5	4	2	5	4	4
u_3	3	5	3	5	4	3	4	3	5	3
Grupo	5	5	5	5	5	4	5	5	5	5

Tabla 1.4: Cálculo de la estrategia de agregación Most Pleasure Strategy (MPS).

1.4.4 Modelos de recomendación basados en aspectos secuenciales

Los registros secuenciales de eventos son usados para determinar comportamientos a largo plazo, por lo que se puede determinar los intereses cambiantes de los usuarios individuales sobre el tiempo [18].

Los SR que están basados en aspectos secuenciales tienen varios retos. Uno de los principales es el ordenamiento de las calificaciones a través del campo *timestamp* para listar las acciones pasadas de un usuario. La salida de estos sistemas es similar a los SR tradicionales, donde se indica la calificación asignada a cada artículo.

Los procesos de computación para los SR basados en aspectos secuenciales se utilizan para encontrar por ejemplo patrones de secuencias, donde el orden de las acciones son relevantes. Existen varios algoritmos que realizan estos procesos de computación. A continuación se describen cuatro algoritmos que se van a utilizar en esta investigación.

1.4.4.1 Algoritmo por Popularidad

El algoritmo de recomendación por Popularidad denominado en inglés *Popularity Recommender*, considera como relevantes los artículos más populares o que más veces han sido

consumidos de todo el conjunto de calificaciones asignadas por un grupo de usuarios. Las tendencias usadas para estos SR pueden ser a largo plazo (artículos que no se vuelven obsoletos con el paso del tiempo, que pueden ser consumidos una y otra vez a pesar de que fecha de lanzamiento al mercado), estacionales o que reflejan picos de popularidad. Simplemente, se recomienda los artículos ordenados por el número de veces que aparecen en el conjunto de entrenamiento, como se muestra en la Figura 1.2 en la que se recomendará el artículo i3 que aparece en el 38 % de registros [23].

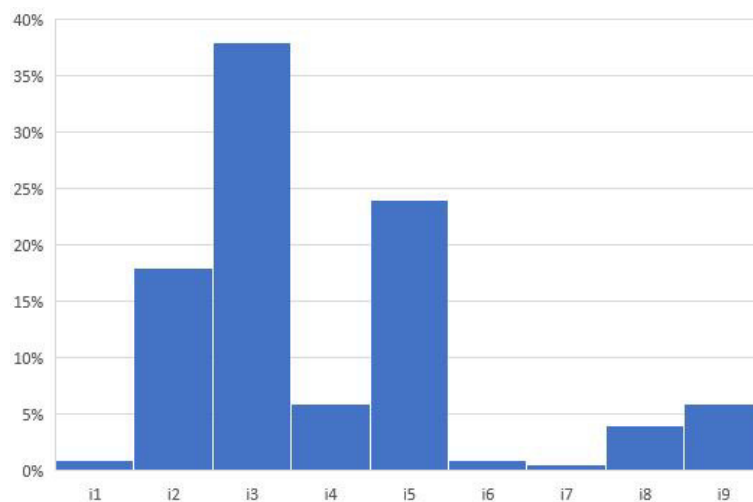


Figura 1.2: Gráfico de barras donde se indican picos de popularidad de nueve artículos.

1.4.4.2 Cadenas de Markov

Los SR basados en una cadena de Markov que en inglés se denomina *Markov Chain* (MC) utilizan datos secuenciales para predecir la siguiente acción de un usuario basado en sus últimas acciones. Por lo tanto, una matriz de transición es la que da la probabilidad de comprar un artículo basado en las últimas compras realizadas por el usuario. La matriz de transición de los modelos de la cadena de Markov es la misma para todos los usuarios.

Los modelos de Markov consideran los datos secuenciales como un proceso estocástico sobre variables discretas aleatorias (o estados). Los procesos se limitan a una historia finita. Por ejemplo, la cadena de Markov de primer orden depende de únicamente del estado anterior. Por otra parte, la cadena de Markov de orden más alto al usar dependencias temporales realiza procesos computacionales más complejos para modelar las relaciones entre sus estados previos.

1.4.4.3 Cadenas de Markov Mixtas

Cuando hablamos de cadenas de Markov Mixtas, estas utilizan algunas técnicas adicionales que ayudan a predecir una calificación. Por ejemplo, aplicar clustering que permite encontrar la similitud entre secuencias, donde la primera secuencia se parece a otra secuencia de todo el conjunto de datos.

1.4.4.4 Cadenas de Markov Factorizadas Personalizadas

Es denominada híbrida, ya que combina métodos flexibles de aprendizaje secuencial con otras técnicas más robustas para el manejo de datos desconocidos, como técnicas de llenado de matrices basadas en factorización matricial.

Cadenas de Markov Factorizadas Personalizadas o *Factorized Personalized Markov Chain* (FPMC) es un método que combina factorización matricial y cadenas de Markov para resolver problemas de recomendación del siguiente artículo dadas por las n últimas interacciones de un usuario [24].

Un proceso secuencial que se va a modelar cuenta con usuarios y cinco artículos (a, b, c, d y e). La tarea final es recomendar artículos en el tiempo t dadas las acciones pasadas B_{t-3}, B_{t-2} y B_{t-1} . Este comportamiento se visualiza en la Figura 1.3 [24].

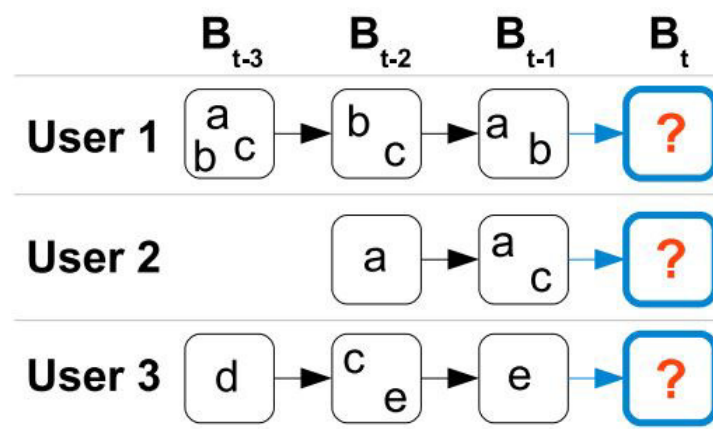


Figura 1.3: Canasta secuencial para determinar la predicción de B_t para cada usuario.

Fuente: Factorizing Personalized Markov Chains for Next-Basket Recommendation [24].

Para cada uno de los usuarios se tiene una matriz de transición individual. Cada matriz contiene las probabilidades de pasar de un artículo a otro. Ejemplos de las matrices de transición para 4 usuarios se muestran en la Figura 1.4 [24]. Los valores perdidos ?, indican

que no hay datos para determinar la probabilidad de transición.

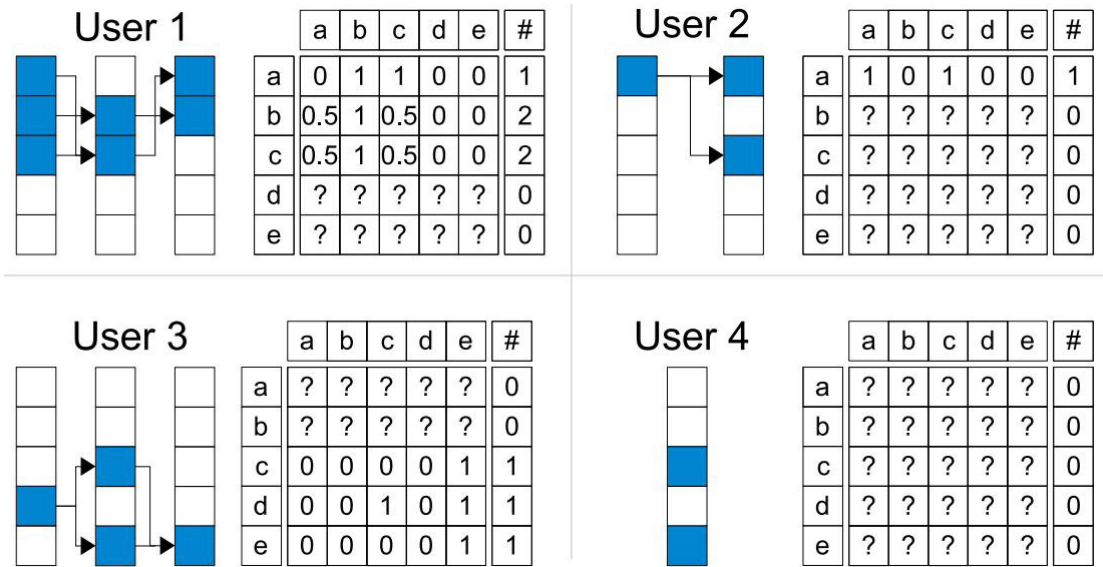


Figura 1.4: Matrices de transición para cada usuario.

Fuente: Factorizing Personalized Markov Chains for Next-Basket Recommendation [24].

El apilamiento de todas las matrices de transición de los usuarios individuales conducen a la formación de un cubo de transición. Este cubo se visualiza en la Figura 1.5.

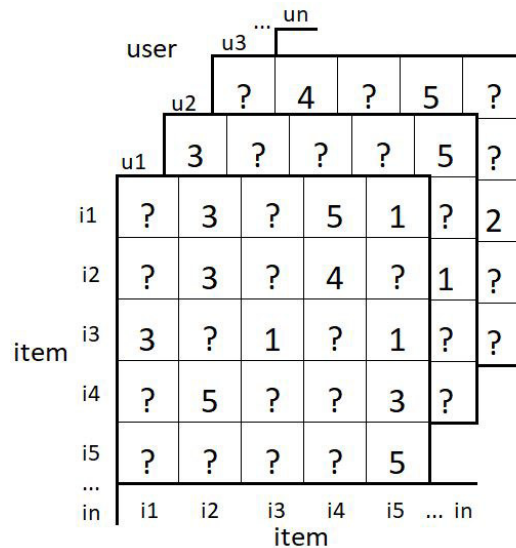


Figura 1.5: Cubo de transición.

Fuente: Factorizing Personalized Markov Chains for Next-Basket Recommendation [24].

FPMC modela las interacciones entre pares de los tres modos, modo U pertenece a los usuarios, I a los artículos del eje x y J los artículos del eje y, por lo que toma en cuenta las interacciones U-I, U-J e I-J.

Al realizar este proceso de interacción por pares, se generan las siguientes matrices de

factorización:

1. Matriz que modela las características del usuario u y el último artículo i .
2. Matriz que modela el siguiente artículo j para el último artículo i .
3. Y finalmente la matriz que modela las características del usuario u para el siguiente artículo j [24].

1.4.4.5 Prod2Vec

El modelo Prod2Vec involucra un proceso de aprendizaje de la secuencia de artículos consumidos donde el método sigue el procesamiento aplicado en NLP. Es decir, la secuencia de interacciones se ve como una “oración” y los elementos de la secuencia se ven como “palabras” [25]. El modelo de predicción usuario-artículo típicamente consideran el historial de compras, los intereses de los usuarios, comportamiento de los usuarios web, contactos sociales, entre otros, para entregar una recomendación [25].

Las recomendaciones se generan devolviendo los k -vecinos más cercanos de los últimos artículos consumidos por un usuario, cuya relevancia se pondera usando un simple decaimiento exponencial (la última calificación de un usuario es la más relevante, y la primera calificación es la menos relevante). La representación de este modelo se visualiza en la Figura 1.6

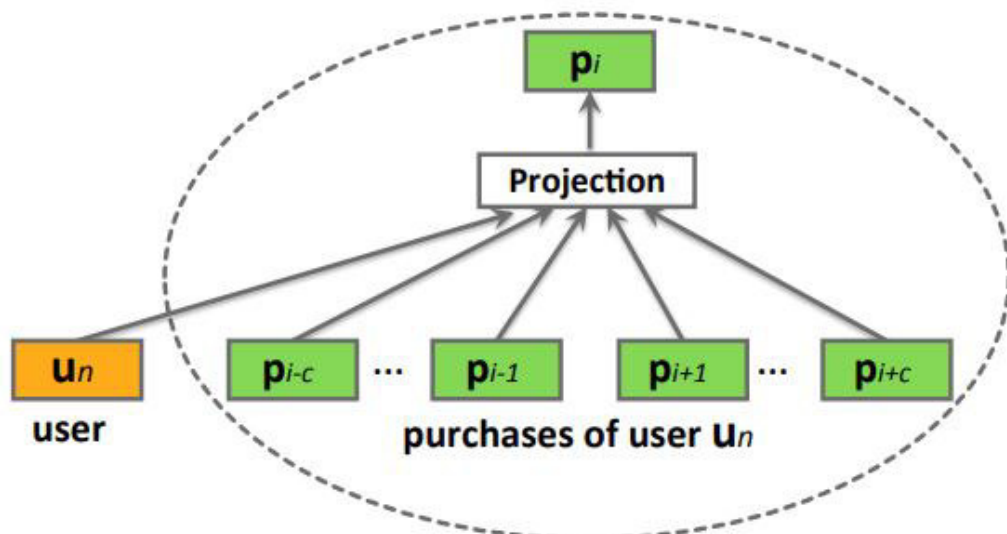


Figura 1.6: Modelo para obtener las predicciones de un artículo para un usuario.
Fuente: E-commerce in Your Inbox: Product Recommendations at Scale [25].

2 METODOLOGÍA

La metodología de investigación que se empleará en este estudio es la conocida como “*Design Science Research*” (Investigaciones basadas en la ciencia del diseño o DSR). Esta es una metodología aplicable principalmente en las investigaciones de tecnología de la información (TI), que se sustenta en los resultados obtenidos en la implementación de pautas específicas que rigen las iteraciones establecidas para obtener el resultado deseado [26].

DSR confluye principalmente en el desarrollo y funcionamiento de “artefactos” diseñados con la intención clara de mejorar el desempeño funcional de dicho artefacto, por lo que, esta metodología se aplica comúnmente a artefactos que incluyen algoritmos, interfaces humano/computadora, metodologías de diseño (incluidos modelos de procesos) y lenguajes, por lo cual, su implementación es más importante en las disciplinas de Ingeniería e Informática, aunque no se limita únicamente a estas [27], [28].

Hevner et al. [29] propusieron un conjunto de pautas para la investigación científica basada en el diseño dentro de la disciplina de Sistemas de Información; en estas se establece que las investigaciones que empleen esta metodología requieren la creación de un artefacto innovador y con propósito para la solución de un problema específico. Dicho artefacto, debe ser evaluado para garantizar así su utilidad en base al problema especificado. Por lo cual, se espera que el artefacto resuelva un problema que aún no esté resuelto o que proporcione una solución más eficaz. Tanto la construcción como la evaluación del artefacto, deben realizarse de manera rigurosa, y los resultados de la investigación deben presentarse de manera efectiva al público asociado a la tecnología.

Ya que nuestro objetivo es construir y evaluar varios modelos para resolver un problema específico referente a los aspectos temporales, mismos que en la actualidad no se toman en cuenta en los SRG, esta metodología se posiciona como la más adecuada para el desarrollo del presente estudio. Otros autores, como Amer-Yahia et al. [9]; Chen et al. [10]; y Verma et al. [30], ya han empleado esta metodología de investigación para el desarrollo de modelos

para SRG.

Para la ejecución de esta investigación se utilizó la metodología de Investigaciones basadas en la Ciencia del Diseño (DSR). Esta metodología propone seis etapas o fases como se muestra en la Figura 2.1.

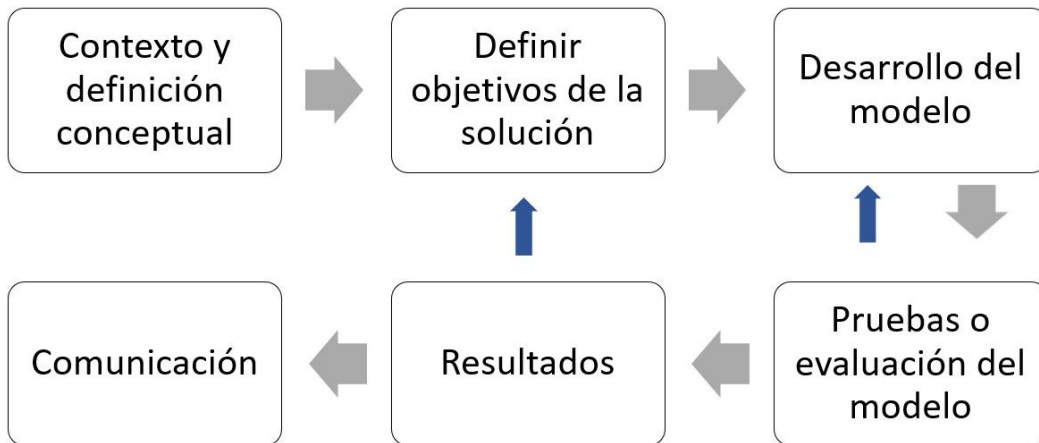


Figura 2.1: Metodología Design Science Research (DSRP).

A continuación se realizará la descripción de cada una de las fases de la metodología DSRP.

FASE 1: La primera etapa consiste en la documentación teórica sobre los aspectos conceptuales relacionados sobre el tema y la literatura relacionada existente en las bases de datos científicas, además la profundización sobre otros estudios que se hayan realizado sobre el tema.

FASE 2: Una vez definida la bibliografía científica como soporte documental, se procederá a la descripción de un modelo adaptativo del tipo dinámico donde se consideren los aspectos de agregación descritos en los parámetros propuestos para el diseño y los diversos aspectos temporales de interés identificados en la literatura. En este caso, se emplearán notaciones matemáticas consistentes y adecuadas con los parámetros a evaluar. Específicamente, se delimitarán conjuntos de pruebas que puedan ser de interés para el estudio. Para conocer las posibles variaciones de interés de los grupos de usuarios, se considera utilizar regresiones lineales sencillas.

FASE 3: Etapa en la que se desarrollará el modelo como tal, para esto, los parámetros calculados previamente, serán programados en las librerías específicas creadas en Python. Adicionalmente, en esta etapa para completar el modelo, se adaptarán librerías de NumPy/Scipy de Python, para que estas contengan los conjuntos completos de parámetros

temporales que se considerarán para el desarrollo de la propuesta.

FASE 4: Etapa de pruebas en el modelo en la que se generarán bases de datos que puedan servir para las respectivas validaciones del modelo. Estos mismos datos se emplearán con el modelo creado y también con sistemas tradicionales de recomendación. Estas evaluaciones y corridas comparativas del modelo creado y los sistemas existentes, se realizarán con base en un estándar de parámetros que surgirá de los resultados obtenidos en las etapas previas de desarrollo del modelo.

FASE 5: En esta etapa de la investigación se presentará los resultados obtenidos, estos incluirán los parámetros finales del diseño, y los cambios realizados a la propuesta inicial que puedan surgir como parte de la adaptación de parámetros al momento de las pruebas respectivas.

FASE 6: En la última etapa se dará a conocer los resultados, generando un artículo académico donde se presente la contribución científica.

2.1 CONTEXTO Y DEFINICIÓN CONCEPTUAL

La creciente importancia de la Web como medio para transacciones electrónicas y comerciales ha servido como motor para el desarrollo de SR, mismos que se han convertido en el corazón de muchas empresas basadas en Internet como Google, YouTube, Facebook, Netflix, LinkedIn, Amazon, entre otras.

Los SR proporcionan sugerencias de elementos que pueden ser de utilidad para un usuario. Las sugerencias están destinadas a apoyar a las personas en varios procesos de toma de decisiones [31]. Para ello, los SR emplean varias fuentes de datos (como entrada) para inferir los intereses del cliente con el fin de generar recomendaciones a un usuario sobre elementos que podrían interesarle [32]. Los diferentes tipos de SR toman distintos datos de entrada y pueden pertenecer a categorías tan amplias como los datos de patrones, los datos demográficos, los datos de producción, los de calificación o ratings, etc. [15].

Dado que es conocido que las preferencias de los usuarios y la popularidad de los artículos a ser recomendados tienden a variar con el tiempo, autores como Koren [5] y Xiang et al. [33] ya han destacado la importancia de un modelado delicado de la dinámica temporal al diseñar modelos de filtrado colaborativo. El análisis de dichos datos debe encontrar el equilibrio adecuado entre el descuento de los efectos temporales que tienen un impacto

muy bajo en el comportamiento futuro, mientras se capturan tendencias a más largo plazo que reflejan la naturaleza inherente de los datos. Modelar los cambios temporales en las preferencias del usuario conlleva desafíos únicos. Un tipo de cambio de concepto en esta configuración es la aparición de nuevos productos o servicios que cambian el enfoque de los clientes; los cuales están relacionados con los cambios estacionales, o feriados específicos, que conducen a patrones de compra característicos. Todos esos cambios influyen en toda la población y están dentro del ámbito de los estudios tradicionales sobre la deriva de conceptos. Sin embargo, muchos de los cambios en el comportamiento de los usuarios se deben a factores localizados. Por ejemplo, un cambio en la estructura familiar puede cambiar drásticamente los patrones de compra. Así mismo, las personas cambian gradualmente su gusto por las películas y la música. Todos esos cambios no pueden ser capturados por métodos que busquen una deriva conceptual global. En cambio, para cada cliente estamos viendo diferentes tipos de deriva de conceptos, cada uno ocurre en un marco de tiempo distinto y se dirige hacia una dirección diferente [5].

En general, según Aggarwal [7] los SR que utilizan los aspectos temporales de la actividad del usuario lo hacen de dos formas diferentes:

1. Calificaciones explícitas: en este caso, las fechas se asocian con calificaciones explícitas. Estas fechas (o *timestamp*) se pueden utilizar para mejorar la precisión del proceso de predicción, ya sea mediante el uso de métodos de predicción o mediante información periódica y estacional (por ejemplo, día de la semana).
2. Comentarios implícitos: estos escenarios corresponden a acciones del cliente, como comprar artículos o hacer clic en páginas web. La secuencia histórica de acciones del usuario se utiliza para hacer predicciones sobre acciones futuras. Los métodos subyacentes a menudo comparten muchas similitudes con los pronósticos basados en patrones secuenciales. Estas técnicas se utilizan con frecuencia en muchos escenarios, como los flujos de clics web o el análisis de registros web. Las técnicas también se pueden utilizar para hacer predicciones sobre el comportamiento de compra futura del cliente.

A pesar de que en la actualidad existe un gran ecosistema de SR, en la que se emplean diversas estrategias y se omiten otras, las estrategias de recomendación basadas en aspectos temporales han sido poco abordadas en los últimos años [17]. No obstante, los aspectos temporales son igualmente importantes, ya que las preferencias pueden cambiar mucho con el tiempo con base en diversos condicionantes, y por ello, es necesario abor-

dar este vacío en la presente investigación y poder obtener una propuesta del sistema de recomendación grupal que considere de manera efectiva los aspectos temporales, para así mejorar la efectividad de estos.

Las preferencias de los usuarios son cambiantes a través del tiempo, de forma que afectan directamente a los SRG. Un ejemplo en el sector tecnológico es que las preferencias por un producto o servicio no son permanentes en el paso de los meses o años, sino que todo evoluciona y se actualiza. Por ejemplo, en un día específico, es posible que un usuario escoja un producto y después de un periodo de tiempo determinado puede seleccionar otro con mejores características.

El paradigma que se pretende abordar incluye la creación de un modelo de seguimiento a lo largo de la vida útil de los datos del comportamiento cambiante en el tiempo. Esto permitiría explotar los componentes relevantes de todas las instancias de datos, descartando lo irrelevante.

En los últimos años, los SRG han sido un gran desafío y se han realizado pocas investigaciones de estos sistemas. Sin embargo, se han solucionado problemas referentes a la automatización de selección de grupos al aplicar técnicas de aprendizaje de máquina [17].

En la actualidad, dichos sistemas, emplean modelos estadísticos que poseen como estructura operativa a las calificaciones dadas por los usuarios sobre algún producto o servicio, si estos usuarios tienen características de compra similares se los incluye en un grupo específico. Se han realizado investigaciones donde solo se consideran variables de tiempo en 0 y 1 para ajustar los modelos con el fin de mejorar la eficiencia del sistema de recomendación grupal [10], [17]. Los SRG actuales pierden eficiencia con el pasar del tiempo si no se actualizan los datos permanentemente.

Con base en esto, considerar los aspectos temporales dentro de los SRG es un gran reto, ya que se necesita conocimiento de varias herramientas para recolectar, analizar, ejecutar y probar el modelo deseado. Para el presente proyecto, la herramienta necesaria para realizar el proceso de recolección de datos, creación, ejecución y pruebas del modelo es el lenguaje de programación Python, que permite el uso de varias librerías disponibles como Pandas, Theano, Numpy y Scipy [34]. Otros lenguajes de programación no tienen librerías con algoritmos de clasificación necesarios para el desarrollo del modelo, solo Python permite realizar este tipo de procesos de forma eficiente y adaptar cambios de forma muy fácil.

2.2 DESCRIPCIÓN MODELO

2.2.1 Sistemas de recomendación adaptativo - dinámico considerando aspectos temporales

En los sistemas de información actuales se tiene muchas variables configurables que deben ser tomadas en cuenta para adaptarse a las necesidades específicas de un caso de uso en particular. En los SR no todos los casos de uso son iguales y por más parecidos que sean dos conjuntos de datos se debe hacer ajustes para entregar mejores resultados [35].

Agregar características de adaptabilidad a un sistema de Tecnológicas de Información y Comunicación(TIC) entrega valor agregado a los usuarios finales y el control para adaptarlo en función de los datos generados de forma constante y creciente dentro de una organización.

Existen varios algoritmos [36], [13], [14], [37] utilizados por los sistemas de recomendación. Cada uno de estos cuenta con variables configurables por el usuario que permiten mejorar los resultados a partir de un conjunto de datos que el SR esté utilizando.

Los sistemas de recomendación basados en aspectos temporales son un derivado de los sistemas de recomendación considerando aspectos contextuales [18]. El tiempo en un SR está formado por la combinación de fecha-hora la cual da información de cuando un usuario proporciona una calificación a un artículo que acaba de consumir y aporta información valiosa que puede ser generada desde cualquier sistema de información que integre un reloj al almacenar información en una base de datos. Existen dos tipos de variables: continuas y categóricas. Las variables continuas indican exactamente el tiempo en el cual un usuario interactúa o da una calificación a un ítem y así como (01 02 2022 / 12:34:50) [38]. Las variables categóricas pueden ser calculadas de acuerdo a un periodo específico de interés, así como (mañana, tarde, noche, durante una temporada del año, durante un año en particular) [38].

Al incluir en un SR que está basado en aspectos temporales la característica de adaptabilidad, indica que el SR es capaz de reaccionar ante ciertas condiciones que el usuario ha identificado al analizar los datos y, por lo tanto, entrega los mejores resultados. Para abarcar los aspectos temporales en un SRG, se formaron secuencias de calificaciones con tres periodos de tiempo para cada una de las series que se forman de la combinación de

artículo-grupo. En el SRG se tomará en cuenta las secuencias de calificaciones formadas para cada serie, estos datos permitirán que el SRG entregue recomendaciones de las siguientes calificaciones para periodos semejantes de tiempo a futuro. Esta característica de secuencias permite agregarle el dinamismo al SR, ya que se puede obtener n periodos de tiempo hacia adelante de la fecha a la cual se utilice el SR para obtener las calificaciones predichas o pronosticadas.

2.2.2 Datasets

2.2.2.1 Conjunto de datos MovieLens

El conjunto de datos de MovieLens es uno de los más comúnmente utilizados para la evaluación de los sistemas de recomendación individuales [36], [39], y se utilizará para evaluar un sistema de recomendación grupal. Este conjunto de datos en particular tiene información sobre la fecha y hora en la que un usuario dio una calificación a una película y se encuentra en el campo timestamp; este campo es esencial para generar las matrices considerando aspectos temporales. MovieLens-25M tiene 25 millones de calificaciones aplicados a 62 mil películas realizadas por 162 mil usuarios^[1]. Existen 19 géneros de películas y los registros de calificaciones van desde el año 1995 a 2019.

2.2.2.2 Conjunto de datos Amazon

El conjunto de datos de Amazon está relacionado con el comercio electrónico en la categoría Electrónica. Tiene información de la fecha y hora en la que un usuario dio una calificación a un artículo electrónico en la página web de Amazon, el cual se almacena en el campo timestamp. Este conjunto de datos tienen 1.3 millones de calificaciones aplicadas a 9560 productos electrónicos realizadas por 1.1 millones de usuarios^[2]. Los registros de calificaciones van desde el año 1999 a 2019.

^[1] <https://grouplens.org/datasets/movielens/25m/>

^[2] https://raw.githubusercontent.com/MengtingWan/marketBias/master/data/df_electronics.csv

2.2.3 Análisis y transformación de datos

En esta sección se realizaron procesos para conocer y comprender los conjuntos de datos utilizados en aspectos como: estructura, tipos de datos y distribución temporal. Luego del análisis se procede a realizar el proceso de transformación de datos para colocarlos en un formato con el que el SRG pueda procesarlos de forma correcta. Las bases de datos utilizadas, al tener un formato .csv son compatibles y se utilizan con la librería Pandas de Python para realizar la lectura y transformación de los datos.

Los dos conjuntos de datos utilizados para realizar esta investigación fueron descargados desde fuentes confiables, además son datos reconocidos en el entorno de los sistemas de recomendación, ya que han sido utilizados previamente en otras investigaciones [36], [39], [40].

El formato de los conjuntos de datos es .csv, que son archivos de formato Excel que permite asociarse, cargar y realizar los procesos necesarios para esta investigación de forma eficiente en Jupyter Notebook 6.3.0^[3].

Por lo general, los datos no tienen un formato que se adapte de forma perfecta a un sistema, por lo que es necesario realizar un proceso de evaluación y transformación tomando en consideración las necesidades del SR. En la primera etapa se eliminaron varias columnas que no tienen datos relevantes para el SR, también se cambiaron de nombres las columnas. La Tabla 2.1 tiene información de los nombres de las columnas, tipo de datos y descripción de los datos que se van a ingresar al SRG.

Para ambos conjuntos de datos, el siguiente paso de preparación de datos consiste en dividir los registros en tres bloques. Cada bloque representa un período de tiempo y tiene los registros cuya marca de tiempo cae en el bloque de tiempo correspondiente. La tabla 2.2 indica la cantidad de registros, elementos y usuarios para los conjuntos de datos de MovieLens y Amazon. Aquí también se muestran los periodos de tiempo correspondientes a cada bloque. Luego, las tres matrices de datos para MovieLens se estructuran de la siguiente manera. Bloque *T1* con calificaciones otorgadas de 2005 a 2009, Bloque *T2* para el

^[3] Es un software libre con protocolos de estándar abierto y con servicios web para interactuar con al rededor de 40 lenguajes de programación en los que se incluye Python y R. Fue desarrollado para ayudar en el campo de la ciencia de datos y procesos computacionales en general. Jupyter Notebook es una aplicación web para crear y compartir documentos computacionales.

Campo	Tipo	Descripción
userId	string	Corresponde al identificador de usuario que ha dado una calificación.
mmovieId/itemId	string	Corresponde al identificador de película o artículo al que el usuario ha calificado.
rating	int	Identifica la calificación en el rango de [0-5] que un usuario ha adjudicado a una película que ha visto o artículo que ha consumido.
timestamp	date	Corresponde a la fecha en la cual un usuario ha calificado a una película o artículo que esta constituido por el día, mes y año.

Tabla 2.1: Descripción de las columnas de los conjuntos de datos.

		T1	T2	T3
		2005-2009	2010-2014	2015-2019
MovieLens	Cantidad de registro	656,291	218,613	113,796
	Cantidad de artículos	7,145	7,145	7,145
	Cantidad de usuarios	1,637	1,637	1,637
		T1	T2	T3
		1999-2013	2014-2016	2017-2019
Amazon	Cantidad de registros	531	1,064	1,174
	Cantidad de artículos	200	200	200
	Cantidad de usuarios	168	168	168

Tabla 2.2: Número de registros, artículos y usuarios, antes de aplicar factorización matricial para MovieLens y Amazon.

período 2010-2014 y Bloque *T3* con calificaciones otorgadas de 2015 a 2019. Los bloques para el conjunto de datos de Amazon son *T1* para el período 1990-2013, *T2* para el período 2014-2016 y *T3* para el período 2017-2019.

2.2.4 Predicción de datos con factorización matricial

El modelo de factorización matricial se utiliza para obtener las calificaciones faltantes de un usuario o grupo de usuarios sobre una película que nunca ha visto o un artículo que no ha consumido antes y carece de calificación. Inferir las calificaciones con este modelo se logra con mayor eficiencia cuando se determina un conjunto de datos más pequeño [41]. Para formar grupos de datos más pequeños se aplicaron filtros en el conjunto de datos de las películas. Para este dataset, se filtró considerando el género de película como por ejemplo drama, comedia o terror. Por otro lado, en el dataset de Amazon se filtró los artículos cuya categoría es "equipos electrónicos".

Además de filtrar el conjunto de datos, se determinaron tres matrices de datos *T1*, *T2* y *T3* para cada uno de los periodos de tiempo determinados después del proceso de extracción,

transformación y carga de datos.

Cabe mencionar que los usuarios y elementos son los mismos en los tres bloques y fueron seleccionados considerando su apariencia permanente en el tiempo. Es decir, se considerará al usuario que otorgó al menos una calificación en cada uno de los bloques de tiempo. De la misma forma, se considerará también el ítem que tenga al menos una calificación en cada uno de los bloques de tiempo. La valoración que un usuario ha dado a un artículo queda registrada en uno de los bloques y permanece estática en los bloques posteriores. Por ejemplo, una calificación que existe en el bloque T1 se mantendrá en los bloques T2 y T3, mientras que una calificación que pertenece al bloque T2 se conserva en el bloque T3, pero no existe en T1 (se va a predecir).

Para dar un caso de uso real, consideremos una familia cuyos miembros ven películas y dan calificaciones después de consumirlas. En 1999, esta familia vio una película clásica y le dio su calificación c_{T1} . Luego, lo volvieron a ver en los años 2004 y 2020 y proporcionaron las calificaciones c_{T2} y c_{T3} correspondientes. El GRS que modela estos aspectos temporales puede predecir cuál será la calificación \hat{c}_{T4} .

Con la consideración anterior, se va a completar parcialmente las calificaciones faltantes en la matriz T3 con las calificaciones originales (sin datos de predicción) que se tiene en T1 y T2, lo que hay que tener claro es que no se va a completar en su totalidad la matriz T3. La matriz T3 va a estar completamente llena luego de aplicar el modelo de factorización matricial. La matriz T2 se va a completar parcialmente con las calificaciones de la matriz T1. Y la matriz T1 va a tener calificaciones faltantes como se muestra en la Figura 2.2.

T1	i1	i2	i3	i4	T2	i1	i2	i3	i4	T3	i1	i2	i3	i4
u1	x	x	x	4.1	u1	x	x	3.1	4.1	u1	x	4.0	3.1	4.1
u2	x	x	5.0	2.9	u2	x	3.0	5.0	2.9	u2	2.0	3.0	5.0	2.9
u3	x	x	x	x	u3	x	x	x	4.0	u3	3.5	x	3	4.0
u4	2.0	3.0	x	x	u4	2	3.0	x	5.0	u4	2	3.0	2.7	5.0

Figura 2.2: Matrices usuario-artículo para las tres temporalidades.

Para las tres matrices T1, T2 y T3 se aplicó el modelo de factorización matricial kernel para determinar las calificaciones faltantes. Se utilizó este modelo, ya que no realiza ninguna división de los datos (no pierde información) y la predicción de una calificación con valores más altos al máximo [5] no es posible [22]. Al aplicar este modelo se va a llenar todas las calificaciones faltantes de las tres matrices T1, T2 y T3 como se muestra en la Figura 2.3.

T1	i1	i2	i3	i4	T2	i1	i2	i3	i4	T3	i1	i2	i3	i4
u1	3.4	4.2	3.0	4.1	u1	3.4	4.0	4.1	4.1	u1	3.6	4.0	3.1	4.1
u2	2.0	2.0	5.0	2.9	u2	2.0	4.7	5.0	2.9	u2	2.0	3.0	5.0	2.9
u3	3.3	4.1	2.5	3.0	u3	3.5	4.3	3.0	2.6	u3	3.5	4.0	3.0	4.0
u4	2.0	3.0	2.8	1.8	u4	2.0	3.0	2.7	2.5	u4	2.0	3.0	2.7	5.0

Figura 2.3: Matrices usuario-artículo para las tres temporalidades.

Para obtener las calificaciones faltantes se hace uso de la librería `matrix_factorization` [4] de la cual se importa `KernelMF`. Además, es necesario obtener dos conjuntos de datos, entrenamiento y pruebas, para entrenar y probar el modelo. La división del conjunto de datos se establece en cada bloque, y está dado por el 20 % de los datos para pruebas y el 80 % para datos de entrenamiento para evitar el sobre ajuste.

2.2.5 Generación de grupos y aplicación de estrategias de agregación

Con las tres matrices T1, T2 y T3 llenas con todas las calificaciones para cada usuario-película o usuario-artículo, se procede a realizar la detección de grupos de usuarios por similitud. A continuación, se procede a aplicar las cuatro estrategias de agregación de calificaciones individuales para encontrar las preferencias del grupo.

Los usuarios que tienen características o preferencias similares van a formar parte de un mismo grupo [2]. Para calcular la similitud entre un usuario y su vecino más cercano se comparan todas las calificaciones dadas a todos los artículos o películas. La similitud se determina con el coeficiente de correlación de Pearson (PCC) y usa regresiones lineales para comparar, e indicar si la calificación de un usuario es similar a la de su vecino para posteriormente determinar si dos usuarios pertenecen a un mismo grupo.

El coeficiente de correlación de Pearson entre un usuario u y un vecino n está dado por la ecuación:

$$SimUsuarios(u, n) = \frac{\sum_{i \in I_{un}} (c_{ui} - \bar{c}_u)(c_{ni} - \bar{c}_n)}{\sqrt{\sum_{i \in I_{un}} (c_{ui} - \bar{c}_u)^2} \sqrt{\sum_{i \in I_{un}} (c_{ni} - \bar{c}_n)^2}} \quad (2.1)$$

[4] <https://pypi.org/project/matrix-factorization/>

donde:

c : determina la calificación que ha entregado un usuario u o n a un artículo i .

Un ejemplo de una matriz de calificaciones es el que se muestra en la Tabla 2.3

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
U_1	C_{11}			C_{14}	C_{15}			C_{18}
U_2		C_{22}	C_{23}	C_{24}		C_{26}	C_{27}	
U_3	C_{31}		C_{33}		C_{35}			C_{38}
U_4	C_{41}	C_{42}		C_{44}		C_{46}		C_{48}
				...				
U_n	C_{n1}	C_{n2}	C_{n3}	C_{n4}	C_{n5}	C_{n6}	C_{n7}	C_{n8}

Tabla 2.3: Matriz de calificaciones [17].

La métrica del coeficiente de correlación de Pearson está en el rango de -1 a 1, si es cercano a 1 existe similitud completa y si es cercano a -1 no existe similitud [17]. Este método se utilizó para determinar la similitud entre las calificaciones de dos usuarios.

Dos o varios usuarios pueden formar grupos por tener gustos o preferencias similares [42], los cuales pueden ser agrupados, ya sea por características particulares como fechas similares en que los usuarios que dieron calificaciones a un artículo en común, edades de los usuarios, preferencias, entre otras.

Cada una de las matrices T1, T2 y T3 tienen los mismos usuarios, por lo que solo se va a utilizar la matriz T1 para obtener los grupos por similaridad. Es importante destacar que al aplicar este método se puede definir el número de miembros por cada grupo. En el presente proyecto se asumió para la aplicación integral del método, que se pueden tener grupos de dos miembros, luego se trabajó con grupos de 3 miembros, después con grupos de 4 y de 5,6, 7 y finalmente los experimentos se hicieron con grupos de 8 miembros.

Para la generación de grupos por similitud se utilizó el método `generate_similar_user_group`, el cual permite definir un rango de 2 a 8 miembros que conformaran cada grupo. Este método procede a realizar el cálculo del coeficiente de correlación de Pearson para determinar los grupos de usuarios. Como resultado de este proceso se generan archivos donde se aprecia el identificador de grupo y los n identificadores de usuario que pertenecen al grupo.

Una vez que se generó los grupos, se van a aplicar 4 estrategias de agregación para obtener las calificaciones grupales y adicional se calcula el error medio cuadrático de cada estrategia aplicada para las 3 matrices T1, T2 y T3.

2.2.6 Modelos de predicción de calificaciones para grupos tomando en cuenta aspectos temporales

Los SR basados en aspectos temporales consideran que las recomendaciones entregadas sobre artículos se basan en información que ha sido generada modelando las acciones pasadas que fueron realizadas por los usuarios. Los SR que toman en cuenta los aspectos secuenciales comparten ciertos puntos en común con los basados en aspectos temporales, ya que para poder conocer cuál es el siguiente artículo que se va a recomendar es necesario conocer la secuencia de cómo fueron seleccionados o consumidos.

Bajo estas premisas se formó secuencias de calificaciones dadas en tres periodos de tiempo T1, T2 y T3 por cada uno de los grupos, y al aplicar algoritmos de SR basados en aspectos secuenciales se busca determinar las siguientes calificaciones que se le entrega a un artículo o película.

Quadrana [18] utiliza varios modelos en su investigación, de todos estos se seleccionaron *popularity recommender*, *Factorizing Personalized Markov Chains (FPMC)*, *Prod2Vec* y *Mixed Markov Chain*. Estos modelos van a permitir generar las siguientes calificaciones a partir de la secuencia entrante para una serie (usuario-artículo).

Para poder realizar el procesamiento es necesario generar un archivo .csv con los campos que se muestran en la Tabla 2.4. Con cada uno de los conjuntos de datos se procede a realizar una división en datos de entrenamiento y pruebas. Para los datos de pruebas solo se considera la última interacción de la secuencia de calificaciones, entregada a por a serie grupo-película o grupo-artículo. Todos los datos restantes van a ser datos de entrenamiento. La función para realizar este proceso se denomina `last_session_out_split`.

2.3 DESARROLLO DEL MODELO

En la Figura 2.4 se describe el esquema que utilizó para el desarrollo del SRG basado en aspectos temporales.

Campo	Tipo	Descripción
GroupID	string	Corresponde al identificador de grupo de usuarios que ha calificado una película o un artículo electrónico.
mmovieId/itemId	string	Corresponde al identificador de película o artículo electrónico.
RatingsGroups	int	Identifica la calificación grupal en el rango de [1-5] que un grupo a entregado a una película que ha visto o artículo electrónico que ha consumido.
ts	date	Corresponde a la fecha en la cual un grupo ha calificado a una película o artículo electrónico. Está constituido por la hora, día, mes y año.
session_id	string	Corresponde al identificador grupo-película o grupo-artículo.

Tabla 2.4: Detalle de los conjuntos de datos después de la formación de grupos.

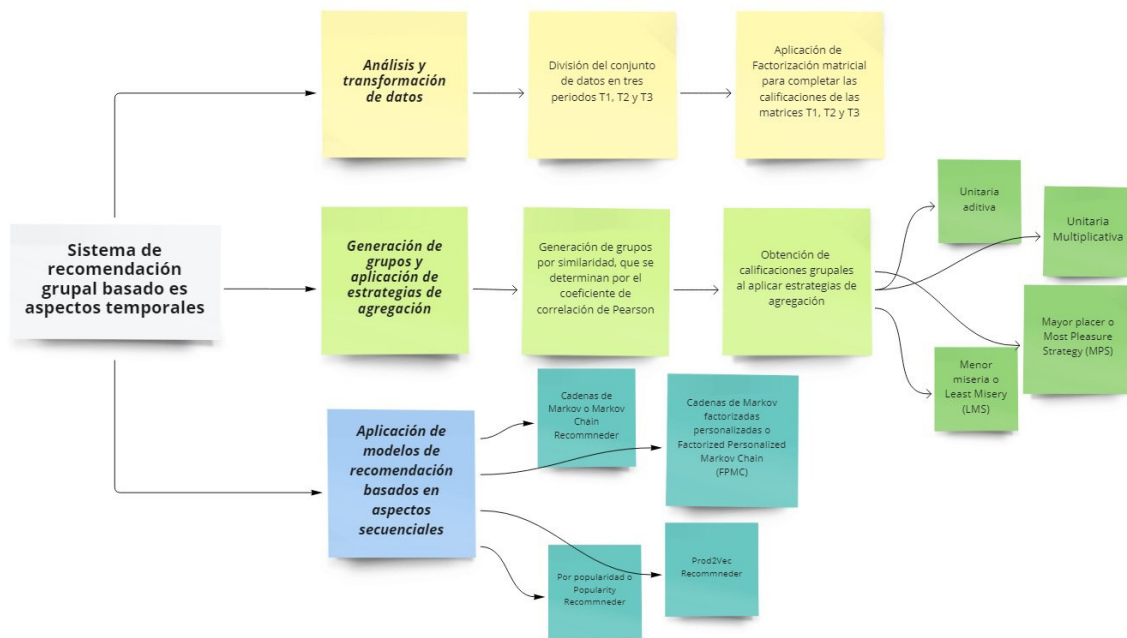


Figura 2.4: Esquema de desarrollo del modelo

2.3.1 Esquema de desarrollo del modelo

2.4 PRUEBAS DEL MODELO

Se presentarán las pruebas realizadas de los procesos utilizados por el SRG que van a ser aplicados a dos conjuntos de datos diferentes.

2.4.1 Pruebas en los conjuntos de datos

Los datos, tablas y gráficas presentadas en esta sección son los que se han generado en los procesos de desarrollo del modelo descritos en la sección anterior.

Después de realizar el proceso de predicción de calificaciones faltantes se muestran en la Tabla 2.5 la cantidad de registros de calificaciones, usuarios y artículos de cada conjunto de datos y de cada una de las temporalidades T1, T2 y T3.

	MovieLens	Amazon Electronics
# Usuarios comunes	1,637	168
# Películas/Ítems comunes	7,076	200
Total de registros	10,88,1486	32,510

Tabla 2.5: Número de registros, usuarios y artículos de cada conjunto de datos T1, T2 y T3.

En la generación de los grupos se crean archivos .csv donde se visualiza cada identificador de grupo con sus respectivos miembros, esto se puede evidenciar en la Figura 2.5. Se puede observar que se generaron 7 archivos con grupos de 2 a 8 miembros, además en el archivo de grupo_2.csv se puede ver cada uno de los identificadores de usuario que pertenecen a cada grupo. Una consideración que se debe tener en cuenta es que un usuario individual debe pertenecer al menos a un grupo y dentro de ese grupo no se puede repetir dos identificadores de usuario. Adicional un usuario puede pertenecer a varios grupos.

Nombre	Tipo	Tamaño
group_2	Archivo CSV	4 KB
group_3	Archivo CSV	5 KB
group_4	Archivo CSV	6 KB
group_5	Archivo CSV	8 KB
group_6	Archivo CSV	9 KB
group_7	Archivo CSV	10 KB
group_8	Archivo CSV	11 KB

GroupID	MembersID
0,16528	42268
1,22501	95893
2,54126	30656
3,68680	30661
4,11517	93904
5,68934	83956
6,28557	106514
7,67491	59478
8,126512	11920
9,156491	39923
10,28708	77523
11,49316	95893
12,90964	93904
13,46206	1732
14,55007	39923

Figura 2.5: Archivos de grupos generados.

Además, se generó archivos de cada una de las estrategias de agregación, aplicadas a las tres temporalidades. En la Figura 2.6 se puede observar los archivos generados para las estrategias *Most Pleasure (MPS)* y *Least Misery (LMS)*. También se visualiza el archivo

group_movies_sizee_LMS que es el resultado de aplicar la estrategia de agregación LMS al conjunto de datos T1 y obtener la calificación grupal para cada identificador de artículo.

GroupID	mmovieId	RatingsGroups
0	2	4.23
1	2	4.28
2	2	4.36
3	2	4.29
4	2	4.35
5	2	4.26
6	2	4.36
7	2	4.19
8	2	4.27
9	2	4.27
10	2	4.35
11	2	4.35
12	2	4.36
13	2	4.32
14	2	4.27
15	2	4.35
16	2	4.27
17	2	4.5
18	2	4.32
19	2	4.31
20	2	4.35
21	2	4.21
22	2	4.32
23	2	4.36
24	2	4.31
25	2	4.39

Figura 2.6: Archivos de calificaciones grupales MPS y LMS.

En la Figura 2.7 se observa 5 registros con las columnas Session_id que es un identificador película-grupo, secuencia que es una matriz de tres elementos donde cada uno representa una calificación dada por el grupo para la película en T1, T2 y T3, a cada una de estas calificaciones se le ha multiplicado por 100 para no perder información de los decimales de los ratings grupales y la última columna es el campo GroupID que representa el identificador de grupo. Las tres calificaciones que se muestran fueron tomadas de las tres matrices T1, T2 y T3 después de aplicar las estrategias de agregación. Siempre se va a tener las calificaciones para cada película o artículo electrónico por grupo, ya que las matrices T1, T2 y T3 están completas y tienen todas las calificaciones para cada artículo.

2.5 RESULTADOS

Con toda la información del algoritmo de predicción, las estrategias de agregación y los métodos de recomendación, se utilizaron para generar un modelo aplicable en SRG que

	session_id	sequence	GroupID
0	0	[399, 367, 348]	0
1	1	[420, 413, 371]	1
2	2	[449, 392, 400]	2
3	3	[409, 394, 387]	3
4	4	[386, 353, 344]	4

Figura 2.7: Registros de secuencias formadas por cada grupo.

considere los aspectos temporales. Los resultados del modelo que se van a presentar representan un análisis comparativo de todas las estrategias de agregación aplicadas y los métodos de recomendación aplicados a dos diferentes conjuntos de datos que responden a las preguntas de investigación planteadas en la introducción de este documento. Para determinar la mejor estrategia de agregación, cada una se somete a una evaluación con la métrica Root Mean Square Error (RMSE) [17]. Luego, para determinar cuál es el mejor método de recomendación para grupos se evaluarán bajo dos métricas RMSE y Normalized Discounted Cumulative Gain (NDCG). Con el fin de garantizar el funcionamiento y los resultados obtenidos del sistema de recomendación, se aplican a dos conjuntos de datos MovieLens y Amazon. Todos los resultados se van a mostrar en la siguiente sección de este documento.

2.6 COMUNICACIÓN

Los resultados de esta investigación fueron enviados a la revista *User Modeling and User-Adapted Interaction Special Issue on "Group Recommender Systems"*^[5].

Esta conferencia invita a académicos que han investigado temas sobre recomendaciones de grupos, desde nuevos algoritmos hasta aplicaciones recientes.

Adicional, los archivos ejecutables necesarios para reproducir los resultados de esta investigación se los almaceno en el repositorio de GitHub <https://github.com/ChrisLex/Time-Aware-Group-System-Recomender>.

^[5] <https://ludovicoboratto.github.io/umuaigroupecsys/>

3 RESULTADOS Y DISCUSIÓN

3.1 RMSE

La calidad de la predicción de las calificaciones grupales está medida por la Raíz del Error Medio Cuadrático con sus siglas en inglés RMSE, a través de la comparación de cada calificación c_{gi} dada por un grupo de usuarios g a una película o artículo electrónico i con la predicción de la calificación p_{gi} entregada por una estrategia de agregación o método de recomendación [17].

El cálculo del RMSE viene dado por la siguiente ecuación:

$$RMSE = \sqrt{\frac{\sum_{i=0}^n (c_{gi} - p_{gi})^2}{n}} \quad (3.1)$$

donde: n : representa a la cantidad de calificaciones disponibles en el conjunto de pruebas.

3.2 EVALUACIÓN DE ESTRATEGIAS DE AGREGACIÓN

Una vez aplicadas las cuatro estrategias de agregación a los dos conjuntos de datos, se realizaron dos gráficas donde se indica la métrica RMSE que pertenece a cada una de las estrategias por cada número de miembros del grupo.

En una primera etapa se obtienen los datos de la métrica RMSE para el conjunto de datos MovieLens, donde se evalúa por cada grupo de 2 a 8 miembros cada una de las estrategias de agregación. Los valores de RMSE van del 0.35 a 0.9, Most Pleasure y Least Misery son las estrategias de agregación que presentan menor RMSE. Al ir incrementando el número de miembros del grupo, el RMSE también aumenta. Esto se evidencia en la Figura 3.1

En una segunda etapa se realiza el mismo proceso para el conjunto de datos de Amazon

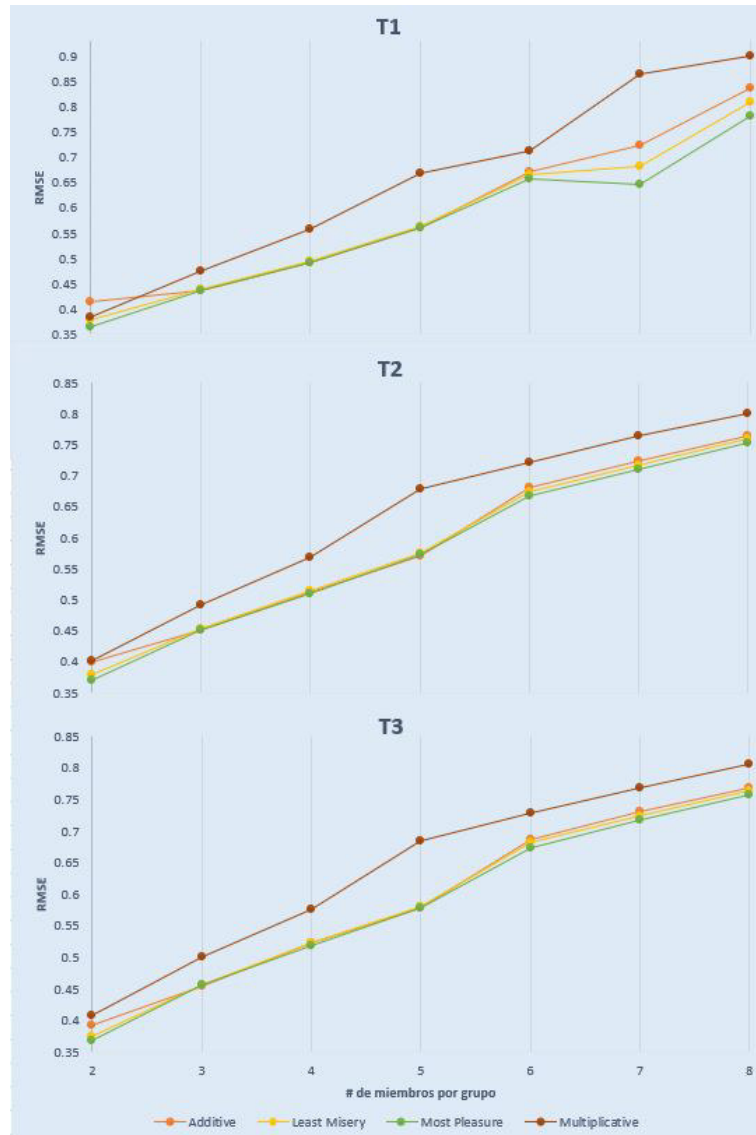


Figura 3.1: Métrica RMSE por número de miembros de grupo al aplicar cuatro estrategias de agregación - MovieLens.

y se aplican las cuatro estrategias de agregación bajo las mismas condiciones que se aplicaron previamente al conjunto de datos de MovieLens. Los valores de RMSE van del 0.1 a 0.26, Most Pleasure y Least Misery son las que presentan menor error al igual que el conjunto de datos de MovieLens. Esto se evidencia en la Figura 3.2.

3.3 NDCG

Los sistemas de recomendación en función de las calificaciones entregan una puntuación de relevancia a cada uno de los artículos proporcionados. La Ganancia Acumulativa o en inglés denominado Cumulative Gain (CG) es la sumatoria de las puntuaciones de relevancia

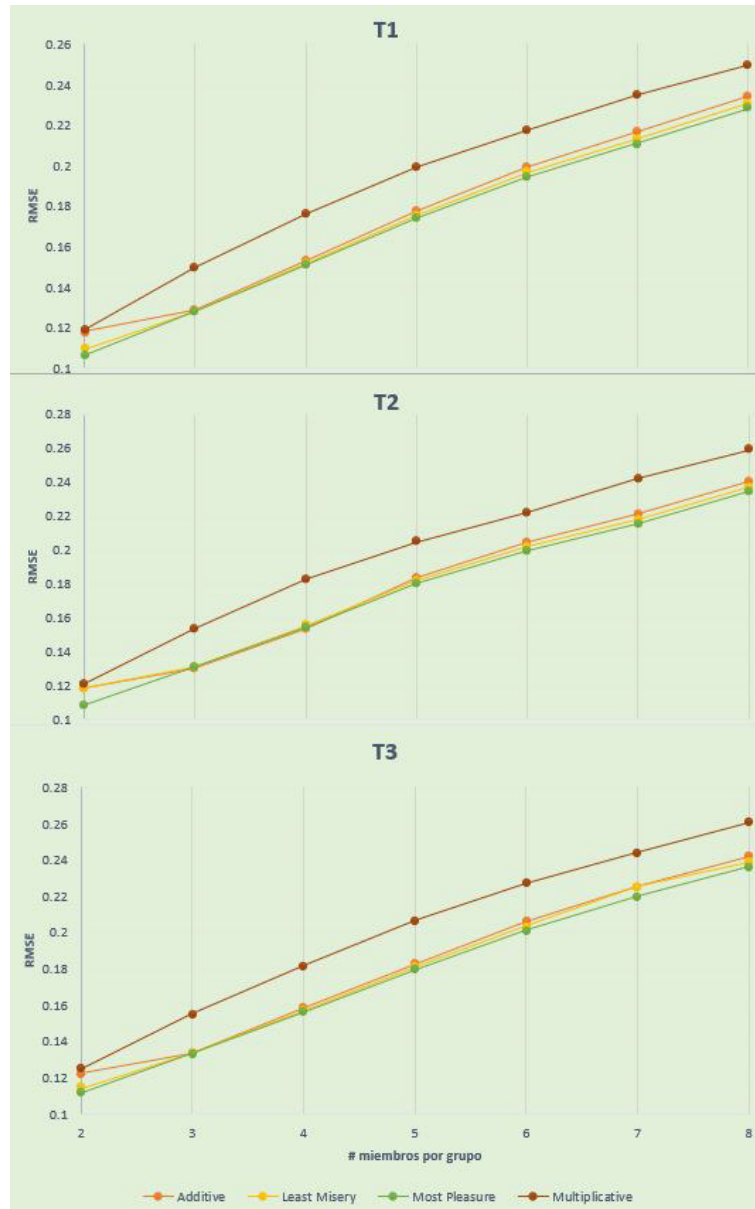


Figura 3.2: Métrica RMSE por número de miembros de grupo al aplicar cuatro estrategias de agregación - Amazon.

de una secuencia de recomendación.

La ganancia acumulativa descontada o en inglés Discounted Cumulative Gain(DCG) permite penalizar degradando los valores de relevancia de forma algorítmica, como se muestra en la siguiente ecuación [43]:

$$DCG@k(g) = \sum_{i=1..k} \frac{2^{relevance(t_i,g)} - 1}{\log_2(1 + i)} \quad (3.2)$$

donde:

k: indica el número de artículos recomendados.

(t_i, g) : indica la relevancia, si el valor es 1 la posición del artículo t_i es relevante para el grupo g y si es 0 no es relevante.

Cuando la longitud de los artículos recomendados para grupos varía, es decir, no hay un valor de k fijo. El DCG tiene que ser normalizado estableciendo el DCG con relación a la ganancia acumulada descontada ideal ($iDCG$) ver Ecuación 3.3. El resultado de esta relación da como resultado del $nDCG$ ver Ecuación 3.4.

$$iDCG@k(g) = \sum_{i=1..k} \frac{1}{\log_2(1+i)} \quad (3.3)$$

$$nDCG@k(g) = \frac{DCG@k(g)}{iDCG@k} \quad (3.4)$$

3.4 EVALUACIÓN DE LOS MODELOS DE RECOMENDACIÓN

Los modelos de recomendación permiten encontrar un número dinámico de calificaciones secuenciales recomendadas por el SRG, y para determinar cuál de los modelos presenta mejores resultados, se realizó una evaluación con las métricas RMSE y $nDCG$.

La métrica RMSE se calcula entre el valor real que es el tercer valor de la secuencia de calificaciones que viene dado en la temporalidad T3 y el valor de la calificación que entrega el SRG. El SRG es dinámico, porque puede entregar n números de secuencias de calificaciones, y estas calificaciones vienen dadas por n separaciones temporales.

En las Figuras 3.3 , 3.4 , 3.5 y 3.6 se tiene mapas de calor con todos los valores del RMSE para la estrategia de agregación Most Pleasure y Least Misery de los conjuntos de datos MovieLens y Amazon electronics, donde se toma en cuenta solo grupos de 2, 5 y 8 miembros como una muestra. Adicional, el recomendador ha entregado un n número de calificaciones por cada uno de los modelos que se han utilizado. Se puede evidenciar que para todos los casos los modelos de recomendación Popularity y Factorization Personalized Markov Chain presentan el menor valor de RMSE. También se puede observar que en todos los modelos del SRG, el RMSE crece en función al número de recomendaciones.

La métrica del $nDCG$ calculada para cada uno de los modelos de recomendación se mues-

MPS 2	# Recomendaciones									
	1	5	10	20	30	50	100	120	150	200
Factorization Personalized Markov Chains Recommender	0.1562	0.1647	0.1684	0.1771	0.1915	0.2302	0.4027	0.4893	0.599	0.764
Popularity recommender	0.1825	0.1882	0.1905	0.1985	0.2127	0.2479	0.3643	0.4132	0.4986	0.6561
Mixed Markov Chains Recommender	0.1832	0.2515	0.2359	0.2643	0.2858	0.3077	0.384	0.3916	0.3916	0.3916
Prod2Vec Recommender	0.2392	0.25	0.254	0.2724	0.2724	0.2724	0.2724	0.2724	0.2724	0.2724
MPS 5	# Recomendaciones									
Factorization Personalized Markov Chains Recommender	0.1538	0.1646	0.1695	0.1786	0.1914	0.2268	0.3985	0.4798	0.585	0.6593
Popularity recommender	0.1686	0.1798	0.1808	0.1839	0.1962	0.237	0.3747	0.4241	0.5093	0.6593
Mixed Markov Chains Recommender	0.264	0.2817	0.2798	0.2671	0.278	0.2799	0.3503	0.3506	0.3506	0.3506
Prod2Vec Recommender	0.27	0.4023	0.4742	0.4746	0.4746	0.4746	0.4746	0.4746	0.4746	0.4746
MPS 8	# Recomendaciones									
Factorization Personalized Markov Chains Recommender	0.1481	0.1596	0.1649	0.1747	0.1879	0.2245	0.4011	0.4839	0.5662	0.5845
Popularity recommender	0.1588	0.1642	0.1728	0.1769	0.1934	0.2353	0.3709	0.4312	0.5089	0.5845
Mixed Markov Chains Recommender	0.2233	0.2504	0.2479	0.2706	0.267	0.2714	0.3372	0.3372	0.3372	0.3372
Prod2Vec Recommender	0.3643	0.4275	0.4979	0.5047	0.5047	0.5047	0.5047	0.5047	0.5047	0.5047

Figura 3.3: Mapa de calor RMSE con Most Pleasure - MovieLens.

LMS 2	# Recomendaciones									
	1	5	10	20	30	50	100	120	150	200
Factorization Personalized Markov Chains Recommender	0.1582	0.1648	0.1685	0.1778	0.1924	0.2328	0.3955	0.4892	0.636	0.811
Popularity recommender	0.1993	0.2056	0.2099	0.2181	0.2315	0.2671	0.3921	0.4368	0.5264	0.6866
Mixed Markov Chains Recommender	0.2204	0.2729	0.2665	0.2728	0.294	0.3397	0.4424	0.4603	0.4605	0.4605
Prod2Vec Recommender	0.27	0.2909	0.3005	0.3368	0.3368	0.3368	0.3368	0.3368	0.3368	0.3368
LMS 5	# Recomendaciones									
Factorization Personalized Markov Chains Recommender	0.1684	0.1778	0.1829	0.1935	0.2076	0.2507	0.4256	0.5225	0.6501	0.8107
Popularity recommender	0.2327	0.2261	0.2272	0.2386	0.247	0.2791	0.4001	0.4676	0.5572	0.7495
Mixed Markov Chains Recommender	0.2303	0.2918	0.3194	0.3603	0.3745	0.3763	0.462	0.4835	0.4856	0.4856
Prod2Vec Recommender	0.3496	0.385	0.4074	0.4537	0.4537	0.4537	0.4537	0.4537	0.4537	0.4537
LMS 8	# Recomendaciones									
Factorization Personalized Markov Chains Recommender	0.2486	0.2573	0.2555	0.2555	0.2663	0.3083	0.5067	0.5929	0.6856	0.7899
Popularity recommender	0.2072	0.2649	0.2535	0.2551	0.2592	0.2872	0.3763	0.4329	0.5181	0.6924
Mixed Markov Chains Recommender	0.1936	0.2875	0.3098	0.3297	0.3562	0.3599	0.4587	0.4952	0.5102	0.5103
Prod2Vec Recommender	0.515	0.6665	0.6877	0.7465	0.7465	0.7465	0.7465	0.7465	0.7465	0.7465

Figura 3.4: Mapa de calor RMSE con Least Misery - MovieLens.

MPS 2	# Recomendaciones									
	1	5	10	20	30	50	100	120	150	200
Popularity recommender	0.0784	0.0927	0.0931	0.1141	0.137	0.1955	0.2857	0.2857	0.2857	0.2857
Factorization Personalized Markov Chains Recommender	0.0916	0.101	0.1101	0.1243	0.1446	0.2031	0.2857	0.2857	0.2857	0.2857
Mixed Markov Chains Recommender	0.1067	0.1304	0.139	0.1403	0.1511	0.1988	0.2133	0.2133	0.2133	0.2133
Prod2Vec Recommender	0.1528	0.1754	0.1807	0.1837	0.1837	0.1837	0.1837	0.1837	0.1837	0.1837
MPS 5	# Recomendaciones									
Popularity recommender	0.0694	0.0786	0.0855	0.1044	0.1319	0.1937	0.2335	0.2335	0.2335	0.2335
Factorization Personalized Markov Chains Recommender	0.0751	0.0896	0.0951	0.111	0.1393	0.2038	0.2335	0.2335	0.2335	0.2335
Mixed Markov Chains Recommender	0.0893	0.0912	0.1118	0.1261	0.1374	0.1885	0.1897	0.1897	0.1897	0.1897
Prod2Vec Recommender	0.1412	0.148	0.1551	0.1569	0.1569	0.1569	0.1569	0.1569	0.1569	0.1569
MPS 8	# Recomendaciones									
Popularity recommender	0.0676	0.078	0.0784	0.0961	0.1273	0.1961	0.2157	0.2157	0.2157	0.2157
Factorization Personalized Markov Chains Recommender	0.0724	0.0863	0.089	0.1022	0.1337	0.2029	0.2157	0.2157	0.2157	0.2157
Mixed Markov Chains Recommender	0.0896	0.097	0.1008	0.1194	0.1342	0.1768	0.1768	0.1768	0.1768	0.1768
Prod2Vec Recommender	0.1062	0.1479	0.1594	0.1613	0.1613	0.1613	0.1613	0.1613	0.1613	0.1613

Figura 3.5: Mapa de calor RMSE con Most Pleasure - Amazon.

tra en las Figuras 3.7, 3.8, 3.9 y 3.10. Estas Figuras son mapas de calor con todos los valores de la métrica nDCG para las estrategias de agregación Most Pleasure (MPS) y Least Misery (LMS) de los conjuntos de datos de MovieLens y Amazon. Nosotros resumimos los resultados para tamaños de grupos de 2,5 y 8 miembros.

LMS 2	# Recomendaciones									
	1	5	10	20	30	50	100	120	150	200
Popularity recommender	0.0793	0.1118	0.0977	0.1048	0.1319	0.1925	0.2825	0.2825	0.2825	0.2825
Factorization Personalized Markov Chains Recommender	0.0917	0.103	0.1123	0.1172	0.1384	0.1993	0.2818	0.2818	0.2818	0.2818
Mixed Markov Chains Recommender	0.0933	0.1041	0.1127	0.126	0.1472	0.1981	0.215	0.215	0.215	0.215
Prod2Vec Recommender	0.1435	0.1762	0.1895	0.1961	0.1961	0.1961	0.1961	0.1961	0.1961	0.1961
LMS 5	# Recomendaciones									
	1	5	10	20	30	50	100	120	150	200
Popularity recommender	0.0676	0.0932	0.0847	0.1022	0.1285	0.1886	0.24	0.24	0.24	0.24
Factorization Personalized Markov Chains Recommender	0.0756	0.0913	0.0952	0.1067	0.1343	0.199	0.24	0.24	0.24	0.24
Mixed Markov Chains Recommender	0.0762	0.0955	0.0967	0.1276	0.1347	0.1937	0.198	0.198	0.198	0.198
Prod2Vec Recommender	0.0988	0.145	0.1557	0.1549	0.1549	0.1549	0.1549	0.1549	0.1549	0.1549
LMS 8	# Recomendaciones									
	1	5	10	20	30	50	100	120	150	200
Popularity recommender	0.0583	0.0691	0.0768	0.0992	0.1246	0.1907	0.2315	0.2315	0.2315	0.2315
Factorization Personalized Markov Chains Recommender	0.0745	0.0879	0.0896	0.1028	0.1305	0.1939	0.2315	0.2315	0.2315	0.2315
Mixed Markov Chains Recommender	0.0808	0.1139	0.1229	0.1185	0.1325	0.1867	0.1894	0.1894	0.1894	0.1894
Prod2Vec Recommender	0.1022	0.1161	0.1321	0.1351	0.1351	0.1351	0.1351	0.1351	0.1351	0.1351

Figura 3.6: Mapa de calor RMSE con Least Misery - Amazon.

Para el cálculo del nDCG el número de artículos recomendados entregados por el SRG es tomado en cuenta; para lo cual se utilizó 3, 5, 10 y 16 recomendaciones.

Valores más altos de nDCG indican que el orden de lista de recomendación proporcionada por el SRG coloca en la parte superior de la lista los elementos relevantes. Por lo tanto, se determina que los modelos de recomendación Prod2Vec y Factorization Personalized Markov Chain son los que presentan mejores resultados.

Además, se observa que mientras más alto es el número de recomendaciones, la métrica nDCG se eleva y se ajusta a los datos reales. También se observa que para la estrategia de agregación Least Misery (LMS) se obtienen valores de nDCG más altos en comparación con Most Pleasure (MPS).

MPS 2	# Recomendaciones			
	3	5	10	16
Prod2Vec Recommender	0.215543	0.243808	0.243808	0.412287
Factorization Personalized Markov Chains Recommender	0.155303	0.172277	0.244411	0.342082
Mixed Markov Chains Recommender	0.037921	0.060325	0.137921	0.235523
Popularity recommender	0.015476	0.039045	0.105169	0.213845
MPS 5	# Recomendaciones			
	3	5	10	16
Prod2Vec Recommender	0.146369	0.197873	0.28651	0.397862
Factorization Personalized Markov Chains Recommender	0.137471	0.167293	0.239027	0.334896
Mixed Markov Chains Recommender	0.077646	0.084158	0.077646	0.230733
Popularity recommender	0.011201	0.031336	0.099372	0.205876
MPS 8	# Recomendaciones			
	3	5	10	16
Factorization Personalized Markov Chains Recommender	0.171807	0.188602	0.249083	0.350724
Prod2Vec Recommender	0.073213	0.121383	0.245339	0.382333
Mixed Markov Chains Recommender	0.0844	0.080877	0.0844	0.220183
Popularity recommender	0.011663	0.027404	0.08974	0.200612

Figura 3.7: Mapa de calor de la métrica nDCG de los cuatro métodos de recomendación con la estrategia Most Pleasure (MPS) para 2, 4 y 8 miembros por grupo - MovieLens.

LMS 2	# Recomendaciones			
	3	5	10	16
Prod2Vec Recommender	0.351505	0.351985	0.349863	0.427052
Factorization Personalized Markov Chains Recommender	0.249132	0.24394	0.271631	0.35026
Mixed Markov Chains Recommender	0.104847	0.110321	0.104847	0.244531
Popularity recommender	0.010581	0.032725	0.110299	0.213441
LMS 5	# Recomendaciones			
	3	5	10	16
Prod2Vec Recommender	0.38033	0.362415	0.341625	0.427979
Factorization Personalized Markov Chains Recommender	0.245651	0.228077	0.255556	0.349913
Mixed Markov Chains Recommender	0.167363	0.142872	0.167363	0.25585
Popularity recommender	0.006766	0.025667	0.108639	0.209304
LMS 8	# Recomendaciones			
	3	5	10	16
Prod2Vec Recommender	0.369863	0.348174	0.321898	0.410807
Factorization Personalized Markov Chains Recommender	0.224295	0.204963	0.243098	0.343896
Mixed Markov Chains Recommender	0.147775	0.124351	0.147775	0.246753
Popularity recommender	0.003475	0.021563	0.108732	0.210024

Figura 3.8: Mapa de calor de la métrica nDCG de los cuatro métodos de recomendación con la estrategia Least Misery (LMS) para 2, 4 y 8 miembros por grupo - MovieLens.

MPS 2	# Recomendaciones			
	3	5	10	16
Prod2Vec Recommender	0.215543	0.243808	0.304529	0.412287
Factorization Personalized Markov Chains Recommender	0.155303	0.172277	0.244411	0.342082
Mixed Markov Chains Recommender	0.037921	0.060325	0.037921	0.235523
Popularity recommender	0.015476	0.039045	0.105169	0.213845
MPS 5	# Recomendaciones			
	3	5	10	16
Factorization Personalized Markov Chains Recommender	0.049376	0.083614	0.191507	0.295369
Prod2Vec Recommender	0.032673	0.059082	0.169406	0.289033
Mixed Markov Chains Recommender	0.023447	0.031967	0.023447	0.2522
Popularity recommender	0.018396	0.030373	0.133011	0.256196
MPS 8	# Recomendaciones			
	3	5	10	16
Factorization Personalized Markov Chains Recommender	0.064584	0.090375	0.171964	0.294176
Prod2Vec Recommender	0.038034	0.07646	0.179778	0.30919
Popularity recommender	0.013092	0.030849	0.126401	0.242396
Mixed Markov Chains Recommender	0.011596	0.0333	0.011596	0.241694

Figura 3.9: Mapa de calor de la métrica nDCG de los cuatro métodos de recomendación con la estrategia Most Pleasure (MPS) para 2, 4 y 8 miembros por grupo - Amazon.

3.5 PREGUNTAS DE INVESTIGACIÓN

3.5.1 Primera pregunta

Los SR basados en aspectos temporales generalmente toman en cuenta las acciones pasadas de un usuario para poder realizar las predicciones. Los SR basado en aspectos secuenciales a menudo se centran en el orden de los eventos en lugar de considerar el momento exacto de las interacciones del usuario. Esto permite crear una secuencia de calificacio-

LMS 2	# Recomendaciones			
	3	5	10	16
Prod2Vec Recommender	0.057869	0.081709	0.18477	0.295604
Factorization Personalized Markov Chains Recommender	0.032419	0.072686	0.161103	0.267676
Mixed Markov Chains Recommender	0.010411	0.034782	0.124113	0.238247
Popularity recommender	0.009791	0.028863	0.118694	0.23837
LMS 5	# Recomendaciones			
	3	5	10	16
Prod2Vec Recommender	0.067462	0.109386	0.195407	0.303225
Factorization Personalized Markov Chains Recommender	0.052931	0.09005	0.158146	0.260221
Mixed Markov Chains Recommender	0.013346	0.022747	0.133458	0.218564
Popularity recommender	0.010665	0.021323	0.108688	0.217556
LMS 8	# Recomendaciones			
	3	5	10	16
Prod2Vec Recommender	0.082614	0.125758	0.202821	0.301537
Factorization Personalized Markov Chains Recommender	0.057361	0.077241	0.155723	0.263655
Popularity recommender	0.006546	0.009863	0.10245	0.210307
Mixed Markov Chains Recommender	0.005051	0.0106	0.005051	0.2131

Figura 3.10: Mapa de calor de la métrica nDCG de los cuatro métodos de recomendación con la estrategia Least Misery (LMS) para 2, 4 y 8 miembros por grupo - Amazon.

nes que considera la temporalidad para poder realizar una predicción de las calificaciones futuras de un artículo objetivo.

Quadrana[1] realizó una investigación de varias estrategias de SR basadas en aspectos secuenciales. Estas estrategias se utilizaron para considerar aspectos temporales a largo plazo y obtener recomendaciones de calificaciones para periodos futuros de tiempo T entre 1 y 50 periodos.

La Figura 3.11 presenta los resultados informados por la métrica RMSE para los cuatro algoritmos basados en secuencias. Después de agrupar a los usuarios en grupos de 2, 5 y 8 miembros y luego aplicar Most Pleasure (MPS), se obtuvieron las siguientes cantidades de recomendaciones basadas en el tiempo: 1, 5, 10, 20 y 50. Es decir, dada una secuencia de calificaciones de grupo-artículo dada por c_{T1} y c_{T2} , la siguiente predicción de calificación sería \hat{c}_1 . Después de 5 períodos de tiempo, la predicción de calificación sería \hat{c}_5 . Después de 10 períodos de tiempo, la predicción de calificación sería \hat{c}_{10} , y así sucesivamente.

El gráfico superior de la figura 3.11 corresponde a los resultados del conjunto de datos de MovieLens. Aquí los valores de RMSE varían entre 0,15 y 0,50. Para el conjunto de datos de Amazon, el gráfico inferior de la figura 3.11 indica que los valores de RMSE oscilan entre 0,07 y 0,20. Los errores obtenidos al aplicar algoritmos basados en aspectos secuenciales sobre SRG para modelar aspectos temporales son, en general, bajos. Y los dos modelos, Popularity Recommender y Factorization Personalized Markov Chains Recommender presentan los mejores resultados con un error entre 0.07 y 0.24.

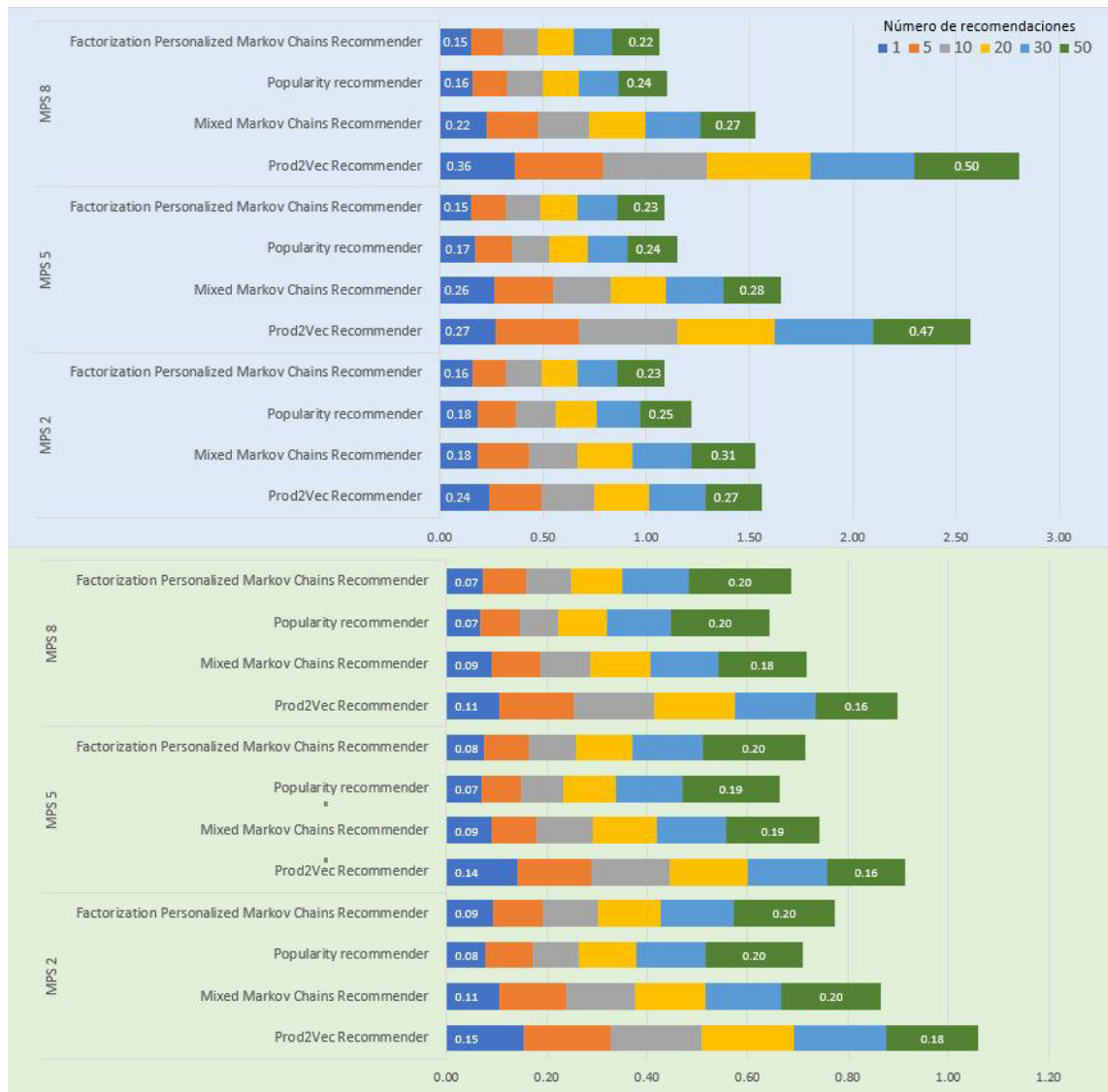


Figura 3.11: Barras horizontales de los valores de Métrica RMSE al aplicar MPS con tamaño de grupo 2, 4 y 8 en los conjuntos de datos MovieLens (superior) y Amazon (inferior), para todos los modelos de recomendación basados en aspectos secuenciales, para números de recomendaciones entre 1 a 50.

3.5.2 Segunda pregunta

La precisión de las recomendaciones proporcionadas por un SRG mide la diferencia entre la calificación pronosticada y la calificación real del grupo-artículo. Para medir la precisión de un modelo, se calcula la métrica RMSE, y cuanto menor sea el RMSE, más preciso será el modelo aplicado.

La figura 3.12 muestra las medidas de RMSE para los conjuntos de datos de MovieLens (gráfico superior en azul) y Amazon (gráfico inferior en verde). Siguiendo el eje horizontal, vemos cuatro bloques, uno para cada método de recomendación basado en secuencias.

Al aplicar cada método, se han proporcionado 1, 5 y 10 recomendaciones conscientes del tiempo. Los colores de las barras representan el número de miembros de los grupos. La evaluación para grupos de 2 integrantes está en verde, los resultados para grupos de 5 están en azul y para grupos de 8 los resultados están en amarillo.

Nuestros hallazgos muestran que al aumentar el tamaño del grupo, Popularity Recommender y Factorization Personalized Markov Chains Recommender aumentan su precisión (RMSE disminuye). Esto se aplica a ambos conjuntos de datos. El modelo de recomendación Mixed Markov Chains y el modelo de recomendación Prod2Vec no tienen un comportamiento específico dependiendo del número de miembros en un grupo. En particular, para el conjunto de datos de Amazon, los cuatro modelos mejoran su precisión si aumenta el tamaño de los grupos.

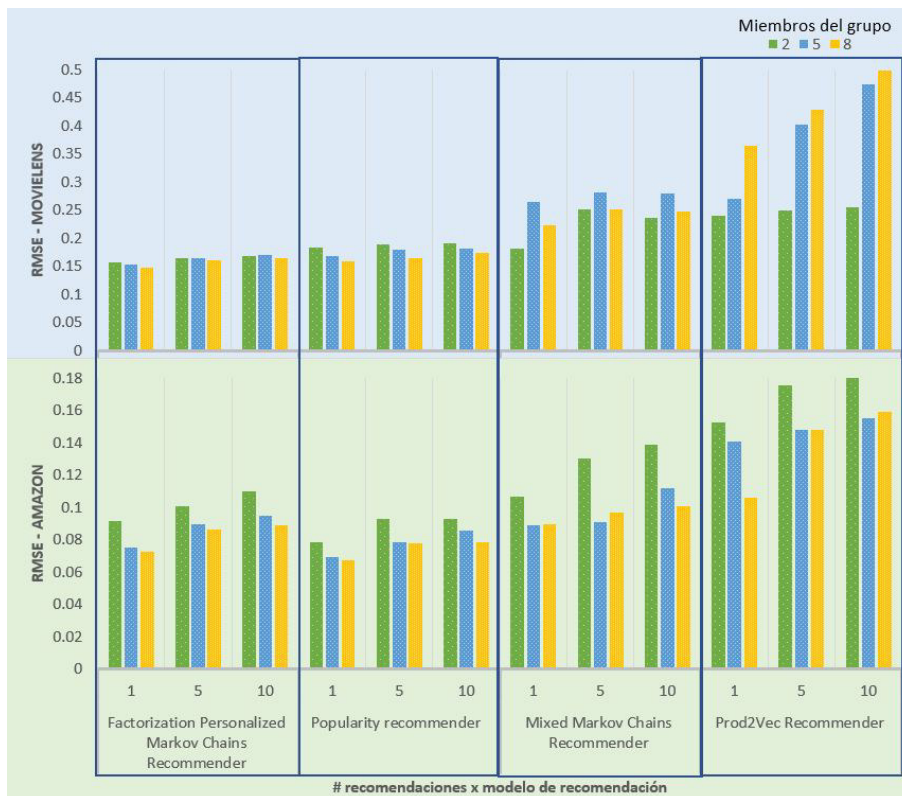


Figura 3.12: Gráficos de barras verticales utilizados para comparar las métricas RMSE con cada cantidad de miembros de grupos 2(verde), 5(azul) y 8(amarillo), para cada uno de los modelos de recomendación basado en aspectos secuenciales con 1, 5 y 10 recomendaciones.

3.5.3 Tercera pregunta

El modelo Factorization Personalized Markov Chains Recommender se utilizó para evaluar como afectan las estrategias de agregación LMS y MPS al rendimiento del mejor modelo

del SRG basado en aspectos temporales. Los resultados se presentan en la Figura 3.13 para los conjuntos de datos de MovieLens (azul) y Amazon (verde).

Al aplicar las estrategias de agregación MPS o LMS, se observa que cuando los períodos de tiempo de las recomendaciones futuras empleadas por el modelo se incrementan de 1 a 5 y luego a 10, el valor de RMSE también aumenta. Este comportamiento parece ser independiente del tamaño del grupo. Por lo tanto, el RMSE depende en mayor medida del número de períodos de tiempo establecidos al generar la predicción de calificación grupal que de la estrategia de agregación de preferencias utilizada. En particular, los valores de RMSE para grupos de 2, 5 y 8 miembros no varían cuando se trabaja con MovieLens y MPS. Sin embargo, al aplicar la estrategia de agregación LMS, RMSE aumenta notablemente para grupos de 8 miembros. Por otro lado, el RMSE para el dataset de Amazon, tanto para estrategias MPS como LMS, sigue el mismo patrón siendo MPS el que menos error genera por muy poca diferencia.

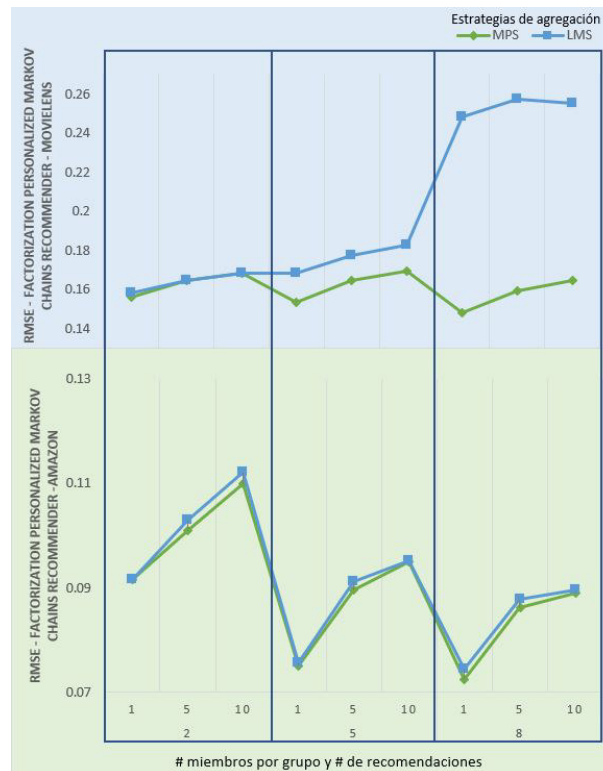


Figura 3.13: Comparación de las estrategias de agregación MPS y LMS, para grupos de usuarios de 2, 5 y 8 miembros con 1, 5 y 10 recomendaciones aplicando el modelo de recomendación Factorization Personalized Markov Chains Recommender.

3.5.4 Cuarta pregunta

Los modelos Popularity Recommender y Factorization Personalized Markov Chains Recommender han sido seleccionados para evaluar cómo la calidad de la recomendación se ve afectada con el tiempo.

Para el conjunto de datos de MovieLens (izquierda) y el conjunto de datos de Amazon (derecha), la Figura 3.14 muestra que el RMSE aumenta a medida que los períodos de tiempo avanzan hacia el futuro y se hacen recomendaciones sensibles al tiempo para los grupos. Este comportamiento es independiente del número de miembros del grupo (2 en barras verdes, 5 en barras azules y 8 en barras amarillas).

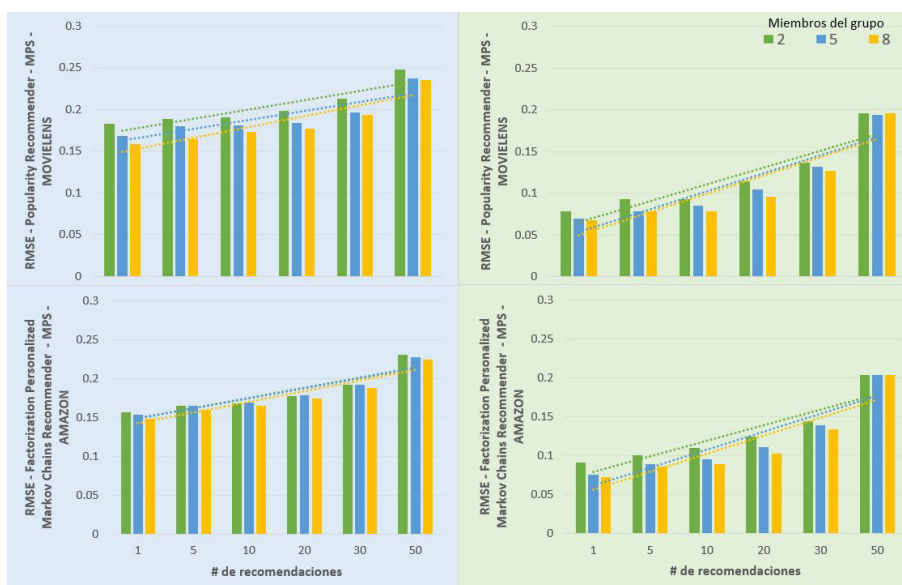


Figura 3.14: Tendencia de la métrica RMSE para los modelos de recomendación Popularity Recommender y Factorization Personalized Markov Chains Recommender con varias cantidades de recomendaciones, aplicando la estrategia de agregación MPS con 2, 5 y 8 miembros por grupo.

3.5.5 Quinta pregunta

Después de considerar diversos tamaños de grupos (de 2 a 8) y aplicar MPS y LMS para el modelado de grupos, obtuvimos las calificaciones de los elementos del grupo en T1, T2 y T3. Considerando la desviación estándar como una métrica de dispersión, la Figura 3.15 presenta los resultados tanto para MovieLens (gráfico superior) como para Amazon (gráfico inferior).

Para MovieLens, se puede ver que MPS tiene valores de desviación estándar más bajos en

un rango de 0,34 a 0,39 en comparación con LMS, que tiene valores entre 0,41 y 0,44 para la temporalidad $T1$. Para $T2$ con MPS hay valores entre 0,24 y 0,28, y con LMS entre 0,31 y 0,35. En la temporalidad $T3$ para MPS la desviación estándar está entre 0.20 y 0.24 y para LMS entre 0.26 y 0.34.

Para el conjunto de datos de Amazon, se puede ver que MPS tiene valores de desviación estándar más bajos en un rango de 0,076 a 0,098 en comparación con LMS, que tiene valores entre 0,078 y 0,098 para $T1$. Para la temporalidad $T2$ con MPS se tienen valores entre 0.072 y 0.094, y con LMS entre 0.074 y 0.096. En la temporalidad $T3$ para MPS la desviación estándar está entre 0.07 y 0.090 y para LMS entre 0.072 y 0.09.

Aquí, los intereses de los grupos a lo largo del tiempo tienden a variar menos. Por lo tanto, cuanto más antiguo sea un artículo dentro de la SRG, más precisa será la predicción de calificación grupal (en períodos de tiempo futuros) para ese elemento. Estos hallazgos nos dan una idea de los cambios de los intereses del grupo a lo largo del tiempo. En realidad, podemos decir que cuanto más interactúa el grupo consumiendo elementos a lo largo del tiempo, más similares se vuelven sus preferencias individuales.

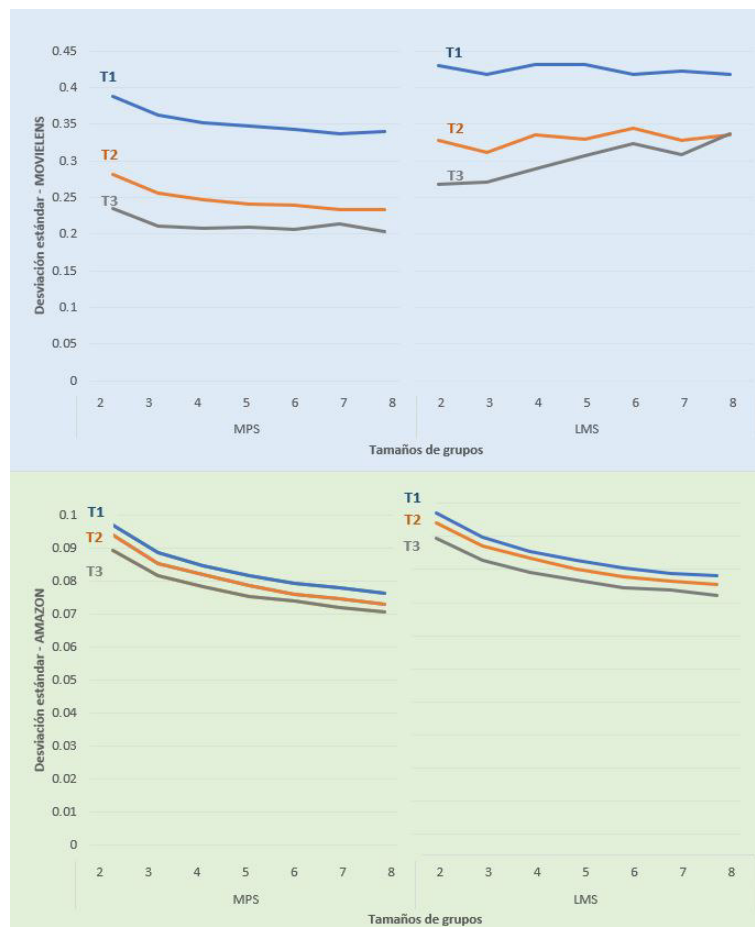


Figura 3.15: Desviación estándar con las estrategias de agregación Most Pleasure (MPS) y Least Misery (LMS) para T1, T2 y T3 de 2 a 8 miembros de grupos en los conjuntos de datos de MovieLens (superior) y Amazon (inferior).

4 CONCLUSIONES

Toda la configuración experimental se ejecutó con las mismas configuraciones para los dos conjuntos de datos, MovieLens y Amazon. Los métodos incluyen estrategias de agregación y sistemas de recomendación basados en secuencias para modelar aspectos temporales. Los SRG sensibles al tiempo se evaluaron utilizando las métricas RMSE y NDCG.

Con respecto a los métodos de agregación, la estrategia multiplicativa reportó el mayor error y Most Pleasure mostró ser la mejor para agregar las preferencias de los miembros del grupo. En términos de comparación, Most Pleasure y Least Misery se aplican para obtener los bloques de grupo T1, T2 y T3 para micro-perfiles de grupo. Esto permite la formación de las secuencias de calificación de grupo-artículo necesarias para luego aplicar los cuatro métodos de recomendación basados en secuencias. Para los cuatro métodos, las métricas de error varían según las calificaciones de predicción dadas por el tiempo T' . De hecho, los períodos de tiempo T' para las predicciones de calificación van desde \hat{r}_1 hasta \hat{r}_{50} .

Factorizing Personalized Markov Chains (FPMC) tiene el mejor comportamiento para capturar las preferencias de los grupos al modelar los aspectos temporales dados por T1 y T2.

Se observó que mediante el uso de estrategias de agregación MPS y LMS, independientemente del número de miembros del grupo o el número de recomendaciones conscientes del tiempo, los modelos Popularity Recommender y Factorization Personalized Markov Chains Recommender informaron los valores de RMSE más bajos. Por tanto, estos dos modelos pueden ser aplicados a pruebas en entornos de producción cuando el SRG necesite generar predicciones de ratings para grupos y disponer de información de ratings de consumos pasados.

Al estudiar la métrica NDCG, evaluamos si las predicciones de calificación del SRG sensible a la temporalidad generan una lista de predicciones similar a la lista de artículos consumidos por el grupo en los datos de prueba $T3$. Con NDCG, fue posible determinar que los

modelos de recomendación Prod2Vec Recommender y Factorization Personalized Markov Chains Recommender obtuvieron los valores más altos, por lo tanto, el mejor desempeño. Adicionalmente, se observó que, al aumentar el número de recomendaciones en la lista de predicciones, el valor de la métrica NDCG aumenta para cualquier estrategia de agregación o modelo de recomendación aplicado.

El uso de SR consciente de secuencias de última generación nos permitió modelar aspectos temporales en un SRG con buenos resultados. En general, cualquiera de los modelos basados en aspectos secuenciales puede utilizarse de manera efectiva. Sin embargo, el Factorization Personalized Markov Chains Recommender se comporta mejor según RMSE y NDCG.

Sobre la precisión de los modelos de recomendación basados en aspectos secuenciales respecto al tamaño del grupo. Se determinó que para los modelos Factorization Personalized Markov Chains Recommender y Popularity Recommender, al aumentar el tamaño de los miembros del grupo disminuye el RMSE, aumentando así la precisión. Y para los modelos Mixed Markov Chains Recommender y Prod2Vec Recommender, no hay un patrón general de comportamiento para los dos conjuntos de datos, ya que difieren a medida que aumenta el tamaño del grupo.

El cálculo de la desviación estándar mostró que los intereses de los miembros del grupo varían menos con el tiempo. Por lo tanto, se determina que las calificaciones otorgadas por los integrantes de un grupo a lo largo del tiempo tienden a ser similares, lo que tiende a presentar menor error. Esto se evidencia en un grupo de amigos donde, por lo general, a lo largo de los años, al consumir productos, tienen gustos similares luego de pasar un tiempo juntos.

Para trabajos futuros, proponemos adaptar el sistema para trabajar con períodos de tiempo más cortos para generar secuencias más largas con T1, T2, T3 y más. Los experimentos pueden considerar otros conjuntos de datos y, dependiendo de esto, ajustar el rango de tiempo según el dominio o contexto de uso del SRG.

5 REFERENCIAS BIBLIOGRÁFICAS

- [1] M. H. Mohamed, M. H. Khafagy y M. H. Ibrahim, «Recommender systems challenges and solutions survey», en *2019 International Conference on Innovative Trends in Computer Engineering (ITCE)*, 2019, págs. 149-155. DOI: 10.1109/ITCE.2019.8646645.
- [2] L. Boratto, S. Carta, A. Chessa, M. Agelli y M. L. Clemente, «Group recommendation with automatic identification of users communities», en *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, IEEE, vol. 3, 2009, págs. 547-550.
- [3] K. Ji, R. Sun, W. Shu y X. Li, «Next-song recommendation with temporal dynamics», *Knowledge-Based Systems*, vol. 88, págs. 134-143, 2015.
- [4] N. Koenigstein, G. Dror e Y. Koren, «Yahoo! music recommendations: Modeling music ratings with temporal dynamics and item taxonomy», en *Proceedings of the fifth ACM conference on Recommender systems*, oct. de 2011, págs. 165-172. DOI: 10.1145/2043932.2043964.
- [5] Y. Koren, «Collaborative filtering with temporal dynamics», en *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, págs. 447-456.
- [6] J. Ding, G. Yu, X. He, Y. Quan, Y. Li, T.-S. Chua, D. Jin y J. Yu, «Improving Implicit Recommender Systems with View Data», jul. de 2018, págs. 3343-3349. DOI: 10.24963/ijcai.2018/464.
- [7] C. Aggarwal, «Time- and Location-Sensitive Recommender Systems», en. mar. de 2016. DOI: 10.1007/978-3-319-29659-3_9.
- [8] J. You, Y. Wang, A. Pal, P. Eksombatchai, C. Rosenberg y J. Leskovec, «Hierarchical Temporal Convolutional Networks for Dynamic Recommender Systems», abr. de 2019.

- [9] S. Amer-Yahia, B. Omidvar-Tehrani, S. Basu Roy y N. Shabib, «Group Recommendation with Temporal Affinities», mar. de 2015. DOI: 10.5441/002/edbt.2015.37.
- [10] J. Chen, Y. Liu y D. Li, «Dynamic group recommendation with modified collaborative filtering and temporal factor», vol. 13, págs. 294-301, ene. de 2016.
- [11] B. Hidasi, A. Karatzoglou, L. Baltrunas y D. Tikk, «Session-based Recommendations with Recurrent Neural Networks», nov. de 2015.
- [12] M. Quadrana, A. Karatzoglou, B. Hidasi y P. Cremonesi, «Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks», ago. de 2017, págs. 130-137. DOI: 10.1145/3109859.3109896.
- [13] A. Oord, N. Kalchbrenner y K. Kavukcuoglu, «Pixel Recurrent Neural Networks», ene. de 2016.
- [14] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký y S. Khudanpur, «Recurrent neural network based language model», vol. 2, ene. de 2010, págs. 1045-1048.
- [15] K. Wei, J. Huang y S. Fu, «A Survey of E-Commerce Recommender Systems», jul. de 2007, págs. 1-5, ISBN: 1-4244-0885-7. DOI: 10.1109/ICSSSM.2007.4280214.
- [16] J. Tang y K. Wang, «Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding», sep. de 2018.
- [17] L. Boratto, S. Carta y G. Fenu, «Discovery and representation of the preferences of automatically detected groups: Exploiting the link between group modeling and clustering», *Future Generation Computer Systems*, vol. 64, págs. 165-174, 2016, ISSN: 0167-739X. DOI: <https://doi.org/10.1016/j.future.2015.10.007>. dirección: <https://www.sciencedirect.com/science/article/pii/S0167739X15003209>.
- [18] M. Quadrana, P. Cremonesi y D. Jannach, «Sequence-aware Recommender Systems», jul. de 2018, págs. 373-374. DOI: 10.1145/3209219.3209270.
- [19] C. Aggarwal, *Recommender Systems*. ene. de 2016, ISBN: 978-3-319-29657-9. DOI: 10.1007/978-3-319-29659-3.
- [20] H. Y. Ayat, E. Fadel y N. Akkari, «Time-aware recommender systems for e-commerce applications», 2020. DOI: 10.21474/IJAR01/10661.
- [21] E. Borba, I. Gasparini y D. Lichtnow, «The Use of Time Dimension in Recommender Systems for Learning», ene. de 2017, págs. 600-609. DOI: 10.5220/0006312606000609.

- [22] S. Rendle y L. Schmidt-Thieme, «Online updating regularized kernel matrix factorization models for large-scale recommender systems», ene. de 2008, págs. 251-258. DOI: 10.1145/1454008.1454047.
- [23] S. Chaudhary y A. C G, «Recommendation System for Big Data Software Using Popularity Model and Collaborative Filtering», en. ene. de 2020, págs. 551-559, ISBN: 978-981-15-0198-2. DOI: 10.1007/978-981-15-0199-9_47.
- [24] S. Rendle, C. Freudenthaler y L. Schmidt-Thieme, «Factorizing personalized Markov chains for next-basket recommendation», ene. de 2010, págs. 811-820. DOI: 10.1145/1772690.1772773.
- [25] M. Grbovic, V. Radosavljevic, N. Djuric, N. Bhamidipati, J. Savla, V. Bhagwan y D. Sharp, «E-commerce in Your Inbox: Product Recommendations at Scale», jun. de 2016. DOI: 10.1145/2783258.2788627..
- [26] J. Van Aken, «Management Research as a Design Science: Articulating the Research Products of Mode 2 Knowledge Production in Management», *British Journal of Management*, vol. 16, págs. 19-36, feb. de 2005. DOI: 10.1111/j.1467-8551.2005.00437.x.
- [27] V. Vaishnavi y B. Kuechler, «Design Science Research in Information Systems», *Association for Information Systems*, ene. de 2004.
- [28] B. Kuechler y V. Vaishnavi, «On theory development in design science research: Anatomy of a research project», *European Journal of Information Systems*, vol. 17, oct. de 2008. DOI: 10.1057/ejis.2008.40.
- [29] A. Hevner, S. March, J. Park y S. Ram, *Design Science in Information Systems Research*, sep. de 2013. DOI: 10.2307/25148625.
- [30] J. Verma, B. Patel y A. Patel, «Big Data Analysis: Recommendation System with Hadoop Framework», *Proceedings - 2015 IEEE International Conference on Computational Intelligence and Communication Technology, CICT 2015*, págs. 92-97, abr. de 2015. DOI: 10.1109/CICT.2015.86.
- [31] F. Ricci, L. Rokach, B. Shapira y P. Kantor, *Recommender Systems Handbook*. ene. de 2011. DOI: 10.1007/978-0-387-85820-3.
- [32] A. Felfernig, G. Friedrich, D. Jannach y M. Zanker, «Developing Constraint-based Recommenders», en. ene. de 2011, págs. 187-215. DOI: 10.1007/978-0-387-85820-3_6.

- [33] L. Xiang, Q. Yuan, S. Zhao, L. Chen, X. Zhang, Q. Yang y J. Sun, «Temporal recommendation on graphs via long- and short-term preference fusion», jul. de 2010, págs. 723-732. DOI: 10.1145/1835804.1835896.
- [34] Y. Soullard, C. Ruffino y T. Paquet, «CTCModel: a Keras Model for Connectionist Temporal Classification», ene. de 2019.
- [35] A. Micarelli y F. Sciarrone, «Anatomy and Empirical Evaluation of an Adaptive Web-Based Information Filtering System», *User Model. User-Adapt. Interact.*, vol. 14, págs. 159-200, jun. de 2004. DOI: 10.1023/B:USER.0000028981.43614.94.
- [36] T. De Pessemier, S. Dooms y L. Martens, «Comparison of group recommendation algorithms», *Multimedia Tools and Applications*, vol. 72, oct. de 2014. DOI: 10.1007/s11042-013-1563-0.
- [37] H. Tercan, C. Bitter, T. Bodnar, P. Meisen y T. Meisen, «Evaluating a Session-based Recommender System using Prod2vec in a Commercial Application», ene. de 2021, págs. 610-617. DOI: 10.5220/0010400706100617.
- [38] P. G. Campos, F. Díez e I. Cantador, «Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols», *User Modeling and User-Adapted Interaction*, vol. 24, págs. 67-119, 2012.
- [39] L. Boratto y S. Carta, «Using Collaborative Filtering to Overcome the Curse of Dimensionality when Clustering Users in a Group Recommender System», vol. 2, ene. de 2014. DOI: 10.5220/0004865005640572.
- [40] M. Z. Ahmed, A. Singh, A. Paul, S. Ghosh y A. Chaudhuri, «Amazon Product Recommendation System», *IJARCCCE*, vol. 11, mar. de 2022. DOI: 10.17148/IJARCCCE.2022.11356.
- [41] P. Castells, N. J. Hurley y S. Vargas, «Novelty and diversity in recommender systems», en *Recommender systems handbook*, Boston, MA: Springer, 2015, págs. 881-918, ISBN: 978-1-4899-7636-9. DOI: 10.1007/978-1-4899-7637-6_26.
- [42] J. Herlocker, J. Konstan, A. Borchers y J. Riedl, «An Algorithmic Framework for Performing Collaborative Filtering», *ACM SIGIR Forum*, vol. 51, págs. 227-234, ago. de 2017. DOI: 10.1145/3130348.3130372.
- [43] A. Felfernig, L. Boratto, M. Stettinger y M. Tkalčič, «Evaluating Group Recommender Systems», en mar. de 2018, págs. 59-71, ISBN: 978-3-319-75066-8. DOI: 10.1007/978-3-319-75067-5_3.