UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DSIC
DEPARTAMENT DE SISTEMES INFORMÀTICS I COMPUTACIÓ

Departamento de Sistemas Informáticos y Computación
Universitat Politècnica de València

Un sistema de recomendación basado en perfiles generados por agrupamiento y asociaciones

Trabajo de fin de máster

Máster Universitario en Ingeniería y Tecnología de Sistemas Software

Alumno: **Enio Walid Ghobar**

Tutora: **María José Ramírez Quintana**

Curso: 2016-2017

Resumen

"Este artículo podría gustarle", "Personas que compraron este artículo también compraron …" o "Esto también podría interesarle" son frases que están cada vez más presentes en la actividad cotidiana de los usuarios consumidores de productos y servicios en tiendas virtuales y otros servicios. Esto se debe a que los sistemas de recomendación se han consolidado como fuerte tendencia para el crecimiento del comercio digital en los últimos años, sobre todo en relación con el *big data*.

Los avances y el abaratamiento de las tecnologías permitieron que muchas empresas construyesen sus entornos virtuales para complementar a su tienda física, o que fuesen creadas en plan puramente virtual y, resulta que, en ambos casos, las interacciones de los clientes con sus entornos virtuales han generado una cantidad enorme de datos acerca de sus preferencias, como por ejemplo, productos buscados y/o comprados, películas o canciones reproducidas o marcadas como favoritas, o incluso eventos capturados por sensores y registrados en una base de datos, dado que la Internet de las Cosas es un agente presente y en plena ascensión que registra datos sobre las personas y sus elecciones de forma involuntaria.

Este trabajo tiene como objetivo principal usar los datos ya existentes sobre las preferencias de los usuarios, y aplicar técnicas de *Machine Learning* para desarrollar un sistema para hacer recomendaciones, sugiriendo nuevos ítems ajustados a los gustos de los usuarios a partir de perfiles de usuario o de productos generados con este fin. Actualmente, diversos proveedores ya ofrecen a las empresas soluciones *SaaS* (*Software as a Service*) con las que integrar recomendaciones personalizadas en su proyecto comercial. Entre los más conocidos se encuentran BrainSINS[1], Barilliance[2] o Certona[3]. A diferencia del servicio ofrecido por estos proveedores, nuestra propuesta es muy poco invasiva en el sentido que usa únicamente los datos del historial de elecciones de los clientes, sin que sean necesarios otros datos sobre los mismos. Para ello, proponemos usar un método híbrido que combina los filtrados colaborativos e ítem a ítem. Además de la seguridad por el anonimato de los clientes y por no demandar la difícil tarea de obtener calificaciones de los ítems por parte de los clientes, nuestra aproximación busca ser capaz de hacer recomendaciones sin requerir muchos datos de entrenamiento, lo que automáticamente la convierte en una solución sencilla y de bajo coste, tanto en el desarrollo como en el mantenimiento del sistema de recomendaciones, que puede

[1] http://www.brainsins.com/es/
[2] https://www.barilliance.com/es/
[3] http://www.certona.com/

ser potencialmente interesante para empresas que todavía permanecen excluidas del mundo de las recomendaciones.

Palabras clave: minería de datos, sistemas de recomendación, agrupamiento, reglas de asociación.

Resum

"Aquest article podria agradar-li", "Persones que van comprar aquest article també van comprar …" o "Açò també podria interessar-li" són frases que estan cada vegada més presents en l'activitat quotidiana dels usuaris consumidors de productes i serveis en tendes virtuals i altres serveis. Açò es deu al fet que els sistemes de recomanació s'han consolidat com a forta tendència per al creixement del comerç digital en els últims anys, sobretot en relació amb el *big data*.

Els avanços i l'abaratiment de les tecnologies van permetre que moltes empreses construïren els seus entorns virtuals per a complementar a la seua tenda física, o que anaren creades en pla purament virtual i, resulta que, en tots dos casos, les interaccions dels clients amb els seus entorns virtuals han generat una quantitat enorme de dades sobre les seues preferències, com per exemple, productes cercats i/o comprats, pel·lícules o cançons reproduïdes o marcades com a favorites, o fins i tot esdeveniments capturats per sensors i registrats en una base de dades, atès que la Internet de les Coses és un agent present i en plena ascensió que registra dades sobre les persones i les seues eleccions de forma involuntària.

Aquest treball té com a objectiu principal usar les dades ja existents sobre les preferències dels usuaris, i aplicar tècniques de *Machine Learning* per a desenvolupar un sistema per a fer recomanacions, suggerint nous ítems ajustats als gustos dels usuaris a partir de perfils d'usuari o de productes generats a aquest efecte. Actualment, diversos proveïdors ja ofereixen a les empreses soluciones *SaaS (Program as a Service)* amb les quals integrar recomanacions personalitzades en el seu projecte comercial. Entre els més coneguts es troben BrainSINS[4], Barilliance[5] o Certona[6]. A diferència del servei oferit per aquests proveïdors, la nostra proposta és molt poc invasiva en el sentit que usa únicament les dades de l'historial d'eleccions dels clients, sense que siguen necessaris altres dades sobre els mateixos. Per a açò, proposem usar un mètode híbrid que combina els filtrats col·laboratius i ítem a ítem. A més de

[4] http://www.brainsins.com/es/
[5] https://www.barilliance.com/es/
[6] http://www.certona.com/

la seguretat per l'anonimat dels clients i per no demandar la difícil tasca d'obtenir qualificacions dels ítems per part dels clients, la nostra aproximació cerca ser capaç de fer recomanacions sense requerir moltes dades d'entrenament, la qual cosa automàticament la converteix en una solució senzilla i de baix cost, tant en el desenvolupament com en el manteniment del sistema de recomanacions, que pot ser potencialment interessant per a empreses que encara romanen excloses del món de les recomanacions.

Paraules clau: mineria de dades, sistemes de recomanació, agrupament, regles d'associació.

Abstract

"This article might interest you", "People who bought this article, also bought ..." or "This might also interest you" are phrases that are increasingly present in the daily activity of users that consume products and services in virtual stores and other services. The reason for it is that recommendation systems have been consolidated as a strong trend for the growth of digital commerce in recent years, especially with regards to big data.

Advances and the cheapening of technologies have allowed many companies to build their virtual environments to complement their physical store, or they were even created in a purely virtual plan and, as a result, in both cases, the interactions of customers with their virtual environments have stored a huge amount of data about their preferences, such as searched and / or purchased products, movies or songs played or marked as favorites, or even events captured by sensors and registered in a database, since the Internet Things is an agent present and in great ascension that records data about people and their choices involuntarily.

The main objective of this work is to use the existing data about users' preferences, and apply Machine Learning techniques to develop a system to make recommendations, suggesting new items adjusted to users' tastes based on users' profiles or products generated for this purpose. Currently, several service providers already offer SaaS solutions (Software as a Service) to companies in order to integrate personalized recommendations into their commercial project. Among the best known are BrainSINS[7], Barilliance[8] or Certona[9]. Unlike the service offered by these providers, our proposal is very non-invasive in the sense that it only uses the data of the clients' election history, without other data about them being necessary. To do this, we propose using a hybrid method that combines collaborative and item-to-item

[7] http://www.brainsins.com/es/
[8] https://www.barilliance.com/es/
[9] http://www.certona.com/

filtering. Besides security due to customers' anonymity and for not demanding the challenging task of obtaining items ratings through customers, our approach aims to be able to make recommendations without requiring much training data, which automatically makes it a simple and low-cost solution, both for developing and maintaining the recommendation system, which can be potentially interesting for companies that remain excluded from the world of recommendations.

Key words: data mining, recommendation systems, clustering, association rules.

Agradecimientos

Agradezco a mi tutora, profesora María José Ramírez Quintana, por haber acreditado en el potencial de la propuesta de este TFM, por aceptar orientarme para su realización y por prestarme todo soporte que necesité en los retos y dificultades encontrados desde la creación del prototipo de recomendación hasta las últimas revisiones de la memoria. Además de todo conocimiento sobre *Data Science* que me ha enseñado en la asignatura del Máster, las productivas reuniones que hicimos durante la elaboración del TFM me han ofrecido una oportunidad valiosa para profundizarme en las técnicas y algoritmos de minería de datos utilizados en nuestro prototipo, y para percibir la dinámica de creación de una solución para un problema de este tipo.

También agradezco por el apoyo de mi mujer Cristina Moraes y de mi hijo Felipe Ghobar por las ideas sugeridas para el trabajo, por el envío de noticias sobre los temas de *Data Science* y recomendaciones, y por la comprensión en los momentos que estuve ausente para trabajar en las actividades necesarias para la conclusión de este TFM.

Por fin, también soy grato también a mi hermana Lilian Ghobar que, entre sus actividades laborales y familiares, ha dedicado su tiempo para aclararme dudas sobre el castellano para que yo pudiera escribir esta memoria de modo más correcto y claro posible.

Índice

Lista de Figuras

Lista de Tablas

1. Introducción

Son notorias e incuestionables la importancia y popularidad que han ganado los sistemas de recomendación en variados ámbitos. Los motivos para este éxito son diversos, pero un factor que es común a todos los entornos en que se los aplican es la **generación de valor** para los usuarios o clientes de la actividad en cuestión. En un contexto mundial donde la cantidad de opciones disponibles aumenta cada vez más, la dificultad para elegir lo que a uno le gusta o le interesa crece en la misma proporción. Como solución escalable a esta tendencia creciente, los sistemas de recomendación ofrecen alternativas de filtrados y personalización de sugerencias a los usuarios y clientes expuestos en el contexto descrito.

La importancia y la popularidad mencionadas de estos sistemas se justifican a través de un cambio en el escenario global producido por Internet, como es bien descrito en el blog *Mente Errabunda*[10]: "Antes de la llegada de Internet, un consumidor de cualquier tipo de producto tenía un acceso limitado a la información relacionada tanto con el producto en sí como con otras posibles opciones. La publicidad se convertía así en prácticamente la única forma de dar a conocer un producto, y el problema del usuario era cómo conseguir una información veraz. En el caso de productos culturales como la música, la radio o las revistas especializadas actuaban como únicos difusores de lo nuevo. Ahora la situación se ha invertido totalmente. De la escasez de información se ha pasado a la saturación. De disponer de algunas estanterías con discos compactos y videos en el centro comercial más cercano se ha pasado a tener acceso a una cantidad inagotable de creaciones culturales en tiendas online o en redes persona a persona. Ahora el problema se ha tornado en cómo separar lo que se quiere de lo que no se quiere encontrar."

Sin embargo, aparte del contexto de consumo descrito, el primer sistema de recomendación surgió en un contexto de filtrado colaborativo de **noticias**: "Según el investigador P. Resnick y sus colegas, en el artículo escrito el año 1995 relacionado con 'GroupLens: Una arquitectura abierta para el filtrado colaborativo de noticias en la red', a principios de la década de los años 1990 empezaron a surgir dentro de los servicios de grupos de noticias, los servicios de filtrado de noticias que permitían a su comunidad de usuarios acceder exclusivamente a aquellas noticias que potencialmente podían ser de su interés. No obstante, el primer sistema de recomendación que apareció fue el llamado 'Tapestry',

[10] Artículo "Sistemas de recomendación" de 23 de abril de 2012, http://menteerrabunda.blogspot.com.es/2012/04/sistemas-de-recomendacion.html

desarrollado por XeroX PARC. Tapestry es un sistema que permite almacenar el conocimiento de los usuarios sobre los artículos o noticias que éstos han leído y posteriormente es utilizado por otros usuarios que aún no han leído el artículo o noticia, para establecer si la información del documento es relevante o no. En un principio este tipo de sistemas fue adoptado con el nombre de filtro colaborativo dado que permite que los usuarios creen filtros a través de sus ítems de interés, en el caso de Tapestry artículos o noticias, y colaborativo pues los usuarios añaden las anotaciones con las opiniones sobre los documentos. Las opiniones añadidas pueden ser utilizadas para las búsquedas de otros usuarios. Los investigadores Resnick y Varian, en el artículo publicado el año 1997 sobre 'sistemas de recomendación', proponen llamar a este tipo de sistemas con el nombre de 'sistemas de recomendación', dado que por esa fecha estos sistemas no sólo se limitaban al filtro de información y habían aparecido nuevos sistemas en el que no se utilizaban las opiniones de otros usuarios."[11]

Todos estos ejemplos ponen de manifiesto que, independientemente del contexto de aplicación, sea para noticias, películas o productos de una tienda, y de la información que se use (proveniente de otros usuarios o de los ítems/productos que se ofertan), los sistemas de recomendación son algoritmos estadísticos o de *Machine Learning* que tienen como principal objetivo ayudar a los usuarios a seleccionar lo que necesitan y les interesan, frente a un escenario de sobrecarga de opciones ofertadas, lo que facilita y agiliza el proceso de decisión, y produce una consecuencia casi tan importante como el hecho de vender para un tienda/servicio, y que es la fidelización de los clientes/usuarios.

1.1 Motivación

"We are our choices". Esta sencilla y concisa afirmación hecha por el filósofo francés Jean Paul Sartre (1905-1980) puede llevarnos a una profunda reflexión sobre quiénes somos, donde cada decisión que tomamos forma parte de la persona en la que nos convertimos y de los gustos que adquirimos. En cierto modo, podemos mapear el perfil de un individuo, e incluso decir que lo conocemos, si sabemos algo sobre sus elecciones pasadas, considerando también su evolución temporal, una vez que las personas están en constante transformación, refinando sus decisiones y adquiriendo nuevos gustos. Al considerar este hecho bajo la perspectiva de la minería de datos, se observa que hay una gran cantidad de información, que se refiere a elecciones de usuarios, almacenadas y disponibles en las bases de datos de diversas entidades,

[11] Artículo "Sistemas de recomendación" de 23 de abril de 2012, http://menteerrabunda.blogspot.com.es/2012/04/sistemas-de-recomendacion.html

con gran potencial para transformarse en conocimiento sobre los perfiles de sus usuarios, y convertirse así en un rico activo para ser explorado.

La motivación de este trabajo es estudiar formas alternativas de extraer el valor existente en los datos sobre la actividad y preferencias de los usuarios (perfiles) para así poder hacer recomendaciones relevantes a sus perfiles. Recomendar, sugerir e influir en las opiniones de los usuarios forma parte de la nueva realidad que extrae valor de las informaciones de modo más potente a través de los recursos de *Machine Learning*. Esta tendencia se refleja de forma muy clara por la frase de Chris Anderson, autor del blog *The Long Tail*[12]: "*We are leaving the information age and entering the recommendation age*". Así como una empresa conoce a sus clientes y sus gustos, el reconocimiento automático de sus perfiles seguido de recomendaciones adecuadas a los mismos permitiría también personalizar la comunicación con los mismos, incluso de forma online, tal y como haría un empleado de forma presencial.

La ventaja, y reto a la vez, presentada en nuestra propuesta, y que la hace diferente de otros modelos de recomendación actuales, es la cantidad reducida de datos que necesitan ser suministrados al sistema para su entrenamiento y utilización. Con respeto a los datos de los usuarios, no se requerirá ningún otro dato adicional aparte de sus elecciones pasadas (productos comprados, canciones oídas, noticias leídas, etc.). Con esto, el suministro de datos se centra exclusivamente en los datos que seguramente las entidades ya poseen, sin que sean necesarios datos personales sobre los usuarios, ni que éstos califiquen los ítems de acuerdo con sus preferencias. Aunque las entidades puedan disponer de los datos personales sobre sus usuarios, nuestra propuesta trabaja bajo el principio del anonimato, lo que la convierte en un sistema de recomendación seguro desde el punto de vista de la privacidad. Obtener valoraciones sobre los ítems del catálogo, o bien un sencillo registro de "me gusta" o "no me gusta", involucra cooperación por parte de los usuarios y, a veces, esfuerzo de las entidades en la creación de encuestas o, incluso, oferta de "premios" a los que la contesten. Otra ventaja de no depender de *feedbacks* de los usuarios para producir recomendaciones está relacionada con el problema generado por el *Cold Start* o "comienzo frio", donde nuevos usuarios o nuevos ítems son introducidos al sistema y de los que no se dispone de historial en base al cual hacer una recomendación. Este problema será brevemente analizado en el capítulo de conclusiones. Además de los datos de clientes ya descritos, también se requerirán datos sobre los ítems del catálogo asociados a sus respectivos tipos, pero que también son datos sencillos y disponibles

[12] "Estamos dejando la era de la información y entrando en la era de la recomendación", Chris Anderson, autor del blog http://www.longtail.com/

en cualquier entidad. En resumen, con nuestra aproximación, el sistema de recomendación generado tendrá las siguientes características:

- **Seguridad,** respecto de los datos sensibles sobre los usuarios
- **Practicidad,** evitándose la tarea de obtención de datos adicionales (a través de cuestionarios, etc.)
- **Sencillez,** en la tarea de suministrar los datos para el entrenamiento del sistema

Como consecuencia directa de las ventajas de practicidad y sencillez mencionadas, se esperan también que haya beneficios en cuanto al tiempo y recursos necesarios para desarrollar y mantener un sistema de recomendación basado en nuestra propuesta.

Adicionalmente, este estudio intenta explorar la búsqueda por ítems frecuentes a través de reglas de asociación en perfiles formados por grupos de usuarios con elecciones similares, incluso aplicando este mismo proceso de forma iterativa a subgrupos formados por perfiles con mayor variabilidad de gustos entre los usuarios, lo que entendemos ser un enfoque distinto al de otros estudios en la literatura que utilizan esta técnica [12, 13,14].

1.2 Objetivo

Este trabajo de tesis tiene como objetivo principal crear un sistema para hacer recomendaciones basado en un perfil de usuario que solo tiene en cuenta el historial de los ítems elegidos por el mismo, sin usar ningún otro dato adicional de carácter personal.

Este objetivo general se desglosa en los siguientes objetivos específicos (OE) para producir las recomendaciones:

OE1. Establecer la representación necesaria de los usuarios para hacer posible el análisis según los algoritmos utilizados.

OE2. Crear un perfil principal, usando técnicas de agrupamiento para descubrir grupos de usuarios similares en cuanto al historial de ítems accedidos en el pasado, para, a continuación, generar reglas de asociación en cada grupo que serán usadas para determinar las recomendaciones que se hacen a los usuarios.

OE3. Crear un perfil secundario, aplicando la misma estrategia seguida para el perfil principal, pero únicamente en aquellos grupos para los que no se hayan generado reglas de asociación durante la construcción del perfil principal.

OE4. Definir un procedimiento de recomendación alternativo para tratar los casos que no recibieron recomendaciones por ninguno de los perfiles anteriores, así como para los casos de “comienzo frio".

El alcance de este trabajo incluye la descripción de la aproximación y el desarrollo de un prototipo basado en el lenguaje R que nos permitirá evaluar nuestra propuesta con 2 conjuntos de datos.

1.3 Estructura del TFM

El resto del trabajo se organiza como sigue:

- El segundo capítulo introduce conceptos básicos de los sistemas de recomendación, así como una revisión de las técnicas más utilizadas en los motores de recomendación. Como ejemplo (y a modo de referencia) se describen las particularidades y técnicas empleadas en los sistemas de dos empresas de gran popularidad, Amazon y Netflix.
- El tercer capítulo describe las técnicas de minería de datos en la que se basa nuestra aproximación, *Clustering* (Agrupamiento) utilizándose el algoritmo de *k-means*, y las Reglas de Asociación que utilizan el algoritmo de Apriori.
- El cuarto capítulo describe nuestra aproximación para hacer las recomendaciones, así como su evaluación experimental con dos conjuntos de datos generalmente utilizados para la evaluación de sistemas de recomendación.
- El último capítulo presenta las conclusiones sobre la propuesta presentada y las extensiones posibles de la misma para trabajos futuros.

2. Sistemas de recomendación

Un sistema para hacer recomendaciones se define como un sistema que analiza y procesa información histórica de los usuarios (edad, compras previas, calificaciones), de los productos o de los contenidos (marcas, modelos, precios, contenidos similares) y la transforma en **conocimiento accionable**, es decir, **determina qué producto puede ser potencialmente interesante para el usuario.**[13]

Hay varios tipos de técnicas de filtrados para hacer recomendaciones, con diferentes aspectos que deben ser evaluados dependiendo del contexto o de lo que se proponen las recomendaciones.

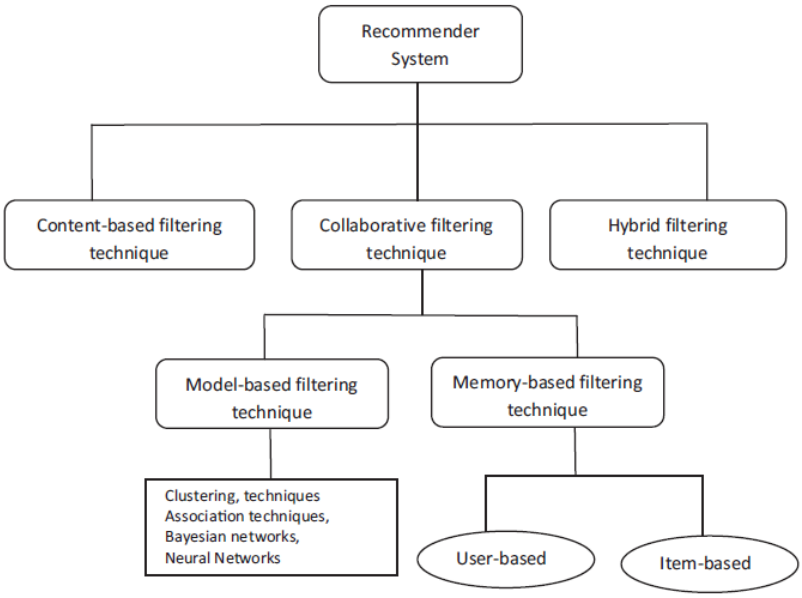


Figura 1 - Técnicas de Recomendación

Los tipos de técnicas son mostrados en la figura 1[4], y descritos como sigue:

- ***Content-based filtering***: También conocida como recomendación ítem-ítem. Esta técnica es la más exitosa cuando se busca recomendar documentos como páginas web, publicaciones y noticias. Las recomendaciones son producidas en base al perfil del usuario el cual es generado usando únicamente las características extraídas de los contenidos que él mismo ha evaluado en el pasado. Los artículos más similares a los positivamente evaluados son los que se recomiendan. La ventaja de este filtrado es que solo necesita de información interna, es decir, sobre los ítems del catálogo, sin

[13] Blog de la empresa CleverData, artículo "Sistemas de recomendación de contenido con Machine Learning", http://cleverdata.io/sistemas-recomendacion-machine-learning/

necesidad de personalizarlo al usuario que interactúa, lo que puede ser interesante para casos en que no haya historial del usuario a evaluar, o éste sea insuficiente. También puede ser ventajoso cuando no hay muchos usuarios registrados, lo que perjudica la formación de perfiles colaborativos. La desventaja es que la propuesta de valor es idéntica para todos los usuarios, perdiendo la posibilidad de personalización, además de que los ítems deben estar valorados por los usuarios.

- ***Collaborative filtering*:** El filtrado colaborativo es una técnica para recomendar contenidos que no pueden ser descritos por metadatos de forma fácil y adecuada. La técnica del filtrado colaborativo funciona construyendo una matriz con las preferencias de los usuarios. Entonces se comparan los usuarios calculándose las similitudes entre sus perfiles (que corresponden a filas de la matriz). Un usuario obtiene recomendaciones de aquellos artículos que no ha calificado antes, pero que ya estaban positivamente calificados por los usuarios más similares a él. La técnica del filtrado colaborativo se puede dividir en dos categorías [4]:
    - *Memory based techniques:* Esta categoría se basa en datos como votos, "me gusta", cliques, y establece correlaciones entre los usuarios del mismo gusto, simplemente comparando los vectores de elecciones entre estos usuarios, calculando el respectivo umbral de similitudes entre ellos para producir la recomendación. Las ventajas de esta categoría es que es fácil de implementar y se adapta bien a los elementos correlacionados, pero como contrapartida depende de las calificaciones humanas y presenta correlaciones sesgadas cuando los datos son escasos.
    - *Model-based techniques:* Esta categoría utiliza valoraciones de los usuarios sobre los ítems para entrenar un modelo a través de aprendizaje automático y generar las recomendaciones. La ventaja de esta categoría es que se basa en datos pre-calculados, lo que ofrece mejor rendimiento que la de *memory-based*, además de presentar resultados de recomendación similares. Pero, por contra, presenta un desarrollo más caro. Ejemplos de tareas de minería de datos utilizados en esta categoría son la Asociación, el *Clustering* y la Regresión.
- ***Hybrid filtering***: La técnica del filtrado híbrido combina diferentes técnicas de recomendación con el fin de obtener una mejor optimización del sistema para evitar algunas limitaciones y problemas de los sistemas de recomendación basados en un único filtrado. La idea detrás de las técnicas híbridas es que una combinación de técnicas proporcionará recomendaciones más precisas y efectivas que una solo técnica

de filtrado, ya que los inconvenientes de un algoritmo pueden ser superados por el otro. El uso de técnicas de recomendación múltiples puede suprimir las debilidades de una técnica individual en un modelo combinado. La combinación de enfoques puede realizarse de cualquiera de las siguientes maneras: implementación separada de algoritmos y combinación del resultado, utilizando un filtrado basado en contenido en un enfoque colaborativo, utilizando un filtrado colaborativo en el enfoque basado en contenido, o creándose un sistema de recomendación unificado con ambos enfoques.[4]

A continuación, se describen los sistemas de recomendación de dos grandes empresas con amplitud y fama mundial, Amazon y Netflix. Debido a la gran complejidad involucrada en el proceso de recomendar, veremos que ambas empresas combinan técnicas distintas para hacer recomendaciones, mostrando que la forma híbrida ofrece la versatilidad y flexibilidad exigidas a esta tarea.

2.1. Recomendaciones de Amazon

Amazon.com es una tienda virtual de amplitud global, con millones de cliente y millones de productos distintos en su catálogo, y utiliza ampliamente los algoritmos de recomendación para personalizar su sitio web a los intereses de cada cliente. Además, requiere alta calidad en sus recomendaciones, mostrando productos de relevancia a los clientes, sea basado en compras anteriores o en los ítems ya elegidos en el carrito. Esta demanda hace que Amazon necesite generar recomendaciones a cada interacción del cliente, es decir, un sistema que compute en tiempo real y de forma rápida recomendaciones para la interacción en marcha (ver figura 2 para un ejemplo de recomendación). [6]

El hecho de que Amazon posee millones de clientes, un gran catálogo de productos y mil millones de registros de compras es un escenario de gran éxito para una tienda virtual. Sin embargo, en el caso de Amazon, la grandeza de su éxito es directamente proporcional a la grandeza de su complejidad cuando se trata de recomendar basado en todo este volumen de datos, añadiéndose el hecho de que también se computan las calificaciones de los productos por sus clientes en este proceso.

Figura 2 - Recomendaciones personalizadas de Amazon [8]

Los enfoques más usuales para sistemas de recomendación, como el filtrado colaborativo, el filtrado basado en contenido o incluso cambiándose a un problema de clasificación a través modelos de *cluster*, no son suficientes para solucionar su problema de escalabilidad. Entre los motivos destacamos la baja posibilidad de computación offline o una computación online costosa o incluso impracticable. Una forma de resolver este problema es reducir la dimensionalidad, efectuando un particionamiento de los datos o muestras, pero esto da lugar a una baja calidad de las recomendaciones.

La solución de Amazon fue la creación de su propio sistema para generar recomendaciones, llamado *item-to-item collaborative filtering* (filtrado colaborativo de ítem a ítem), lo cual se escala a grandes conjuntos de datos y produce recomendaciones de alta calidad en tiempo real.

En lugar de hacer coincidir el usuario con clientes similares, el filtrado colaborativo de ítem a ítem compara cada uno de los artículos comprados y calificados por el usuario a elementos similares y, a continuación, combina esos elementos similares en una lista de recomendaciones. Para determinar la coincidencia más parecida para un producto dado, el algoritmo construye una tabla de elementos similares encontrando elementos que suelen ser vendidos conjuntamente. Se construye una matriz producto-producto iterando a través de todos los pares de elementos y calculando la similitud para cada par. Sin embargo, muchas parejas de productos no tienen clientes comunes y, por lo tanto, el enfoque es ineficiente en términos de tiempo de procesamiento y uso de memoria. Para resolver esta deficiencia, Amazon calcula la similitud entre un único producto y todos los productos relacionados de la siguiente forma [6]:

```
Para cada ítem en el catálogo de producto, I1
      Para cada cliente C que compró I1
            Para cada ítem I2 comprado por el cliente C
                  Registrar que un cliente compró I1 e I2
      Para cada ítem I2
            Calcular la similitud entre I1 e I2
```

La similitud entre los productos I1 e I2 se calcula de forma vectorial aplicándose la similitud basada en el coseno formado por los ángulos entre los vectores que describen los productos. Si los vectores son coincidentes, el ángulo será 0º, y su respectivo coseno será 1, lo que representa total similitud. Para vectores perpendiculares, el coseno es 0, lo que representa ninguna similitud. La similitud por coseno es aplicada en espacios positivos, donde los rangos varían entre 0 y 1. Por lo tanto, vectores diametralmente opuestos, los cuales presentan coseno negativo (-1) se interpretan de la misma forma que el 1 donde tienen total similitud. [6]

Dada una tabla de ítems similares, el algoritmo encuentra ítems similares a cada una de las compras y calificaciones del usuario, agrega esos ítems y, a continuación, recomienda los que son más populares o correlacionados. Este cálculo es muy rápido, dependiendo solamente del número de artículos que el usuario ha comprado o calificado.

2.2. Recomendaciones de Netflix

Netflix es una empresa pionera en la popularización de la televisión por internet, lo que ofrece a sus usuarios la libertad de elegir qué, cuándo y dónde ver una película o programa de televisión. Su principal producto y fuente de ingresos es el servicio de suscripción que permite a los miembros ver cualquier vídeo de su colección de películas y programas de televisión en cualquier momento en una amplia gama de dispositivos conectados a Internet. En 2015, Netflix ya tenía más de 65 millones de miembros inscritos, que consumían más de 100 millones de horas de películas y programas de televisión por día. Un pilar clave de su producto es el sistema que hace recomendaciones el cual ayuda a sus usuarios a encontrar videos para ver en cada sesión. Una investigación sobre sus consumidores sugiere que un usuario típico de Netflix pierde interés después de quizás de 60 a 90 segundos en el proceso de elección, habiendo revisado de 10 a 20 títulos (tal vez 3 en detalle) en una o dos pantallas. Por eso, su sistema de recomendación no es un algoritmo, sino una colección de diferentes algoritmos que sirven a

diferentes casos de uso que se unen para crear lo que llaman "la experiencia completa de Netflix", la cual tiene como propósito principal evitar el abandono del cliente y garantizar que algo de su interés esté en las dos primeras pantallas. [7]

Históricamente, el problema de la recomendación de Netflix se ha considerado equivalente al problema de predecir el número de estrellas con las que una persona calificaría un video después de mirarlo, en una escala de 1 a 5. Incluso se ha organizado un concurso destinado a mejorar el algoritmo que predice las calificaciones conocido como "Netflix Prize 2009". Sin embargo, con las transmisiones de contenido y grandes cantidades de datos acumulados sobre cada usuario, las recomendaciones pasaron a estar no más centradas en calificaciones, sino en los datos que describen lo que mira cada cliente de Netflix, cómo mira (por ejemplo, el dispositivo, hora del día, día de la semana, intensidad de la visualización), el lugar en el que cada vídeo fue descubierto, e incluso las recomendaciones que se mostraron, pero que no se reproducen en cada sesión.

Estos datos y las experiencias resultantes mejoraron el producto Netflix, enseñándoles que hay maneras mucho mejores de ayudar a las personas a encontrar videos para ver que centrarse sólo en aquellos con una alta calificación de estrellas predichas.

Su sistema de recomendación consiste en una variedad de algoritmos que definen colectivamente la experiencia de Netflix, la mayoría de los cuales se reúnen en la página principal de Netflix. Esta es la primera página que un usuario de Netflix ve al ingresar a su perfil de Netflix en cualquier dispositivo (TV, *tablet*, teléfono o navegador), es la presentación principal de las recomendaciones, donde se descubren 2 de cada 3 horas transmitidas en Netflix.

Hay típicamente 40 filas en la página principal (dependiendo de la capacidad de cada dispositivo) y hasta 75 videos por fila. Los videos de cada fila son generados por algoritmos distintos, descritos con más detalle abajo [7]:

- ***Personalized Video Ranker (PVR)*:** este algoritmo ordena todo el catálogo de videos (o subconjuntos seleccionados por género u otro filtrado) para cada perfil de usuario de una manera personalizada. El orden resultante se utiliza para seleccionar el orden de los videos en el género (por ejemplo, generando una fila llamada "Películas de Suspense") y otras filas, y es la razón por la que la misma fila de género mostrada a los diferentes usuarios a menudo tiene vídeos completamente diferentes. También se utiliza este algoritmo combinado con ítems populares del catálogo (no personalizada) para generar las recomendaciones en la fila "Populares en Netflix".

- ***Top-N Video Ranker***: El objetivo de este algoritmo es encontrar un número reducido de las mejores recomendaciones personalizadas en todo el catálogo para cada usuario, es decir, centrarse sólo en las que estén en la cabecera del ranking (ver figura 3).

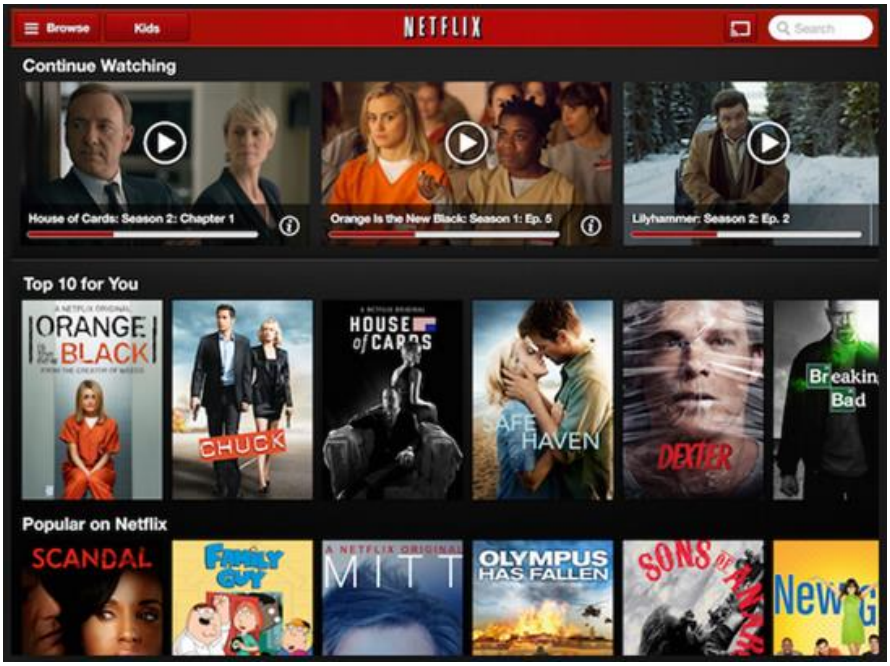


Figura 3 - Ejemplo de los resultados del algoritmo “Top-N video ranker” y “Continue Watching”

- ***Trending Now***: Esta fila muestra las tendencias del momento. Su algoritmo utiliza tendencias temporales, desde minutos hasta días. Como ejemplo de tendencias que se repiten a cada periodo de meses, podemos citar la mayor visualización de películas románticas durante el *Valentine´s day* en Norteamérica y, como ejemplo de tendencias de corto plazo, podemos mencionar la mayor visualización de documentales sobre huracanes si hay alguna región del mundo afectada por este evento del que se informa en los periódicos.
- ***Continue Watching***: El algoritmo de la fila “Continúe viendo”, al contrario de los otros que sugieren videos basados en títulos no vistos, ordena los títulos ya iniciados de acuerdo con lo que tenga más probabilidad de interés por el usuario, y para eso considera el tiempo transcurrido desde la visualización, el punto de abandono (medio del programa vs. comienzo o final), si se han visto diferentes títulos desde entonces, y los dispositivos utilizados (ver figura 3).
- ***Video-Video Similarity***: El algoritmo de similitud video-video genera las filas tituladas “Porque has visto” (*Because you watched - BYW*) combinado con el

nombre de alguna película vista por el usuario bajo la cual se basará la recomendación. Este algoritmo procesa la similitud entre películas (selección no personalizada) y la personaliza antes de mostrar en la fila BYW según los gustos del usuario y las películas ya vistas. (ver figura 4)

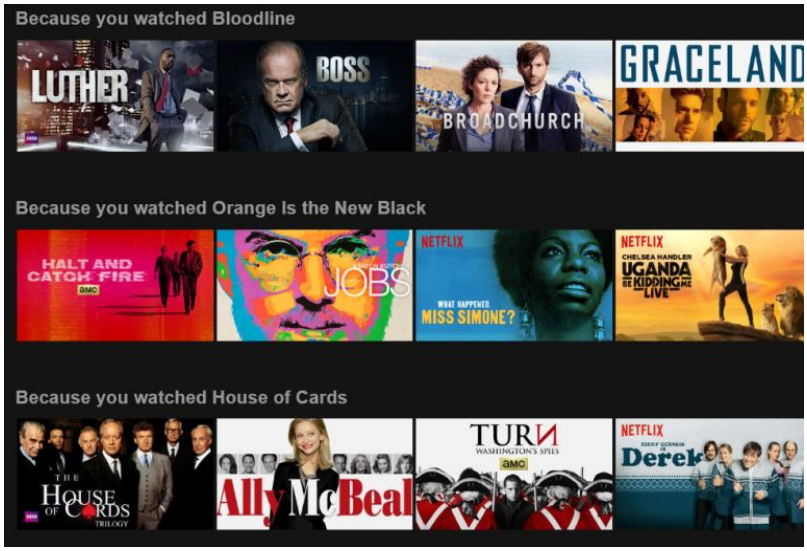


Figura 4 - Ejemplo de los resultados del algoritmo "*Because you watched*"

- ***Page Generation - Row Selection and Ranking***: La filas que son mostradas en la página de Netflix para cada usuario también son generadas a través de un algoritmo totalmente personalizado y matemático que puede seleccionar y ordenar filas de un gran grupo de películas para crear un orden optimizado por relevancia y diversidad. Este algoritmo no utiliza una plantilla, por lo tanto, es más libre para optimizar la experiencia; por ejemplo, puede elegir no tener ninguna fila BYW para una página de inicio o bien dedicar la mitad de una página a ellas.

Todos los algoritmos definidos hasta ahora forman parte de la mencionada "Experiencia Netflix". Sin embargo, también hay algoritmos complementarios como es el caso de "*Evidence*", que evalúa todos los posibles ítems de evidencia (cantidad de estrellas, premios de la película, elenco y otros metadatos) de cada recomendación y selecciona los más relevantes para mostrar al usuario, si es similar a otra película recientemente vista e incluso cual es la mejor versión de imagen de la película para soportar la recomendación. [7]

El sistema de recomendación de Netflix se utiliza en la mayoría de sus pantallas, pero también más allá de la página principal y su influencia total es responsable de aproximadamente un 80% de las horas transmitidas en Netflix. El 20% restante proviene de la herramienta de búsqueda, que requiere su propio conjunto de algoritmos. Los usuarios frecuentemente buscan

por videos, actores o géneros, pero que a menudo no están en el catálogo. Toda información introducida por el usuario es aprovechada para que otro algoritmo empiece su trabajo, recomendando otros videos relevantes como resultados alternativos para la búsqueda fallida. [7]

Todos los algoritmos de recomendación de Netflix están soportados por técnicas estadísticas y de aprendizaje automático, donde se incluyen los enfoques supervisados (clasificación y regresión) y no supervisados (reducción dimensional a través de agrupamiento o compresión).

3. Técnicas de minería de datos empleadas

3.1. Agrupamiento

3.1.1. La tarea de agrupamiento

Hay varias formas de estructurar un agrupamiento. La más conocida y usual forma grupos de acuerdo con la similitud de los atributos de cada ítem, por ejemplo, sexo, edad o profesión si se trata de un cliente, o una combinación entre ellos.

La tarea de agrupamiento consiste en partir un conjunto de datos, que forma un único grupo, en un conjunto de subgrupos, a partir de uno o más campos previamente elegidos, conforme a la similitud de estos atributos [1]. Por ejemplo, si se desea agrupar los tiquetes de compra de una tienda por fecha de venta y tipo de producto comprado, estos dos campos serán comparados registro a registro y, los que coinciden o son similares formarán parte del mismo grupo.

Agrupamiento es una tarea no supervisada, pues su objetivo no es crear un modelo. Es una tarea que crea una segmentación de los datos. Un buen método de agrupamiento producirá *clusters* (grupos) de alta calidad en la que la similitud *intra-cluster* es alta, mientras que la similitud *inter-cluster* es baja. La alta similitud *intra-cluster* significa que hay gran proximidad y cohesión entre los elementos de cada *cluster*. Y la baja similitud *inter-cluster* significa que los puntos centrales (centroides) de cada *cluster* se alejan de forma satisfactoria representando un buen aislamiento de los grupos.

Entre las técnicas para realizar esta tarea, encontramos el agrupamiento jerárquico y el agrupamiento por medias. El agrupamiento jerárquico es una manera sencilla de visualizar mediante una estructura jerárquica (un árbol) cómo los datos se agrupan entre sí (de acuerdo con la distancia de base), pero puede ser costoso de construir y mostrar el árbol completo para un gran volumen de datos. Por eso en este trabajo se ha elegido el agrupamiento por medias, en específico, el algoritmo *k-means*, por ofrecer un tratamiento más eficiente que el agrupamiento jerárquico cuando las observaciones presentan una gran cantidad de atributos, y también por ofrecer una presentación sencilla de los resultados para su interpretación. [1]

3.1.2. Algoritmo *k-means*

K-means es un algoritmo de agrupamiento por medias, donde se realiza una búsqueda iterativa de los k grupos de puntos representando cada grupo mediante su centroide (punto central del grupo) y asignando cada ejemplo al grupo de centroide más próximo, como ilustrado en la figura 5.

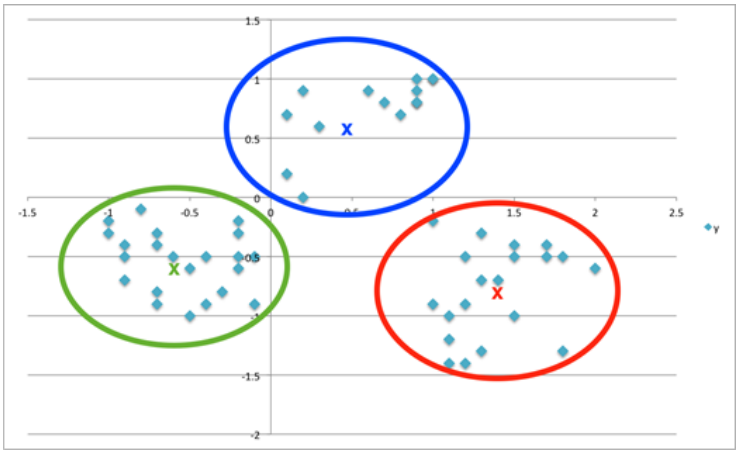


Figura 5 - Ejemplo de *clusters* generado por *k-means*

La utilización de *k-means* presupone la determinación de un parámetro de entrada por parte del usuario (el valor *k*), que indica el número de grupos que se desea crear. Este es un punto crítico en la aplicación de *k-means* una vez que es difícil predecir cuál sería la *k* ideal para generar el número óptimo de *clusters*, con respeto a las medidas de evaluación antes mencionadas (similitud *intra* e *inter-clusters*). Para este trabajo, se ha elegido el método "Elbow" para la elección de la *k*, que será mejor detallado al final de este capítulo.

A continuación, se describe el algoritmo de *k-means*:

1. Basado en el valor *k* elegido por el usuario, *k* puntos aleatorios son elegidos como centroides.
2. A continuación, el algoritmo calcula la distancia de cada punto a los centroides, y asigna el punto al *cluster* donde la distancia a su respectivo centroide sea la más pequeña. Si los atributos son numéricos, seguramente se usa la distancia Euclídea.
3. Tras haber asignado todos los puntos a un *cluster*, nuevos centroides son calculados para cada *cluster* basado en los puntos asignados al mismo.
4. Los pasos 2 y 3 se repiten hasta que todos los centroides se mantengan estables, o sea, no se modifican con relación a la iteración anterior.

Como resultado de este procesamiento, obtenemos el *cluster* asignado a cada observación del *dataset* (conjunto de datos suministrados al sistema), y algunas medidas útiles sobre los *clusters* generados tales como:

1. ***betweenss*** : suma de los cuadrados (en inglés, sum of squares, lo que explica las ss al final del nombre) entre los centroides de los *clusters*, o sea, la distancia euclídea *inter-cluster*.
2. ***withinss***: un vector con la distancia euclídea entre los puntos de cada *cluster*.
3. ***totwithinss*** : suma total de las distancias euclídeas *intra-cluster*.

Una vez comentado sobre estos 3 datos de salida del procesamiento, tomamos el **totwithinss** como protagonista del método "Elbow" comentado anteriormente para la definición de la *k* ideal del procesamiento. Siendo el **totwithinss** la distancia *intra-cluster* (umbral de cohesión interno de cada *cluster*), se observa una dependencia de esta distancia en relación con el número de *clusters*. O sea, cuanto más grande la k, más grande es la cohesión entre los puntos, lo que muestra *clusters* cada vez más compactos, como en el grafo de ejemplo de la figura 6 [14]:

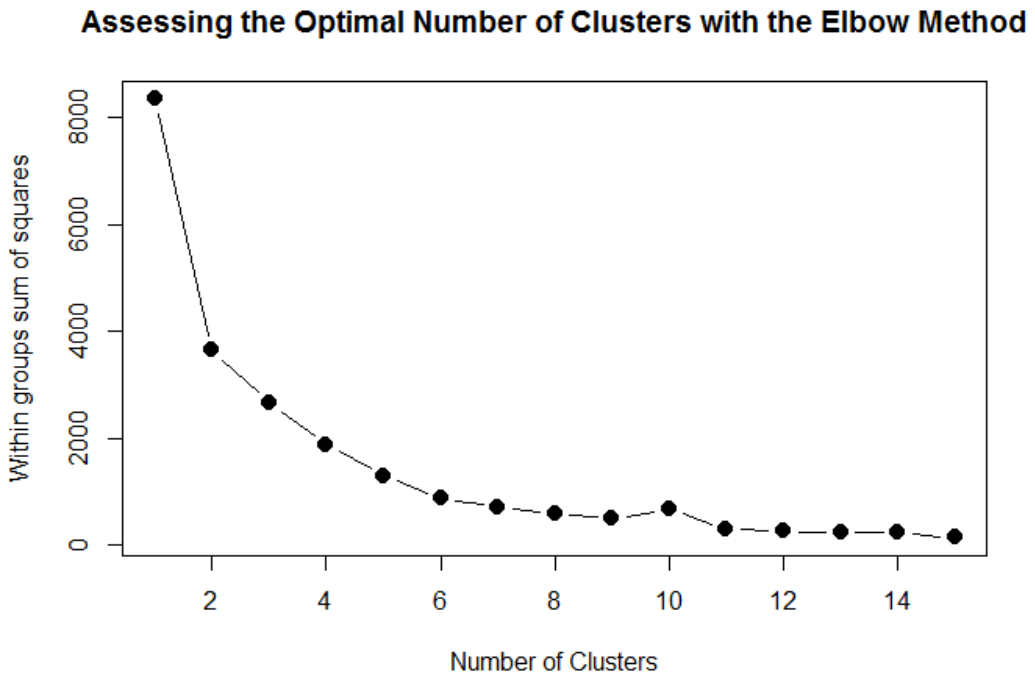


Figura 6 - Ejemplo del método "Elbow" para determinar el número óptimo de *clusters*

[14] Felipe Rego, artículo "Example of K-Means Clustering with R" en RPubs, https://rpubs.com/FelipeRego/K-Means-Clustering, julio de 2015.

En esta figura, se muestra el valor de **totwithinss** para valores de k variando desde 1 hasta 15. Como se puede observar, el valor de k en la gráfica que corresponde a un codo (Elbow) representa el punto a partir de lo cual no va a producirse una variación significativa en la cohesión. En el ejemplo, k = 6 representa el codo y por lo tanto el número ideal de *clusters*. Es interesante observar que este método ofrece una alternativa para evitar la creación de una cantidad demasiada, o insuficiente, de *clusters*, al tiempo que indica la k mínima a partir de la cual se estabiliza la similitud entre las observaciones de cada *cluster*.

3.2. Reglas de asociación

3.2.1. La tarea de asociación

En la tarea de asociación se buscan reglas que relacionan los atributos/ítems que ocurren de forma frecuente entre las instancias de un *dataset*. Por ejemplo, si las instancias representan los *tiques* de la compra, la asociación busca productos comprados de forma simultánea. Análogamente, también se podría pensar en un cliente y todos los productos que buscó en una sección de internet, o en un oyente y todas las canciones de su *playlist* o las escuchadas en una sección de internet.

Sea un conjunto A = $\{a_1, a_2 \dots a_n\}$ el conjunto que representa todos los posibles ítems que el usuario/cliente puede seleccionar, y sea D = $\{d_1, d_2 \dots d_m\}$ el conjunto de instancias del *dataset* y que representan transacciones, donde cada instancia $d_j = \{a_{ij}, \dots a_{nj}\}$ es una tupla de n componentes siendo a_{ij}=1 si el ítem a_i está en la transacción d_j, y 0 en caso contrario. Una regla de asociación es una implicación de la forma X => Y donde:

- $X, Y \subseteq A$, o sea, los ítems expresados en la regla son elementos del conjunto A.
- $X \cap Y = \emptyset$, o sea, el mismo ítem nunca estará en la parte anterior y posterior de la regla.
- X es llamado **antecedente** o **parte izquierda (lhs)** de la regla, mientras que Y es llamado **consecuente** o **parte derecha (rhs)**.

Un ejemplo de regla para el supermercado podría ser {mantequilla, pan} ⇒ {leche}, que significa que, si la mantequilla y el pan han sido comprados juntos, entonces los clientes también compraron leche. Basado en este ejemplo, uno podría a través de la regla saber qué

producto podría tener sus ventas afectadas en caso de que faltara otro, o también, qué producto combinado con pan podría estimular la venta de leche. La regla de asociación del ejemplo se denomina booleana ya que expresa la ausencia o presencia de ítems en las transacciones. También existen las reglas de asociación cuantitativas, que expresan la relación entre rangos de valores. Por ejemplo, una regla Edad=25..30 => Coches =1..2 significa que entre 25 y 30 años, la población posee entre 1 y 2 coches.

El soporte (*support*) de una regla permite medir cuántas transacciones contienen el subconjunto de ítems formado por las partes antecedente y consecuente de una regla. Si el soporte de la regla es igual o más grande que el soporte mínimo definido, la regla es considerada válida, sino es excluida. Formalmente, el soporte de una regla X => Y se define como:

$$support\,(\mathrm{X} => \mathrm{Y}) = \frac{\text{número de transacciones que contienen X e Y}}{\text{número de total de transacciones}}$$

Como ejemplo, consideremos la siguiente base de *tiques* (tabla 1) que usamos para calcular el soporte de la regla {mantequilla} => {leche}:

Trans.ID	Lista de ítems
1	mantequilla, pan
2	mantequilla, pan, café, leche
3	pan, cerveza, leche
4	mantequilla, azúcar, leche
5	mantequilla, pan, leche

Tabla 1 - Ejemplo de tiquetes de compra

$$support\,(\{\text{mantequilla}\} => \{\text{leche}\}) = \frac{\text{3 trans. contienen antecedente y consecuente}}{\text{5 es el número de total de trans.}} = 60\%$$

La confianza de una regla permite medir cuántas transacciones contienen el subconjunto de ítems formado por las partes antecedente y consecuente de una regla, con relación a las transacciones que sólo contienen el antecedente. La confianza de una regla X => Y se define como:

$$\textit{confidence}\,(\mathrm{X} => \mathrm{Y}) = \frac{\mathrm{support}(\mathrm{X} => \mathrm{Y})}{\mathrm{support}(\mathrm{X})} = \frac{\text{número de transacciones que contienen ambos X e Y}}{\text{número de transacciones que contienen X}}$$

Basado en las mismas reglas del ejemplo anterior, tendremos las siguientes medidas de confianza:

$$\textit{confidence}(\{\text{mantequilla}\} => \{\text{leche}\}) = \frac{\text{3 trans. contienen antecedente y consecuente}}{\text{4 trans. contienen antecedente}} = 75\%$$

Para seleccionar reglas interesantes del conjunto de todas las reglas posibles, se pueden utilizar restricciones sobre diversas medidas de significación e interés. Las restricciones más conocidas son los umbrales mínimos de soporte y confianza, que son los parámetros especificados por el usuario y que deben ser satisfechos al mismo tiempo para que la regla de asociación sea seleccionada. Siguiendo el ejemplo de la tabla 1, si el usuario establece 50% como soporte mínimo, la regla {mantequilla} => {leche} sería considerada válida. Pero, si consideramos la regla {mantequilla, pan} => {leche}, obtendremos un soporte de ⅖ (40%), por lo que sería excluida ya que no supera el soporte mínimo. De la misma forma que el soporte, si la confianza de la regla es igual o más grande que la confianza mínima definida por el usuario, la regla es considerada válida, sino excluida.

Si el usuario ha establecido una confianza de 70%, por ejemplo, la regla {mantequilla} => {leche} también sería seleccionado. Sin embargo, para la otra regla del ejemplo {mantequilla, pan} => {leche}, obtendremos una confianza de 2/3 (66,7%), que no supera el umbral establecido.

Si una regla presenta ambos soporte y confianza por encima de los umbrales, entonces es considerada una regla fuerte. El conjunto formado por todas las reglas fuertes es el resultado del procesamiento de la tarea de asociación.

También hay otras medidas alternativas para evaluar las reglas generadas, lo que puede aclarar cuáles reglas son más usables y que complementan las medidas de soporte y confianza. Entre las medidas alternativas, es relevante comentar el *Lift*, que es una medida que representa el umbral de independencia entre el antecedente y el consecuente de una regla. Si una regla tuviera *Lift* igual a 1, implicaría que la probabilidad de ocurrencia del antecedente y la del consecuente son independientes entre sí, donde no se puede extraer ninguna regla de asociación que involucra estos dos eventos. Sin embargo, si el *Lift* es más grande que 1, esto nos permite concluir que el antecedente y consecuente ocurren de forma más frecuente conjuntamente, o sea, presentan dependencia entre sí, lo que hace con que las reglas que involucran ambos sean

potencialmente útiles para predecir el consecuente. Al contrario, para un *Lift* más pequeño que 1, tenemos el caso de una asociación negativa, lo que revela que la frecuencia del antecedente hace que la frecuencia del consecuente sea menor y, por lo tanto, sea una regla más débil.

Hay una serie de algoritmos utilizados para generar reglas de asociación tales como Apriori, Eclat y FP-growth. Este trabajo aplica el algoritmo Apriori en el proceso de recomendación pues, al contrario de los otros mencionados, es un algoritmo especialmente diseñado para generar reglas de asociación en base de datos transaccionales.

3.2.2. Algoritmo Apriori

El algoritmo Apriori consiste en generar primero conjuntos de ítems frecuentes y después les ordena para generar las reglas.

El principio fundamental es que cualquier subconjunto de un conjunto frecuente también es frecuente. O sea, una transacción que contenga el conjunto {A, B, C} también contiene {A, B}. Si {A, B, C} es frecuente, entonces {A, B} también los es. Además, también asume como principio que ningún superconjunto de cualquier conjunto de ítems poco frecuentes debe ser generado o probado, lo que es extremadamente ventajoso para el desempeño computacional del algoritmo, pues puede ignorar ramas enteras basada en la frecuencia de los primeros conjuntos probados, como resaltado por la línea discontinua en la figura 7 [15]. Por ejemplo, si el conjunto {A, B} indicado no es suficientemente frecuente, Apriori no evaluará ningún superconjunto conteniendo {A, B} por considerarlo insuficientemente frecuente [9]. Esto permite generar los conjuntos de ítems frecuentes mediante un proceso iterativo, creando primero conjuntos de tamaño 1, que se incrementa de uno en uno cada iteración.

[15] Imagen de la diapositive "Mining Association Rules" de autor no identificado, https://paginas.fe.up.pt/~ec/files_0506/slides/04_AssociationRules.pdf

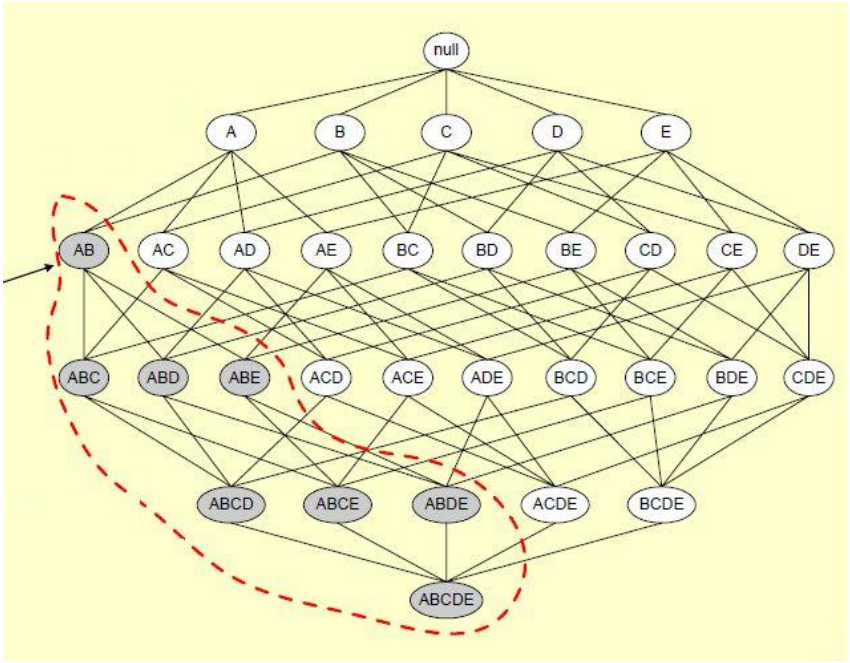


Figura 7 - Ejemplo de conjuntos de ítems frecuentes

Por tanto, basado en el mínimo soporte y confianza definidos por el usuario la búsqueda de los ítems frecuentes empieza considerando toda la base de datos como candidatos al conjunto de elementos frecuentes, y entonces cuenta la ocurrencia de cada elemento, y selecciona los que superan el mínimo soporte y confianza definidos. El paso siguiente se basa en el conjunto formado en el paso anterior y lo adopta como conjunto de candidatos que combina entre sí se buscando conjuntos frecuentes de 2 elementos que superan los umbrales mínimos. De forma iterativa, este proceso repite hasta tener un único grupo. El ejemplo mostrado abajo en la figura 8 ilustra de forma sencilla la interacción entre las fases de Apriori, considerando la restricción de soporte mínimo de 2 transacciones:

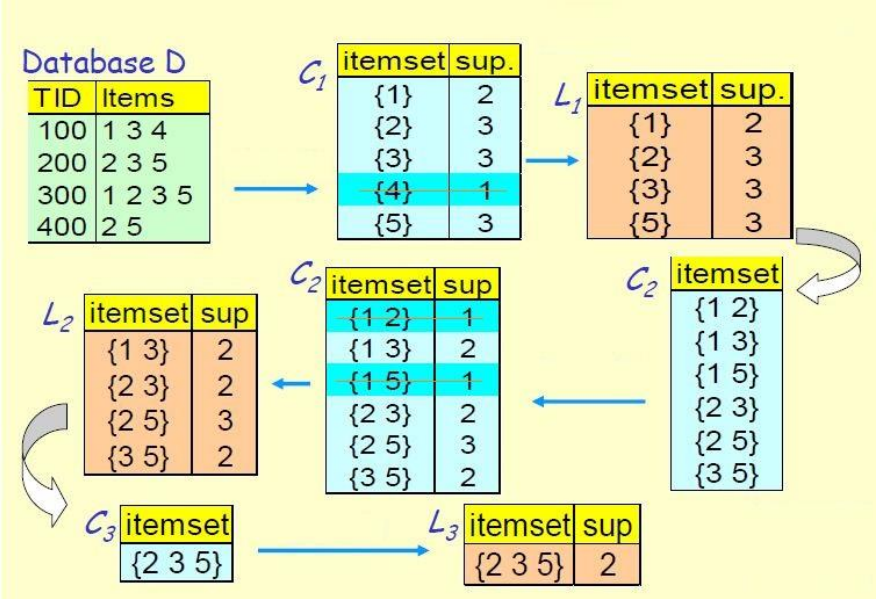


Figura 8 - Ejemplo de aplicación de Apriori [16]

Al final del procesamiento, Apriori genera las reglas a partir de los conjuntos de ítems frecuentes (seleccionando parte de los ítems como antecedente y el resto como consecuentes) indicando para cada regla su soporte, confianza y *Lift*, como en el ejemplo que sigue:

```
      lhs                rhs    support    confidence lift
[1]   {}              => {289} 0.9305136 0.9305136  1.000000
[2]   {}              => {89}  0.8972810 0.8972810  1.000000
[3]   {}              => {288} 0.8942598 0.8942598  1.000000
[4]   {89}            => {289} 0.8549849 0.9528620  1.024017
[5]   {289}           => {89}  0.8549849 0.9188312  1.024017
[6]   {288}           => {289} 0.8429003 0.9425676  1.012954
[7]   {289}           => {288} 0.8429003 0.9058442  1.012954
[8]   {}              => {300} 0.8308157 0.8308157  1.000000
[9]   {288}           => {89}  0.8157100 0.9121622  1.016585
[10]  {89}            => {288} 0.8157100 0.9090909  1.016585
[11]  {}              => {292} 0.7915408 0.7915408  1.000000
[12]  {300}           => {289} 0.7854985 0.9454545  1.016057
[13]  {289}           => {300} 0.7854985 0.8441558  1.016057
[14]  {288,89}        => {289} 0.7824773 0.9592593  1.030892
[15]  {288,289}       => {89}  0.7824773 0.9283154  1.034587
[16]  {289,89}        => {288} 0.7824773 0.9151943  1.023410
[17]  {300}           => {89}  0.7613293 0.9163636  1.021267
[18]  {89}            => {300} 0.7613293 0.8484848  1.021267
[19]  {292}           => {289} 0.7462236 0.9427481  1.013148
[20]  {289}           => {292} 0.7462236 0.8019481  1.013148
```

[16] Autor no identificado, Mining Association Rules, https://paginas.fe.up.pt/~ec/files_0506/slides/04_AssociationRules.pdf

4. Un sistema de recomendación basado asociaciones y agrupamiento

En este capítulo presentamos una nueva aproximación a los sistemas de recomendación que se caracteriza por estar basada en la generación de dos filtros, uno colaborativo y otro por contenido. Estos filtros, como explicamos a lo largo de este capítulo, responden a la siguiente idea básica: el **filtro colaborativo** consiste en agrupar los clientes teniendo en cuenta la similitud de los ítems previamente comprados/vistos por los clientes y luego generar reglas de asociación en cada grupo. Estas reglas mostrarán los ítems más frecuentes y serán las usadas para hacer las recomendaciones. Este constituye lo que denominamos **perfil principal** del filtro colaborativo. En el caso de que no se generen reglas en un grupo, definimos un **perfil secundario** consistente en aplicar el mismo procedimiento del perfil principal pero solo en determinados grupos; el **filtro por contenido** (al que denominamos **Perfil 2**) consiste en agrupar los ítems de acuerdo a sus características y hacer recomendaciones entre los ítems similares. Por lo tanto, del mismo modo que los sistemas de recomendación de referencia, como Netflix y Amazon, nuestro sistema utiliza más de un método de filtrado, caracterizándose por lo tanto como un modelo híbrido (*Hibrid filtering*), donde se utiliza el filtrado colaborativo como motor principal de la recomendación, y el filtrado basado en contenido (ítem-ítem) como alternativa para recomendaciones en segunda instancia. Además, también se utilizará *"Model-based techniques*" como categoría del filtrado colaborativo, una vez que se combinarán las tareas de agrupamiento y de asociación para la construcción del sistema. Para la evaluación experimental del prototipo desarrollado, han sido desarrollados scripts en lenguaje R utilizándose el entorno RStudio, con el soporte adicional de los siguientes paquetes:

- **plyr**: paquete que ofrece herramientas para que se pueda dividir un problema en partes manejables, operar cada pieza y luego volver a unir todas las piezas.
- **dplyr**: paquete para manipulación de *dataframes* (conjunto de datos) como objetos, pudiéndose ya estar en memoria o no.
- **arules**: paquete que provee la infraestructura para representación, manipulación y análisis de datos transaccionales y patrones (conjunto de ítems frecuentes y reglas de asociación).
- **pmml**: abreviación de *Predictive Model Markup Language*, es un paquete basado en lenguaje XML que proporciona una forma para que las aplicaciones definan

modelos estadísticos y de minería de datos para compartir modelos entre aplicaciones compatibles con PMML.

Como ya hemos mencionado anteriormente, al contrario de muchos estudios hechos en el ámbito de las recomendaciones y de las soluciones actualmente ofrecidas por empresas, nuestra aproximación se caracteriza por la simplicidad de los datos empleados que requiere escasa información de los clientes y ninguna interacción posterior con los mismos, por lo que podemos considerarla muy poco invasiva. En concreto, nuestra aproximación es capaz de generar recomendaciones a partir únicamente de los ítems previamente seleccionados/visitados/comprados por los usuarios, junto con la información del tipo asociado a cada ítem.

A partir de estos datos, el algoritmo de entrenamiento del sistema genera sus bases de recomendación, creándose distintas alternativas para abarcar el mayor número de situaciones en la aplicación del sistema. En una eventual utilización de este prototipo para la creación de un sistema de recomendación completo, los escenarios de recomendación podrían ser evaluados en cada ámbito que se piense utilizar, siendo posible que se personalice el comportamiento del sistema, cambiando parámetros como, los niveles de soporte y confianza para las reglas o bien desactivando algún o todos los flujos alternativos, si así fuera más adecuado a su contexto y estrategia de negocio. Si el sistema no produce ninguna recomendación, sea por insuficiencia de datos de entrenamiento o por definiciones personalizadas restrictivas del motor de recomendaciones, aun podrían ser ofrecidas otras alternativas de personalización tal y como mencionamos en el capítulo de conclusiones de este trabajo.

4.1. Definición del sistema

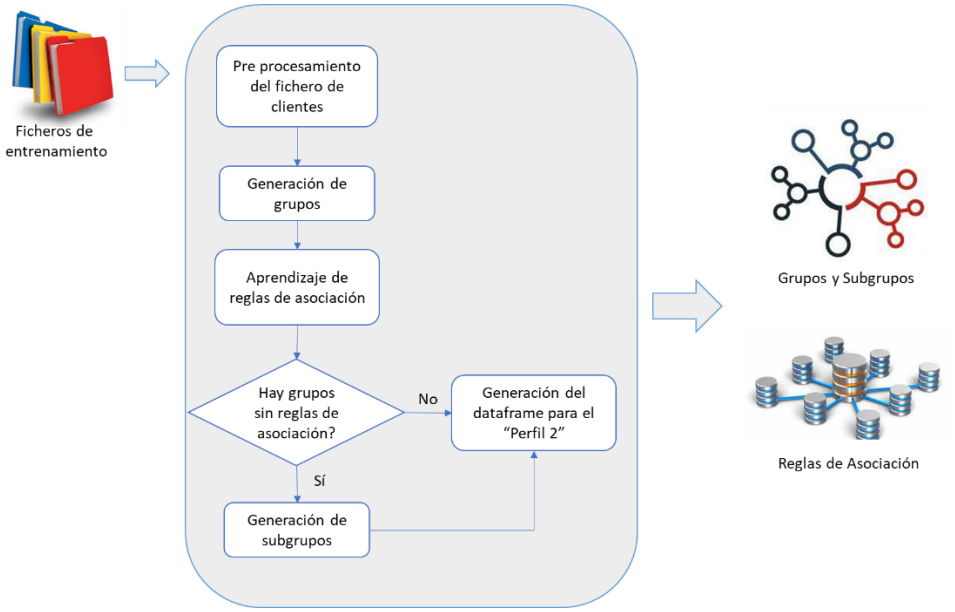


Figura 9 – Esquema del sistema de recomendación: generación de los filtros colaborativos y por contenido

La figura 9 muestra el esquema de nuestro sistema de recomendación que consta de las siguientes etapas:

1. Preparación de datos
2. Generación del perfil principal (filtro colaborativo)
3. Generación del perfil secundario (filtro colaborativo)
4. Generación del perfil 2 (filtro ítem-ítem)

1. Preparación de los datos

La primera etapa se dedica al preprocesamiento del fichero de clientes que debe pasar por una transformación inicial, si no está en el formato CSV (*comma separated value*) como esperado por el sistema. Este fichero debe contener un código identificador del cliente y un código identificador de su ítem de preferencia donde cada fila representará una pareja con estos dos códigos, pudiendo haber repetición del código de cliente para cada una de sus preferencias. A continuación, el fichero es convertido a una matriz formada por columnas que representan cada ítem del catálogo y una fila por cliente. Cada fila de la matriz representa un vector binario cuyas componentes tendrán el valor 1 si corresponde a un ítem elegido por el cliente ó 0 en caso contrario y, por eso, esta parte del preprocesamiento se denomina **vectorización**. Como última acción de esta etapa, se hace una partición de los datos en **conjunto de entrenamiento**,

y **conjunto de prueba (test)**. El conjunto de entrenamiento contiene los datos que serán utilizados exclusivamente para el entrenamiento del sistema, donde se identifican los patrones y se generan las reglas que formarán parte del mismo. Mientras que el conjunto de prueba contiene los datos que se usan para la evaluación experimental, donde se comprueba la eficiencia de las reglas formadas aplicándolas a los datos de prueba y verificando la ocurrencia del patrón (como veremos en la sección 4.2). Para eso, se crea una muestra fraccionada aleatoria de la matriz de datos utilizándo la función *sample_frac* del paquete dplyr donde se ha definido para este trabajo que el 70% de los clientes se destinen al conjunto de entrenamiento y, el 30% restante al conjunto de pruebas. A partir de este punto ya no se utiliza el conjunto de pruebas en ninguna etapa de entrenamiento, garantizándose el completo aislamiento de los datos para la etapa de evaluación.

2. Generación del perfil principal

Una vez concluido el preprocesamiento, se procede a la generación del perfil principal que consta de 2 etapas: generación de grupos y la obtención de reglas de asociación. En la primera, los grupos son generados a partir de los perfiles de clientes, basándose en la similitud entre los vectores de cada cliente, utilizándose el algoritmo de *k-means*. En esta tarea, el reto principal está en la definición de la cantidad de grupos más adecuada, que corresponde al parámetro *k*. Para la definición de la *k* ideal en este trabajo se han combinado las siguientes técnicas:

- Método del codo (*Elbow Method*): conforme sugerido por el método, se ha generado un grafo a partir de ejecuciones de *k-means* utilizando nuestra base de datos entrenamiento con la *k* variando desde 1 hasta 10, y se toma el punto que donde se nota el codo como la *k* de nuestro proceso de agrupación. (tal y como explicamos en la sección 3.1.2)
- Una vez que se tiene un posible valor para usarse como parámetro *k*, seguimos con una prueba de ejecución de Apriori sobre los grupos creados, evaluándose si este algoritmo termina con éxito. Se ha adoptado este procedimiento combinado Elbow-Apriori para determinar la *k*, pues hay valores de k con los que se crean grupos que hacen que Apriori genere miles, o incluso millones, de reglas, colapsando el procesamiento. Se ha observado que estos grupos normalmente contienen una cantidad muy reducida de transacciones (que en nuestro caso representan los

clientes) en comparación a los otros grupos, y que logran cumplir el soporte y confianza mínimos establecidos para casi todas las combinaciones posibles entre los ítems de cada transacción, y por eso el volumen tan alto de reglas. Si esta etapa de pruebas con Apriori termina con éxito, adoptamos la *k* sugerida por el codo. En caso contrario, restamos 1 de la *k* sugerida y volvemos a procesar Apriori, repitiéndose este proceso hasta que tengamos reglas generadas con éxito. El motivo por lo cual se vuelve atrás para probar un valor inferior de *k* es que los valores mayores siempre mantienen el grupo reducido de transacciones que producen el colapso de Apriori al intentar generar las reglas.

Tras las ponderaciones presentadas y la k ideal ya definida, se ejecuta el algoritmo de *k-means,* y luego se etiqueta cada usuario en el grupo al que ha quedado asignado.

A continuación, pasamos a la etapa de producción de reglas de asociación, donde se ejecuta el algoritmo Apriori del paquete *arules*, el cual producirá las reglas bajo las cuales se basarán las recomendaciones que denominamos "Perfil Principal", por el hecho de que son fruto de la estrategia principal de recomendación de este trabajo. Para la producción de dichas reglas, los valores umbrales de soporte y confianza se han establecido en 30% y 40%, respectivamente, para que se pueda explorar un rango mayor de posibilidades de recomendación. Una vez generadas las reglas, eliminamos las que son redundantes usando la función *is.redundant* del paquete *arules*, la cual mantiene las reglas más generales y de mayor confianza, descartándose las demás. Finalmente, el conjunto de reglas resultante es ordenado de forma decreciente por el valor de soporte y grabado en un fichero XML utilizándose la función *write.pmml* del paquete *pmml*.

3. Generación del perfil secundario

Si es el caso de que ninguna regla haya sido generada para algún grupo, lo que a menudo se ha observado en *clusters* que agrupan un gran número de clientes de gustos más particulares, se pasa a la siguiente etapa del procesamiento donde se generan subgrupos, también denominados *subclusters*, aplicándose la etapa 2 solamente a los datos del grupo sin reglas. Para los subgrupos, se procede con la definición de la *k* ideal de la misma forma que se hace para los grupos de la primera etapa. Con esto, se forma el primer flujo alternativo para lograr éxito en la recomendación, una vez que es muy común encontrar personas con gustos más diversificados y distintos a los perfiles generales. Ambos perfiles principal y secundario

constituyen lo que denominamos “Perfil 1”. La figura 10 muestra un ejemplo de generación de los perfiles:

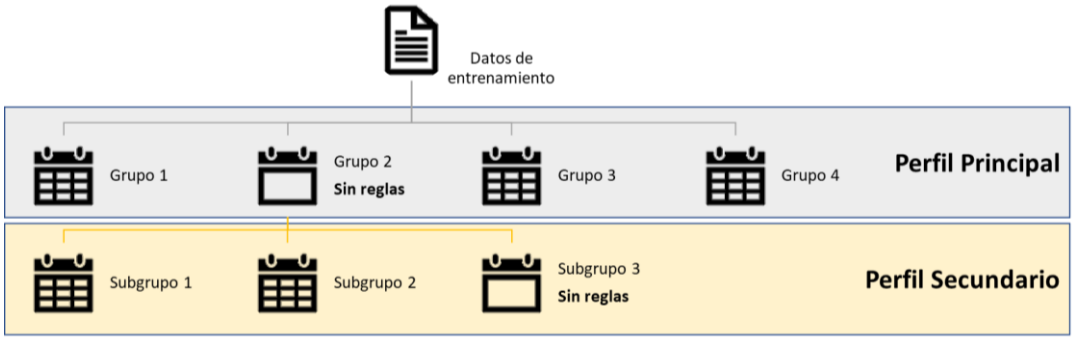


Figura 10 – Ejemplo de generación de los perfiles principal y secundario (filtro colaborativo).

4. Generación del “Perfil 2”

Igual a lo que pasa a los grupos, se observó que entre los subgrupos también se pueden generar uno o más grupos que concentran una cantidad más grande de clientes de gustos muy particulares y donde, nuevamente, no se generan reglas de asociación, como en el ejemplo del subgrupo 3 de la figura 10, lo que no permitiría ninguna recomendación. Además, también hay posibilidad de que existan reglas de asociación, pero que los antecesores de las mismas no coincidan con ítems del historial del cliente, lo que también nos llevaría al escenario de no recomendar nada. Como segundo flujo alternativo, hemos definido un filtro ítem-ítem que denominamos “Perfil 2”, que se basa en la generación de grupos de ítems que suelen llevar los mismos **tipos/características asociadas**, por ejemplo, artistas que están entre los mismos géneros musicales. En esta etapa, los datos con lo que se crea el perfil contienen los ítems del catálogo asociados a sus respectivos tipos. Un mismo ítem podrá aparecer varias veces en el conjunto entrenamiento, dependiendo de cuantos tipos tenga asociado. A estos datos aplicamos el mismo proceso de vectorización descrito anteriormente, creándose un vector binario para cada ítem y sus respectivos tipos/características asociados. A continuación, se aplica *k-means* para la generación de grupos de ítems que presentan más similitudes entre sus tipos asociados, realizándose la evaluación de la k ideal según las mismas técnicas ya descritas anteriormente en este capítulo, con excepción de que no se ejecuta Apriori en esta etapa y, por lo tanto, se saltará el criterio de evaluación de reglas generadas. Una vez realizada la agrupación, se asigna el respectivo número de grupo calculado a cada ítem y, en complemento a la estrategia del “Perfil 2”, también se asigna la frecuencia en la que el ítem aparece en el conjunto de entrenamiento (es decir, cuantas veces dicho ítem fue elegido por un cliente) para que, además

de encajar el cliente al grupo de tipos más ajustado a su perfil, también se le recomienden los ítems más frecuentes. Esta etapa finaliza el proceso de entrenamiento del sistema, y también marca su característica híbrida pues permite que el cliente que no obtenga recomendaciones a partir del filtrado colaborativo bajo el "Perfil 1", pueda obtenerlas a través del mapeo de sus gustos por el filtrado basado en contenido, a partir de los tipos de los ítems. En otros términos, si el sistema no consigue recomendar algo basado los ítems elegidos por otros usuarios similares, sube un nivel en la escala de generalización, y busca una recomendación basada en contenido a través de la similitud entre tipos de ítems.

5. Flujo base para producir recomendaciones

Una vez concluida la etapa de entrenamiento donde se han generados los grupos y subgrupos con sus respectivas reglas de asociación, y también los grupos del "Perfil 2", ya se puede ejecutar el flujo para hacer recomendaciones, el cual se define a través de las siguientes etapas mostradas en la figura 11:

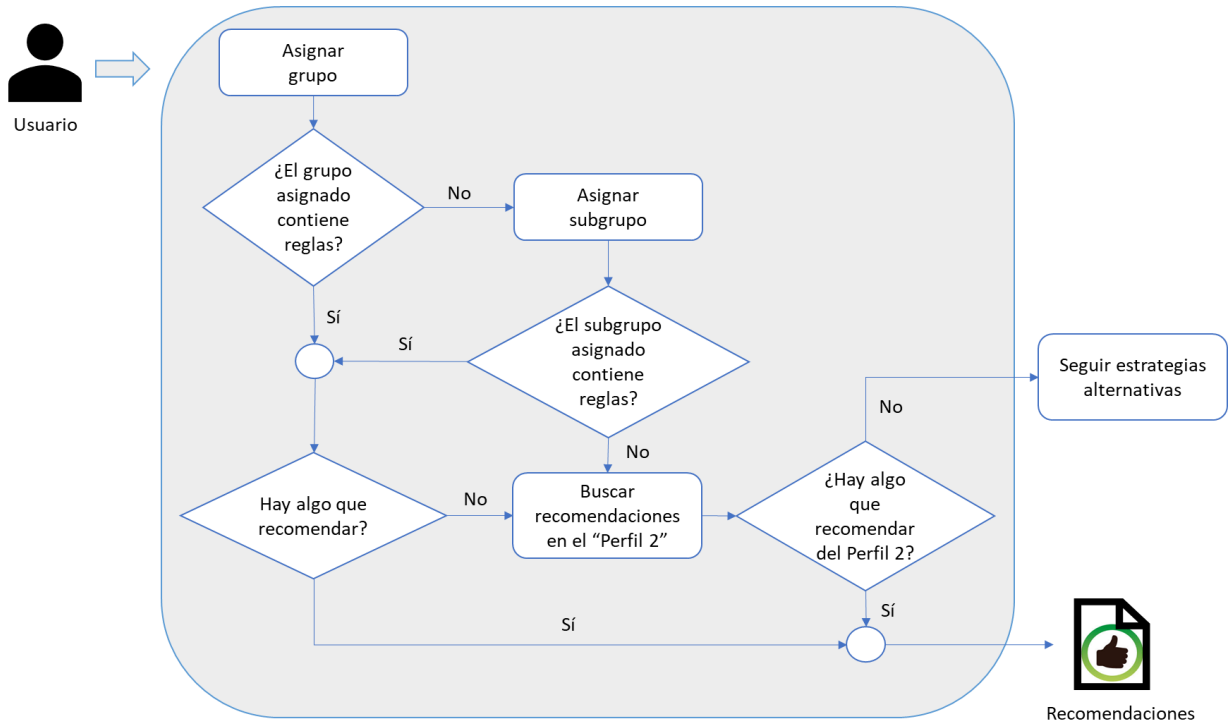


Figura 11 - Flujo de recomendación

Para hacer recomendaciones a un cliente C representado por el vector v_C, el sistema procede de la siguiente forma:

1. **Asignación de grupo:** El sistema calcula la distancia euclídea entre v_C y los centroides de cada grupo. Para este cálculo se utiliza la función *dist*, nativa del lenguaje R, que retorna la distancia entre dos vectores (filas de una matriz) basándose en una distancia de base especificada, que en este caso es *euclidean*. La menor distancia encontrada indica el grupo que se asigna a C.
2. **Asignación de subgrupo:** Si el sistema detecta que el grupo asignado a C no contiene reglas de asociación, automáticamente se procede a comparar la distancia euclídea entre v_C y los centroides de los subgrupos.
3. **Búsqueda por recomendaciones en las reglas de asociación:** Una vez que el grupo, o subgrupo, haya sido determinado, el sistema selecciona el conjunto de reglas de asociación correspondiente y prueba si los antecesores de cada regla es un subconjunto de v_C (es decir, v_C tiene un valor igual a 1 en las componentes que corresponden a los antecesores). En caso positivo, verifica si el sucesor no pertenece a v_C (es decir, la correspondiente componente de v_C tiene el valor 0) y, en este caso, lo selecciona para recomendación. En este punto del procesamiento, se podría optar por diferentes estrategias de recomendación, recomendar todos los ítems posibles (usando todas las reglas aplicables) o solo los ítems de las tres reglas de mayor confianza y soporte, o un único ítem. O sea, se podrían definir reglas de decisión de forma versátil incorporando parámetros personalizados para ajustarse al contexto de aplicación.
4. **Búsqueda por recomendaciones en el "Perfil 2":** Si el sistema detecta que ninguna recomendación ha sido encontrada para C dentro del "Perfil 1", automáticamente el "Perfil 2" es accionado. Para este perfil, creamos un vector de tipos asociados a las elecciones del cliente vt_C, donde la distancia euclídea entre vt_C y los centroides de los grupos creados en el "Perfil 2" son calculadas. El grupo que presenta la menor distancia es el que se asignará al cliente. Una vez que determinado, el sistema elige los tres primeros ítems de mayor frecuencia en dicho grupo, y que no sean ítems pertenecientes a v_C como recomendaciones. Nuevamente, se podría parametrizar este perfil para que se pudieran recomendar más ítems dependiendo del contexto.

4.2. Evaluación experimental

4.2.1. *Datasets* utilizados

Para la evaluación experimental de nuestro sistema de recomendación han sido utilizados dos *datasets* de dominio público extraídos del sitio web grouplens.org[17], con el propósito de probar la efectividad de las recomendaciones bajo dos contextos distintos. A continuación, se describen los *datasets* y los ficheros utilizados de cada caso:

- LastFM es una colección de datos del sitio web LastFM[18]. Del conjunto de datos disponibles, ha sido utilizado el conjunto de datos "user_artists.txt" que contiene datos sobre las preferencias por artistas musicales (17.632 artistas) de un conjunto de 1.892 usuarios, y también el fichero "user_taggedartists-timestamps.txt" que contiene las asignaciones de género hechas por cada usuario de la plataforma.
- MovieLens es una colección de datos de calificación del sitio web MovieLens[19]. De los muchos conjuntos de datos disponibles, ha sido utilizado el conjunto de datos "ratings.txt" que contiene más de 10 millones de clasificaciones y 95.580 etiquetas aplicadas a 10.681 películas por 71.567 usuarios, y también el fichero "movies.txt" que contiene los géneros asignados a cada película.

4.2.2. Marco experimental: definición de los experimentos y medidas de evaluación

La evaluación experimental utilizando los *datasets* de LastFM y MovieLens se ha realizado a través de 2 experimentos que nos han permitido comprobar la efectividad de nuestra propuesta. Ambos utilizan como base el flujo para producir recomendaciones ya descrito anteriormente en la sección anterior. El experimento 1 permite evaluar el grado de acierto en las recomendaciones en una situación simulada en la que se evaluará si el sistema es capaz de recomendar ítems a los que luego el cliente efectivamente accedió. En el experimento 2, evalúa la capacidad del sistema de recomendar nuevos ítems.

Para medir la calidad de una recomendación, se ha elegido un criterio de puntos, lo cual denominaremos *score* de una regla, y que es calculado a partir de las medidas de soporte y confianza de la regla aplicada, a través de la siguiente fórmula:

[17] https://grouplens.org/datasets/hetrec-2011/
[18] http://www.last.fm
[19] https://movielens.org/

$$Score_{regla} = Soporte_{regla} * Confianza_{regla}$$

Se ha utilizado el producto del soporte por la confianza como criterio de puntuación de una regla pues se interpretan estas dos medidas como representaciones de la calidad de una regla, lo que tiene influencia directa en su uso para hacer recomendaciones, es decir, cuantos mayores sean el soporte y la confianza, mejores serán las recomendaciones hechas por la regla. [3]

Uno de los factores de éxito de los sistemas de recomendación es el filtrado de ofertas que hacen a los usuarios, que a menudo encaran una gran cantidad de ítems en los catálogos, no logrando, a veces, encontrar los que sean de su gusto. Y para ser coherente a su propósito, los propios sistemas de recomendación no pueden reproducir este escenario ofreciendo una gran cantidad de sugerencias. Siguiendo esta línea de pensamiento, también se ha decidido explorar en nuestros experimentos los indicadores para la recomendación de los tres ítems de mayor *score* entre los seleccionados por el sistema, visto que esta estrategia podría ser también interesante en ciertos contextos (por ejemplo, en las ventas online). Para facilitar la distinción de otros indicadores evaluados en los experimentos, esta modalidad será denominada "*top 3*" de aquí en delante.

En ambos experimentos, estaremos constantemente evaluando las cantidades absolutas y relativas (porcentajes) de recomendaciones obtenidas en cada perfil, incluso con subtotales de las recomendaciones que sugirieron 1 ítem, 2 ítems y 3 o más ítems. Además, también medimos el **promedio del score general,** que significa la media del *score* de todos los ítems encontrados para la recomendación, y el **promedio del score de los "*top 3*"**, que significa la media del *score* de los 3 ítems de mayor *score* encontrados para la recomendación.

Para el experimento 2, en los casos en los que se aplica el "Perfil 2", también evaluamos los motivos por los que el "Perfil 1" ha fracasado a la hora de ser aplicado, midiéndose la cantidad absoluta y relativa (porcentaje) de casos en los que esto pasa. Estos motivos son:

- **sin reglas**: no hay reglas de asociación en los grupos o subgrupos asignados a los usuarios.
- **sin recomendación**: hay reglas de asociación en los grupos o subgrupos asignados a los usuarios, pero los ítems del vector del cliente no corresponden a los antecesores de ninguna regla.

4.2.2.1. Experimento 1 – Omitir una elección conocida

El experimento 1 simula la aplicación de nuestras recomendaciones sobre los ítems conocidos lo que nos permitirá registrar cuantos aciertos/fracasos habríamos cometido en las recomendaciones. Para ello, seguimos la siguiente estrategia: dado un usuario, de forma iterativa ocultamos cada ítem de su vector y tratamos de recomendar ítems basándonos en los que restan. Al final de todas las iteraciones, se obtendrán los éxitos y fracasos referidos al hecho de haber sido capaces de recomendar o no justo el ítem que hemos ocultado.

La ventaja de este experimento es que, sabiendo el ítem ocultado, podemos comparar de forma consistente si los mecanismos de agrupaciones en perfiles y reglas generadas han sido efectivos para recomendarlo. Aunque el enfoque de este experimento sea comprobar la recomendación del ítem omitido a través del "Perfil 1", también se producen resultados de recomendación del "Perfil 2" para la evaluación de este flujo alternativo.

Se explica la secuencia lógica de este experimento en el pseudocódigo abajo, el cual utiliza lenguaje natural y no intenta reflejar íntegramente el código original, sino la lógica aplicada. La función obtieneCluster utilizada en las explicaciones a continuación devuelve el grupo al que pertenece un cliente, tal y como se ha descrito en el apartado 4.2.1.

```
vector_items = selección de los ítems del cliente informado
Para cada ítem del vector_items
        matrix_items = convierte vector_items en matrix
        matrix_items[ítem] = 0              # para omitir el ítem que se intentará recomendar
        cluster = obtieneCluster(matrix_items,kmeans_output_cluster)
        Si no hay reglas para cluster entonces
                subcluster = obtieneCluster(matrix_items,kmeans_output_subcluster)
        Fin Si
        Si hay reglas para cluster o para subcluster
                reglas = leer fichero XML       #se cargan las reglas del XML correspondiente

                #se filtran reglas donde el ítem omitido es igual al sucesor
                subset_rhs = subset(reglas, rhs = ítem)

                éxitos = 0
                Para cada conjunto_antecesores del subset_rhs

                        #Si los antecesores de una regla son subconjunto del vector de ítems,
                        #entonces la regla recomendaría el ítem omitido
                        Si conjunto_antecesores está contenido en vector_items entonces
                                éxitos = éxitos + 1
                                max_sop_conf = máximo soporte + confianza entre las reglas
                        Fin Si
```

```
                Fin Bucle
                Si éxitos > 0 entonces
                        añadir ítem y max_sop_conf en tabla_items_exito
                Fin Si
        Fin Si
        Si tabla_items_exito no es vacía entonces
                Se muestra "Cantidad de ítems recomendados con éxito: " & éxitos
                Se muestra "Cantidad de ítems sin éxito: " & longitud(vector_items) - éxitos
                Se muestra los 3 ítems de mayor max_sop_conf
        Sino
                cluster_perfil_2 = obtieneCluster(matrix_items,kmeans_output_perfil2)

                # selección de ítems del dataframe del Perfil 2, asociados al cluster
                # calculado
                vector_recom_perfil2 = selección de ítems del cluster_perfil_2

                # se eliminan los ítems ya elegidos por el cliente del vector de
                # recomendaciones
                vector_recom_perfil2 = filtro de los ítems del vector_recom_perfil2
                                        que ya estén en vector_items
                Si vector_recom_perfil2 no es vacío entonces
                        Se muestra los 3 ítems de vector_recom_perfil2 con mayor soporte +
                                                                        confianza
                Sino
                        Se muestra "No hay recomendaciones."
                Fin Si
        Fin Si
Fin Bucle
```

4.2.2.2. Experimento 2 – Recomendar nuevo ítem

El experimento 2 simula la recomendación de ítems nuevos, es decir, ítems que no constan en el vector del usuario, lo que nos permitirá evaluar la capacidad del sistema de ofrecer novedades. A partir del vector de un usuario dado, el sistema mostrará todas las recomendaciones posibles de nuevos ítems a ese usuario, siguiendo el flujo del apartado 4.1. Además de mostrar todos los ítems nuevos posibles, también se mostrará una lista solamente con los "*top 3"* y sus respectivos scores.

El objetivo de este experimento es comprobar la cantidad de ítems nuevos que se podrían recomendar y sus respectivos *scores*. Al contrario del experimento anterior, no se puede medir aciertos/fracasos cuando se recomiendan ítems nuevos, por lo tanto, haremos una evaluación cuantitativa que se medirá a través de los porcentajes de usuarios que han recibido alguna recomendación, recomendaciones hechas por cada perfil, casos en que el sistema recomienda 1, 2, 3 o más ítems, y también a través de los promedios del score de todas las

posibles recomendaciones y de los "*top 3*". Las reglas donde el antecesor sea vacío no son consideradas en este experimento pues siempre generan recomendación, y lo que buscamos es obtener resultados que reflejan exactamente la capacidad del sistema para recomendar basado en el historial de los usuarios.

Se explica la secuencia lógica de este experimento en el pseudocódigo abajo, el cual utiliza lenguaje natural y no intenta reflejar íntegramente el código original, sino la lógica aplicada.

```
vector_items = selección de los ítems del cliente informado
matrix_items = convierte vector_items en matrix
cluster = obtieneCluster(matrix_items,kmeans_output_cluster)
Si no hay reglas para cluster entonces
        subcluster = obtieneCluster(matrix_items,kmeans_output_subcluster)
Fin Si
éxitos = 0
Si hay reglas para cluster o para subcluster
        reglas = leer fichero XML       #se cargan las reglas del XML correspondiente
        Para cada regla en reglas

                # Si los antecesores de la regla son subconjunto del vector de ítems y
                # el sucesor ya no ha sido elegido, entonces es una regla de éxito
                Si conjunto_antecesores está contenido en vector_items entonces y
                   rhs_regla no está contenido en vector_items
                        éxitos = éxitos + 1
                        Si rhs_regla ya existe en la tabla_items_exito
                                Se actualiza el score tabla_items_exito si es mayor que el
                                anterior
                        Sino
                                Se añade el ítem y su score en la tabla_items_exito
                        Fin Si
                Fin Si
        Fin Bucle
        Si éxitos > 0 entonces
                Se muestra todos los ítems de la tabla_items_exito
                Se muestra los 3 ítems de mayor score de la tabla_items_exito
        Fin Si
Fin Si
Si no hay reglas para cluster o para subcluster o éxitos = 0

        cluster_perfil_2 = obtieneCluster(matrix_items,kmeans_output_perfil2)

        # selección de ítems del dataframe del Perfil 2, asociados al cluster
        # calculado
        vector_recom_perfil2 = selección de ítems del cluster_perfil_2

        # se eliminan los ítems ya elegidos por el cliente del vector de
        # recomendaciones
```

```
        vector_recom_perfil2 = filtro de los ítems del vector_recom_perfil2
                                        que ya estén en vector_items
        Si vector_recom_perfil2 no es vacío entonces
                Se muestra los 3 ítems de vector_recom_perfil2 con mayor score
        Sino
                Se muestra "No hay recomendaciones."
        Fin Si
Fin Si
```

5. Resultados y Análisis de la evaluación experimental

5.1. Resultados de LastFM

5.1.1. Preprocesamiento de los datos

El *dataset* "user_artists.txt" de LastFM contiene una cantidad de datos que ha podido ser gestionada en las tareas críticas del entrenamiento, como la ejecución de *k-means*, de acuerdo con la capacidad computacional disponible para las pruebas de este trabajo. El preprocesamiento ha consistido en quitar las columnas innecesarias y quedarse con las de usuarios y artistas de preferencia. El *dataset* "user_taggedartists-timestamps.txt" ha presentado error asociado al volumen de datos en la ejecución de *k-means* de tal modo que se ha hecho una selección eliminando artistas de frecuencia menor o igual a uno, ya que no iban a tener ningún peso en las recomendaciones dadas por el "Perfil 2".

5.1.2. Entrenamiento del sistema

Como recomendación de referencia con la que comparamos, se ha generado un "baseline" consistente en generar reglas de asociación a partir de todos los usuarios, es decir, considerando un único grupo. El resultado del baseline es que ninguna regla ha sido generada, utilizándose 30% de soporte y 40% de confianza, conforme el patrón definido. Esto demuestra la necesidad de especialización de los perfiles a través de agrupamiento antes de buscar asociaciones entre las elecciones.

Así como se ha descrito en el apartado 4.1, iniciamos el proceso de entrenamiento creando los conjuntos de entrenamiento y de pruebas, de acuerdo con los porcentajes definidos para este estudio, 70% y 30%, respectivamente,

Para la ejecución de *k-means* en este *dataset*, se ha elegido una k igual a 6 para agrupación de perfiles de usuarios, utilizándose como referencia los datos del grafo del método del codo, probando desde 1 hasta 10 clusters (figura 12 y tabla 2).

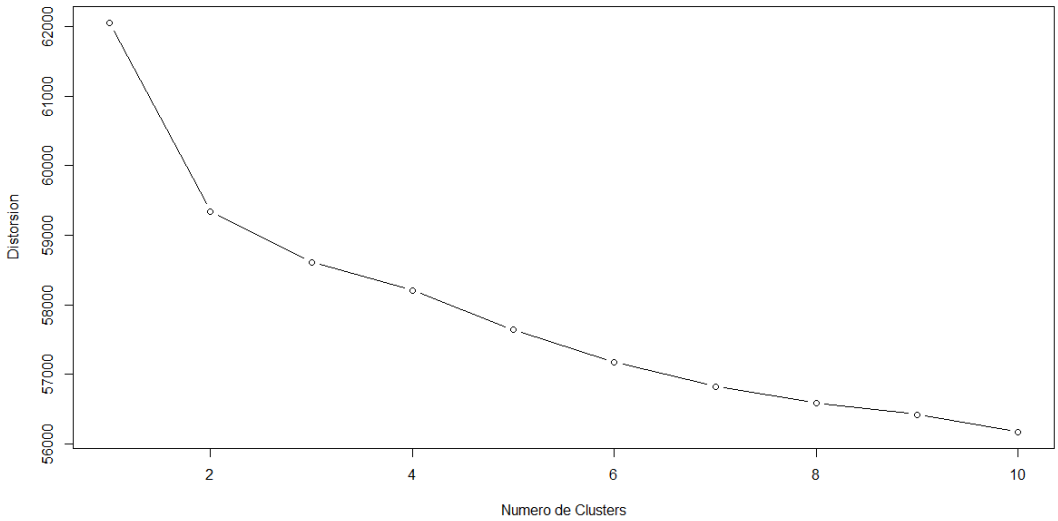


Figura 12 - Grafo del "Elbow Method" para el dataset de LastFM

Cantidad de grupos	Distribución de usuarios por grupo									
	1	2	3	4	5	6	7	8	9	10
1	1324	–	–	–	–	–	–	–	–	–
2	976	348	–	–	–	–	–	–	–	–
3	678	304	342	–	–	–	–	–	–	–
4	618	120	338	248	–	–	–	–	–	–
5	130	110	334	227	523	–	–	–	–	–
6	125	130	331	195	431	112	–	–	–	–
7	126	139	168	195	424	97	175	–	–	–
8	96	86	165	195	389	94	175	124	–	–
9	119	114	159	153	320	65	177	115	102	–
10	115	108	125	110	328	46	129	115	101	147

Tabla 2 - Distribución de usuarios para las diferentes cantidades de grupos probadas

Aunque el valor de la k=2 muestra un codo más acentuado, todavía notamos que hay una grande variación de la distorsión con relación a las próximas k, así que optamos por la k=6.

Tras la ejecución de *k-means*, el proceso de entrenamiento sigue con la ejecución de Apriori con los parámetros mínimos de soporte y confianza en 30% y 40%, respectivamente, establecidos para este estudio. La cantidad de reglas de asociación generadas indican una gran oportunidad de ofrecer recomendaciones para todos los grupos, lo que también refuerza la adecuación de la cantidad de grupos elegida (tabla 3).

	1	2	3	4	5	6
Cantidad de usuarios por grupo	125	130	331	195	431	112
Cantidad de reglas de asociación generadas	63	11	2293	46	0	13

Tabla 3 – Cantidad de usuarios y reglas de asociación generadas por grupo

En estos resultados, también se ha podido observar el patrón ya mencionado anteriormente de que el grupo con la cantidad más grande de usuarios, en este caso el grupo 5,

en verdad agrupa a los usuarios que poseen gustos particulares y que se destacan de los gustos más populares, lo que se evidencia pues no se ha producido ninguna regla de asociación en este grupo en particular. A partir de esta verificación, el sistema sigue con el procesamiento para generación de los subgrupos del grupo 5, utilizando los mismos criterios para determinar la mejor k, pero sólo utilizando datos de este grupo. En la figura 13 se puede notar un ligero codo en el valor de k=5, pero el nivel de distorsión sigue bajando de forma consistente para valores arriba de este punto. Así que seguimos la evaluación de la k observando la distribución de usuarios entre los subgrupos.

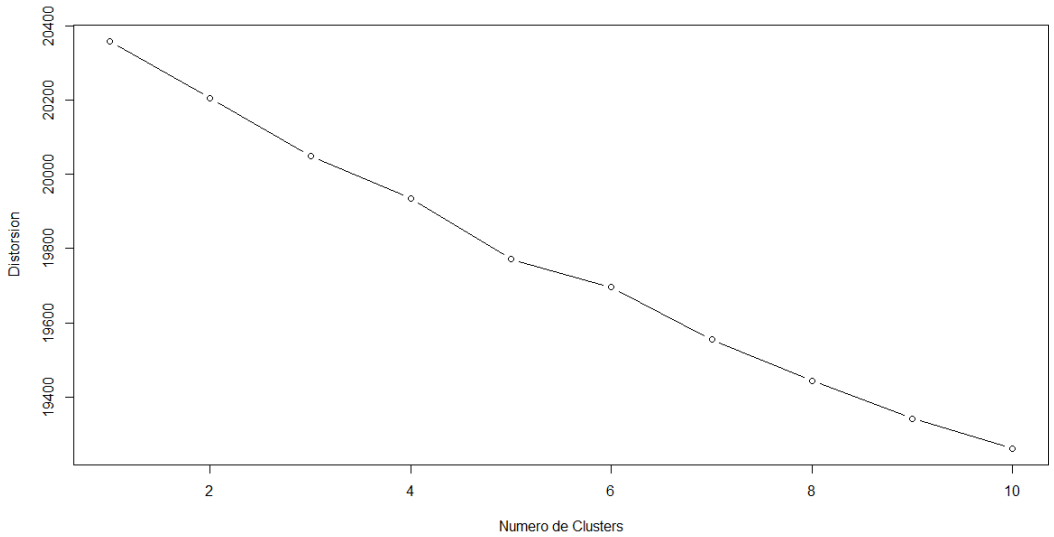


Figura 13 - Grafo del "Elbow Method" para los datos del grupo 5 del dataset de LastFM

En la tabla 4, se puede observar que a partir de la k igual a 6, surge un subgrupo con 3 elementos. Como ya comentamos anteriormente, cuando se ejecuta Apriori en un grupo con pocas transacciones, se genera una cantidad muy grande de reglas debido a los miles de combinaciones entre sus ítems que cumplen con los mínimos de soporte y confianza establecidos, pero que son innecesarias para nuestro propósito y, además, forman un perfil que no aporta valor a las pruebas.

Cantidad de subgrupos	Distribución de usuarios por subgrupo									
	1	2	3	4	5	6	7	8	9	10
1	431	–	–	–	–	–	–	–	–	–
2	401	30	–	–	–	–	–	–	–	–
3	343	27	61	–	–	–	–	–	–	–
4	329	27	45	30	–	–	–	–	–	–
5	54	25	38	20	294	–	–	–	–	–
6	54	25	38	20	291	3	–	–	–	–
7	57	24	19	20	263	3	45	–	–	–
8	67	25	51	17	203	3	46	19	–	–
9	65	24	26	41	153	3	53	18	48	–
10	35	25	23	41	170	3	42	18	48	26

Tabla 4 – Distribución de usuarios en las pruebas de subgrupo

Aunque se puedan gestionar los mínimos de soporte y confianza de forma dinámica en cada interacción para aumentarlos conforme se encuentren grupos con baja cantidad de transacciones, no se consideró hacerlo en este trabajo pues, al definirse la k = 5 para subdividir los elementos de este grupo, las reglas de asociación creadas en estos subgrupos (tabla 5) se mostraron suficientes para hacer recomendaciones en este experimento, y también porque ya se presentaba como posible k por el método del codo.

	1	2	3	4	5
Cantidad de usuarios por subgrupo	54	25	38	20	294
Cantidad de reglas de asociación generadas	5	37	0	27	0

Tabla 5 - Cantidad de usuarios y reglas de asociación generadas por subgrupo

También se ha podido observar que los subgrupos 3 y 5 no generaron ninguna regla de asociación, siendo que solamente el 5 ha producido el patrón descrito anteriormente, pues posee el mayor volumen de transacciones.

Para el entrenamiento del "Perfil 2", el agrupamiento por genero de los artistas ha sido probado con k desde 1 hasta 10 (como las anteriores aplicaciones de *k-means*), y los resultados indicaron que la k = 8 es un valor suficiente para la realización del experimento. Para este valor, la distribución por grupos, mostrada en la tabla 6, es:

K	Distribución de géneros por grupo							
	1	2	3	4	5	6	7	8
8	227	223	820	2768	219	418	328	116

Tabla 6 - Distribución de los géneros por grupo

Se asume como satisfactoria la distribución de la tabla 6 ya que, todos los grupos con perfil de género presentan una cantidad de artistas asociados que ofrecen alta probabilidad de recomendación, así como se mostrará a continuación.

5.1.3. Resultados de los experimentos

Las pruebas fueron ejecutadas con los 568 usuarios del conjunto de test, los cuales presentaron los siguientes resultados en cada experimento:

Experimento 1: los 568 usuarios han obtenido alguna recomendación del sistema con respeto al ítem omitido. En las pruebas de este experimento, se ha notado que, por el hecho de que cada usuario contenga aproximadamente 50 ítems correspondientes a artistas que ha escuchado, al

quitarse un elemento del vector, no se produce cambio de grupo o subgrupo, de forma que se pasó a realizar el cálculo para obtención de grupo solo en la primera iteración, lo que se volvió en gran beneficio para las pruebas pues aumentó considerablemente la eficiencia de la ejecución. La distribución porcentual de éxitos de recomendación separada por estrategia de recomendación sigue conforme la figura 14:

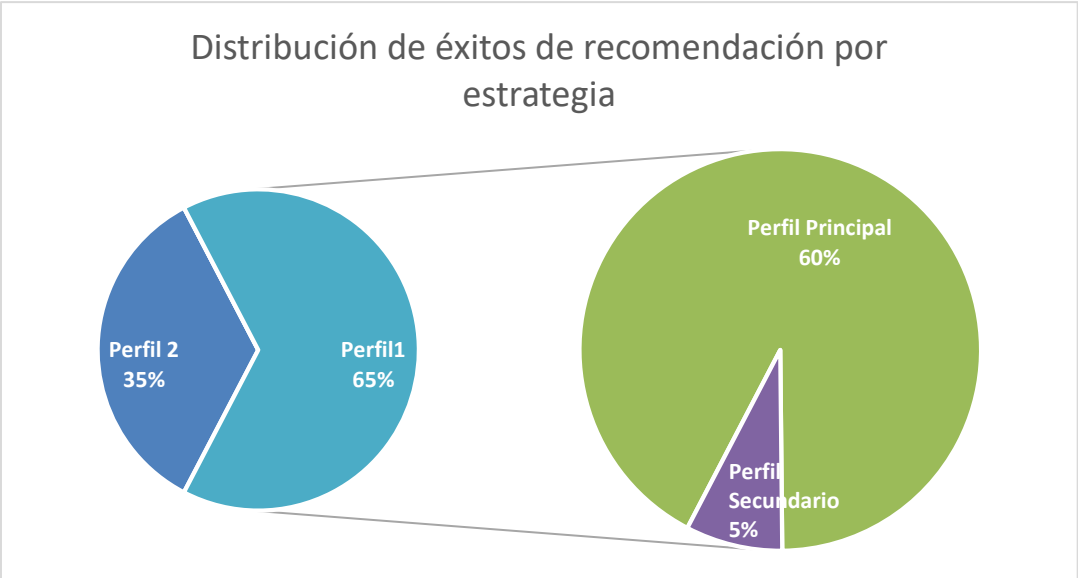


Figura 14 – Grafo de distribución del experimento 1 para LastFM

- **Perfil 1**: 371 usuarios han obtenido recomendación del ítem omitido a través del "Perfil 1", lo que corresponde al 65,32% de los casos probados. El promedio del score general con respecto al criterio de *score* ha sido 37,40% y, para los "*top 3*", el promedio del *score* pasa a 51%. Aún basado en este total, podemos verificar que:
    - **Perfil principal**: La gran mayoría de recomendaciones ha sido de 3 ítems o más que corresponde a casi un 85% de los casos, destacándose el promedio del *score* de los "*top 3*" que es de casi 54%, conforme muestra la tabla 7:

Recomendación de	Cantidad	%	Promedio del *score* general	Promedio del *score* de los "*top 3*"
1 ítem	22	5,93%	36,98%	36,98%
2 ítems	34	9,16%	34,19%	34,19%
3 o más ítems	315	84,91%	37,79%	53,79%

Tabla 7 – Tabla comparativa de la cantidad de ítems recomendados y sus scores

    - **Perfil secundario**: Hemos tenido éxito para hacer recomendaciones a través de subgrupos en 29 casos de prueba que corresponde al 7,81% de las sugerencias hechas por el "Perfil 1". Se ha observado que la concentración

mayor de éxitos de recomendación por subgrupo está entre las que sugieren 1 y 2 ítems, como muestra la tabla 8:

Recomendación de	Cantidad	%	Promedio del *score* general	Promedio del *score* de los "*top 3*"
1 ítem	9	31,03%	34,61%	34,61%
2 ítems	12	41,38%	32,25%	32,25%
3 ítems	3	10,34%	34,32%	34,32%
4 ítems	4	13,79%	29,54%	31,63%
5 ítems	1	3,45%	30,88%	34,77%

Tabla 8 – Tabla comparativa de la cantidad de ítems recomendados y sus scores

- **Perfil 2**: 197 (34,68%) usuarios no recibieron recomendaciones del "Perfil 1", pero han sido atendidos completamente por el flujo alternativo del "Perfil 2", donde un mínimo de 3 ítems ha sido recomendado a cada usuario.

Experimento 2: los 568 usuarios han obtenido recomendación de algún nuevo ítem, y la distribución porcentual de éxitos de recomendación separada por estrategia de recomendación sigue conforme la figura 15.

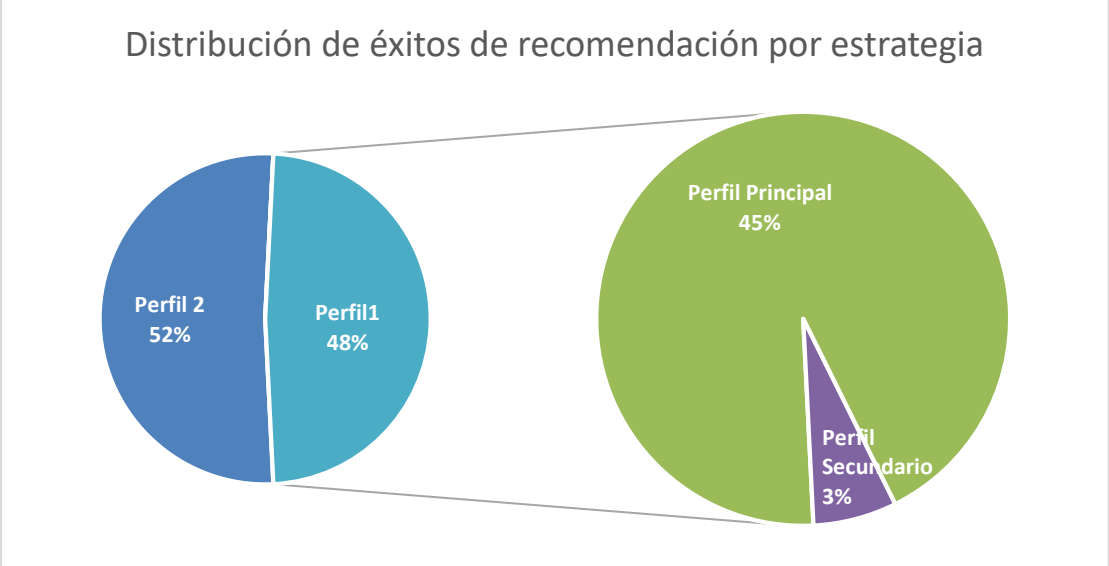


Figura 15 - Grafo de distribución del experimento 2 para LastFM

- **Perfil 1**: 275 usuarios han obtenido recomendación de un nuevo ítem a través del "Perfil 1", lo que corresponde al 48,42% de los casos probados. El promedio general con respecto al *score* ha sido 34,32%, y el promedio de los "*top 3*" sube más de 5%, llegando a 39,55%. Aún basado en este total, podemos verificar que:

- **Perfil principal**: En la tabla 9, podemos ver una gran cantidad de recomendaciones de 3 o más ítems, que corresponden al 72% de los casos, con promedio del score de los "*top 3*" de 44,06%:

Recomendación de	Cantidad	%	Promedio del *score* general	Promedio del *score* de los "*top 3*"
1 ítem	44	16,00%	28,07%	28,07%
2 ítems	33	12,00%	27,83%	27,83%
3 o más ítems	198	72,00%	36,79%	44,06%

Tabla 9 – Tabla comparativa de la cantidad de ítems recomendados y sus scores

- **Perfil secundario:** Hemos tenido éxito en hacer recomendaciones a través de subgrupos en 18 casos de prueba que corresponden al 6,54% de las sugerencias hechas por el "Perfil 1". Así como en el experimento 1, también se observa una gran concentración de éxitos para recomendar 1 ítem a través de los subgrupos, como muestra la tabla 10:

Recomendación de	Cantidad	%	Promedio del *score* general	Promedio del *score* de los "*top 3*"
1 ítem	11	61.11%	29,57%	29,57%
2 ítems	3	16.66%	25,68%	25,68%
3 ítems	2	11.11%	25,68%	25,68%
4 ítems	2	11.11%	25,49%	26,87%

Tabla 10 – Tabla comparativa de la cantidad de ítems recomendados y sus scores

- **Perfil 2**: 293 (51,58%) usuarios no recibieron recomendaciones del "Perfil 1", pero han sido atendidos completamente por el flujo alternativo del "Perfil 2", donde un mínimo de 3 ítems nuevos han sido recomendados a cada uno. También se constata a través de la tabla 11 que el motivo que más ha contribuido para el fracaso en recomendar por el "Perfil 1" corresponde a usuarios que fueron asignados a grupos o subgrupos con ninguna regla de asociación disponible para hacerlo:

Motivo	Cantidad	%
Sin reglas	179	61,09%
Sin recomendación	114	38,91%

Tabla 11 – Motivos de los usuarios que pasaron al "Perfil 2"

5.2. Resultados de MovieLens

5.2.1. Preprocesamiento de los datos

El fichero "ratings.txt" de Movielens contiene un volumen de datos muy grande y requiere una capacidad computacional más grande de lo que había disponible, de forma que ha sido necesario aplicar una técnica de reducción de los datos para realizar el entrenamiento. Se han limitado los códigos de usuario hasta 2000 y códigos de películas hasta 1500.

El fichero "movies.txt" contiene los géneros de películas concatenados en el propio registro de la película. Ha sido necesario convertir la disposición de los datos donde cada registro corresponda a una película y género, pudiendo repetir la misma película para los géneros a los que esté asociado. También se ha filtrado este *dataset* de géneros quitando películas de frecuencia menor o igual a uno, así como se ha hecho con el *dataset* de LastFM, los cuales no hacen falta en el proceso de recomendación de los más frecuentes del "Perfil 2".

5.2.2. Entrenamiento del sistema

El procedimiento inicial para la generación del *baseline* ha producido los mismos resultados que el *dataset* de LastFM lo que confirma la necesidad de especialización de perfiles a través de agrupamiento, tal y como se ha definido en este trabajo.

Así como se ha descrito en el apartado 4.1, iniciamos el proceso de entrenamiento creando los conjuntos de entrenamiento y de pruebas, de acuerdo con los porcentajes definidos para este estudio, 70% y 30%, respectivamente,

Para la ejecución de *k-means* en este *dataset*, se ha elegido la k=6 para agrupación de perfiles de usuarios (perfil principal), utilizándose como referencia los datos del grafo del método del codo (figura 16), probándose la creación desde 1 hasta 10 grupos.

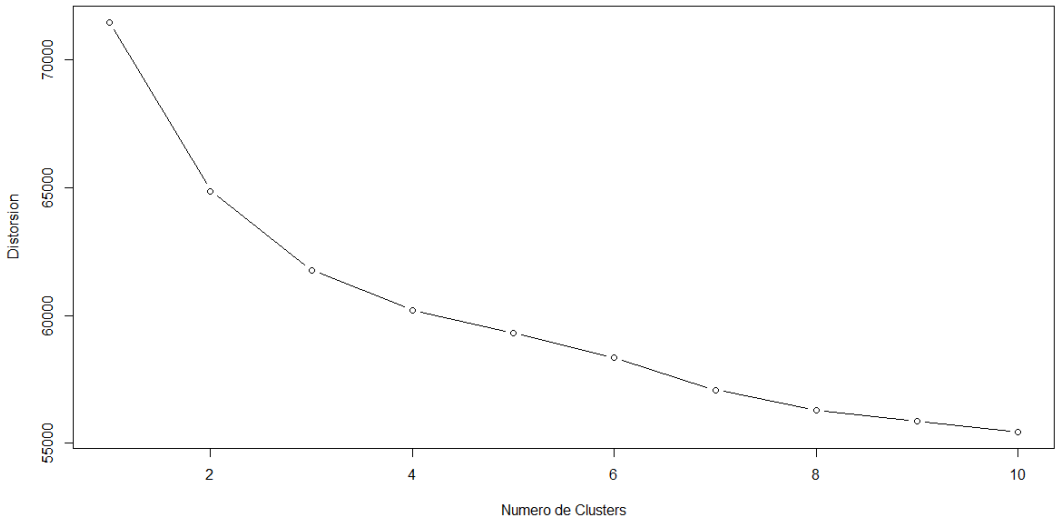


Figura 16 - Grafo del "Elbow Method" para el dataset de MovieLens

Cantidad de grupos	Distribución de usuarios por grupo									
	1	2	3	4	5	6	7	8	9	10
1	1334	-	-	-	-	-	-	-	-	-
2	1014	320	-	-	-	-	-	-	-	-
3	834	257	243	-	-	-	-	-	-	-
4	721	247	114	252	-	-	-	-	-	-
5	244	201	91	83	715	-	-	-	-	-
6	206	197	70	81	709	71	-	-	-	-
7	201	189	70	81	610	71	112	-	-	-
8	199	103	68	83	551	18	112	200	-	-
9	202	53	61	80	538	83	112	193	12	-
10	188	54	61	69	517	58	106	138	12	131

Tabla 12 - Distribución de usuarios para las diferentes cantidades de grupos probadas

Combinado al método del codo, también se ha evaluado la cantidad de transacciones generadas para cada valor de k, conforme la tabla 12, y verificado cuantas reglas generan en cada caso (tabla 13), de modo que, así como en las pruebas de LastFM, el valor de k=6 se ha demostrado adecuado para este *dataset*.

	1	2	3	4	5	6
Cantidad de usuarios por grupo	206	197	70	81	709	71
Cantidad de reglas de asociación generadas	16	211	8416	66	0	81530

Tabla 13 – Cantidad de usuarios y reglas de asociación generadas por grupo

Para la ejecución de Apriori en este *dataset* con los umbrales mínimos de soporte y confianza de 30% y 40%, respectivamente, diversos intentos mostraron que una cantidad de millones de reglas eran generadas en algunos grupos, lo que impidió la terminación del procesamiento con los recursos disponibles. Por este motivo, ambos parámetros fueron

modificados fijándose a 60% y 70%, respectivamente, para la etapa de generación de grupos y sus reglas asociadas (perfil principal). Para el proceso de subgrupos, los umbrales iniciales han resultado adecuados. En este *dataset*, se ha observado el mismo patrón de comportamiento que en el anterior dominio: un grupo contiene la cantidad más grande de usuarios no produciéndose reglas de asociación para el mismo, el cual, por coincidencia, también es el grupo 5. Según los mismos criterios para determinación de la mejor k, repetimos el proceso para los datos del grupo 5 y evaluamos la distribución y la cantidad de reglas de asociación generadas para k = 7 sugerida por el método del codo, que resultó el valor más adecuado, conforme la figura 17:

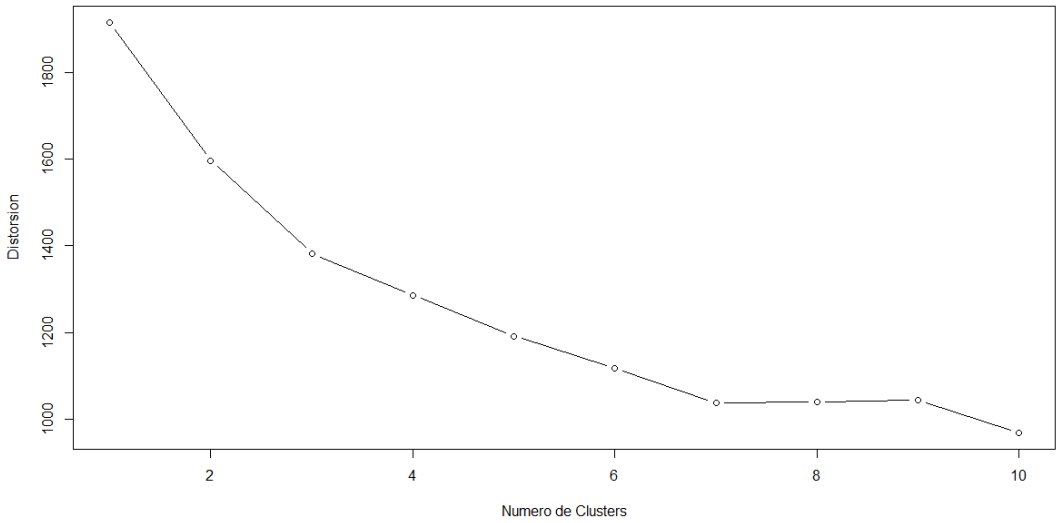


Figura 17 - Grafo del "Elbow Method" para los datos del grupo 5 del dataset de MovieLens

	1	2	3	4	5	6	7
Cantidad de usuarios por subgrupo	289	23	106	75	22	77	117
Cantidad de reglas de asociación generadas	0	251	4209	80	822	215	34

Tabla 14 - Cantidad de usuarios y reglas de asociación generadas por subgrupo

En este entrenamiento de subgrupos, el cluster 1, con la cantidad más grande usuarios, ha confirmado el patrón observado en el *dataset* de LastFM, y tampoco ha generado reglas, tal y como se observa en la tabla 14, por lo que se espera que usuarios asignados al grupo 5 y subgrupo 1 reciban recomendaciones por el "Perfil 2".

Para el entrenamiento del "Perfil 2", el agrupamiento de los ítems (películas) ha sido probado con k desde 1 hasta 10, y los resultados indicaron que k = 9 es el valor más adecuado, como se muestra en la figura 18. La distribución de ítems se muestra en la tabla 15.

K	Distribución de ítems por grupo								
	1	2	3	4	5	6	7	8	9
9	898	709	404	703	737	1805	760	1044	3621

Tabla 15 - Distribución de ítems por grupo

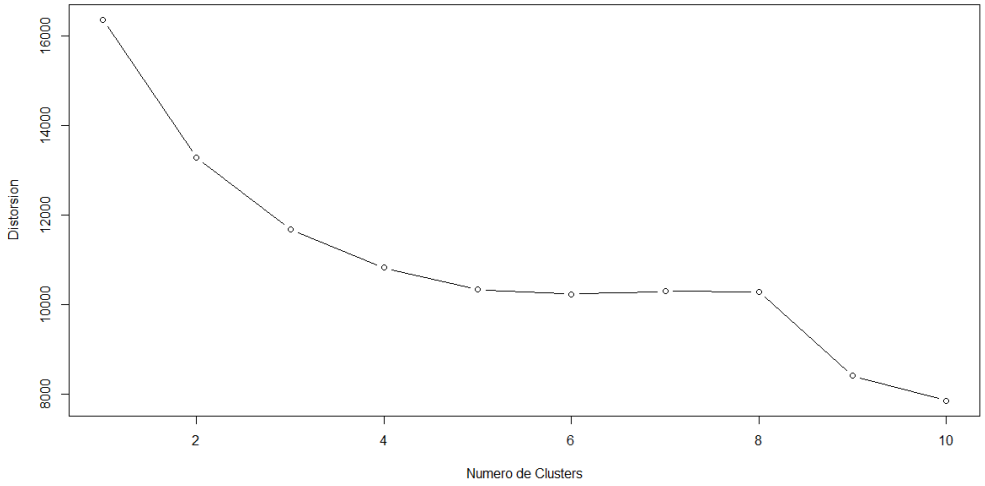


Figura 18 - Grafo del "Elbow Method" para el “Perfil 2”

5.2.3. Resultados de los experimentos

Las pruebas fueron ejecutadas con los 571 usuarios del conjunto de test, los cuales presentaron los siguientes resultados en cada experimento:

Experimento 1: los 571 usuarios han obtenido alguna recomendación del sistema con respeto al ítem omitido. Diferente de las pruebas con LastFM, para el *dataset* de MovieLens se ha recalculado el grupo y subgrupo al que pertenece el cliente a cada iteración, pues la cantidad de ítems cambiaba mucho entre los usuarios, pudiéndose que ocurriera un cambio de grupo, dependiendo del ítem omitido. La distribución porcentual de éxitos de recomendación separada por estrategia de recomendación sigue conforme la figura 19:

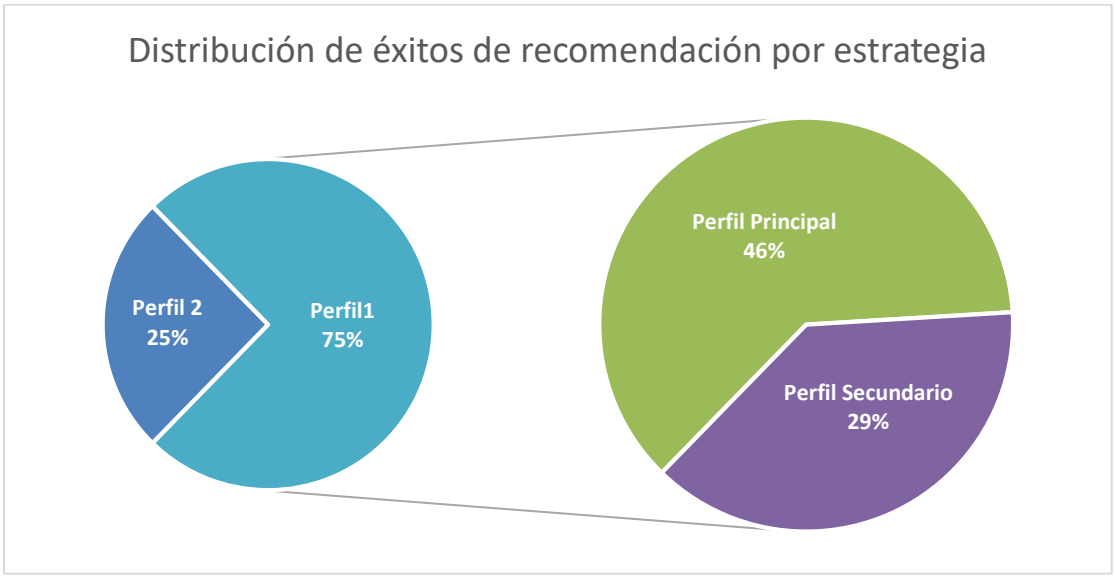


Figura 19 - Grafo de distribución del experimento 1 para MovieLens

- **Perfil 1:** 426 usuarios han obtenido recomendación del ítem omitido a través del "Perfil 1", lo que corresponde 74,61% de los casos probados. El promedio general del *score* ha sido 49,41% y, considerando el escenario de los "*top 3*", el promedio sube considerablemente a 63,02%. Aún basado en este total, podemos verificar que:
    - **Perfil principal:** en este perfil, la gran mayoría de recomendaciones han sido de 3 ítems o más, lo que corresponden al 95,54% de los casos sugeridos por el "Perfil 1", destacándose el promedio del score del "*top 3*" de 64,32%, conforme muestra la tabla 16:

Recomendación de	Cantidad	%	Promedio del *score* general	Promedio del *score* de los "*top 3*"
1 ítem	5	1,17%	24,34%	24,34%
2 ítems	14	3,29%	38,92%	38,92%
3 o más ítems	407	95,54%	50,08%	64,32%

Tabla 16 – Tabla comparativa de la cantidad de ítems recomendados y sus scores

    - **Perfil secundario:** Hemos tenido éxito para hacer recomendaciones a través de subgrupos en 163 casos, siendo responsable del 38,26% del total de sugerencias del "Perfil 1". Para este *dataset*, los subgrupos fueron extremamente exitosos como estrategia alternativa de recomendación pues, 94 de los 163 casos, o sea, más de 50%, recomiendan entre 3 y 11 ítems.
- **Perfil 2**: 145 (25,39%) usuarios no recibieron recomendaciones del "Perfil 1", pero han sido atendidos completamente por el flujo alternativo del "Perfil 2", donde un mínimo de 3 ítems han sido recomendados a cada uno.

Experimento 2: los 571 usuarios han obtenido recomendación de algún nuevo ítem, y la distribución porcentual de éxitos de recomendación separada por estrategia de recomendación sigue conforme la figura 20:

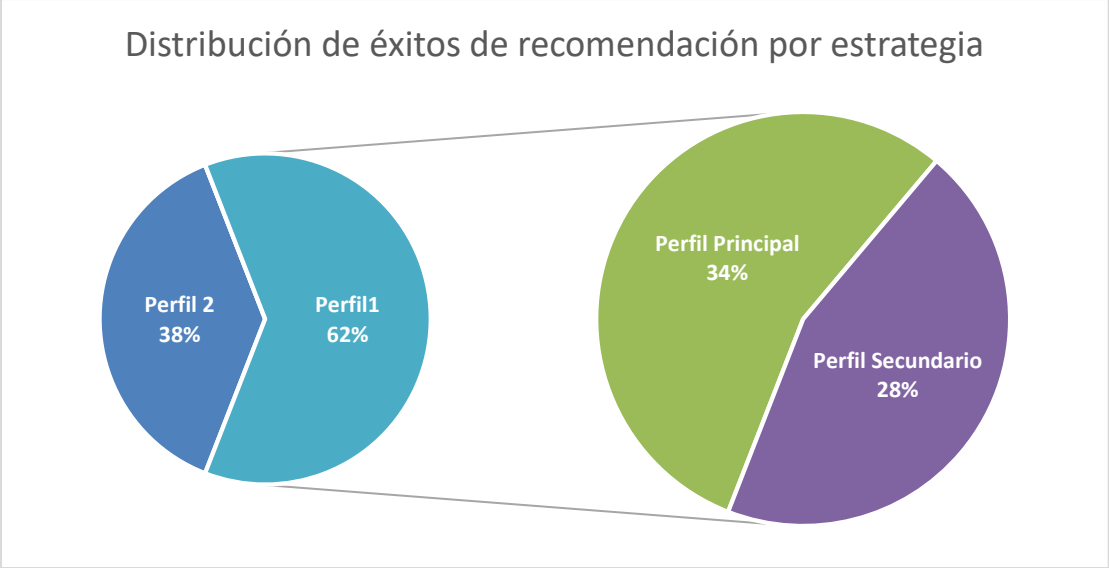


Figura 20 - Grafo de distribución del experimento 2 para MovieLens

- **Perfil 1**: 353 usuarios han obtenido recomendación de un nuevo ítem a través del "Perfil 1", lo que corresponde 61.82% de los casos probados. El promedio general del *score* ha sido 47.09% y el promedio de los "*top 3*" salta a 52,53%. Aún basado en este total, podemos verificar que:
    - **Perfil principal**: Repitiendo el éxito del experimento 1, la gran mayoría de recomendaciones han sido de 3 ítems o más en casi un 61% de los casos, destacándose el promedio del score del "*top 3*" de casi 56%, conforme muestra la tabla 17:

Recomendación de	Cantidad	%	Promedio del *score* general	Promedio del *score* de los "*top 3*"
1 ítem	79	22,38%	49,05%	49,05%
2 ítems	59	16,71%	45,45%	45,45%
3 o más ítems	215	60,91%	46,82%	55,75%

Tabla 17 – Tabla comparativa de la cantidad de ítems recomendados y sus scores

    - **Perfil secundario:** Hemos tenido éxito para hacer recomendaciones a través de subgrupos en 158 casos, siendo responsable por 44,75% de las sugerencias del "Perfil 1", lo que resalta todavía más la efectividad de esta alternativa para la recomendación si comparamos al experimento 1, que ya había sido muy expresiva.
- **Perfil 2:** 218 (38,18%) usuarios no recibieron recomendaciones del "Perfil 1", pero han sido atendidos completamente por el flujo alternativo del "Perfil 2", donde un

mínimo de 3 ítems nuevos han sido recomendados a cada uno. También se constata a través de la tabla 18, que el motivo que más ha contribuido para el fracaso en recomendar por el "Perfil 1" corresponde a usuarios que fueron asignados a grupos o subgrupos con ninguna regla de asociación disponible para hacerlo:

Motivo	Cantidad	%
Sin reglas	132	60,55%
Sin recomendación	86	39,45%

Tabla 18 – Motivos de los usuarios que pasaron al "Perfil 2"

5.3. Análisis y discusión de los resultados

Con estos dos *datasets* (pertenecientes a dominios diferentes) hemos podido demostrar la efectividad de nuestro sistema de recomendación. Si consideramos las recomendaciones bajo el "Perfil 1" en el experimento 1, 65,32% para LastFM y 74,61% para MovieLens, podemos notar la gran capacidad de predicción del sistema, una vez que estamos probando recomendar ítems que seguramente fueron elegidos por los usuarios. Y este número se vuelve aún más atractivo si pensamos que el sistema solo ha sido entrenado con elecciones previas de los usuarios, sin ninguna otra información sobre ellos.

Las recomendaciones de los "*top 3*" se mostraron muy interesantes con promedios del *score* significativamente superiores en algunos casos, comparándose a los promedios del *score* general. Además, también mostramos que la gran mayoría de recomendaciones han sido de 3 ítems o más, lo que permitiría la utilización de esta estrategia en los ámbitos en que pueda aplicarse.

El hecho de probarse el prototipo de recomendación con dos *datasets* distintos se ha mostrado esencial pues hemos podido obtener respuestas más relevantes sobre una estrategia de un *dataset* que de otro, y también confirmar tendencias sobre el tema de las preferencias, que se manifestaron en ambos. Por mencionar un ejemplo, un grupo formado por usuarios de gustos particulares y elecciones diversificadas, donde no se generan reglas de asociación, nos muestra que en la mayoría de sus subgrupos pueden generarse reglas de asociación, pero que todavía puede presentarse un subgrupo sin reglas, lo que nos ha permitido ver la necesidad de crearse más alternativas para el intento de recomendar. También pudimos constatar que el valor de *k* más adecuado para decidir la cantidad de grupos y subgrupos es muy dependiente del dominio y que su determinación afecta a la efectividad de nuestras recomendaciones. Y por eso consideramos una buena práctica en nuestros experimentos probar distintos valores de la *k*

además de lo sugerido por los métodos convencionales, como el método del codo utilizado en nuestro trabajo.

Gracias a la estrategia ítem a ítem aplicada a los perfiles de género, combinada a la selección de los ítems más frecuentes, el "Perfil 2" se ha mostrado muy eficiente al conseguir atender el 100% de los casos que le han sido pasados, ofreciendo al mínimo 3 opciones distintas en todos ellos.

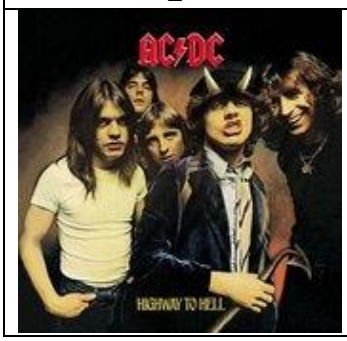
Además de un análisis cuantitativo sobre los éxitos de recomendación bajo cada estrategia, buscamos alternativas para hacer una comprobación cualitativa de los resultados, intentando verificar la coherencia de algunas recomendaciones producidas por el sistema.

La primera forma de comprobación cualitativa ha sido buscar ejemplos coincidentes de recomendación del sistema comparando con los de los propios sitios web de LastFM y MovieLens, que son proveedores de los *datasets* utilizados en este trabajo. La idea es verificar similitudes entre las recomendaciones de ambos, no siendo muy restrictivo con respeto a la expectativa de este método, dado que hay diferencias entre las estrategias de nuestro prototipo y los motores de recomendación de ambas plataformas, además de los datos previos suministrados a cada uno. Puestas estas consideraciones, hicimos un acceso a cada uno de los sitios sin que haya ningún historial de navegación o cookies registradas. A continuación, elegimos un ítem y miramos si se apuntan en los ítems similares alguno que nuestro sistema también ha recomendado.

En la segunda forma de comprobación cualitativa, aunque sea más subjetiva e imprecisa, pues no tenemos un "indicador de coherencia", nos basamos en nuestro propio conocimiento general sobre los datos de ambos *datasets*, seleccionando una muestra de casos de prueba que contengan artistas y películas con mayor popularidad en el ámbito del cine y de la música, para evaluar la coherencia de la recomendación ofrecida por nuestro sistema. Con esto, se intenta acreditar que el sistema pueda ser fiable y razonable al ofrecer recomendaciones, aproximándose a la experiencia de recibir sugerencias de un servicio que tenga muchos años de experiencia atendiendo a sus clientes y pleno dominio del catálogo de ítems.

A continuación, mostramos la muestra de usuarios seleccionados y las respectivas recomendaciones del nuestro sistema según el "*top 3*", tomando un ejemplo de cada estrategia de recomendación:

- El usuario 377 de LastFM tiene entre los ítems de su historial: *Slayer, Iron Maiden, Megadeth, Dream Theater, Led Zeppelin, Anthrax, Dave Matthews Band, Avenged Sevenfold y Rage Against the Machine.* Y nuestro sistema le ha recomendado los siguientes artistas a través del perfil principal del "Perfil 1":

1º	2º	3º
		

Si seleccionamos un artista de gran popularidad de su historial, como *Iron Maiden*, y lo buscamos en el sitio web de LastFM, entre los artistas similares, nos ha mostrado *Metallica* en la primera página (figura 21) y *AC/DC* en la segunda (figura 22), lo que coincide con 2 de las recomendaciones de nuestro sistema.

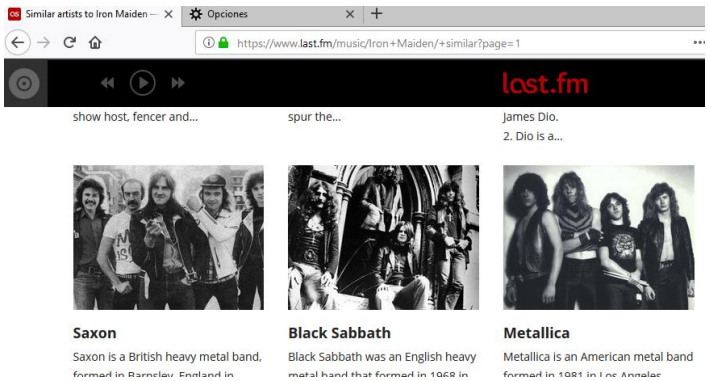


Figura 21 - Recomendación de Metallica por LastFM

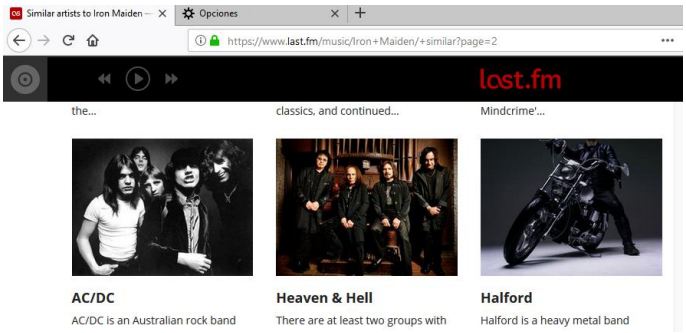
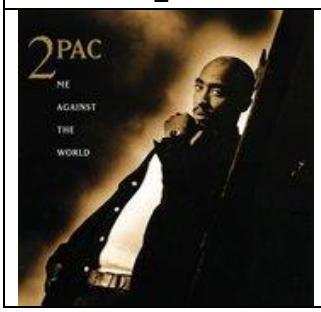


Figura 22 - Recomendación de AC/DC por LastFM

- El usuario 842 de LastFM tiene entre los ítems de su historial: *Kanye West, The Roots, Zomby, Jay-Z, Atmosphere, De La Soul, DJ Krush, Ellie Goulding y The Sonics.* Y nuestro sistema le ha recomendado los siguientes artistas a través del perfil secundario del "Perfil 1":

1º	2º	3º
		

Visto que se trata de una recomendación basada en el perfil secundario, los gustos de este usuario pueden no ser de conocimiento general, y por eso buscamos por uno de los artistas de popularidad más destacada en este estilo musical entre los que estaban presentes su historial. Seleccionamos *Jay-Z* y lo buscamos en el sitio web de LastFM, nos ha mostrado la recomendación de *2Pac* como artista similar en la tercera página de navegación (figura 23), lo que coincide a nuestra primera recomendación.

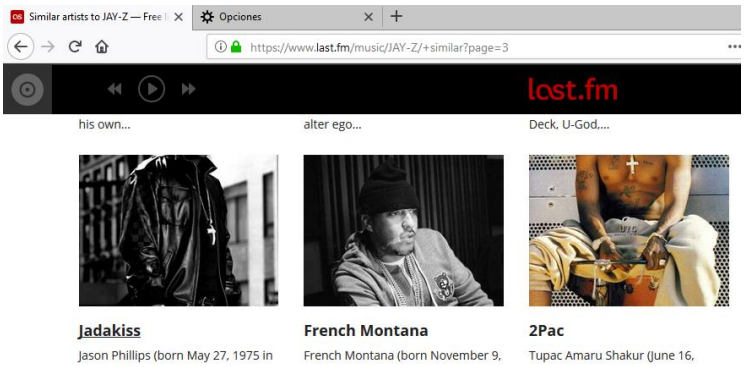
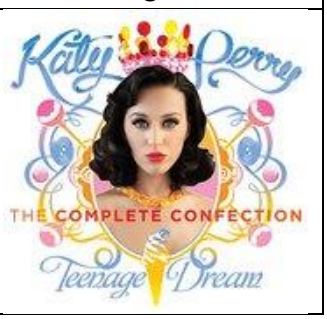


Figura 23 - Recomendación de 2Pac por LastFM

- El usuario 27 de LastFM tiene entre los ítems de su historial: *Duran Duran, Madonna, David Guetta, Laura Pausini, Shakira, Yanni, Blondie, Frank Sinatra y Earth, Wind & Fire.* Y nuestro sistema le ha recomendado los siguientes artistas a través del "Perfil 2":

1º	2º	3º
		

Si seleccionamos un artista de gran popularidad de su historial, como *Madonna*, y lo buscamos en el sitio web de LastFM, una de las primeras recomendaciones es *Britney Spears* (figura 24), lo que coincide a la segunda opción recomendada por nuestro sistema.

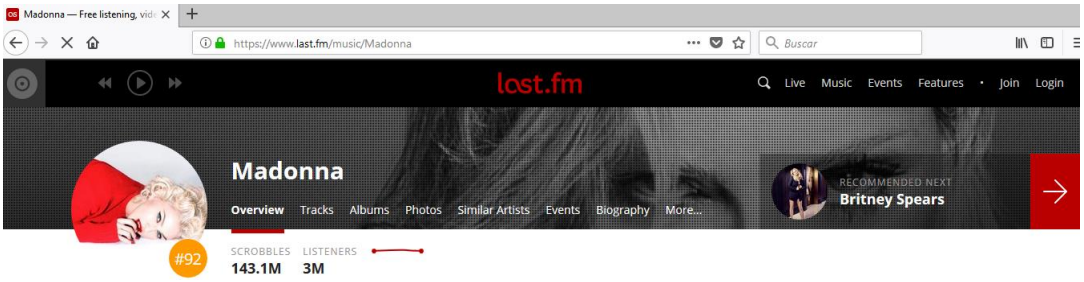
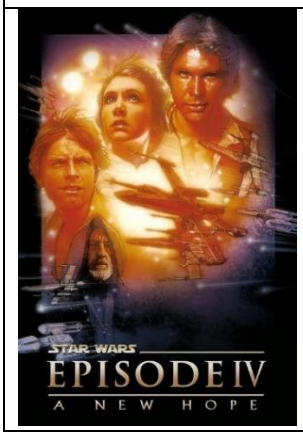
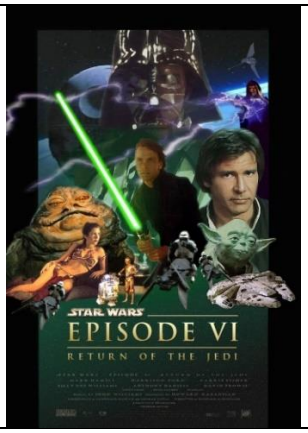
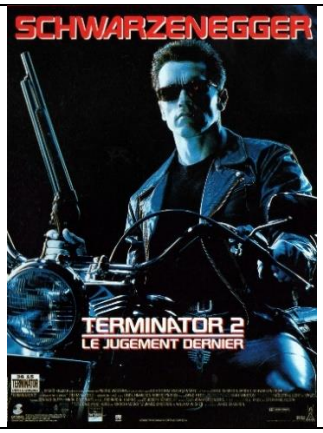


Figura 24 - Recomendación de Britney Spears por LastFM

- El usuario 149 de MovieLens tiene entre los ítems de su historial: *Star Wars: Episode V - The Empire Strikes Back, Apollo 13, Pulp Fiction, Forrest Gump, Alien, Blade Runner, Batman, The Godfather y Star Trek: First Contact*. Y nuestro sistema le ha recomendado las siguientes películas a través del perfil principal del "Perfil 1":

1º	2º	3º
		

Si seleccionamos una de las películas de gran popularidad de su historial, como *Star Wars: Episode V - The Empire Strikes Back*, y la introducimos en el sitio web de *MovieLens*,

la primera página de recomendaciones (figura 25) nos ofrece las mismas 3 recomendaciones que nuestro sistema.

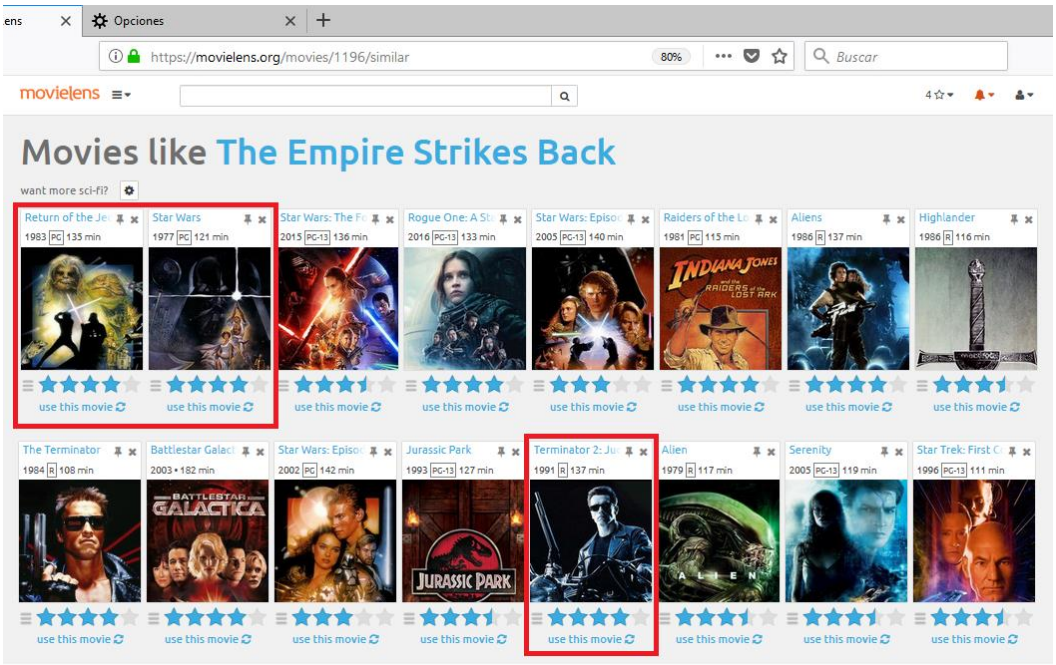
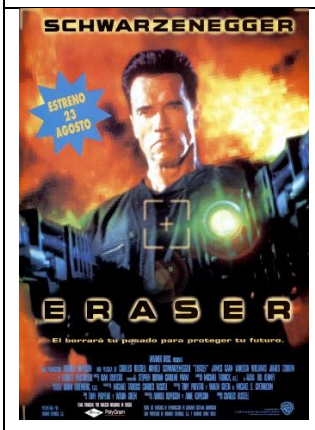
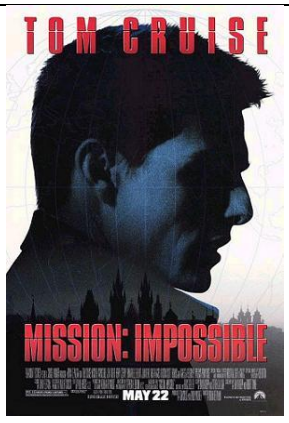
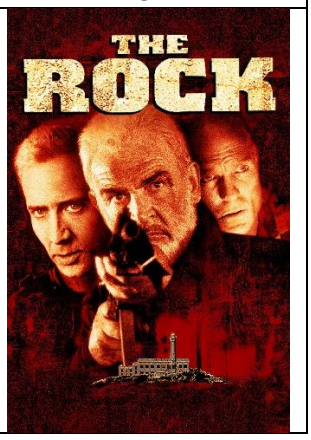


Figura 25 - Recomendación de películas por MovieLens

- El usuario 120 de MovieLens tiene entre los ítems de su historial: *Broken Arrow*, *Star Wars: Episode IV - A New Hope*, *Twister*, *Independence Day*, *Willy Wonka & the Chocolate Factory*, *Star Wars: Episode VI - Return of the Jedi* y *Jerry Maguire*. Y nuestro sistema le ha recomendado las siguientes películas a través del perfil secundario del "Perfil 1":

1º	2º	3º
		

Siendo un usuario que presenta gustos diversificados, y por eso ha sido recomendado por el perfil secundario, no encontramos coincidencias con nuestras recomendaciones al buscar por las películas de más popularidad en el sitio web de MovieLens, como las dos películas de

Star Wars o *Independence Day*. Pero encontramos la misma recomendación que nuestro sistema para *The Rock* (figura 26), cuando buscamos por el ítem *Broken Arrow*.

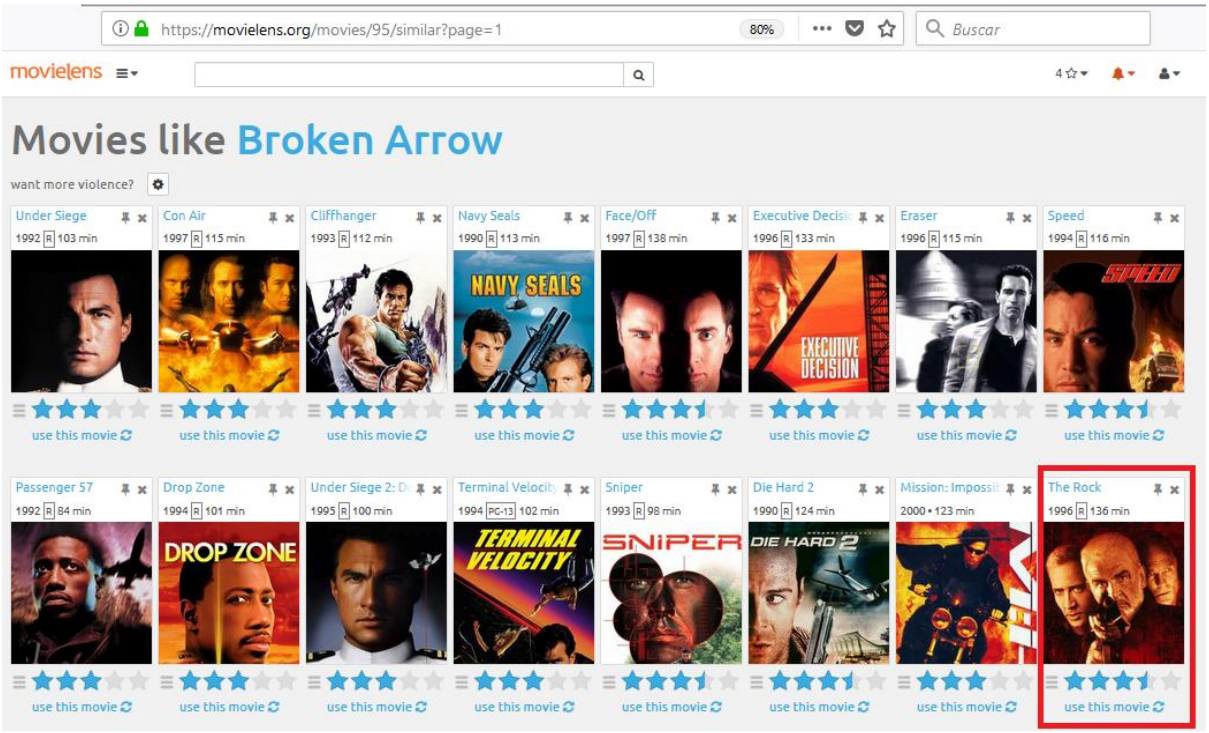
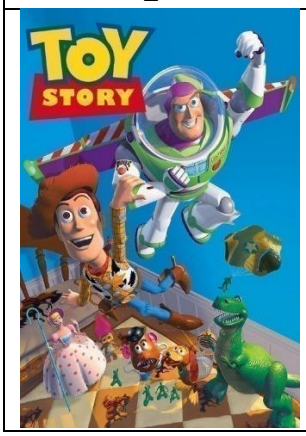
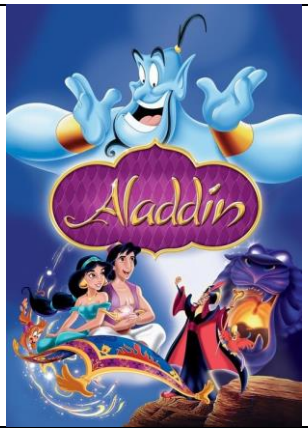
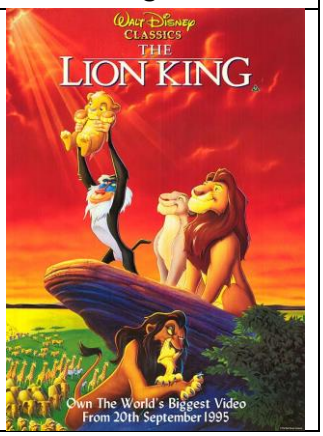


Figura 26 - Recomendación de *The Rock* por MovieLens

- El usuario 51 de MovieLens tiene entre los ítems de su historial: *The Wizard of Oz, Mary Poppins, The Return of the Pink Panther, Cinema Paradiso, Fantasia y Grease.* Y nuestro sistema le ha recomendado las siguientes películas a través del "Perfil 2":

1º	2º	3º
		

Si seleccionamos una película de gran popularidad de su historial, como *Fantasia*, y la

buscamos en el sitio web de MovieLens, la tercera página de recomendaciones (figura 27) ofrece *Alladin* y *Lion King*, que coinciden con la segunda y la tercera recomendación de nuestro sistema.

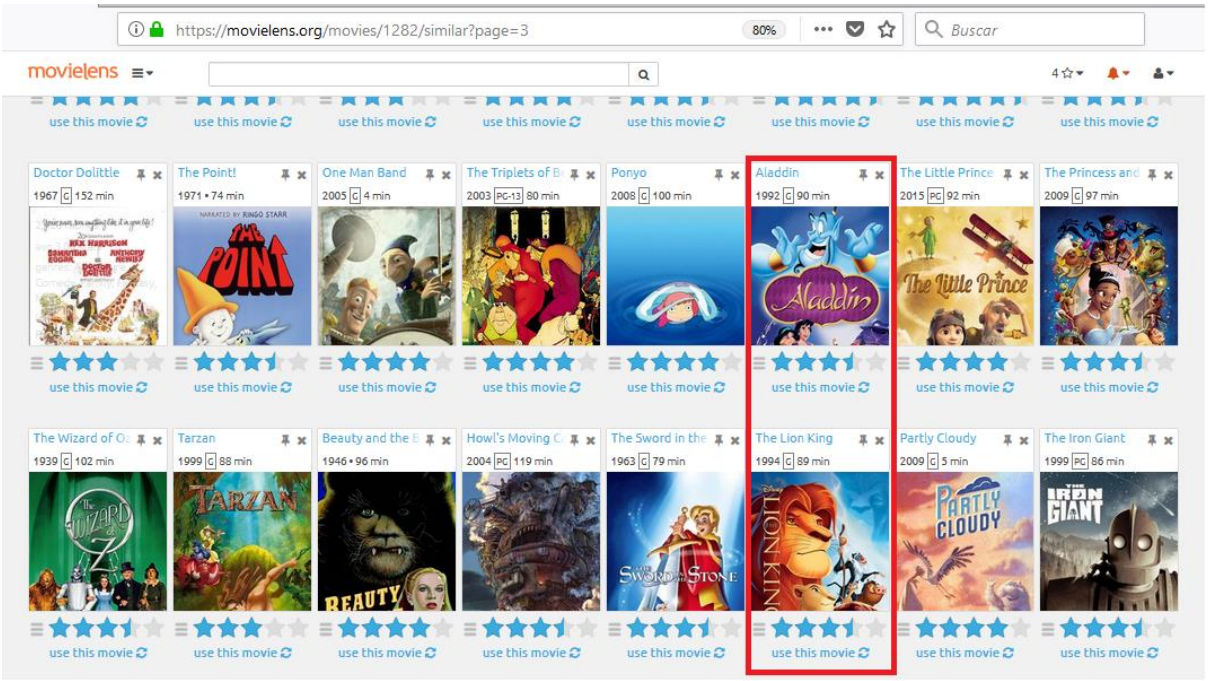


Figura 27 - Recomendación de películas por MovieLens

Con base en estos ejemplos, podemos verificar que encontramos diversos ítems similares entre las recomendaciones de nuestro sistema de recomendación y ambos sitios web, lo que satisface la demostración de coherencia y credibilidad que se intentaba obtener. Además, consideramos las elecciones previas de cada ejemplo muy alineadas a las recomendaciones en términos de género y tipos de gustos detectados entre ellas.

6. Conclusiones y trabajos futuros

La construcción de un prototipo híbrido para recomendaciones, utilizando filtrado colaborativo asociado al filtrado por contenido, siguiendo la misma orientación de los modelos adoptados por las empresas de referencia en el tema como Amazon y Netflix, ha demostrado ser muy eficiente para la generación de recomendaciones tanto para los escenarios más triviales en términos de detección perfiles de gustos e ítems frecuentes, cuanto para los escenarios de gustos más diversificados y particulares.

El filtrado colaborativo de nuestro modelo ha utilizado el *Model-based techniques* pues compone las tareas de agrupamiento y asociación para la generación de recomendaciones en primera y segunda instancia, es decir, en cascada. La primera instancia hace recomendaciones basada en grupos generados por *k-means* y reglas de asociación producidas por Apriori, mientras que la segunda instancia se utiliza de la misma combinación de tareas, pero aplicada a un grupo que se ha quedado sin reglas de asociación en primera instancia.

El reto presentado por la tarea de agrupamiento, realizado usando el algoritmo de *k-means*, nos ha conducido a buscar un método para determinar el número adecuado de grupos (parámetro *k*) que se reveló muy efectivo y adecuado para los fines perseguidos por este trabajo. La combinación del método del codo, de la evaluación de los tamaños de grupos generados por *k-means* para el rango de 1 hasta 10 grupos, y la cantidad de reglas generadas por Apriori para cada grupo forman parte del método que ha solucionado este reto. Al utilizarlo, logramos producir una cantidad suficiente y satisfactoria de reglas de asociación para los grupos responsables por las recomendaciones del "Perfil 1". También, hemos percibido que adoptar un porcentaje más bajo para el soporte y confianza en las tareas de asociación, ejecutadas a través de Apriori, ha sido una estrategia acertada para la exploración de más asociaciones entre ítems frecuentes, incluso por el hecho de que a veces se generaba una gran cantidad de reglas pero que era compensado por el proceso de eliminación de reglas redundantes, lo cual descartaba las reglas menos generales y de menor confianza, sin causar ningún perjuicio al sistema.

El flujo principal de recomendación, denominado "Perfil 1" basado en grupos o perfil principal, ha sido responsable del 60% de las sugerencias hechas a los 568 usuarios de prueba en el experimento 1 de LastFM, y del 48% de las sugerencias hechas a esta misma masa de pruebas en el experimento 2. En el caso de MovieLens, los resultados de recomendación bajo

este perfil han sido más bajos, pero igualmente satisfactorios para el objetivo de estudio planteado en este trabajo.

Los grupos que no produjeron reglas de asociación aplicando el perfil principal, lo que suponía no realizar recomendación alguna para ellos, fueron sometidos a un nuevo proceso de agrupamiento y generación de reglas de asociación, lo que denominamos perfil secundario. Los grupos que en el "Perfil 1" no han producido reglas de asociación son muy heterogéneos y contienen clientes con gustos diversificados y particulares, los cuales se distinguen de perfiles de gustos más comunes y populares recogidos en los otros grupos. Clientes con ese perfil son redireccionados al perfil secundario para recibir sus recomendaciones. Este flujo alternativo de recomendación ha demostrado gran potencial de solución para los casos particulares, por haber sugerido a 5% y 3% del total de usuarios probados en LastFM para los experimentos 1 y 2, respectivamente. Y, de forma aún más evidente, las recomendaciones a través de los subgrupos han solucionado 29% y 28% de los casos de prueba de MovieLens en estos mismos experimentos, lo que es un porcentaje alto si pensamos en grupos y reglas formadas para gustos particulares.

Pero también hemos visto que no todos los perfiles de usuario recibían recomendación a través del "Perfil 1" por dos motivos posibles:

- el usuario podría ser asignado a un grupo con reglas, pero que ninguna le pudiera recomendar algo por no coincidencia entre su historial y los antecesores de las reglas
- el usuario podría ser asignado a un subgrupo del flujo alternativo que no contiene reglas

Para tratamiento de estas dos excepciones no resueltas en los flujos de primera y segunda instancia, se ha creado una instancia final basada en filtrado de contenidos, denominada "Perfil 2", que combina el agrupamiento de ítems y la frecuencia en que estos son elegidos por los usuarios. La combinación entre perfiles de tipos de ítems y frecuencias demostró gran capacidad para hacer recomendaciones de calidad, una vez que ha ofrecido recomendaciones a 100% de los casos de excepción mencionados. Este flujo alternativo ha sido responsable del tratamiento del 35% y 52% del total de usuarios probados en LastFM para los experimentos 1 y 2, respectivamente. Para MovieLens, debido al gran éxito del tratamiento por los subgrupos, solamente 25% y 38% del total de usuarios probados recibieron recomendaciones por el "Perfil 2" para los mismos experimentos.

Por los resultados cuantitativos y cualitativos verificados en este estudio, el prototipo creado ha demostrado plena capacidad de convertirse en un sistema de recomendaciones poco

invasivo, sencillo, barato y que podría usarse en aquellos ámbitos en los que se recogen pocos datos de clientes e ítems.

Trabajos futuros

Además de lo mencionado, otra posible e interesante extensión de este trabajo sería adaptar el sistema creado a los *datasets* con características de tiques de compra. Este es un ámbito con una gran proyección (como demuestra Amazon). Los sistemas de recomendación que tienen en cuenta el tiempo deberían ser considerados en este estudio para que el orden de adquisición de los ítems sea un factor de calidad para las recomendaciones. Por ejemplo, se podría recomendar a un cliente una impresora u otros accesorios informáticos, si se sabe que la última compra del mismo fue un ordenador. Es fácil comprender la necesidad de tener en cuenta el tiempo si planteamos el ejemplo al revés, es decir, recomendar una impresora o accesorio a un cliente que ni siquiera se sabe si tiene un ordenador puede generar en el cliente un efecto disonante y pérdida de credibilidad en el sitio, al contrario de lo que se intenta a través de las recomendaciones.

Otra posible extensión sería el propio estudio del desarrollo de una solución completa basada en este prototipo y en las funcionalidades de personalización sugeridas a lo largo de este trabajo. El estudio sobre el sistema completo podría explorar todos los parámetros y configuraciones de comportamiento que se podrían desarrollar, como las siguientes sugerencias:

- adopción de porcentaje de soporte y confianza fijos o dinámicos, de acuerdo con la cantidad de transacciones de cada grupo.
- elegir el flujo alternativo para cada caso de excepción identificado, con personalización a través de parámetros de los usuarios.
- opción para recomendar todos los ítems encontrados o los “top x” que acumulan mayor *score*, donde x seria definido por el usuario.
- si no hay un mínimo de x ítems para recomendar, se acciona otro flujo, donde x sería definido por el usuario
- parámetro para recomendar ítems ya adquiridos (ejemplo, ver una película repetida).
- parámetros para mínimo y máximo de ítems a recomendar por el perfil 2.

Adicionalmente a lo indicado sobre este sistema y los resultados obtenidos, una gran preocupación durante la creación del sistema ha sido evitar al máximo que un cliente se quede sin alguna recomendación. Otras opciones para el tratamiento de casos de excepción para los que no se generan recomendaciones, podrían ser también incorporadas al sistema de recomendación:

- **Ítems independientes de antecesor**: A través de la ejecución de Apriori, hay reglas de asociación generadas donde el antecesor es vacío, o sea, la frecuencia del consecuente es alta independiente de cualquier ítem. Si el usuario es nuevo, o sea, sin historial de elecciones, no hay como asignarlo a un grupo. En este caso, este flujo alternativo podría recoger las reglas donde el antecesor es vacío en todos los grupos formados en el entrenamiento, y se recomendarían sus respectivos ítems consecuentes. Pero, si el usuario no es nuevo, entonces ha sido asignado a un grupo sin reglas de recomendación. En este caso, el flujo alternativo podría recoger todas las reglas del grupo asignado donde el antecesor es vacío y recomendar la lista de consecuentes.
- **Los más elegidos**: esta recomendación sería una selección pura de los ítems más elegidos en la tienda según el criterio de tiempo definido por la misma, por ejemplo, los 10 productos más elegidos en la última semana. Esta es una acción disponible en la mayoría de sistemas de recomendación actuales.
- **Ítems nuevos**: Como solución al problema del "comienzo frío" para los nuevos ítems del catálogo, se podría enseñar las novedades del catálogo o bien añadir una subsección a las alternativas anteriores, siendo una oportunidad de divulgarles y estimular la generación de sus historiales de elección.
- **No recomendar**: aunque esta alternativa parezca contradictoria al propósito de este trabajo, hay que considerarla como una opción también válida para no generar rechazo o disgusto entre los clientes ante recomendaciones no deseadas.

A continuación, resumimos en la tabla 19 algunos posibles escenarios y las acciones que se podrían realizar en el ámbito de nuestra aproximación:

Escenario	Acciones
Es un usuario nuevo y no hay historial de elecciones para determinar su perfil.	Recomendar: • Los más elegidos + ítems nuevos • Ítems independientes de antecesor

No es un usuario nuevo, pero tiene un historial pequeño con una cantidad de elecciones considerada insuficiente para encajarlo en un grupo, y hacer una recomendación adecuada. La cantidad mínima de elecciones podría ser un parámetro por determinar dependiendo del ámbito.	Si no se ha recomendado nada por el "Perfil 2", recomendar: • Los más vistos + ítems nuevos • Ítems independientes de antecesor de los grupos en general
El usuario fue asignado a un grupo donde ninguna de sus elecciones o combinación de elecciones anteriores corresponde a un antecesor de reglas del grupo.	Si no se ha recomendado nada por el "Perfil 2", recomendar: • Los más vistos + ítems nuevos • Ítems independientes de antecesor de su grupo

Tabla 19 – Tabla de escenarios de excepción y acciones sugeridas

7. Referencias

[1] Narendra Sharma, Aman Bajpai, Mr. Ratnesh Litoriya, Comparison the various clustering algorithms of weka tools, ISSN 2250-2459,Volume 2, Issue 5, May 2012.

[2] Toby Segaran, Collective Intelligence, O'Reilly, ISBN-10: 0596529325, ISBN-13: 978-0596529321.

[3] Weiyang Lin, Association Rule Mining for Collaborative Recommender Systems, A Thesis Submitted to the Faculty of the WORCESTER POLYTECHNIC INSTITUTE

[4] F.O. Isinkaye, Y.O. Folajimi, B.A. Ojokoh, Recommendation systems: Principles, methods and evaluation.

[5] Ms.Aarti Patil, Ms. Seem Kolkur, Ms.Deepali Patil, Advanced Apriori Algorithms, International Journal of Scientific & Engineering Research, Volume 4, Issue 8, August-2013, ISSN 2229-5518

[6] Greg Linden, Brent Smith, and Jeremy York, Amazon.com Recommendations Item-to-Item Collaborative Filtering, JANUARY • FEBRUARY 2003 Published by the IEEE Computer Society 1089-7801/03/$17.00©2003 IEEE IEEE INTERNET COMPUTING

[7] CARLOS A. GOMEZ-URIBE and NEIL HUNT, The Netflix Recommender System: Algorithms, Business Value and Innovation, Netflix, Inc., ACM Transactions on Management Information Systems, Vol. 6, No. 4, Article 13, Publication date: December 2015.

[8] Manisha Hiralall and Wojtek Kowalczyk, Recommender systems for e-shops, Business Mathematics and Informatics paper, 2011.

[9] Andrew Kusiak, Association Rules: The Apriori Algorithm, http://user.engineering.uiowa.edu/~comp/Public/Apriori.pdf

[10] J. Manimaran1 and T. Velmurugan, Analysing the quality of Association Rules by Computing an Interestingness Measures, Indian Journal of Science and Technology, Vol 8(15), July 2015, ISSN (Print): 0974-6846, ISSN (Online): 0974-5645

[11] Medhat H A Awadalla and Sara G El-Far, Aggregate Function Based Enhanced Apriori Algorithm for Mining Association Rules, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 3, No 3, May 2012, ISSN (Online): 1694-0814

[12] Andrei Toma, Radu Constantinescu, Floarea Nastase, Recommendation system based on the clustering of frequent sets, ISSN: 1790-0832, Issue 5, Volume 6, May 2009

[13] Abhishek Saxena, Navneet K Gaur, Frequent Item Set Based Recommendation using Apriori, International Journal of Science, Engineering and Technology Research (IJSETR), Volume 4, Issue 5, May 2015, ISSN: 2278 – 7798.

[14] Karandeep Singh Talwar, Abhishek Oraganti, Ninad Mahajan, Pravin Narsale, Recommendation System using Apriori Algorithm, IJSRD - International Journal for Scientific Research & Development| Vol. 3, Issue 01, 2015 | ISSN (online): 2321-0613.