Trabajo Fin de Máster

Recomendación de producto

José Raúl Orgeira Beiro

Máster en Técnicas Estadísticas

Curso 2023-2024

Propuesta de Trabajo Fin de Máster

Título en galego: Recomendación de producto
Título en español: Recomendación de producto
English title: Product recommendation
Modalidad: Modalidad B
Autor/a: José Raúl Orgeira Beiro, Universidade da Coruña
Director/a: Jose Ameijeiras Alonso, Universidade de Santiago de Compostela; Paula Saavedra Nieves, Universidade de Santiago de Compostela
Tutor/a: Jorge López Muñiz, Hijos de Rivera, S.A.U. (Estrella Galicia)
Breve resumen del trabajo: Se introducirán y definirán los diferentes sistemas de recomendación existentes y su aplicabilidad en base a los datos disponibles y se desarollará uno de ellos para Estrella Galicia.
Recomendaciones:
Otras observaciones:

Don Jose Ameijeiras Alonso, profesor ayudante doctor LOU de la Universidade de Santiago de Compostela, doña Paula Saavedra Nieves, profesora contratada doctor de la Universidade de Santiago de Compostela y don Jorge López Muñiz, Head of Business Analytics de Hijos de Rivera, S.A.U. (Estrella Galicia), informan que el Trabajo Fin de Máster titulado

Recomendación de producto

fue realizado bajo su dirección por don José Raúl Orgeira Beiro para el Máster en Técnicas Estadísticas. Estimando que el trabajo está terminado, dan su conformidad para su presentación y defensa ante un tribunal.

En A Coruña, a 11 de enero de 2024.

AMEIJEIRAS ALONSO JOSE - 39450236S
Firmado digitalmente por AMEIJEIRAS ALONSO JOSE - 39450236S
Fecha: 2024.01.10 16:34:54 +01'00'

El director:
Don Jose Ameijeiras Alonso

SAAVEDRA NIEVES PAULA - 44847364D
Firmado digitalmente por SAAVEDRA NIEVES PAULA - 44847364D
Fecha: 2024.01.11 11:32:19 +01'00'

La directora:
Doña Paula Saavedra Nieves

El tutor:
Don Jorge López Muñiz

El autor:
Don José Raúl Orgeira Beiro

Declaración responsable. Para dar cumplimiento a la Ley 3/2022, de 24 de febrero, de convivencia universitaria, referente al plagio en el Trabajo Fin de Máster (Artículo 11, Disposición 2978 del BOE núm. 48 de 2022), **el/la autor/a declara** que el Trabajo Fin de Máster presentado es un documento original en el que se han tenido en cuenta las siguientes consideraciones relativas al uso de material de apoyo desarrollado por otros/as autores/as:

- Todas las fuentes usadas para la elaboración de este trabajo han sido citadas convenientemente (libros, artículos, apuntes de profesorado, páginas web, programas,...)
- Cualquier contenido copiado o traducido textualmente se ha puesto entre comillas, citando su procedencia.
- Se ha hecho constar explícitamente cuando un capítulo, sección, demostración,...sea una adaptación casi literal de alguna fuente existente.

Y, acepta que, si se demostrara lo contrario, se le apliquen las medidas disciplinarias que correspondan.

Agradecimientos

Esta memoria está dedicada a todos los profesores que durante este año y medio me han orientado en mi formación y han hecho posible el llegar hasta aquí; a mis tutores, José Ameijeiras Alonso, Paula Saavedra Nieves y Jorge Lopez Muniz por su guía y apoyo constante en este trabajo y, por supuesto, a mis padres.

Índice general

Resumen

Resumen en español

En las próximas páginas se presenta la memoria de prácticas del alumno José Raúl Orgeira Beiro. En ellas se introduce el problema propuesto por la empresa Hijos de Rivera para desarrollar un sistema de recomendación. En el Capítulo 1 se realiza una introducción al sector, a la empresa y a los datos utilizados para la realización del trabajo. En el Capítulo 2 se realiza una introducción teórica a los modelos de recomendación. Se introducen conceptos generales de los métodos de recomendación como sus distintos enfoques y objetivos. Posteriormente se explican detalladamente los 4 grandes tipos de sistemas de recomendación. En el Capítulo 3 se desarrolla un sistema de filtrado colaborativo basado en usuarios y otro sistema de filtrado colaborativo que utiliza la técnica de factorización matricial conocida como descomposición en valores singulares (SVD) para solucionar los problemas del primero. Por último, en el Capítulo 4 se exponen las conclusiones y posibles extensiones del proyecto.

English abstract

The following pages present the internship report of the student José Raúl Orgeira Beiro. In them, the author introduces the problem proposed by the company Hijos de Rivera to develop a recommender system. Chapter 1 contains an introduction to the industry, the company and the data used to carry out the work. Chapter 2 provides a theoretical introduction to recommender models. It introduces general concepts of recommendation methods and their different approaches and objectives. Subsequently, the 4 main types of recommender systems are explained in detail. In Chapter 3, a user-based collaborative filtering system and a collaborative filtering system using the matrix factorization technique known as singular value decomposition (SVD) are developed to solve the problems of the former. Finally, Chapter 4 presents the conclusions and possible extensions of the project.

Capítulo 1

Introducción

Con la llegada de internet, y su consecuente aparición del comercio electrónico, han cambiado los hábitos de consumo de los clientes. A su vez, estos cambios en los hábitos de consumo han dado pie a la creación de nuevas estrategias de venta. Es sencillo identificar situaciones en las que tras comprar un producto a través de una página web, la propia página recomienda productos similares o que le podrían resultar de interés al comprador. Estas recomendaciones son realizadas por los denominados sistemas de recomendación. Un sistema de recomendación es una herramienta cuyo objetivo es predecir productos que puedan ser de interés para el comprador a la par que incrementa las ventas de la compañía que lo ha implementado.

El objetivo del presente trabajo es revisar los principales sistemas de recomendación de producto existentes en la literatura y aplicar uno de ellos en base a los datos de la compañía Hijos de Rivera.

Con carácter previo a la presentación de los distintos sistemas de recomendación, se hará una contextualización, tanto del sector cervecero como de Hijos de Rivera, para motivar el problema que se ha abordado. De esta forma, en primer lugar se expondrán algunas de las características más relevantes del sector a nivel global y a nivel regional para, posteriormente, presentar a la empresa Hijos de Rivera y ubicar sus actividades en el sector presentado.

Finalmente, se describirá con detalle el problema planteado por Hijos de Rivera a partir de la base de datos facilitada por la misma.

1.1. Sector cervecero

En relación al sector cervecero es preciso mencionar que, a pesar de la crisis de 2020 originada por la pandemia, el mercado experimenta un proceso de expansión en la gran mayoría de países, tanto en producción como en consumo; aproximando el valor del mismo en torno a los 794.000 millones de dólares estadounidenses en 2022. La cerveza es la bebida alcohólica por excelencia pues, en 2022 su consumo superó en más de 65.000 millones de litros al resto de alternativas etílicas en conjunto, siendo así la opción más popular entre los consumidores de bebidas alcohólicas de todo el mundo y logrando mantener su producción anual alrededor de los 2.000 millones de hectolitros, incluso en situaciones desfavorables como la pandemia de 2020 (Orus, 2022).

Relativo a su producción en el mercado internacional destacan países como China, que en 2022 produjo cifras cercanas a los 360 millones de hectolitros; seguida por Estados Unidos (204 millones), Brasil (143 millones) y México (135 millones) (Cerveceros, 2023).

En el mercado Europeo el mayor país productor de cerveza es Alemania, que a nivel mundial se sitúa en el quinto puesto, con una producción en 2022 de 88 millones de hectolitros; seguido por España (41 millones de hectolitros) y Polonia (38 millones de hectolitros), situados, respectivamente, como noveno y décimo mayores productores de cerveza mundiales.

En términos de consumo China y Estados Unidos lideran la clasificación, con ingestas de 38 y 24 millones de hectolitros. Sin embargo, esta situación experimenta un cambio radical cuando la atención se centra en el consumo per cápita en lugar de en cifras absolutas; pues República Checa es, con 184 litros por cabeza en 2021, el país que encabeza la clasificación, superando por 85 litros anuales por cabeza al país situado en segundo lugar.

En referencia a España, el país se sitúa como segundo productor de cerveza a nivel europeo y noveno a nivel mundial, tras superar en 2022 a Polonia y Reino Unido. Dentro del panorama agroalimentario del país, pese a no alcanzar en hostelería en 2022 las cifras de ventas previas a la pandemia, el consumo total de cerveza ha aumentado en 1 millón de hectolitros respecto a 2019 (42,3 vs 41,3 millones de hectolitros), alcanzando cifras superiores de consumo en hogar a cualquiera de los años prepandemia.

El sector cervecero es fundamental en el país mediterráneo pues en 2022 no sólo ha prácticamente igualado en hostelería los niveles de consumo prepandemia sino que, además, genera 450.000 puestos de trabajo y un valor de producción cercano a los 4.000 millones de euros, suponiendo aproximadamente una cuarta parte del total del sector bebidas. Esta recuperación ha sido gracias al incremento del turismo que experimentó España respecto a 2021 y al mantenimiento de las pautas de consumo mediterráneas. Dichas pautas hacen que el consumo de cerveza se asocie en alrededor del 90 % de ocasiones con momentos de consumo de otros alimentos y que alrededor de la mitad de personas que consumen cerveza lo hagan diariamente o, al menos, en dos ocasiones a la semana.

Dentro del panorama industrial, los grupos Mahou San Miguel, Damm e Hijos de Rivera, principales productores de cerveza en España, forman parte de los 40 mayores grupos productores a nivel mundial.

1.2. El grupo cervecero

Hijos de Rivera es una de las empresas del sector agroalimentario más importantes de España, dedicada principalmente a la producción, comercialización y distribución de bebidas. La compañía, con origen y sede central en Galicia, cuenta con alrededor de 1500 empleados y concluyó el ejercicio 2022 con unos ingresos récord de 724 millones de euros. (García Ropero, 2023).

Pese a que la denominación de la empresa es Hijos de Rivera, S.A.U., ésta es más comúnmente conocida como *Estrella Galicia*, el nombre de su cerveza más popular. La compañía fue fundada en 1906 por José María Rivera Corral tras emigrar a Cuba y a México, iniciándose en ese momento el vínculo familiar que se mantiene a día de hoy entre la corporación y la familia Rivera.

La compañía ha experimentado un crecimiento prácticamente constante desde su creación. Este crecimiento fue interrumpido únicamente en 1941 debido al desabastecimiento de agua y cebada tras el final de la guerra civil española. Con carácter posterior a dicha situación, la producción y el crecimiento volvieron a la corporación, produciéndose entre 1949 y los primeros años del S. XXI avances importantes. Por ejemplo, el traslado de la fábrica desde Cuatro Caminos al polígono de A Grela, el cambio de nombre del producto principal al afamado *Estrella Galicia* y el lanzamiento de otra de sus cervezas más populares, *1906 Reserva Especial*.

Sin embargo, ha sido a partir de 2007, año en el que Ignacio Rivera Quintana es nombrado nuevo

Director General de Hijos de Rivera, que la compañía comienza a experimentar un proceso de expansión en el ámbito nacional. En lo que respecta al mercado internacional, éste sigue la misma senda que el nacional y para 2012 ya vendían sus productos en más de 30 países de todo el mundo. El crecimiento que ha acompañado a la compañía desde su creación ha permitido a Hijos de Rivera gozar de un catálogo de productos altamente variado; incluyendo, además de productos de elaboración propia como la cerveza *Estrella Galicia*, marcas de distribución como *CocaCola* o *ColaCao*.

1.3. Descripción del proyecto

El gran catálogo de productos enumerado al final de la sección anterior y la desigual venta de los mismos justifican la necesidad de desarrollar un sistema de recomendación propio para Hijos de Rivera.

En la Figura 1.1 se muestra un diagrama de barras (frecuencias absolutas anuales de compra) para los productos más vendidos de la compañía. Con el fin de mantener la privacidad de los datos proporcionados por la empresa, se han enmascarado todos aquellos datos que puedan resultar de información confidencial. A pesar de que en la Figura 1.1 no se muestran los valores de las frecuencias o los nombres de los productos, en ella se puede identificar que la mayoría de productos tienen una frecuencia de compra mucho menor que los productos de las marcas A,B,C,D y E. La aplicación de un sistema de recomendación podría ayudar a incrementar las ventas de aquellos productos menos comprados. El objetivo de este trabajo es la implementación de dicho sistema de recomendación.

En Hijos de Rivera disponen de diferentes herramientas para visualizar los productos que está comprando cada establecimiento. Mientras que en algunas de ellas se diferencia el formato, unidades y ediciones de cada producto, en otras simplemente se agrupan los productos por marca. Como el objetivo del sistema de recomendación es incrementar las ventas de aquellos productos con menor frecuencia de ventas, el recomendador se realizará a partir de bases de datos donde los productos están agrupados por marca.

Los sistemas de recomendación son un conjunto de técnicas y herramientas que proporcionan sugerencias de productos que pueden ser de interés para un usuario particular (Ricci et al., 2022). Para ello se utilizan técnicas estadísticas que se aplican a un conjunto de datos, como pueden ser un histórico de ventas o las diferentes características de los productos que se pretendan recomendar.

El sistema de recomendación a implementar puede variar en función de la información disponible

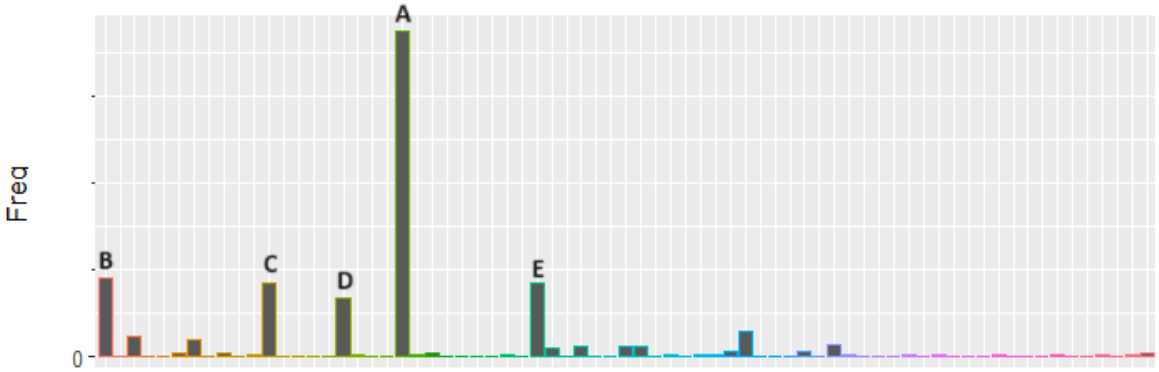

Figura 1.1: Frecuencias anuales de compra de los 74 productos más vendidos

y de los objetivos fijados. En este caso, para el desarrollo del trabajo se ha tenido acceso a diferentes datos, entre ellos el histórico de pedidos y la información sobre las características de los locales. En Hijos de Rivera todas estas bases de datos están divididas en grupos y canales bien diferenciados. Así, a pesar de que analizando las ventas de Hijos de Rivera se pueden identificar diversos canales y mercados, solo se desvelarán aquellos cuya información ha sido utilizada para desarrollar el sistema de recomendación, estos son, el canal HORECA y el canal Alimentación.

- **HORECA**: acrónimo de Hoteles, Restaurantes y Cafeterías. Hace referencia, dentro del sector de los servicios de comidas, a todos aquellos negocios y empresas que brindan servicios de comida y bebida para su consumo fuera del hogar. En este canal están presentes negocios situados en la Península, independientemente de si su localización es en España o en Portugal.
- **Alimentación**: etiqueta que engloba las ventas realizadas a negocios mayoristas y grandes distribuidores que hacen de intermediarios entre el fabricante y el usuario intermedio o minorista.

Cabe destacar que no todos los canales y mercados tienen la misma importancia para la empresa, siendo el canal HORECA y el mercado español los más importantes de los datos utilizados.

Los datos han sido recolectados de las bases de datos de la compañía almacenadas en la plataforma SQL Server, descargados en formato CSV y transcritos a R; aunque una conexión directa entre SQL Server y R también hubiera sido viable. Dichos datos han sido extraídos de distintas tablas maestras, concretamente, los albaranes correspondientes al año 2022, las características de los locales y las características de los productos. Las variables extraídas de los datos de la compañía son las siguientes:

- *Articulo ID*: identificador de producto.
- *Marca ID*: identificador de marca.
- *Marca desc*: nombre de la marca correspondiente a la variable `MARCA ID` .
- *Grupo material desc*: tipo de producto vendido independientemente de la marca. Por ejemplo, el campo correspondiente en *Grupo material desc* a una venta de cervezas 1906 sería “cerveza” y el correspondiente a una venta de agua Cabreiroá sería “agua”.
- *Agrupacion envase ID*: número de unidades que hay en el paquete vendido dependiendo del id del artículo.
- *Ventas EUR*: precio total de venta de cada pedido realizado.

- *Ventas LTS*: cantidad total (en litros) vendida en cada pedido.
- *Ventas UDS*: cantidad de unidades vendidas en cada pedido.
- *Ventas UDS INV*: muestra las unidades de inventario vendidas en cada pedido. Es preciso mencionar que, en el caso de ventas de cerveza,en lugar de indicar el número de unidades vendidas (40 paquetes de 24 botellines, por ejemplo) indica el número de litros.
- *Local ID*: identificador de cliente.
- *Tipo instalacion ID*: indicador de la instalación del local (barril, tanque o ninguna).
- *Exclusiva establecimiento ID*: presencia o ausencia de acuerdos de rappel anticipados.
- *Acuerdo promo ID*: presencia o ausencia de acuerdos promocionales. Dichos acuerdos pueden incluir descuentos porcentuales (5 % de descuento a los pedidos realizados), acuerdos cruzados (si se compra A se obtiene un descuento en B) o modulares (si se compran 3 unidades solo se pagan 2).
- *provincia ID*: código de provincia correspondiente al identificador del cliente.

Una vez depurados los datos se han creado nuevas variables. Por ejemplo:

$$\text{Cabreiroá}_i = \begin{cases} 1, & \text{Si el usuario } i \text{ ha comprado productos de la marca Cabreiroá} \\ 0, & \text{En caso contrario} \end{cases}$$

El contenido del trabajo se estructura de la siguiente manera:

En el Capítulo 2 se realiza una revisión de los sistemas de recomendación. Se introducen los modelos de recomendación, las formas en las que se puede formular el problema y los objetivos del sistema. Posteriormente, se desarollan los 4 tipos de modelos de recomendación: métodos de *filtrado colaborativo*, métodos *basados en contenido*, métodos *basados en conocimiento* y métodos *híbridos*. Para cada uno de dichos métodos se explica su funcionamiento, requisitos y distintas versiones de cada uno de los modelos y se discuten sus ventajas y desventajas. En el Capítulo 3 se implementa un sistema de recomendación de producto para Hijos de Rivera que se adecúe a los objetivos planteados y a los datos disponibles y se analizan los resultados obtenidos. Finalmente, en el Capítulo 4 se exponen las conclusiones y posibles extensiones para mejorar los resultados obtenidos en futuras líneas de trabajo.

Capítulo 2

Modelos de recomendación

Este capítulo inicial se centrará en introducir conceptos básicos de los modelos de recomendación así como en revisar algunos de los sistemas de recomendación más utilizados y que han sido considerados en la realización del trabajo. Las principales fuentes de consulta han sido Aggarwal, 2016; Dietmar et al., 2011 y Mohanty et al., 2020.

2.1. Introducción a los modelos de recomendación

Supongamos que un usuario visita una tienda web para buscar un producto determinado que le resulta de especial interés y que una vez el usuario ha escrito el nombre de dicho producto en la barra de búsqueda de la página, el producto aparecerá como uno de los posibles resultados. Una vez el usuario ha comprado el producto se encuentra con una sección en la página con título "Otros usuarios que compraron este producto también han mostrado interés en" que enseña una variedad de productos similares al comprado en los que el usuario podría tener interés. La situación descrita puede identificarse con el proceso de compra de productos o consumo de servicios que experimentan la mayoría de las personas al utilizar páginas como Amazon, Ebay, Zalando, Netflix o incluso Youtube. El programa que determina qué objetos han de ser mostrados a cada persona se denomina sistema de recomendación. (Dietmar et al. 2011, pp: 2-8)

Habitualmente, el objetivo principal de los sistemas de recomendación es incrementar las ventas; sin embargo, otras razones por las que se suelen implementar modelos de recomendación incluyen:

- Satisfacción de clientes: mejorar la experiencia de los consumidores mediante recomendaciones útiles, interesantes y/o relevantes contribuye a la satisfacción de los mismos.
- Lealtad de clientes: de la misma forma que con la satisfacción de los consumidores, un sistema que conozca sus preferencias y los trate como un cliente valorado o importante contribuye a mejorar su lealtad hacia dicho portal web.
- Entender mejor las necesidades de los clientes: analizar las preferencias y el comportamiento de los consumidores en una determinada tienda o portal web, ayuda a orientar el desarrollo de futuros productos o la oferta de futuros servicios.

A su vez, los sistemas de recomendación también cuentan con objetivos técnicos y/o operacionales orientados a producir recomendaciones de calidad. Dichos objetivos técnicos acostumbran a ser los siguientes:

- Relevancia: es decir, recomendar productos que un usuario valore y considere interesantes.
- Primicia: que el sistema recomiende productos que el usuario no ha visto antes.

- Casualidad: este concepto hace referencia a que los productos recomendados sean de alguna forma inesperados y que por lo tanto exista un elemento modesto de "suerte" en oposición a recomendaciones más obvias de productos. Más información acerca de la causalidad puede consultarse en Good et al., 1999.

Los sistemas de recomendación utilizan distintas fuentes de datos obtenidos de los clientes para poder identificar qué productos les pueden resultar de interés. Para ello, las recomendaciones están basadas en interacciones pasadas de los usuarios con distintos productos y clientes, pues intereses pasados y proclividades de los clientes suelen ser buenos indicadores de elecciones futuras. (Aggarwal, 2016, pp:1-28). La forma en la que se recopilan todos los datos necesarios sobre los clientes para poder recomendar productos varía de sistema a sistema; los datos pueden ser recogidos ya sea de forma implícita, donde simplemente se supervisa o controla el comportamiento de los distintos usuarios, o de forma explícita, donde a los usuarios se les pregunta de forma proactiva cuáles son sus preferencias.

Como los datos sobre el comportamiento de los consumidores son habitualmente utilizados para determinar futuras preferencias, los sistemas de recomendación generalmente asumen que existen fuertes dependencias entre usuarios y objetos. Un ejemplo intuitivo de cómo se utilizan los datos de los consumidores es la recomendación colaborativa. En ella, el sistema identifica qué productos han consumido los usuarios, películas, por ejemplo, para posteriormente encontrar usuarios con historiales de compra parcialmente solapados y recomendar los productos que un usuario ha consumido y el otro no. Es decir, si un usuario a ha visto "Interestellar", "Inception" y "Matrix" y un usuario b ha visto "Interestellar", "Tenet" y "Matrix", el sistema le recomendará "Inception" al usuario b.

Para poder hacer las recomendaciones es necesario guardar de alguna forma las preferencias o el historial de consumo de cada uno de los usuarios. Con este propósito, la mayoría de los sistemas de recomendación utilizan lo que se denomina como matriz de *ratings* R. Una matriz de *ratings* R es una matriz de dimensión $n \times m$ donde n es el número de usuarios, m el número de productos y cada $r_{a,j}$, $a \in 1, ..., n$, y $j \in 1, ..., m$ representa la valoración o *rating* que un determinado usuario le ha otorgado a un determinado producto. En caso de que un usuario no haya valorado o consumido un determinado producto, su $r_{a,j}$ correspondiente se fija a 0. Además, cuanto mayor sea la cantidad de *ratings* cubiertos en la matriz, mayor será la fiabilidad de las predicciones.

En el párrafo anterior se ha dado un ejemplo de cómo los sistemas de filtrado colaborativo utilizan las preferencias de los usuarios para generar recomendaciones. Sin embargo, sistemas diferentes de recomendación utilizarán la información de formas distintas. Otras opciones de sistemas de recomendación son los recomendadores basados en contenido, los basados en conocimiento y lo métodos híbridos.

En los distintos recomendadores el problema de recomendación podrá ser formulado de una forma u otra. Generalmente, los dos enfoques más utilizados son la versión predictiva del problema y la versión con rankings. La versión predictiva del problema consiste en predecir una puntuación o *rating* para un determinado producto, por ejemplo, predecir la puntuación que un usuario a le daría a un libro a partir de las puntuaciones que usuarios similares le han dado. La predicción podría calcularse como, por ejemplo, la media de las puntuaciones de los 10 usuarios más similares a a. En este enfoque se suele asumir que se dispone de una base de datos de entrenamiento que incluya las preferencias o valoraciones de los usuarios por distintos productos, recogidas en una matriz de *ratings* $n \times m$ incompleta. Por otro lado, en la versión con rankings del problema, las puntuaciones que un usuario le dará a un determinado producto son más bien irrelevantes, pues el objetivo es recomendar top$-k$ productos a un determinado usuario o top$-k$ usuarios a un determinado producto, siendo más común el primero de los métodos.

Como ya se ha mencionado anteriormente, en este proyecto el sistema de recomendación se realiza

para Hijos de Rivera y tiene el objetivo de aumentar la frecuencia de compra de aquellos productos más desconocidos. Hijos de Rivera no es una compañía dedicada al sector servicios por lo que obtener valoraciones de los clientes es más complicado que en otras empresas en las que la valoración forma parte de la experiencia del cliente como *Netflix*, *Goodreads* o *Rotten Tomatoes*. Sin embargo, haciendo uso de una matriz de datos con valoraciones en forma de litros vendidos, se implementará la versión predictiva del problema.

2.1.1. Medidas de error

A lo largo de la Sección 2.1 se ha hablado de los objetivos de los sistemas de recomendación, se ha dejado entrever como pueden funcionar y se han mencionado los distintos enfoques que se le pueden dar al problema. Sin embargo, para la implementación de un sistema de recomendación no es suficiente con construirlo, es necesario evaluarlo. Por ello, con carácter previo a la explicación de los tipos de modelos de recomendación, se mencionarán los distintos criterios de error utilizados para evaluar su calidad. Borchers et al. (1999) distinguen dos formas distintas en las que es posible evaluar un sistema de recomendación: su alcance o cobertura (*coverage*) y su precisión.

Primeramente, el alcance es una medición del porcentaje de ítems para los que el sistema da recomendaciones. Generalmente, se utiliza como métrica el porcentaje de ítems que han sido retornados como predicciones; es decir, si se evalúa la cobertura del sistema en 100 usuarios, el alcance será la proporción de ítems recomendados de los ítems totales ofrecidos por la compañía. En segundo lugar, se puede evaluar la precisión de las recomendaciones. Las métricas que evalúan la precisión de los modelos pueden dividirse en dos categorías, las medidas de precisión estadística y las medidas de apoyo a la toma de decisiones (*decision-support accuracy metrics*). Las primeras comparan los valores predichos de una valoración con los respectivos valores reales y en ellas se encuentran medidas estadísticas como el Error Absoluto Medio (*MAE*), el Error Cuadrático Medio (ECM) o la Raíz del Error Cuadrático Medio (*RMSE*):

El error absoluto medio calcula la desviación de la predicción de su valor original y se calcula de la siguiente forma:

$$\text{MAE} = \frac{1}{|\Upsilon|} \sum_{(u,j)\in\Upsilon} |r_{uj} - \hat{r}_{uj}|,$$

donde Υ es el conjunto de todas las parejas usuario-ítem (u, j) para las que se tiene un *rating* predicho ($\hat{r}_{uj}$) y un *rating* conocido r_{uj} y $|r_{uj} - \hat{r}_{uj}|$ representa el valor absoluto de la diferencia entre ambos.

El error cuadrático medio mide el promedio de los errores al cuadrado y se calcula de la siguiente forma:

$$\text{ECM} = \frac{\sum_{(a,j)\in\Upsilon}(r_{uj} - \hat{r}_{uj})^2}{|\Upsilon|},$$

donde $|\Upsilon|$ es el número de todas las parejas usuario-ítem para las que se tiene un *rating* predicho. Por último, el RMSE se calcula como la Raíz del ECM calculado anteriormente:

$$\text{RMSE} = \sqrt{\frac{\sum_{(a,j)\in\Upsilon}(r_{uj} - \hat{r}_{uj})^2}{|\Upsilon|}}.$$

De las métricas mencionadas, el RMSE penaliza más que el MAE errores más grandes de predicción. Es preciso destacar que las tres métricas dependen de las unidades de los datos para calcular los errores. En este caso no supone un problema pues los valores predichos de los *ratings* se devuelven sobre la misma matriz y están en las mismas unidades. En caso de querer comparar modelos en distintas

unidades, sería adecuado utilizar criterios como el Error Porcentual Medio Absoluto; véase Hyndman et al. (2006) y De Myttenaere et al. (2016).

Por otro lado, las medidas de apoyo a la toma de decisiones evalúan la calidad de los ítems recomendados. A pesar de que hay varias alternativas posibles, lo más habitual es utilizar la sensibilidad ROC cuando se quiere evaluar este aspecto del modelo de recomendación.

Dependiendo de los objetivos a conseguir con el sistema de recomendación, el enfoque del problema y el tipo de modelo construído, varían las dimensiones en las que evaluar el modelo y las métricas utilizadas. Puede consultarse literatura existente en Borchers et al., 1999; Herlocker et al., 2004; Herlocker et al., 2004 y Bellogín et al., 2013.

2.2. Tipos de modelos de recomendación

Como ya se ha mencionado anteriormente, dependiendo del tipo de problema que se quiera solucionar se aplicará un sistema de recomendación u otro. Generalmente se trabaja con dos tipos de datos: el historial y los comportamientos del consumidor, y las propiedades (atributos) de los clientes y productos. Dentro de los sistemas de recomendación existentes, los que utilizan el historial del consumidor son parte de los llamados *métodos de filtrado colaborativo* y los que usan los atributos de los clientes o productos forman parte de los *métodos basados en contenido* (aunque estos métodos en la mayoría de los casos también utilizan el historial del consumidor). A mayores, podemos hablar también, de *métodos basados en conocimiento* y de *métodos mixtos o híbridos*.

2.2.1. Métodos de filtrado colaborativo

La recomendación colaborativa se basa en la idea de si dos usuarios han estado interesados en productos similares en el pasado, dichos usuarios tendrán gustos semejantes en el futuro. Por ejemplo, si dos usuarios tienen un historial de compras parcialmente solapado, ambos usuarios también disfrutarán los productos que el otro ya ha comprado y ellos todavía no. Los sistemas de filtrado colaborativo han sido ampliamente utilizados en los últimos años, especialmente en tiendas online de venta minorista y su desempeño, sus ventajas y sus desventajas han sido objeto de estudio desde la segunda mitad de la década de los 90 (Zhang et al., 2008) . A continuación se presentarán los distintos tipos de filtrado colaborativo, sus ventajas y sus desventajas.

2.2.1.1 User-Based Neighborhood Models

Uno de los primeros métodos creados, cuya idea subyacente es simple y ya ha sido mencionada a lo largo del trabajo: dada una matriz de *ratings* y un usuario objetivo a, identificamos usuarios similares, es decir, usuarios que han puntuado productos de forma parecida al usuario a. Una vez se han determinado dichos usuarios, para cada producto que han valorado y el usuario a no, se calcula una predicción basada en la media ponderada de las puntuaciones de los k vecinos más similares.

Como en los métodos de filtrado colaborativo utilizamos únicamente la matriz de *ratings*, disponemos de: un conjunto de usuarios $U = \{u_1, ..., u_n\}$, un conjunto de productos (comúnmente referidos como ítems) $P = \{p_1, ..., p_m\}$ y una matriz de *ratings* R de dimensión $n \times m$. Una vez tenemos identificados los usuarios y los ítems, podemos definir como obtener el conjunto de usuarios similares. Para ello, generalmente se utiliza el coeficiente de correlación de Pearson, de forma que la similitud de dos usuarios a,b ($\text{sim}_P(a, b)$) dada una matriz R es la siguiente: (Dietmar et al., 2011, pp:13-18)

$$\text{sim}_P(a, b) = \frac{\sum_{p \in P}(r_{a,p} - \overline{r_a})(r_{b,p} - \overline{r_b})}{\sqrt{\sum_{p \in P}(r_{a,p} - \overline{r_a})^2}\sqrt{\sum_{p \in P}(r_{b,p} - \overline{r_b})^2}},$$

donde $\overline{r_a}$ se corresponde con las puntuaciones medias del usuario a
Es preciso mencionar que la formula anterior, a pesar de ser una de las más utilizadas, es sólo una de las formas para calcular la similitud entre usuarios propuestas en Ekstrand et al., 2011 . Dentro de las otras propuestas podemos encontrar la similitud coseno o el coeficiente de correlación de Spearman. De acuerdo al análisis empírico llevado a cabo por Borchers A. y demás en Herlocker et al., 1999, las medidas pueden ser más o menos adecuadas dependiendo de la situación que se tiene que afrontar. Por ejemplo, en caso de que se quiera predecir valoraciones en una escala pequeña de números discretos, el coeficiente de correlación de Spearman da mejores resultados (en términos de precisión estadística); mientras que si las valoraciones son continuas es preferible utilizar el coeficiente de correlación de Pearson.

Una vez se tienen calculadas las similaridades de todos los usuarios con respecto al usuario a, es necesario identificar qué usuarios se van a tener en cuenta y cómo se valorará su opinión. Una fórmula para computar la predicción del usuario a por el ítem p que tenga en cuenta la proximidad relativa de los vecinos más cercanos y el *rating* medio de a es la siguiente:

$$\text{pred}(a,p) = \overline{r_a} + \frac{\sum_{b \in N} \text{sim}_P(a,b) \cdot (r_{b,p} - \overline{r_b})}{\sum_{b \in N} \text{sim}_P(a,b)},$$

donde U es el conjunto de usuarios disponible.

Calcular las predicciones para un usuario a teniendo en cuenta todos los usuarios de la base de datos puede ser muy costoso computacionalmente y puede disminuir la precisión de dichas predicciones. Para solucionar este problema un procedimiento común es considerar sólo algunos usuarios (un vecindario) en el sumatorio (Borchers et al., 1999). Dicho vecindario puede estar compuesto por aquellos usuarios que tengan una correlación positiva con el usuario objetivo; sin embargo, como el vecindario puede llegar a resultar demasiado grande, la técnica más común para reducir la dimensión del mismo consiste, simplemente, en seleccionar los k vecinos más cercanos al usuario objetivo. Generalmente, valores de k entre 10 y 15 suelen ser adecuados (Borchers et al., 1999). A pesar de que el valor de k es dependiente de la base de datos a utilizar, el método k-NN ha probado tener un buen funcionamiento (O'Mahony et al., 2003).

2.2.1.2 Item-Based Neighborhood Models

Uno de los problemas que afrontan los modelos de recomendación basados en usuarios se da lugar en comercios online con un número extremadamente grande de productos y clientes, pues cuando hay millones de consumidores, valorar el número total de potenciales vecinos dificulta la realización de predicciones en tiempo real. Para solucionar este problema se pueden utilizar los modelos de recomendación basados en ítems o productos (Dietmar et al., 2011).

Los modelos de recomendación basados en productos funcionan de una manera similar a aquellos basados en usuarios, con la diferencia de que en lugar de calcular la similitud entre usuario para, posteriormente, recomendar los productos que han consumido usuarios similares, se calcula la similitud entre los distintos ítems. Así, el primer paso para construir uno de estos modelos es encontrar una medida adecuada de calcular la similitud. En este caso, la medida estándar para computarla es la similitud coseno. En ella se divide el producto escalar de los vectores de *ratings* para los productos i,j por el producto de sus normas euclídeas:

$$\text{sim}_C(\vec{r_i}, \vec{r_j}) = \frac{\overline{r}_i \cdot \vec{r_j}}{|\vec{r_i}||\vec{r_j}|},$$

donde $\vec{r_i}$ y $\vec{r_j}$ son los vectores de *ratings* para los productos i,j y $|\vec{r_i}|$ es la norma euclídea del vector, definida como la raíz cuadrada del producto escalar del vector consigo mismo. (Dietmar et al., 2011,

pp:18-22)

La similitud coseno no tiene en cuenta las diferencias entre las distintas formas de valorar productos que tienen los usuarios. Para solucionarlo se puede utilizar la similitud coseno ajustada en la que simplemente restamos a los vectores de valoraciones la media de las valoraciones de dicho usuario. De esta forma, el resultado que se obtendría oscilaría entre -1 y 1 en lugar de entre 0 y 1. Es preciso mencionar que, aunque en los modelos basados en ítems también es posible utilizar la correlación de Pearson para calcular la similitud entre ítems, la similitud coseno es simple, rápida y otorga una mayor precisión predictiva. (Borchers et al., 1999)

A continuación, de la misma forma que se hacía con los modelos anteriores, el siguiente paso sería predecir el *rating* que un usuario a le otorgaría a un producto i. Para ello, una vez identificado un conjunto S de productos similares a i, $\text{pred}(a, i)$ se computa de la siguiente forma:

$$\text{pred}(a, i) = \frac{\sum_{j \in S} \text{sim}_C(i, j) * r_{a,j}}{\sum_{j \in S} |\text{sim}_C(i, j)|},$$

donde S es típicamente seleccionado como los k ítems más similares a j que a también ha valorado para un vecindario de tamaño k (Ekstrand et al. 2011). Los ítems más similares son aquellos con mayor valor de $\text{sim}_C(\vec{r_i}, \vec{r_j})$.

La idea básica de los algoritmos de recomendación basados en productos consiste en aprovechar las valoraciones que el usuario ha realizado sobre artículos similares para realizar la predicción.(Aggarwal, 2016, pp:40-41). Por esta misma razón, los modelos basados en ítems suelen dar recomendaciones más relevantes y acostumbran a tener una mayor precisión. No obstante, esa misma precisión puede ser una desventaja pues, al recomendar productos similares, si el usuario no encuentra relevante la primera recomendación, tampoco encontrará relevantes el resto de recomendaciones. En este aspecto en concreto, los modelos basados en usuario tienen mayor ventaja, pues, aunque las predicciones sean menos precisas, la diversidad de ítems recomendados es mayor y fomentan la "causalidad" mencionada en los objetivos secundarios de los modelos de recomendación.

2.2.1.4 Sistemas de filtrado colaborativo basados en modelos

Tanto los métodos basados en usuarios como los métodos basados en ítems son considerados métodos basados en memoria, pues la matriz de *ratings* original está guardada en la memoria y se utiliza para generar recomendaciones. Sin embargo, dentro del filtrado colaborativo existen algunos métodos basados en modelos en los que primero se preprocesan los datos. Este preprocesado puede ser realizado ya sea aplicando análisis de componentes principales (PCA), aplicando una descomposición en valores singulares (SVD), agrupando los ítems o usuarios en distintos clusters, utilizando un enfoque probabilístico con el teorema de Bayes o simplemente filtrando determinados usuarios o ítems en la base de datos (Dietmar et al., 2011, pp: 26-40).

Dentro de los métodos basados en modelos, los que aplican análisis de componentes principales o descomposición en valores singulares son denominados modelos de *factores latentes* o de *factorización matricial*, pues el preprocesado que se aplica a los datos consiste en utilizar técnicas de reducción de la dimensión. Las técnicas de reducción de la dimensión han sido aplicadas de forma exitosa desde 1990, tras la publicación de Deerwester et al., 1990; y, tras la publicación de Sarwar B. et al., 2000, se concluyó que los métodos que aplican de descomposición en valores singulares funcionan mejor que los modelos basados en usuario.

A continuación se explicarán dos de las técnicas de preprocesado más utilizadas, el sistema de filtrado colaborativo basado en el clasificador bayesiano ingenuo (*Naïve Bayes Classifier*) y los métodos basados en descomposición de valores singulares. Puede consultarse literatura existente en Miyahara

et al.,2002; Breese et al., 2013; Dietmar et al., 2011 y Valdiviezo-Diaz et al., 2019.

Métodos basados en el clasificador de Bayes

Para explicar el modelo de filtrado colaborativo basado en el clasificador de Bayes se asumirá que hay un número pequeño de valoraciones posibles y que pueden ser tratados como variables categóricas; de esta forma, se dispone de l valores posibles para cada valoración, que se denotarán por $v_1, ..., v_l$. De nuevo, se dispone de una matriz R de dimensión $n \times m$ que contiene las valoraciones de n usuarios para los distintos m productos. El modelo de Bayes utiliza inferencia para predecir valores faltantes tratando a los productos como atributos o características y a los usuarios como instancias del problema a las que se han de asignar dichas características. Al enfocar el modelo de esta forma para un modelo de filtrado colaborativo, el problema principal es que cualquier producto puede ser la clase objetiva en el filtrado y que hay que trabajar con valores faltantes. Para solucionar esto, se pueden añadir modificaciones menores al modelo ingenuo de Bayes. A continuación veremos un ejemplo:

Si se considera el usuario a que ha valorado un conjunto de productos P_a de la forma que si el usuario a ha valorado los productos 1, 2 y 9, $P_a = \{1, 2, 9\}$. Si se quiere predecir la valoración que el usuario a hará para el producto j (r_{aj}) dentro de $\{v_1, ..., v_l\}$, entonces se quiere determinar la probabilidad de que dicho usuario otorgue una valoración u otra dentro de los valores posibles condicionada a los valoraciones de los productos de P_a. Así, para cada valor de $s \in \{1, ..., l\}$ se quiere identificar la probabilidad $P(r_{aj} = v_s|\text{valoraciones de } P_a)$ que puede, usado el teorema de Bayes, ser representada de la siguiente forma:(Aggarwal, 2016, pp:82-86)

$$P(r_{aj} = v_s|\text{valoraciones de } P_a) = \frac{P(r_{aj} = v_s) \cdot (P(\text{valoraciones de } P_a|r_{aj} = v_s))}{P(\text{valoraciones de } P_a)}.$$

Así, necesitamos determinar cuál de los valores de $s \in \{1, ..., l\}$ tiene una mayor probabilidad y, como el denominador de la parte derecha de la ecuación es independiente de s, para ello se puede expresar la ecuación anterior en términos de una proporcionalidad constante:

$$P(r_{aj} = v_s|\text{valoraciones de } P_a) \propto P(r_{aj} = v_s) \cdot P(\text{valoraciones de } P_a|r_{a,j} = v_s).$$

Así, $P(\text{valoraciones de } P_a|r_{a,j} = v_s)$ se estima haciendo uso de la *naive assumption* que está basada en la independencia condicional entre las valoraciones (las valoraciones de a para los productos de P_a son independientes entre si, condicionados al hecho de que r_{aj} fue v_s) de la siguiente forma:

$$P(\text{valoraciones de } P_a|r_{aj} = v_s) = \prod_{k \in P_a} P(r_{ak}|r_{aj} = v_s).$$

Donde $P(r_{ak}|r_{aj} = v_s)$ se estima como la proporción de usuarios que han dado la valoración r_{ak} para el k-ésimo producto, teniendo en cuenta que han asignado la valoración v_s al j-ésimo producto. Si se introducen las ecuaciones anteriores a la ecuación expresada en términos de proporcionalidad constante, se puede obtener la probabilidad *a posteriori* de que el usuario a valore el producto j de la forma:

$$P(r_{a,j} = v_s|\text{valoraciones de } P_a) \propto P(r_{aj} = v_s) \cdot \prod_{k \in P_a} P(r_{ak}|r_{aj} = v_s).$$

Finalmente, el valor estimado de la probabilidad *a posteriori* de la valoración r_{aj} puede usarse para estimar su valor de varias formas. Una de dichas formas consiste en calcular todas las expresiones del lado derecho de la ecuación anterior para cada s y determinando cual de dichos s es mayor, es decir:

$$\hat{r}_{aj} = \text{argmax}_{v_s} P(r_{a,j} = v_s|\text{valoraciones de } P_a) = \text{argmax}_{v_s} P(r_{aj} = v_s) \cdot \prod_{k \in P_a} P(r_{ak}|r_{aj} = v_s).$$

Es preciso mencionar que este enfoque sólo es útil en el supuesto establecido anteriormente, es decir, que hay un número pequeño de valoraciones posibles, pues trata las valoraciones como valores categóricas.

Descomposición en valores singulares (SVD)

A lo largo de este apartado se ha mencionado que uno de los métodos de preprocesado más eficaces para construir un sistema de recomendación de filtrado colaborativo es aplicar una descomposición en valores singulares. Dicho método forma parte de los modelos de factores latentes y ayuda a solucionar los problemas derivados de los modelos de filtrado colaborativo cuando hay una matriz dispersa. A continuación, se explicará dicha técnica en mayor detalle, de acuerdo a la descripción de Aggarwal Charu, 2016.

La descomposición en valores singulares es un método de factorización matricial del tipo $R = U\Sigma V^t$ en el que las columnas U y V son forzadas a ser mutuamente ortogonales. En matrices completamente especificadas, la descomposición matricial es relativamente sencilla, pues podemos factorizar (aproximadamente) la matriz de valoraciones R usando la descomposición en valores singulares *truncada* del rango $g \ll \text{mín}\{m, n\}$. La descomposición en valores singulares truncada se realiza de la siguiente forma:

$$R \approx Q_g \Sigma_g W_g^T.$$

Donde Q_g, Σ_g y W_g son matrices de tamaño $m \times g$, $g \times g$, y $n \times g$, respectivamente y g es un valor mucho menor que $\text{mín}\{m, n\}$. La matriz Q_g se corresponde con la matriz de los factores de usuarios y W_g con la matriz de los factores de los productos. Además, las matrices Q_g y W_g contienen los g eigenvectores más grandes de RR^T y R^TR, respectivamente, mientras que Σ_g contiene las raíces cuadradas (no negativas) de los g eigenvectores más grandes de cualquiera de las dos matrices anteriores en su diagonal.

Es preciso mencionar que los eigenvalores distintos de 0 de las matrices RR^T y R^TR son los mismos, aunque si $m \neq n$ tendrán un número diferente de valores distintos a 0.

La matriz W_g es la representación de base reducida (*reduced basis representation*) requerida para la reducción de la dimensión del espacio de las filas. W_g contiene los eigenvectores más grandes de R^TR que, a su vez, contienen información de las direcciones de las correlaciones producto-producto entre las distintas valoraciones; permitiendo así representar a cada usuario en un número reducido de dimensiones. Además, la matriz $Q_g\Sigma_g$ contiene la representación $m \times g$ transformada y reducida de la matriz de valoraciones original en las bases que corresponden a W_g. Así, partiendo de la ecuación mostrada anteriormente ($R \approx Q_g \sum_g W_g^T$) se ve que la descomposición en valores singulares es una factorización matricial en 3 matrices en lugar de 2 ($R = UV^t$). De todas formas, la matriz diagonal Σ_g podría ser absorbida en cualquiera de las matrices de los factores de usuarios Q_g o de los productos W_g. Así, por convención, dichas dos matrices se definen:

$$U = Q_g \Sigma_g,$$

$$V = W_g.$$

Ahora, la factorización de la matriz de valoraciones R se define como $R = UV^T$. De esta forma, el objetivo del proceso de factorización es determinar las matrices U y V con columnas ortogonales. Para ello, las técnicas SVD pueden ser formuladas como un problema de optimización de la forma:

$$\text{Minimizar J} = \frac{1}{2}||R - UV^T||^2.$$

sujeto a:

Las columnas de U son mutuamente ortogonales
Las columnas de V son mutuamente ortogonales

La única diferencia con la factorización matricial sin restricciones es la presencia de las restricciones de ortogonalidad, es decir, que se minimiza la misma función objetivo pero en un espacio menor de soluciones. Además, la presencia de dichas restricciones no aumenta el error de J en las aproximaciones y, cuando la matriz de valoraciones está completamente especificada, el valor óptimo de J es el mismo utilizando técnicas SVD y técnicas de factorización matricial sin restricciones. Esto no es necesariamente cierto cuando la matriz de valoraciones R no está completamente especificada, casos en los que la factorización matricial sin restricciones comúnmente tendrá errores menores en los input o entradas observados. Para dichos casos, existen diversas técnicas que permiten solucionar el problema SVD cuando la matriz R no está completamente especificada. Uno de dichos métodos podría ser un enfoque iterativo como el siguiente:

En primer lugar, se centra en relación a la media cada fila de R (restándole a cada dato de la fila a la valoración media del usuario a), denominando a la matriz con dichos datos R_c. A continuación las entradas faltantes de R_c se fijan a 0 y se aplica el método SVD a R_c para obtener la descomposición $R_c \approx Q_g \Sigma_g W_g^T$, de forma que los factores resultantes de productos y usuarios están dados por $V = W_g$ y $U = Q_g \Sigma_g$. Si se deja que la a-ésima fila de U sea un vector g-dimensional denotado como $\bar{u_a}$ y la j-ésima fila de V el vector g-dimensional denotado como $\bar{v_a}$, entonces, el *rating* $\hat{r}_{aj}$ del usuario a para el producto j se estima de la siguiente forma:

$$\hat{r}_{aj} = \bar{u}_a \cdot \bar{u}_j + \mu_a,$$

donde μ_a se añade para compensar el paso anterior en el que se centraron los datos respecto a la media.

El principal problema de este enfoque es que puede llevar a tener un sesgo considerable. Para solucionar dicho sesgo se puede usar una gran variedad de técnicas, como realizar la estimación por máxima verosimilitud. Otro enfoque posible para reducir el sesgo podría ser el siguiente proceso iterativo:

- Inicialización: Se inicializan las entradas faltantes de R en la a-ésima fila para que sean μ_a y crear, así, la matriz R_f.
- Paso 1: realizar una descomposición en valores singulares de rango-k en R_f de la forma $Q_g \Sigma_g W_g^T$
- Paso 2: ajustar únicamente las entradas faltantes originalmente en R_f a los valores correspondientes de $Q_g \Sigma_g W_g^T$ y volver al paso 1.

Donde los pasos 1 y 2 se ejecutan hasta conseguir convergencia.
Aunque en este método el proceso de inicialización causa sesgo en las iteraciones iniciales de descomposición de valores singulares, iteraciones posteriores producen estimaciones más robustas.

2.2.1.3 Escasez de datos e inicio en frío

En algunas situaciones puede ser complicado conseguir valoraciones de los consumidores sobre ciertos productos. Por ejemplo, en un sistema de recomendación para una tienda de libros es muy probable que se disponga de pocos usuarios que han valorado muchos libros y muchos usuarios que sólo han valorado 3 o 4 libros. Cuando se da esta situación, el cálculo de similitudes se complica para aquellos usuarios que han valorado pocos libros, pues no se encuentran usuarios similares, y provoca que no se puedan generar recomendaciones para ellos.

En aplicaciones prácticas, es bastante común tener poca información de muchos consumidores y, en consecuencia, matrices de *ratings* dispersas. Una de las formas con las que se puede solucionar este problema es utilizando técnicas de *factores latentes* como la ya explicada SVD que realiza una

descomposición matricial para poder aproximar valores faltantes (*Singular Value Descomposition*) (Banirostam T. et al, 2021; Aggarwal, 2016; Dietmar et al., 2011). Adicionalmente, la escasez de datos se puede solucionar utilizando datos adicionales tanto de los productos como de los consumidores, es decir, covariables; como pueden ser la edad, el género, o cualquier dato que nos permita agrupar a los usuarios. De esta forma, los métodos pasan a ser métodos híbridos en lugar de modelos puramente colaborativos.

Otro de los problemas inherentes de los modelos de filtrado colaborativo es el conocido "cold start" o inicio en frío. Dicho problema ocurre cuando el sistema no puede hacer inferencia en usuarios o productos de los que aún no se ha obtenido información y suele darse en tres casos:

- Cuando se implementa el sistema de recomendación en una nueva compañía que aunque tenga información de los productos no tiene información de los consumidores por lo que no puede obtener una matriz de *ratings*.
- Cuando se incorpora un producto nuevo, pues es un ítem que ningún usuario ha adquirido o valorado.
- Cuando se incorpora un usuario que no ha interactuado con ningún ítem al sistema

Aunque, realmente el problema de inicio en frío puede verse como un subproblema dentro del obstáculo que es la escasez de datos (Huang et al., 2004) .

Sin embargo, dada la prevalencia del problema, ambos casos han sido ampliamente estudiados y se han presentado diversas soluciones como pueden ser el empleo de los ya mencionados modelos híbridos, requisitos de creación de perfil o usuario (es decir, que como requerimiento para poder introducir un usuario en el sistema este valore antes algunos ítems) o aplicando algunas técnicas de preprocesado de datos. (Xuanhhat et al., 2008 ; Schein et al., 2002).

2.2.2. Métodos basados en contenido

Los sistemas de recomendación basados en contenido se caracterizan por utilizar los atributos de los ítems para realizar las recomendaciones. A partir de interacciones previas de los usuarios, el sistema recomienda ítems analizando los atributos de los mismos. Por ejemplo, si nos basamos en las valoraciones que un usuario ha hecho de diferentes géneros literarios, el sistema llegará a recomendar libros del género literario que ha sido valorado positivamente por el usuario. A diferencia de los modelos de filtrado colaborativo que calculan la similitud entre distintos usuarios, los métodos basados en contenido tratan de emparejar los intereses del usuario con los atributos de los ítems. (Mohanty et al., 2020, pp: 8-12).

De esta forma, los modelos basados en contenido, en su versión más básica, dependen de dos tipos de datos relacionados con los productos y los usuarios. En primer lugar, dependen de los distintos atributos que definen a los productos o ítems a recomendar. Por ejemplo, si los ítems fueran libros, los atributos podrían ser el género, el año de publicación, el autor y si el libro pertenece a alguna saga de libros o no. Adicionalmente dependen del perfil de usuario generado a partir de las interacciones con los distintos ítems. La forma más sencilla de considerar estos datos son los *ratings* de los que hablábamos en los modelos de filtrado colaborativo; sin embargo, en los métodos basados en contenido también podemos considerar algún tipo de datos recogidos de forma implícita.

En estos modelos, las valoraciones que han realizado usuarios distintos al usuario objetivo no tienen importancia alguna a la hora de realizar recomendaciones. Dadas las grandes cantidades de datos necesarios para poder implementar un modelo de recomendación basado en contenido, éstos suelen ser implementados cuando se dispone de una cantidad grande de atributos de ítems, especialmente cuando

estos atributos son simplemente palabras clave.

A grandes rasgos, los métodos basados en contenido están formados por 3 etapas:

1. Preprocesado de los datos o análisis del contenido: generalmente incluye la extracción de las características de los ítems y otra información relevante y la estructuración de los mismos para que puedan ser utilizadas en los pasos siguientes.
2. Fase de aprendizaje: involucra la construcción de un modelo de usuario a partir de feedback implícito (historial de compras) y explícito (comportamiento en la página o actividades) del usuario, la construcción de un modelo de aprendizaje con dicho *feedback* y los atributos de los ítems y la obtención de un modelo final al que comúnmente se denomina como *perfil de usuario* o *"user profile"*.
3. Filtrado: fase en la que se hace uso del perfil de usuario para recomendarle ítems en base a distintas métricas de similitud.

En las secciones siguientes se explicará con mayor detalle cada una de las etapas, pero antes es preciso mencionar que los pasos y fórmulas presentados a continuación están sacados de las explicaciones de Aggarwal, 2016; otros libros como por ejemplo Mohanty et al., 2020. o Brusilovsky et al., 2007 pueden identificar pasos dentro de las etapas o métodos diferentes.

2.2.2.1 Preprocesado de los datos

El primer paso para construir un modelo de recomendación basado en contenido consiste en extraer las características o atributos de los datos, generalmente representadas por palabras clave (por ejemplo, el género literario de un libro). Esta parte es altamente dependiente tanto del portal para el que se quiera construir el sistema de recomendación como de la base de datos disponible.

Una vez se han extraído distintas "bolsas de palabras" (conjuntos de palabras existentes en el diccionario) de los datos es necesario limpiarlas y representarlas de forma que sean adecuadas para el procesado de datos. Dentro de los pasos incluidos en este apartado, nos podemos encontrar con los siguientes:

- Eliminación de palabras vacías: muchas de las descripciones obtenidas de los productos van a contener palabras que lejos de representar alguna característica clave del producto, son palabras altamente utilizadas en un idioma. Por ejemplo, si extraemos descripciones en español de un determinado producto, una cantidad elevada de las palabras extraídas van a ser conjunciones, preposiciones o pronombres; dependiendo de los productos con los que se esté trabajando, estas palabras serán más o menos comunes.
- Stemming: consiste en concentrar distintas variaciones o sinónimos de una palabra en una misma palabra raíz.
- Extracción de palabras clave: el objetivo es identificar palabras que de manera recurrente se presentan juntas y tienen un significado diferente a los significados individuales de las palabras que las forman, es decir, locuciones, como puede ser "media naranja".

Una vez se han ejecutado dichos pasos, las palabras clave son transformadas en una representación en un espacio-vectorial, en la que distintos documentos están representados como bolsas de palabras con sus respectivas frecuencias. Un modelo espacio vectorial representa documentos en lenguaje natural de una manera formal utilizando vectores en un espacio lineal multidimensional. Generalmente se utiliza una expresión vectorial en la que las dimensiones del vector representan términos, frases o conceptos que aparecen en el documento; puede consultarse Salton et al., 1975.

Con los atributos de los ítems ya recogidos, se procede a recoger la información de los usuarios. Esta información puede ser recogida en forma de *ratings*, *feedback* implícito (como por ejemplo búsquedas recurrentes de un mismo producto aunque el usuario no lo haya comprado), opiniones en formato escrito (como pueden ser los comentarios que un usuario ha dejado en un producto tras comprarlo) o los denominados "*cases*". Los "cases" son ejemplos de ítems especificados por los usuarios en los que podrían estar interesados y suelen ser recolectados con el modelo de feedback relevante de Rocchio, aunque también se pueden utilizar árboles de decision u otros modelos probabilísticos. (Mohanty et al., 2020, pp: 165-197).

Independientemente del tipo de información que hayamos recogido de los usuarios, ésta se representa de forma numérica, ya sea unaria, binaria, por intervalos o como un *rating*.

Para finalizar con el preprocesado de los datos, es necesario identificar cuáles de los atributos de los ítems son lo suficientemente relevantes para retener en la representación del espacio vectorial mencionado anteriormente. Para ello se pueden utilizar tanto técnicas de selección como técnicas de ponderación, con la diferencia de que las primeras eliminan atributos que otorgan poca información mientras que las segundas simplemente le otorgan un peso pequeño. La mayoría de los métodos utilizados con este propósito tienen en cuenta los *ratings* de los usuarios y seleccionan las variables evaluando la sensibilidad de la variable dependiente respecto a una característica para evaluar la "informatividad" de la misma (Aggarwal, 2016, pp: 139-167). Para seleccionar estos atributos se utilizan, generalmente, el índice de Gini y el valor de entropía.

El índice de Gini es el método más utilizado pues es simple y fácil de entender; sin embargo, está principalmente pensado para variables binarias, valoraciones ordinales o valoraciones que no se distribuyen en un número grande de intervalos. Generalmente no suele haber problemas en ninguno de los casos pues, como el número de valoraciones posibles suele ser pequeño (de 1 a 5 estrellas, por ejemplo) puede discretizarse fácilmente. De esta forma, si l es el número de valores posibles de un ráting y dentro de los documentos que contienen una palabra particular w, $j_1(w), ..., j_l(w)$ son la proporción de los ítems valorados en cada uno de esos valores posibles; entonces el indice de Gini es:

$$\text{Gini}(w) = 1 - \sum_{i=1}^{l} j_i(w)^2.$$

Y toma valores entre $(0, 1 - 1/l)$, siendo los valores cercanos a 0 los más indicativos de poder discriminativo.

Por otro lado, en relación al valor de entropía, este se computa de forma similar al índice de Gini. Si partimos otra vez de un número l de valores posibles de una valoración y $j_1(w), ..., j_l(w)$ como la fracción de documentos que contienen una palabra particular w y que están valorados a cada uno de los posibles valores de l, entonces la entropía de la palabra w es:

$$\text{Entropia}(w) = -\sum_{i=1}^{l} j_i(w) \log(j_i(w)),$$

El valor de entropía se encuentra siempre entre 0 y 1, siendo los valores más cercanos a 0 aquellos que indican un mayor poder discriminativo. El valor de entropía tiene resultados similares al índice de Gini, con la diferencia de que, a cambio de ser más difícil de interpretar, tiene una base más solida en los principios matemáticos de la teoría de la información (Aggarwal, 2016).

Adicionalmente, existen otros métodos como pueden ser el estadístico χ^2 o la desviación normalizada. Se puede encontrar más información de estos métodos en Aggarwal, 2016.

2.2.2.2 Fase de aprendizaje y filtrado

Una vez se ha identificado cuáles son los atributos de los productos y otra información relevante para realizar recomendaciones, para cada uno de los usuarios hay que construir un modelo con sus preferencias para, posteriormente, utilizarlas como base de la recomendación. Esta fase está altamente relacionada con problemas de clasificación y regresión, dependiendo del tipo de datos. En caso de que los datos disponibles sean valores discretos, se estará en una situación similar a la de clasificación de textos mientras que si los datos son entidades numéricas, entonces se estará en un problema similar a la regresión. Sea como fuere, se asume que se tiene un conjunto de D_L documentos que están etiquetados por un usuario a al que se denomina usuario activo (es, también, el usuario para el que se obtendrán recomendaciones).

Los documentos se corresponden con las descripciones de diferentes ítems y incluen las valoraciones que el usuario activo a otorgado a cada ítem. Con estos datos se construye un modelo de aprendizaje para ese usuario a. Posteriormente, se define un conjunto de documentos DO_a sin valorar en los que se testeará el modelo. Dicho modelo puede ser ajustado para que dé una respuesta con la valoración que el usuario activo podría hacer de los documentos que no ha valorado o para que la respuesta sea una lista de las top-k recomendaciones para dicho usuario.

Los métodos de aprendizaje más comunes son la clasificación de vecinos más cercanos, los métodos basados en reglas y los métodos basados en regresión.

En la clasificación de vecinos más cercanos el primer paso para la clasificación sería definir una función de similitud (generalmente se utiliza la similitud coseno). Dicha función de similitud está definida de forma que si $\vec{X} = (x_1, ..., x_d)$ y $\vec{Y} = (y_1, ..., y_d)$ son un par de documentos en los que las frecuencias normalizadas de la palabra i-ésima están dados por x_i e y_i, entonces la similitud coseno es la siguiente:

$$Coseno(\bar{X}, \bar{Y}) = \frac{\sum_{i=1}^{d} x_i y_i}{\sqrt{\sum_{i=1}^{d} x_i^2}\sqrt{\sum_{i=1}^{d} y_i^2}}.$$

La similitud coseno es especialmente útil haciendo predicciones para ítems de los que desconocemos la preferencia del usuario activo. Para cada documento en DO_a se determinan sus k-vecinos más cercanos utilizando la similitud coseno y posteriormente se calcula el valor medio de los mismos. Dichos valores medios son los valores predichos para las valoraciones de cada uno de los documentos. A pesar de ser un método simple y fácil de implementar, su principal desventaja es la complejidad computacional y el tiempo necesario para encontrar las predicciones (Aggarwal, 2016, pp: 150-151; Aggarwal, 2015).

En segundo lugar están los métodos basados en reglas (Aggarwal, 2016). A pesar de que existe una gran variedad de clasificadores basados en reglas, dada su simplicidad, se dará una explicación de aquellos basados en reglas asociativas. Para ello, primero es necesario definir los conceptos de "apoyo" y "confianza". El apoyo de una regla hace referencia a la fracción de filas (la representación en palabras clave de un ítem) que satisfacen tanto los antecedentes como las consecuencias de una regla, mientras que la confianza de una regla es la fracción de filas que satisfacen las consecuencias de una regla de aquellas filas que ya se sabe que satisfacen los antecedentes de la misma. En los sistemas de recomendación basados en contenido, los "antecedentes" de una regla hacen referencia a las palabras clave que están presentes en la descripción de un ítem, mientras que las "consecuencias" hacen referencia a la valoración que ese usuario ha otorgado a ese ítem (Aggarwal, 2014; Aggarwal, 2015). De esta forma, una regla puede verse de la siguiente forma:

La descripción del ítem contiene el conjunto de palabras A $\Rightarrow$ Valoración

Así, si un antecedente satisface una fila particular, significa que todas las palabras contenidas en el antecedente están en dicha fila; mientras que una fila satisface la consecuencia de la regla si la

valoración de la consecuencia encaja con la valoración de la variable dependiente en esa fila (es decir, el *rating*).

La idea subyacente en estos métodos consiste en identificar todas las reglas del usuario activo y determinar, para un conjunto dado de ítems que no ha valorado, cuáles de esas reglas son útiles o se "encienden", es decir, que las palabras clave de la regla están incluidas en la descripción del ítem no valorado. Finalmente, la valoración para el nuevo ítem, será la media de las valoraciones de las reglas que se han encendido en dicho ítem.

Como pasaba con el método de los k vecinos más cercanos, esta técnica tendría que repetirse para cada uno de los usuarios, pues el conjunto de reglas que se identifican es exclusivo del usuario activo en ese momento.

En tercer lugar se encuentran los métodos basados en regresión. Estos métodos pueden ser utilizados para varios tipos de valoraciones, ya sean binarias, numéricas o basadas en intervalos. A pesar de que se pueden utilizar distintos tipos de regresiones para los métodos, se explicara el modelo con regresión lineal.

En este caso se dispone de M_L, una matriz $o \times d$ en la que se recogen los o documentos etiquetados en el conjunto D_L y que tienen un lexicon o repertorio de tamaño d, además, $\vec{y}$ es un vector columna de dimensión o que contiene las valoraciones del usuario activo par cada uno de los documentos. La idea subyacente de los métodos basados en regresión es que las valoraciones pueden ser modeladas como una función (lineal en este caso) de las frecuencias de las palabras. Así, si $\bar{W}$ es un vector fila de dimensión d que contiene los coeficientes de cada palabra en la función lineal que relaciona las frecuencias con las valoraciones, entonces el modelo lineal asume que las frecuencias de palabras en M_L se relacionan a los vectores de valoración de la siguiente forma (Aggarwal 2016, pp: 158-159):

$$\vec{y} \approx M_L \bar{W}^T.$$

Para maximizar la calidad de las predicciones, se debe minimizar la norma al cuadrado del vector $(M_L \bar{W}^T - \bar{y})$. A mayores, para reducir el sobreajuste se suele realizar la *regularización de Tikhonov* o *ridge regression* (véase Van Weasel. W, 2023).

Otras técnicas que se pueden utilizar para crear el modelo de aprendizaje son el clasificador de Bayes (que, en el contexto de los sistemas de recomendación basados en contenido se enfoca como un problema de clasificación de textos), los árboles de decisión o se puede, incluso, recuperar el Modelo de Feedback Relevante de Rocchio utilizado en la fase de extracción de información. Puede consultarse Dietmar et al., 2011, Mohanty et al., 2020 y Aggarwal, 2016.

2.2.2.3 Comparación con los modelos de filtrado colaborativo

La principal ventaja de los modelos basados en contenido en comparación a los modelos de filtrado colaborativo, es que, al estar basados en los atributos de los productos, es posible saber cuáles son las razones por las que se ha recomendado un ítem u otro. Al recomendar un determinado producto, estos sistemas hacen posible que al usuario activo se le den dichas razones. Por ejemplo, si un usuario ha comprado el libro "Palabras radiantes" se le podría enviar un mensaje similar a:

Se ha recomendado el libro "Juramentada" porque es de fantasía, forma parte de una saga literaria y porque el usuario ha leído el libro "Palabras radiantes".

Este tipo de mensajes proporciona una mayor transparencia y le da al usuario razones adicionales para comprar un producto o consumir un servicio (Dietmar et al., 2011, pp: 74-79).

Otra de las ventajas de los sistemas basados en contenido respecto a los sistemas de filtrado colaborativo se da con el anteriormente mencionado "inicio en frío". A pesar de que los sistemas basados en contenido no llegan a solucionar del todo este problema, sí que presentan algún beneficio adicional. Siempre que el usuario no sea nuevo (es decir, que no haya valorado ningún producto) se le van a recomendar ítems que le puedan gustar, independientemente de si éstos son nuevos o no. Mientras que los sistemas de filtrado colaborativo suelen tener problemas tanto con nuevos ítems como con nuevos usuarios, los sistemas basados en contenido solo experimentan dificultades con los últimos. Además, como al generar recomendaciones para cada uno de los usuarios solo se tienen en cuenta datos del usuario activo, los sistemas basados en contenido ofrecen independencia entre usuarios (Dietmar et al., 2011, pp: 74-79).

Con todo esto, el problema de generar recomendaciones para usuarios nuevos puede llegar a ser mucho mayor en estos modelo, pues, para evitar el sobreajuste lo máximo posible, la cantidad de documentos necesarios para la clasificación de texto es muy elevada (Dietmar et al., 2011, pp: 74-79).

En relación a las desventajas de estos sistemas, es interesante analizar aquellas derivadas de la personalización de las recomendaciones. Así como las recomendaciones más "personalizadas" de los sistemas de filtrado colaborativo pueden ser vistas como una ventaja, en la mayoría de las situaciones se da el caso contrario, pues tienden a sobreajustar los datos. En el contexto de los recomendadores basados en contenido este problema se denomina como *"sobreespecialización"*. En la Sección 2.1 se mencionaba que dos de los objetivos de los sistemas de recomendación eran la primicia (recomendar productos nuevos) y la causalidad (recibir recomendaciones inesperadas). Debido a la "sobreespecialización" los sistemas basados en contenido suelen recomendar siempre ítems con atributos muy similares a aquellos que ha valorado el usuario activo en el pasado, de forma que no alcanzan los objetivos mencionados.

Otro de los problemas en comparación con los modelos de filtrado colaborativo es la dificultad de obtener el *feedback* de distintos usuarios. Mientras en los sistemas colaborativos la única información que se utiliza son las valoraciones históricas de los usuarios, los sistemas basados en contenido necesitan, adicionalmente, información sobre los atributos de los productos o ítems y el comportamiento del consumidor, aumentando la complejidad del proceso de recolección de datos. (Poonam et al., 2009)

Por último, cabe mencionar que ambos métodos pueden llegar a tener problemas con los tiempos de computación, pues los dos modelos requieren de una fase en la que se calculen los vecinos más cercanos, ya sea para los usuarios o para los ítems. Este proceso de cálculo puede llevar demasiado tiempo cuando se tiene un número de usuarios o ítems demasiado grande. Por ejemplo, si el número de usuarios n es del orden de unos cuantos cientos de millones, el tiempo de ejecución ($O(n^2 \cdot m')$, donde $m' \ll m$ es el número máximo de valoraciones de un ítem) puede acabar resultando impráctico (Aggarwal, 2016, pp: 45-47).

En conclusión, ambos métodos de recomendación presentan sus propias ventajas y desventajas respecto al otro. Mientras los modelos de filtrado colaborativo atacan los objetivos de los sistemas de recomendación (primicia y casualidad), éstos tienen problemas con el inicio en frío. Por otro lado, a pesar de que los sistemas basados en contenido solucionan parcialmente el problema de inicio en frío y otorgan una mayor transparencia en las recomendaciones, la cantidad de información que requieren es elevada y suelen realizar un sobreajuste de los datos. Por estas razones, uno de los enfoques que se puede dar a los sistemas de recomendación es la realización de un modelo híbrido que permita combinar las ventajas de otros métodos para acabar con un modelo más robusto.

2.2.3. Métodos basados en conocimiento

A lo largo de este capítulo se ha visto como los modelos de de filtrado colaborativo y los modelos de recomendación basados en contenido necesitan de una gran cantidad de datos para funcionar

adecuadamente, de lo contrario se encontrarán con el problema de inicio en frío. Además, estos modelos pueden dar problemas en situaciones en las que, a pesar de no tener usuario nuevos, no hay demasiada información. Estas situaciones pueden darse cuando lo que se pretende recomendar son, o bien, productos que no se suelen comprar muy frecuentemente, como una casa o un ordenador, o con productos altamente personalizables como pueden ser los coches. En dichas situaciones hay factores a tener en cuenta como el tiempo entre compras o valoraciones (un ordenador que se valoró hace 10 años, no tendrá la misma valoración a día de hoy) y los requisitos que pueden llegar a definir los consumidores (el coche x tiene que ser de color negro y el precio máximo es y) que los modelos basados en contenido o los de filtrado colaborativo no tienen en consideración. (Dietmar et al., 2011, pp:81-124)

Los sistemas de recomendación basados en conocimiento permiten afrontar ambas situaciones a la vez, principalmente porque no utilizan las valoraciones históricas de los consumidores para generar las recomendaciones. En su lugar, para recomendar productos, estos sistemas utilizan la similitud entre los requisitos del cliente y las características de los productos (sistemas basados en casos o "case-based recommender systems") o funcionan bajo unas reglas explícitas de recomendación (sistemas basados en restricciones o "constraint-based recommender systems"). Los sistemas de recomendación basados en conocimiento dependen de las descripciones de los productos en forma de atributos relacionales (atributos que representan una propiedad de una relación), a diferencia de los sistemas basados en contenido que lo hacían en forma de palabras clave.

Como el proceso de recomendación en los sistemas basados en conocimiento es altamente interactivo, hay un ligero cambio en la interpretación de los sistemas de conocimiento como modelos de recomendación. Más que recomendar un producto específico, lo que suelen hacer es "guiar al consumidor a determinados productos de acuerdo a sus especificaciones y requerimientos". Todos aquellos sistemas de recomendación que necesitan datos no explotados por los sistemas de filtrado colaborativo o los métodos basados en contenido son considerados como modelos basados en conocimiento (Burke, 2000).

Dada la alta interactividad de los sistemas basados en conocimiento, independientemente de si están basados en casos o en restricciones, el usuario puede interactuar con el recomendador de diversas formas. Dependiendo de cómo se interactúe con el sistema, los sistemas de recomendación pueden ser:

- Sistemas conversacionales: se utilizan cuando el ámbito de recomendación es complejo y las preferencias del usuario han de ser dadas mediante bucles de *feedback* pues éstas solo pueden ser determinadas a través del contexto de varias conversaciones.

- Sistemas basados en búsqueda: en ellos las preferencias del usuario se determinan tras consultarle una serie de preguntas previamente definidas.

- Sistemas basados en navegación: en este caso, se llega a las recomendaciones finales haciendo cambios a las recomendaciones actuales mostradas al usuario activo. Una vez el usuario obtiene x recomendaciones, este puede cambiar o concretar los requisitos que ha establecido previamente para obtener otras nuevas. Tras repetir este proceso varias veces, el usuario puede llegar a encontrar el ítem deseado.

Previamente se han mencionado los dos tipos de sistemas de recomendación basados en conocimiento (basados en restricciones y basados en casos). Ambos son muy similares pues en ambos se asume que el usuario no tiene la capacidad de definir exactamente el producto deseado y ambos funcionan a través de varias interacciones con él. Sin embargo, la diferencia radica en el modo que el usuario tiene de interactuar con el sistema e introducir los requisitos.

En los sistemas basados en restricciones el usuario especifica requisitos del producto y posteriormente va añadiendo, modificando o eliminando el conjunto original de requisitos. Por otro lado, en

los sistemas basados en casos, el usuario especifica algunos atributos del producto deseado y, a través de interacciones sucesivas, va determinando que atributo de los resultados ha de ser modificado y de qué forma; generalmente este proceso iterativo implica un sistema conversacional, más que una simple modificación de los atributos.(Aggarwal, 2016, pp: 172-181).

2.2.3.1 Métodos basados en restricciones

Los modelos basados en restricciones permiten a los usuarios establecer distintos requisitos o restricciones en los atributos del producto. Posteriormente, se utiliza un conjunto de reglas para emparejar dichos requisitos con los atributos. Sin embargo, a pesar de que en algunas situaciones los requisitos establecidos por el usuario sí coinciden con los atributos del producto (por ejemplo el número de habitaciones al buscar casas) esto no siempre es así. Cuando se da el caso de que las restricciones no coinciden con el producto es necesario "mapear" dichas restricciones con los atributos de los productos, para que sea posible filtrar que productos ofrecer. Dependiendo del caso, ese "mapeo" puede ser más o menos obvio (el riesgo de una inversión se puede obtener cuando el usuario marca *inversiones conservadoras*). Para evitar este problema se utilizan las denominadas *bases de conocimiento*, que contienen reglas adicionales que permitan realizar el "mapeo"(Aggarwal, 2016, pp: 172-181). Un ejemplo de las reglas puede ser:

$$\textit{Inversiones conservadoras} = \textit{Bajo riesgo} \Rightarrow \textit{Renta fija} = \langle\ \textit{Lista de inversiones relevantes}\ \rangle$$

Debido a la función de "mapeo" que cumplen las reglas, comúnmente son denominadas como *condiciones filtro*. En el ejemplo anterior se puede observar como dichas reglas son altamente dependientes del ámbito en el que se esté desarrollando el sistema de recomendación y que no siempre va a ser posible identificar los atributos del producto de forma sencilla. En este caso la información podría encontrarse, por ejemplo, en la página de letras del tesoro pero en otros casos será necesario utilizar técnicas de minado de datos con bases de datos específicas.

A mayores, existen las llamadas *condiciones de compatibilidad*. Dada la alta personalización que permiten los sistemas basados en conocimiento, en muchos casos se puede dar que uno de los atributos requeridos solo esté disponible con algunos productos específicos. Algunos ejemplos de esta situación podrían ser un coche de gama baja que no está disponible con un motor muy potente o un libro que no tiene edición de tapa dura. En esas situaciones son las *condiciones de compatibilidad* las que rápidamente detectan posibles inconsistencias en los requerimientos del usuario.

Otra situación a destacar en la que no coinciden los requisitos o restricciones con los atributos del producto se da cuando dichas restricciones vienen de datos personales del usuario. Un ejemplo podría ser:

$$\textit{Edad} = \textit{20-25} \Rightarrow \textit{Riesgo} = \textit{Elevado}$$

En dichos casos la información se obtiene a través de los datos históricos (específicos del ámbito en el que se desarrolla el trabajo) que están disponibles. En el ejemplo anterior, el banco ha podido determinar a través de inferencia en sus bases de datos que, generalmente, las personas en un rango de edad de 20-25 años suelen tener poca aversión al riesgo.

Así, los diferentes tipos de inputs que se pueden dar en un sistema de recomendación basado en restricciones son los siguientes:

1. Atributos que describen propiedades inherentes del usuario y requerimientos específicos de los productos. En la mayoría de los casos, los usuarios no especifican sus propiedades o requerimientos de forma consistente, por lo que las recomendaciones varían de uso a uso del sistema.

2. Bases de conocimiento que "mapean" los requisitos de los clientes con los atributos de los productos. Esto puede hacerse de forma directa (*Inversiones conservadoras = Bajo riesgo* $\Rightarrow$ *Renta fija* = $\langle$ *Lista de inversiones relevantes* $\rangle$) o indirecta (*Edad = 20-25* $\Rightarrow$ *Riesgo = Elevado*). En el caso de que el "mapeo" se realice de forma indirecta, las reglas pueden verse como una forma de relacionar atributos del consumidor con atributos del producto atendiendo a lo que típicamente se espera del consumidor. Comúnmente estos datos son obtenidos a partir de bases de datos públicas, la experiencia de la compañía o a través del minado de bases de datos históricas.

3. El catálogo de productos que contiene los distintos atributos y características de cada uno de los productos o servicios ofertados.

2.2.3.1.1 Realización de la recomendación

Una vez se ha explicado qué son las restricciones y cuáles son los tipos de restricciones a los que el sistema se puede enfrentar, para determinar que productos enseñar solo habría que identificar cuáles las cumplen y mostrarlos. De esta forma, lo que se hace es ver dichas reglas como una serie de consultas en el catálogo que permiten filtrar los productos.

En primer lugar para cada requerimiento o restricción que ha introducido el usuario, se comprueba si cumple el antecedente de alguna de las reglas presentes en la base de conocimiento. En caso de que cumpla dicho antecedente, la consecuencia de una norma será activada. Por ejemplo, si el usuario introduce su edad, se activaría la siguiente consecuencia:

$$Edad = 20\text{-}25 \Rightarrow Riesgo = Elevado$$

La consecuencia activada (que la inversión tenga un riesgo elevado) ha de añadirse al conjunto de restricciones y requerimientos que había seleccionado inicialmente el usuario. Posteriormente, se vuelve a comprobar cuáles de dichos requerimientos activan una consecuencia. Si la condición de que el riesgo de la inversión sea elevado no activa ninguna consecuencia, entonces el primer paso acaba; si dicha condición activa alguna consecuencia, éstas se incluirán en la base de datos y se volverá a hacer la comprobación.

Seguidamente, se construye la consulta en la base de datos. Dicha consulta es la intersección de todos los requerimientos a los que se ha llegado en el paso anterior.

Y, finalmente, se muestran los resultados de la consulta, obteniendo así un conjunto o muestra inicial de recomendaciones.

Es preciso mencionar que una de las características de seleccionar los productos relevantes de esta manera es que las recomendaciones no son persistentes. Un usuario en una misma sesión podría obtener recomendaciones totalmente distintas aunque haya cambiado sólo un requerimiento y dos usuarios que hayan marcado exactamente las mismas restricciones obtendrán las mismas recomendaciones.

Tras haber realizado los pasos anteriores se obtiene una muestra inicial de recomendaciones. Sin embargo, generalmente la muestra inicial de productos no coincide con las recomendaciones finales, sino que es sólo uno de los pasos del proceso de interacción del usuario con el sistema.

A continuación se detallan los tres pasos del proceso completo de interacción con el usuario:

En el primer paso, el usuario especifica sus requerimientos iniciales a través de una interfaz interactiva. En algunos casos, en lugar de pedir los requerimientos de forma explícita, los sistemas simplemente realizan una serie de preguntas preestablecidas para poder obtener dichos requerimientos de forma implícita.

A continuación, en el segundo paso se sigue el procedimiento explicado en los párrafos anteriores para mostrarle al usuario una lista de recomendación inicial. Es ya en este paso donde los sistemas aprovechan una de sus ventajas respecto a los métodos de filtrado colaborativo y muestran, junto con las recomendaciones, una explicación de por qué cada producto ha sido recomendado (de forma similar a los modelos basados en contenido). Algunos resultados posibles son que los requisitos que haya introducido el usuario no devuelvan ningún producto (el conjunto es vacío) o que los requisitos establecidos sean demasiado débiles y el sistema devuelva demasiados productos.

En caso de que una consulta no devuelva ningún producto, al usuario se le suelen ofrecer dos opciones: empezar desde el inicio escogiendo un conjunto distinto de requerimientos o relajar las restricciones para el siguiente proceso interativo (Paso 3). Si el usuario ha decidido relajar los requerimientos, el sistema generalmente otorga una serie de sugerencias para relajar. Dichas recomendaciones son denominadas *propuestas de reparación* y su funcionamiento recae en determinar el conjunto mínimo de requerimientos incompatibles y presentárselos al usuario. De esta forma, si al usuario se le muestra que *Riesgo = bajo* y *Mercado = Georgia* son incompatibles, él entenderá que debe o bien asumir más riesgo o cambiar de mercado. Para determinar éstos conjuntos se puede utilizar una búsqueda *bottom-up* de las combinaciones de atributos seleccionados y, en caso de que haya un número demasiado grande de requisitos (no suele ser común), se pueden utilizar procedimientos como "QUICKXPLAIN" o "MINRELAX". Puede consultarse literatura existente en Jannach D., 2006; Rodler P., 2022 y Felfernig et al., 2004.

Si, por el contrario, se da la situación de que el sistema devuelve demasiados productos, lo recomendable es proveer al usuario de recomendaciones de restricciones para añadir. Comúnmente dichas sugerencias se suelen obtener a través del análisis de registros históricos, ya sea de todos los usuarios o del usuario activo (otorga recomendaciones más personalizadas pero es más difícil de obtener). Dichos registros históricos se utilizan para recomendar restricciones populares. Por ejemplo, si un usuario especifica los requerimientos de riesgo y mercado de una inversión, se buscan los registros que contengan ambos requerimientos (con los mismos valores especificados por el usuario) y se identifican las top-k sesiones vecinas con atributos comunes. Es decir, si en dichos top-k vecinos se determina que la restricción más popular es la del tipo de inversión, al usuario se le recomendará especificar el tipo de inversión (Aggarwal, 2016, pp: 172-181).

Finalmente en el tercer y último paso, el usuario modifica sus requerimientos iniciales añadiendo nuevas restricciones o modificando las restricciones iniciales; después se vuelve al primer paso.

2.2.3.1.2 Conocimiento del usuario

En determinadas situaciones, el usuario objetivo puede no ser capaz de determinar las restricciones suficientes para generar el bucle de interacciones explicado en la sección anterior. Por ejemplo, en el caso de las inversiones, el usuario puede ser capaz de cubrir el riesgo pero no el tipo de inversión que está deseando realizar. Para poder operar bajo esas situaciones, los sistemas basados en reglas acostumbran a obviar atributos no especificados o utilizar valores por defecto.

En el primer caso, se obvian los atributos que el usuario no ha especificado y se devuelven todas las recomendaciones que cumplan las pocas condiciones que el usuario ha establecido.

En el segundo caso, los valores por defecto pueden ser utilizados de dos formas: cubriendo con los valores por defecto aquellos campos o atributos que el usuario no haya seleccionado, o mostrándole al usuario dichos valores por defecto como sugerencias para que pueda ir especificando atributos adicionales. En Aggarwal (2016), se sugiere que dichos valores por defecto se utilicen únicamente para guiar al usuario, de forma que no se incurra en ningún tipo de sesgo al realizar la búsqueda de productos con

restricciones que el usuario no ha seleccionado. Es preciso mencionar que, en muchos casos, los valores por defecto en un atributo pueden influir en los valores por defecto de otro atributo. Por ejemplo, si al buscar una casa el valor por defecto es que tenga 5 habitaciones, dicha selección puede hacer que el valor por defecto del número de baños sea de 3 en lugar de 2.

En el supuesto de que se opte por utilizar valores por defecto, es necesario determinar cómo se van a calcular dichos valores para poder introducirlos de forma explícita en la base de conocimiento. Uno de los enfoques más utilizados es usar los datos históricos de los usuarios a partir de los valores medios de los mismos. A mayores, como también se dispone de los datos de cada sesión realizada por un usuario en concreto, conforme vaya aumentando el número de sesiones del usuario, los valores por defecto pueden ser sustituidos por aquellos introducidos previamente por el usuario.

Finalmente, el último aspecto a tener en cuenta en el proceso de recomendación es el orden en el que mostrar los productos que cumplen con las condiciones establecidas por el usuario. Para ello existen una gran variedad de métodos y uno de los más simples y utilizados es que el usuario, en el momento de establecer los requerimientos, introduzca un orden de importancia de cada uno de los atributos. En el ejemplo anterior de las inversiones, el usuario podría ordenar, por ejemplo, el riesgo de la inversión, su tipo, el sector y el país en el que se realizará la inversión. Así, los productos se ordenan atendiendo al atributo más importante.

Sin embargo, la principal desventaja de utilizar un único atributo es que se resta importancia a otros atributos. Para solucionarlo, un enfoque común es el de utilizar funciones de utilidad que permitan ordenar los ítems que cumplen las restricciones. Si $\vec{V} = (v_1, ..., v_d)$ es un vector que contiene los valores de los atributos que definen a los productos seleccionados, la dimensión del espacio de contenido es d. Así, se definen las funciones de utilidad como funciones ponderadas de las utilidades de los atributos individuales. A cada atributo se le asigna un peso pe_j y una contribución definida por $f_j(v_j)$ dependiendo del valor v_j del atributo emparejado (Aggarwal, 2016, pp: 172-181). De esta forma, la utilidad del producto seleccionado ($U(\bar{V})$) es la siguiente:

$$U(\bar{V}) = \sum_{j=1}^{d} pe_j \cdot f_j(v_j).$$

Para poder determinar las utilidades de los productos, pe_j y $f_j(\cdot)$ han de ser especificados. Por ello, el diseño de funciones de utilidad competentes necesita del conocimiento específico de datos inherentes al ámbito en el que se desarrolla el sistema de recomendación, o que existan datos históricos de usuarios que permitan identificar las preferencias de los usuarios. Con el objetivo de obtener datos de entrenamiento, se suele pedir a algunos usuarios que ordenen según sus preferencias unos productos de ejemplo para, posteriormente, mediante modelos de regresión, identificar y establecer las funciones de utilidad.

2.2.3.2 Métodos basados en casos

Los modelos de conocimiento basados en casos utilizan métricas de similitud para poder obtener productos similares a los que ha especificado el usuario; dichos productos especificados son lo que se denomina como casos. A diferencia de los métodos basados en restricciones, aquí no se establecen restricciones estrictas; el usuario establece sus preferencias y, a través de las métricas de similitud mencionadas anteriormente, se determina que productos son más similares al producto preferido por el usuario y se muestran. Por esta razón que los métodos basados en casos no tienen el problema que tenían los métodos basados en restricciones de devolver un conjunto vacío de ítems.

Otra de las grandes diferencias de los sistemas basados en casos respecto a los modelos basados en restricciones es que, mientras los segundos siempre han necesitado que se relajen, añadan o eliminen las restricciones iniciales, los modelos basados en casos inicialmente abogaban únicamente por la modificación de las mismas. A raíz del desarrollo de esta idea, surgió la técnica llamada *crítica*. La idea subyacente de dicha técnica consiste en que los usuarios seleccionen uno o varios de los objetos devueltos por el sistema de recomendación y especifiquen nuevas condiciones de la siguiente forma: (Aggarwal, 2016, pp: 181-195)

"Devuélveme más ítems similares a X, pero que sean distintos en el atributo(s) Y según las direcciones Z"

Dependiendo de si se elige uno o varios atributos distintos a los ítems X, existen grandes variaciones en el funcionamiento del sistema. Además, el enfoque basado en casos generalmente provoca que los ítems devueltos por el sistema vayan disminuyendo de un paso al siguiente. Aún así, existe la posibilidad de aumentar el alcance del sistema haciendo que las consultas se hagan sobre la base de datos entera en lugar de sólo sobre los ítems que ha devuelto el paso anterior. Sin embargo, ampliar el alcance del sistema puede provocar no sólo que los ítems se vuelvan a cada paso más distintos que los del paso actual, sino también que las recomendaciones se vuelvan irrelevantes conforme se avanza en el proceso iterativo.

Uno de los beneficios que presenta este proceso iterativo es que, al ser un modelo de aprendizaje para el usuario, éste puede acabar encontrando ítems que no habría encontrado en caso de no haberse vuelto gradualmente más consciente de las opciones que tiene disponibles en cada uno de los pasos. Dicho aspecto es, posiblemente, una de las razones por las que se utilizan los sistemas basados en casos ya que no es extraño que los usuarios no sean conscientes de algunos atributos al principio del proceso. Por ejemplo, siguiendo el ejemplo de las inversiones de la sección anterior, el usuario en el primer paso puede no ser consciente de cuál es un nivel de riesgo aceptable para una inversión. Dicho de otra forma, a través de este proceso iterativo se hace posible cerrar la brecha existente entre el conocimiento del usuario y la disponibilidad de productos.

Como ya se ha mencionado, los sistemas basados en casos dependen del diseño de las métricas de similitud y de las *críticas*. A continuación, se entrará más en detalle en cada uno de dichos aspectos.

2.2.3.2.1 Métricas de similitud

El diseño de las métricas de similitud y la incorporación de la importancia de los distintos atributos en ellas es esencial si se quieren obtener resultados relevantes tras cada interacción. Los sistemas iniciales utilizaban atributos ordenados en un orden decreciente de importancia; ordenados previamente de acuerdo al primer criterio, seguido del segundo y así sucesivamente (Burke et al., 1997).

Dicho de otra forma, tenemos un sistema en el que cada producto está descrito por d atributos y queremos determinar la similitud entre dos vectores parciales de atributos definidos en el subconjunto C de los d atributos ($|C| = c \leq d$). Sean $\vec{AT} = (at_1, ..., at_d)$ y $\vec{T} = (t_1, ..., t_d)$ dos vectores de atributos de dimensión d que pueden estar parcialmente especificados (y asumiendo que, como mínimo, el subconjunto de atributos $C \subseteq \{1...d\}$ está especificado en ambos vectores), entonces la función de similitud $f(\vec{T}, \vec{AT})$ entre $\vec{T}$ y $\vec{AT}$ donde $\vec{T}$ representa al ítem objetivo es la siguiente (Aggarwal, 2016, pp: 172-181):

$$f(\vec{T}, \vec{AT}) = \frac{\sum_{i \in C} pe_i \cdot \text{sim}_{at}(t_i, at_i)}{\sum_{i \in C} pe_i}.$$

Donde $\text{sim}_{at}(t_i, at_i)$ representa la similitud entre los valores t_i y at_i, y pe_i representa el peso del i-ésimo atributo y regula su importancia relativa.

Al calcular $f(\vec{T}, \vec{AT})$, la determinación de la importancia relativa de los atributos puede ser complicada. Para identificar dicha importancia se puede optar por que un experto en el campo de trabajo introduzca los pesos a mano o por aprender los valores a través del feedback de usuarios (presentándoles y pidiéndoles a los usuarios que valoren distintas parejas de ítems para después hacer regresión y determinar valores de pe_i).

Por otro lado, la similitud $\text{sim}_{at}(t_i, at_i)$ entre los atributos puede determinarse de diversas formas; pero antes es preciso mencionar que los atributos pueden ser tanto cuantitativos como categóricos, así como simétricos o asimétricos. La simetría de los atributos hace referencia a que, en algunos supuestos, devolver un ítem con un valor en un atributo superior al requerido puede implicar cosas diferentes dependiendo del atributo. Por ejemplo: precio y hercios en un monitor de ordenador donde no se quiere productos más caros pero si se aceptan productos con más hercios.

Para calcular la similitud entre dos atributos cuantitativos puede utilizarse la siguiente función propuesta por Aggarwal C., 2016:

$$\text{sim}_{at}(t_i, at_i) = 1 - \frac{|t_i - at_i|}{\text{máx}_i - \text{mín}_i},$$

donde máx_i y mín_i representan los valores posibles máximos y mínimos del i-ésimo atributo

También sería posible incluir la desviación típica de los datos históricos en la fórmula o, en caso de que tengamos atributos asimétricos, incluir una *recompensa asimétrica* (Aggarwal, 2016, pp: 172-181):

$$\text{sim}_{at}(t_i, at_i) = 1 - \frac{|t_i - at_i|}{max_i - min_i} + \alpha_i \cdot I(at_i > t_i) \cdot \frac{|t_i - at_i|}{max_i - min_i},$$

donde α_i es un parámetro preestablecido por el usuario e $I(at_i > t_i)$ una función indicadora que toma el valor 1 si $at_i > t_i$ y 0 en caso contrario. Dicha función actúa únicamente cuando el valor del atributo (at_i) es mayor que el valor especificado por el usuario (t_i).

La función indicadora presente en la ecuación anterior tiene sentido para atributos en los que el usuario acepta productos con valores superiores al especificado (la resolución de una pantalla, por ejemplo). En caso de que se prefirieran valores bajos (como el precio) dicha función indicadora pasaría a ser $I(at_i < t_i)$. De esta forma, la función de similitud podría llegar a verse como una función de utilidad.

Si los atributos son categóricos, la determinación de valores de similitud es más complicada. Por ello, generalmente se suelen construir jerarquías atendiendo al sector de los productos a recomendar, de forma que valores más cercanos en la jerarquía implican mayor similitud. Dependiendo de dicho sector, las jerarquías pueden estar disponibles en diversas fuentes o habrá que construirlas específicamente para el sistema de recomendación (Aggarwal, 2016, pp: 181-194) . En la Figura 2.1 se muestra un ejemplo de una clasificación jerárquica para el ejemplo de los monitores de ordenador mencionado en la simetría de los atributos.

Finalmente, otro aspecto a tener en cuenta al establecer las métricas es la recomendación de ítems similares pero suficientemente diversos (uno de los objetivos secundarios de los sistemas de recomendación). Para ello, generalmente se utiliza el procedimiento denominado *bounded greedy selection strategy*. En él, se empieza con los $b \cdot k$ casos más similares a un ítem objetivo ($b > 1$) y con un conjunto vació RV para poder crear una métrica de calidad que combine similitud y diversidad entre casos y mostrar los casos con más calidad respecto al caso objetivo. Puede consultarse Smyth et al., 2001.

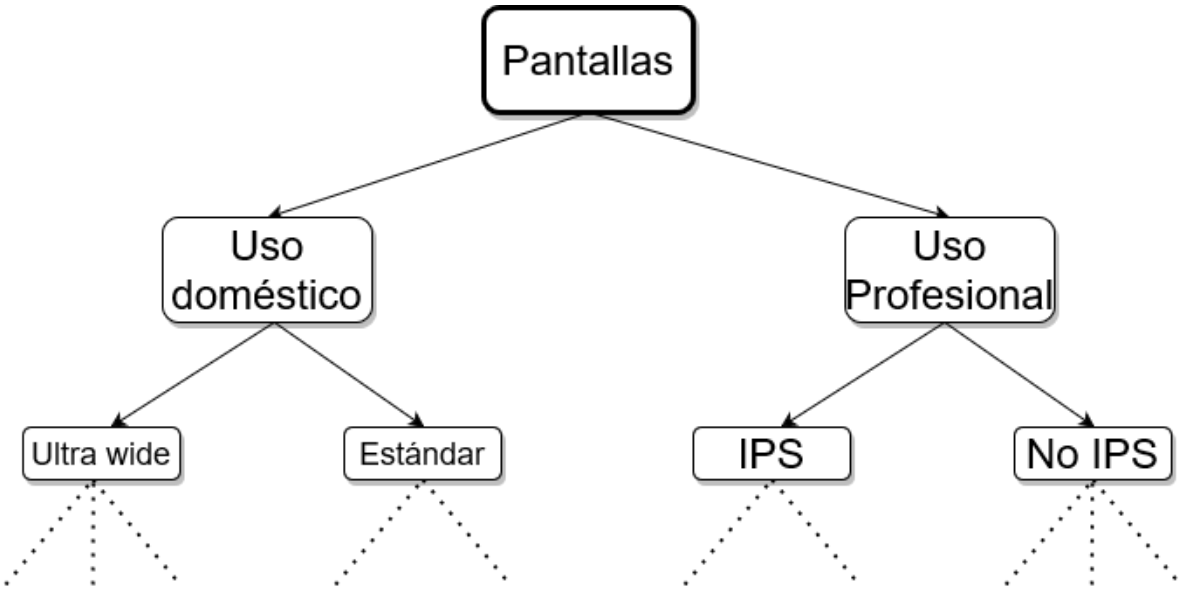

Figura 2.1: Ejemplo de clasificación jerárquica de atributos

2.2.3.2.1 Formas de realizar las críticas

Como se ha mencionado a lo largo de la Sección 2.2.3.2, muchas veces los usuarios no tienen claras sus preferencias o requisitos exactos en la búsqueda inicial y a veces tampoco pueden traducir de una forma adecuada sus preferencias para introducirlas como criterios de búsqueda.

El objetivo de las críticas es que los usuarios, después de realizar la búsqueda inicial, puedan modificar dichas preferencias iniciales para mejorar sus recomendaciones. Una vez se introduce una crítica, en lugar de únicamente mostrar productos adicionales, se eliminan todos aquellos productos que no cumplen la condición de la crítica y se devuelven los productos más similares a los seleccionados. Cuando el usuario acaba especificando varias críticas a través de distintos pasos de recomendación, el sistema da prioridad a las críticas más recientes. Según Aggarwal C., 2016 (pp: 181-194), dependiendo de como se introduzcan las críticas, podemos encontrar tres tipos distintos: Críticas simples, compuestas y dinámicas.

En las críticas simples, una vez se le ha devuelto al usuario la lista de productos recomendados, éste ha de hacer un único cambio en los atributos de los productos que se le han recomendado. Dependiendo del caso, el usuario podrá establecer un cambio específico (en el ejemplo de la inversión podría escoger el riesgo exacto de la misma; *riesgo=medio*) o simplemente incrementar o reducir un atributo en concreto ($riesgo \geq medio$), lo que se denomina como *crítica direccional*.

Las críticas direccionales eliminan únicamente los productos situados en el lado "equivocado" de la crítica (inversiones de riesgo < medio) y son particularmente útiles en todos esos ámbitos o mercados en los que los usuarios no tienen muy claros los atributos de los productos. Además, como generalmente suelen ser implementadas de una forma más conversacional, tienen una mejor recepción por parte de los usuarios. Sin embargo, las críticas simples afrontan problemas varios como pueden ser los ciclos de interacciones extremadamente largos (pues las restricciones se modifican de una en una) o dificultades para mantener valores de atributos constantes a lo largo de varias interacciones (pues al introducir nuevas preferencias, las preferencias iniciales pueden dejar de ser posibles).

Para hacerle frente al problema de los ciclos de interacciones excesivamente largos surgieron las críticas compuestas. Con estas críticas los usuarios pueden establecer varias modificaciones a sus requerimientos en un único paso (sigue habiendo los dos enfoques anteriores: establecer cambios específicos o simplemente incrementar o reducir un atributo). Introducir varios cambios en cada uno de los casos le permite al usuario hacer saltos más grandes en el catálogo de producto y tener un mayor control del proceso de recomendaciones. No obstante, no se puede concluir que las críticas compuestas ayuden al usuario a entender mejor el catálogo de productos y los atributos de los mismos. Su principal desventaja es que, al modificar varios parámetros a la vez, devuelven nuevas recomendaciones que no están basadas en los resultados anteriores, estas recomendaciones son denominadas comúnmente como recomendaciones *estáticas*(Aggarwal, 2016, pp: 181-194).

Por último, las críticas dinámicas surgen para solucionar el problema principal de las críticas compuestas, las recomendaciones estáticas. La idea subyacente es la de utilizar técnicas de minado de datos para determinar cuáles pueden ser las combinaciones de modificaciones más fructíferas o convenientes basadas en las recomendaciones actuales. Posteriormente, se le permite al usuario realizar alguna modificación de aquellas combinaciones identificadas. Para detectar dichas combinaciones e identificar las más relevantes, se recupera el concepto de *apoyo* visto en la Sección 2.2.2.2

A modo de breve recapitulación, los sistemas de recomendación basados en conocimiento son especialmente útiles en aquellos ámbitos o mercados en los que los productos son altamente personalizables y funcionan bien cuando el usuario no tiene mucho conocimiento sobre los atributos del producto. Como estos sistemas utilizan los requisitos del usuario y no los datos históricos para devolver las recomendaciones, no experimentan dificultades a la hora de afrontar situaciones que en otros modelos resultarían con problemas de inicio en frío. Esa ventaja puede ser, a su vez, una desventaja pues no utilizan los datos históricos de los usuarios para devolver mejores recomendaciones.

2.2.4. Métodos híbridos

A lo largo de este capítulo se han explicado los distintos métodos "puros" (que no combinan distintos modelos) más destacados, se ha visto como obtienen la información y como siguen diferentes enfoques para realizar las recomendaciones. Además, se ha comentado sobre las distintas ventajas y desventajas que tiene cada uno y de su diferente rendimiento a pesar de tener todos el mismo objetivo, devolver recomendaciones personalizadas al usuario.

Así, se observó como los modelos de filtrado colaborativo utilizan datos derivados de la comunidad para recomendar productos, mientras los modelos basados en contenido utilizan los atributos de los productos y los modelos basados en conocimiento sacan conclusiones a partir de los requisitos del cliente y las características de los productos. Se advirtió, también, como mientras los dos primeros métodos tienen problemas con la escasez de datos (*inicio en frío*), los últimos requieren un mayor esfuerzo para obtener las bases de conocimiento necesarias e implementar los modelos. Sin embargo, ninguno de estos modelos es capaz de solucionar todos los problemas a la vez. Es por esta razón que uno de los objetivos actuales es el desarrollo e investigación de los modelos híbridos; modelos que permiten combinar las fortalezas de los 3 métodos anteriores mientras se reducen sus respectivas desventajas.

Los métodos híbridos son "enfoques técnicos" que combinan varias implementaciones diferentes de algoritmos o sistemas de recomendación. Podría decirse que hay tres maneras o enfoques distintos para diseñar dichos sistemas de recomendación: el *diseño agrupado*, el *diseño monolítico* y el *diseño mixto*.

En primer lugar, se encuentran los sistemas de *diseño agrupado o ensemble design.* En este caso se utilizan distintos métodos de recomendación estándar y se combinan sus resultados para poder proporcionar una predicción más robusta. Existe una gran diversidad y variación en las metodologías que se utilizan para realizar la combinación de recomendaciones.

En los sistemas de diseño agrupado, $\hat{R}_h$ es una matriz $n \times m$ que contiene las valoraciones de los n usuarios para los m ítems a partir del h-ésimo algoritmo donde $h \in \{1, ..., q\}$. Así, se pueden usar q algoritmos para computar las predicciones. Es preciso mencionar que las valoraciones observadas de cada uno de los usuarios se mantienen en cada una de las matrices $\hat{R}_h$, variando únicamente los valores predichos. Así, el resultado final se obtiene combinando las predicciones $\hat{R}_1, \hat{R}_2, ..., \hat{R}_q$ en una salida única. Una de las formas de realizar esta combinación es a partir de la media ponderada de las predicciones. Estos sistemas se caracterizan por usar sistemas de recomendación estándar ya existentes y por proporcionar una valoración única.

Dichos sistemas se dividen en: sistemas secuenciales y sistemas de diseño paralelos, junto con sus respectivas subdivisiones. (Aggarwal, 2016, pp: 199-224)

Por otro lado, se encuentran los sistemas de *diseño monolítico o monolithic design.* En este caso, a diferencia del diseño agrupado, se crea un sistema de recomendación integrado utilizando distintos tipos de datos. En muchos casos los algoritmos de filtrado colaborativo o aquellos basados en contenido tendrán que ser modificados para poder utilizarse. Como este tipo de diseño integra distintos tipos de datos, los componentes individuales no pueden verse fácilmente.(Dietmar et al., 2011, pp: 129-134)

Por último, están los sistemas de *diseño mixto*. De forma similar a los sistemas de diseño agrupado, se utilizan distintos métodos de recomendación como *cajas negras*, con la diferencia de que los ítems que recomienda cada uno de los métodos se presentan juntos. De esta forma, ver las recomendaciones individuales de ítems no tiene mucho sentido, pues es el conjunto entero lo que hace la recomendación.(Aggarwal, 2016, pp:220-224)

2.2.4.1 Sistemas de diseño agrupado

Como se ha mencionado en la sección anterior, dentro de los sistemas de diseño agrupado podemos encontrar aquellos de diseño secuencial y de diseño paralelo.

- **Sistemas de diseño secuencial:** los sistemas híbridos de diseño secuencial implementan un proceso por partes en el que distintas técnicas se construyen secuencialmente las unas sobre las otras para obtener las recomendaciones. Los subsistemas derivados de aquellos de diseño secuencial se diferencian los unos de los otros principalmente en la salida que otorgan de una etapa a otra; esto es, por ejemplo, si la salida de una etapa es un modelo o una lista de recomendaciones a modificar. Dentro de dichos subsistemas se encuentran los *"cascade hybrids" y los "meta-level hybrids"*. A continuación, se explicarán únicamente los "híbridos cascada" o *cascade hybrids* pues su procedimiento es más intuitivo:

  Los "híbridos cascada" son sistemas de diseño secuencial en los que la salida de un paso a otro es en forma de una lista de recomendaciones que ha de ser depurada. Formalmente, se tiene una lista de q técnicas donde rec_1 representa la función de recomendación de la primera técnica y rec_q la de la última. Así, la valoración final de cada producto la calcula la q-ésima técnica. Sin embargo, a la h-ésima técnica solo se le sugiere un producto si la técnica anterior le ha dado una valoración positiva. Por inducción, esto se puede aplicar a todas las técnicas salvo a la primera (Dietmar et al., 2011, pp: 138-139):

  $$rec_{\text{cascada}}(a, i) = rec_q(a, i),$$

  donde $\forall h \geq 2$ se debe cumplir:

  $$rec_h(a, i) = \begin{cases} rec_h(a, i), & \text{si } rec_{h-1}(a, i) \neq 0 \\ 0, & \text{en otro caso} \end{cases}$$

De esta forma, en los modelos secuenciales todos las técnicas excepto la primera sólo pueden cambiar los productos que ha determinado su predecesor, sin poder introducir nuevos productos en las recomendaciones. Por esta razón, uno de los principales problemas de los "híbridos cascada" es que reducen potencialmente las recomendaciones, resultando en sistemas que no devuelven el número mínimo de recomendaciones requerido.

- **Sistemas de diseño paralelo:** estos sistemas emplean diferentes métodos de recomendación lado a lado, es decir, de forma paralela. Dentro de los subsistemas de diseño paralelo se encuentran los sistemas ponderados (weighted) y los sistemas "switching". Algunos autores como Burke, 2000 incluyen a los sistemas de diseño mixto como subsistema de los de diseño paralelo; sin embargo, en este trabajo se seguirá el enfoque de Charu C. Aggarwal, 2016 (pp:199-224) y se explicarán por separado. Dada su sencillez, en esta sección se explicarán únicamente los "switching hybrids".

  En dichos sistemas es preciso determinar un "oráculo" o herramienta que permita decidir que sistema de recomendación es necesario o adecuado en cada situación, dependiendo de los datos disponibles en ese momento, de la calidad de la predicción... Uno ejemplo podría ser:

$$\exists_h : 1...n \text{ rec }_{\text{switching}}(a, i) = rec_h(a, i).$$

  Donde h es determinado por la condición de cambio (*switching condition*). Un ejemplo simple podría ser establecer una condición en la que si inicialmente hay una matriz de valoraciones dispersa, se utilice un sistema basado en conocimiento y cuando ya haya suficientes datos como para poder evitar el inicio en frío, se utilice un sistema de filtrado colaborativo o uno basado en contenido. El "oráculo" que establece las condiciones de cambio puede ser muy variado, mientras que en Billsus et al., 2000 se propone un sistema que combine dos métodos basados en contenido, en Zanker M. et al., 2009 proponen un sistema similar al ejemplificado anteriormente que combine un método de filtrado colaborativo con algún método basado en conocimiento.

2.2.4.1 Sistemas de diseño monolítico

Mientras los sistemas de diseño mixto y los de diseño agrupado se basan en combinar los resultados de varios métodos de recomendación, los sistemas de diseño monolítico optan por construir un único sistema que combine múltiples enfoques proporcionados por diversos modelos a través del preprocesado y la combinación de varias fuentes de conocimiento. Generalmente, es necesario modificar los algoritmos implementados y realizar transformaciones específicas dadas por los datos para que puedan ser utilizados por el algoritmo.

Las subdivisiones de los sistemas de diseño monolítico comúnmente incluyen a los métodos de combinación de características (*feature combination*) y los métodos de aumento de características (*feature augmentation*). Sin embargo, dependiendo del autor los métodos de aumento de características podrían ser considerados como diseño secuencial y los métodos meta-nivel mencionados anteriormente como de diseño monolítico; ambas interpretaciones son válidas. En esta sección se hará un bosquejo los métodos de combinación de características pues los métodos de aumento de características utilizan transformaciones complejas de los datos. Puede consultarse literatura existente en Melville et al., 2002; Burke R., 2007 y Dietmar et al., 2011, pp: 129-134

Los modelos de combinación de características utilizan una gran variedad de datos de entrada y, dependiendo de los sistemas que se quieran combinar, éstos pueden ser muy diferentes entre sí. De esta forma, uno de las técnicas propuestas en Basu et al., 1998 combina aspectos de los métodos de filtrado colaborativo con los métodos basados en contenido y lo ejemplifica bajo un recomendador de libros.

En dicho método, aparte de considerar las valoraciones históricas de los usuarios (filtrado colaborativo) y los géneros de los libros (atributos de los modelos basados en contenido) considera características

híbridas basadas tanto en los usuarios como en los atributos de los productos. Basu et al., 1998 calificaron dichas características híbridas como "el esfuerzo común de la ingeniería humana que implica inventar buenas características que permitan un aprendizaje satisfactorio". En su ejemplo, considera que si dos tercios de las compras totales de un usuario son sobre libros del mismo género, se incluye la variable de que al usuario a le gustan muchos libros del género X en la matriz de usuarios/ítems. Posteriormente, a través de transformaciones en dicha matriz, se determina la similitud entre usuarios utilizando los conjuntos de características valoradas y la regla inductiva de aprendizaje *Ripper* (Cohen W., 2000).

Otra posibilidad en los modelos de combinación de características sería implementar un modelo de filtrado colaborativo utilizando diferentes tipos de *feedback* de valoraciones basados en su disponibilidad y precisión predictiva, como se propueso en Zanker M. et al., 2009

2.2.4.1 Sistemas de diseño mixto

Por último, los sistemas de diseño mixto combinan las recomendaciones realizadas por varios métodos de recomendación en la presentación de las recomendaciones para obtener valoraciones "conjuntas" por cada una de las técnicas. Por esta razón, los sistemas de diseño mixto podrían ser considerados como sistemas de diseño paralelo y sus recomendaciones tienen más sentido cuando se evalúan de forma conjunta que cuando se evalúan de forma individual. De esta forma, las recomendaciones para el usuario a y el ítem j en un sistema de diseño mixto es el conjunto de tuplas (score,h) para cada uno de sus q que conforman los recomendadores rec_h:

$$rec_{mixto}(a, j) = \bigcup_{h=1}^{q} (rec_h(a, j), h).$$

Posteriormente, los ítems con las top-k puntuaciones se muestran al usuario. Dependiendo del ámbito en el que se aplique el sistema de diseño mixto puede haber dificultades a la hora de mostrar los resultados de los recomendadores de forma unificada (por ejemplo en Cotter et al., 2000, se proponen el caso de la programación de la televisión). En dichos casos, para devolver las recomendaciones será necesario aplicar alguna forma o técnica de resolución de conflictos (Dietmar et al., 2011).

Una vez indicados los modelos de recomendación, las diferentes formulaciones del problema y los objetivos del sistema y, tras comentar detalladamente los 4 tipos de modelos de recomendación, es posible concluir que no hay un único sistema que responda a todas las necesidades de recomendación existentes.

Ya en 1997 Macready W. y Wolpert D. establecieron con la publicación de los teoremas *No Free Lunch* que, en el campo de la optimización y de aprendizaje supervisado, no existe un algoritmo óptimo universal para todos los problemas. Dicha idea puede ser extendida al campo de los sistemas de recomendación; así, dependiendo del tipo de datos disponible, de los objetivos del sistema de recomendación o de cómo se formule el problema, será más adecuado un modelo u otro. Incluso teniendo en cuenta la existencia de modelos híbridos que combinan las ventajas de múltiples sistemas, diferentes situaciones requerirán diferentes métodos.

Capítulo 3

Construcción de los modelos

El objetivo de este capítulo es, a partir de los datos facilitados por la compañía Hijos de Rivera, desarrollar un sistema de recomendación aplicando las técnicas adecuadas. En primer lugar se describirá el modelo inicial que se planteó al comienzo del proyecto y los problemas que éste acarreaba. A posteriori se explicará teóricamente la solución aplicada a dicho modelo y, finalmente, se construirán y compararán ambos modelos.

3.1. Primeras ideas

Con carácter previo a la selección del modelo a desarrollar es necesario reconocer los datos disponibles. Recordando lo mencionado en la Sección 1.3, para la construcción del sistema de recomendación se han usado los datos correspondientes al ejercicio 2022. Dichos datos están dispuestos en formato albarán, de forma que cada entrada de la base de datos representa una compra realizada por un local determinado. En dicha entrada, se pueden identificar las variables `Articulo ID`, `Marca ID`, `Marca desc`, `Grupo material desc`, `Agrupación envase ID`, `Ventas EUR`, `Ventas LTS`, `Ventas UDS INV`, `Local ID`, `Tipo de instalacion ID`, `Exclusiva establecimiento ID`, `Acuerdo promo ID` y `Provincia ID`.

Debido a limitaciones computacionales del uso de un ordenador personal, no será posible desarrollar el sistema utilizando todas las observaciones disponibles. Así, el modelo se desarrollará utilizando una muestra de 70.000 datos extraídos aleatoriamente del conjunto total de observaciones.

El primer paso para realizar el sistema de recomendación es identificar qué tipo de modelo se ajusta más a los objetivos del sistema y los datos disponibles. En este caso, el objetivo es desarrollar un sistema de recomendación para mayoristas; de forma que cuando se vaya a contabilizar el pedido de un local determinado, aparezcan recomendaciones de productos distintos de los que compra habitualmente y que le puedan interesar. Por otro lado, los datos disponibles pueden dividirse en tres grupos: historial de compras, características de clientes y características de productos.

Con ambos aspectos en mente, el planteamiento inicial podría incluir la construcción de un sistema basado en contenido o la de un sistema basado en conocimiento, pues ambos permitirían tomar ventaja de los atributos de los productos. Sin embargo, sólo se disponen de dos atributos de cada producto (el tipo de producto y el número de unidades vendidas en cada paquete) y en algunos casos dichos campos no están cubiertos. Además, la interfaz de los sistemas basados en conocimiento comprometería la rapidez del proceso de recomendación, pues se necesitan demasiadas interacciones para llegar a un conjunto de recomendaciones adecuado. Por estas razones se ha optado por utilizar una técnica diferente, construyendo un modelo de filtrado colaborativo.

Como se ha mencionado en la Sección 2.2.1, los modelos de filtrado colaborativo pueden estar basados en usuarios o basados en ítems. En este caso, la escasez de variables que indiquen atributos de producto complica la obtención de similitudes entre ellos, por lo que se realizará un modelo basado en usuarios. A continuación, se muestran los pasos seguidos para la creación del modelo inicial.

El primer aspecto a resolver para la construcción del modelo es la obtención de la matriz de valoraciones R. El propósito de está matriz es identificar cuáles son los productos en los que los usuarios están interesados. Generalmente los recomendadores de libros o películas ya cuentan con valoraciones realizadas por los usuarios una vez han leído o visto el producto correspondiente. A pesar de que en Hijos de Rivera no se tienen unas valoraciones explícitas realizadas por cada uno de los consumidores, sí se dispone de sus datos de compra. Así, para determinar qué productos interesan a cada individuo se utilizará su histórico de compras, de forma que cuantos más litros de un producto haya comprado más interesado estará en él.

Utilizando las variables `Local ID`, `Ventas LTS` y `Marca desc`, se ha creado una matriz de valoraciones R de dimensión $n \times m$ que recoge las ventas que ha realizado cada local de dicho producto; en ella cada fila representa un identificador de la variable `Local ID` y cada columna una marca de producto. En este caso se dispone de información acerca de $n = 39000$ locales y $m = 74$ productos. Como el volumen de compra de los distintos establecimientos varía dependiendo del tamaño de los mismos, para reducir la influencia de las "valoraciones" individuales se han escalado los datos restando la media de cada individuo.

Una vez se dispone de la matriz de *ratings* R, se seleccionan usuarios similares. Para ello se ha utilizado como función de similitud la correlación de Pearson. A modo ilustrativo, en la Figura 3.1 se muestran gráficamente las correlaciones entre 14 usuarios seleccionados al azar. En ella arcos de color azul indican correlación positiva y arcos de color rojo indican correlación negativa.

Es precisamente en el cálculo de correlaciones donde comienzan los problemas pues la matriz de datos es muy dispersa y no se pueden computar todas las correlaciones. Adicionalmente, en la Sección 1.3 ya se mencionaba que varios usuarios sólo compran 1 o 2 productos. Por ello, al calcular las correlaciones, un grupo elevado de usuarios tendrá como vecinos más cercanos (usuarios con los que tiene mayor correlación) establecimientos que compran exactamente los mismos productos. En términos estrictos, el sistema sigue funcionando, pues se usan dichos usuarios para predecir la "valoración" que ese establecimiento hará de un producto. Es decir, si el establecimiento a compra 350 lts. de cerveza *EG Especial* todos los meses, el sistema utiliza los vecinos más cercanos para poder predecir cuantos litros de esa cerveza volverá a comprar el usuario, pero no recomendará ningún producto adicional.

Ambos problemas derivados de la escasez de datos pueden ser solucionados utilizando, entre otros, modelos de factores latentes. Concretamente, en este trabajo se ha utilizado la técnica conocida como *descomposición en valores singulares* (SVD) vista en la Sección 2.2.1.4 .

3.2. Construcción de los modelos

Para la construcción de los modelos de recomendación se hará uso del paquete `recommenderlab` (Hahsler M., 2023) que provee de la infraestructura necesaria para desarrollar y evaluar algoritmos de recomendación de filtrado colaborativo. El primer algoritmo a desarrollar será el algoritmo de filtrado colaborativo sin descomposición matricial. Concretamente, se desarrollará el modelo que utiliza vecindarios basados en usuarios introducido en la Sección 2.2.1.1. A pesar de que construyendo el modelo a mano se ha visto que las recomendaciones no son adecuadas, es posible calcular y comparar los errores con el modelo creado con descomposición de valores singulares a fin de ver las diferencias. Para el

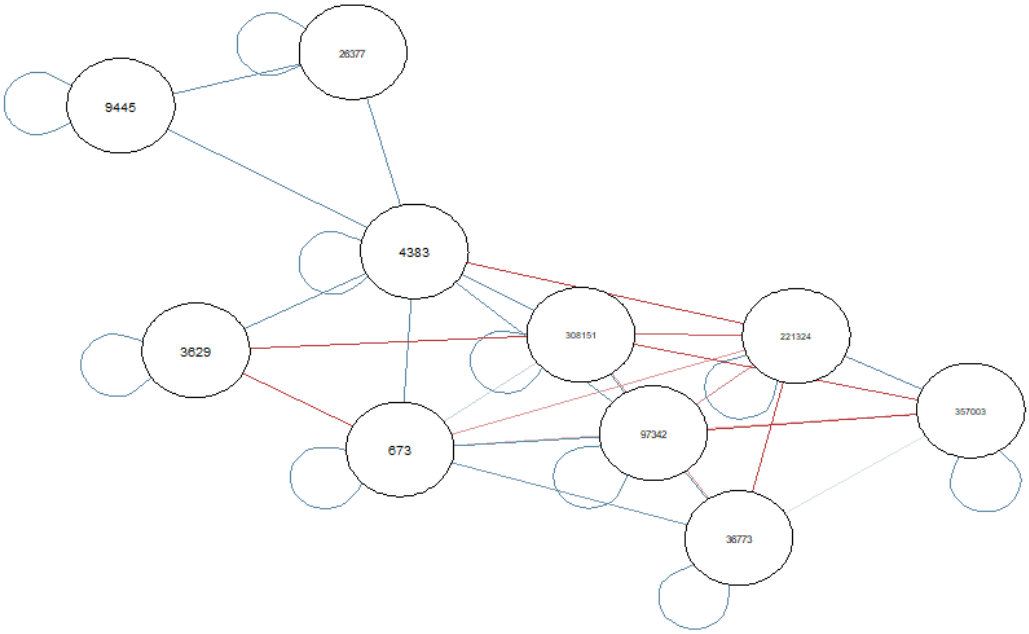

Figura 3.1: Correlaciones entre usuarios, en azul las correlaciones positivas y en rojo las negativas.

modelo de filtrado colaborativo se hará un modelo de k-vecinos más cercanos, comprobando valores de k de 10, 15 y 20; pues valores más altos reducen la calidad de las predicciones (Borschers et al., 1999). Posteriormente se construirá el modelo de filtrado colaborativo usando la técnica de descomposición en valores singulares (SVD).

Los modelos se evaluarán en dos dimensiones distintas, el alcance del modelo y su precisión estadística. Para evaluar el alcance se calculará la fracción de ítems recomendados respecto a los ítems totales. En la Sección 1.3 se indicó que en Hijos de Rivera hay muchos productos con baja frecuencia de ventas, por ello, la recomendación de más ítems distintos se valorará positivamente. Además, se tendrá en cuenta que los ítems más recomendados no pertenezcan a los productos (A, B, C, D y E) indicados en dicha sección.

En relación a la precisión estadística, se hará uso del Error Absoluto Medio (*Mean Absolute Error* o MAE), de la Raíz del Error Cuadrático Medio (*Root Mean Square Error* o RMSE) y del Error Cuadrático Medio (*Mean Square Error* o MSE) (Herlocker et al., 2004 y Gunawardana et al., 2009). Para evaluar la precisión estadística se utilizará validación cruzada en η grupos. En ella se dividirá la muestra total en η grupos del mismo tamaño (aproximadamente). Para cada uno de esos grupos se calcularán las medidas de error, usando el resto de grupos como muestra de entrenamiento para construir el modelo de recomendación. Finalmente, se promedian los valores obtenidos de cada una de las métricas (Tabla 3.1).

MÉTRICA	UCBF-10	UCBF-15	UCBF-20	SVD-CF
MAE	1,079	1,087	1,003	0,606
MCE	4,414	4,642	3,911	2,081
RMSE	2,042	2,003	1,921	1,420
Alcance	$\frac{52}{74}$	$\frac{62}{74}$	$\frac{63}{74}$	$\frac{74}{74}$
Recomendaciones totales	41 355	50 795	56 202	138 483

Tabla 3.1: Tabla comparativa de los errores y la cobertura del modelo de filtrado colaborativo sin utilizar SVD (UBCF-10, UBCF-15 y UBCF-20) y utilizando SVD (SVD-CF)

La Tabla 3.1 muestra los resultados obtenidos tanto en términos de precisión estadística como de cobertura de los modelos de filtrado colaborativo desarrollados. Respecto a las métricas de precisión estadística, se observa como el modelo de filtrado colaborativo utilizando descomposición de valores singulares presenta errores menores en todas las métricas. A priori cabría esperar que los modelos de filtrado colaborativo sin utilizar descomposición matricial tuvieran una mayor precisión pues, como se ha mencionado en el capítulo anterior, para aquellos usuarios que consumen únicamente un producto los sistemas sólo recomiendan el ítem que ya habían consumido; eliminando la posibilidad de que haya error en los *ratings* predichos de otros productos. Sin embargo, dichos modelos de filtrado colaborativo presentan valores del Error Absoluto Medio y del Error Cuadrático Medio cercanos al doble de aquellos obtenidos con el sistema de filtrado colaborativo utilizando SVD.

Analizando los resultados en términos de cobertura o alcance y las recomendaciones totales realizadas por el sistema, el modelo de filtrado colaborativo utilizando SVD vuelve a tener mejores resultados que los sistema de filtrado colaborativo sin SVD. A diferencia de como pasaba con las métricas de precisión estadística, en este caso los resultados eran esperables; pues desde la construcción de los modelos ya se sabía que los modelos sin utilizar SVD tenían problemas para recomendar productos a aquellos establecimientos (usuarios) que hubieran comprado únicamente un ítem de los 74 disponibles. Asimismo, pese a que dichos sistema han conseguido recomendar 52, 62 y 63 de los 74 productos disponibles, la cantidad total de recomendaciones ha sido muy inferior a las recomendaciones totales realizadas por el sistema de filtrado colaborativo utilizando SVD (41 355, 50 795 y 56 202 frente a las 138 483 realizadas por el sistema con SVD).

Teniendo en cuenta ambas dimensiones de forma conjunta, se puede concluir que el sistema de filtrado colaborativo utilizando SVD ha solucionado los problemas derivados de la escasez de datos que presentaban los sistemas de filtrado colaborativo iniciales. A su vez, se puede observar que dicho sistema cumple los objetivos iniciales del trabajo, puesto que produce recomendaciones para todos los establecimientos y recomienda productos diferentes a los 5 productos más populares mostrados en la Sección 1.3; cumpliendo los requisitos de primicia y causalidad. Aún así, es preciso destacar que el problema de inicio en frío sigue presente, pues no se llegarían a recomendar nuevos ítems que no hayan sido comprados. Además, es adecuado mencionar que el modelo implementado no es manifiestamente novedoso, abundando las fuentes relativas a él.

3.3. Modelo adicional

En la Sección 2.2.2.3 se ha mencionado que uno de los posibles problemas que podían tener los métodos basados en vecindarios (ya sean modelos de filtrado colaborativo o modelos basados en contenido) es que para un número de usuarios o de productos demasiado grande (cientos de millones) los tiempos de computación pueden ser demasiado elevados, dado que es necesario calcular todos los vecinos posibles del a-ésimo usuario o del j-ésimo producto. Una de las formas de solucionar este problema es emplear los métodos basados en clusters mencionados en la Sección 2.2.1.4.

La idea subyacente en dichos métodos es sencilla, en primer lugar se aplica una técnica de agrupación, como puede ser k-medias, y posteriormente se aplica un método de recomendación; con la diferencia de que en lugar de utilizar todos los usuarios de la base de datos para recomendar productos al usuario a-ésimo, se utilizan únicamente aquellos usuarios que pertenecen al mismo cluster que dicho usuario. A cambio de un procedimiento, a priori, más rápido, se sacrifica precisión de las predicciones pues, los top-k vecinos del clúster son de menor calidad que los top-k vecinos de la base de datos completa (Dietmar et al., 2011).

En este aspecto, uno de los modelos considerado inicialmente ha sido un sistema de recomendación a partir de una técnica de clústering basada en la estimación no paramétrica de regiones de elevada densidad. Generalmente, las técnicas de agrupación basadas en densidad se apoyan en la idea de asociar grupos a los componentes conexos de conjuntos de niveles de densidad subyacentes a los datos, que serán estimados por métodos no paramétricos. La técnica utilizada, propuesta por Azzalini y Menardi (2014) utiliza un procedimiento alternativo, midiendo la extensión de posibles valles de la densidad a lo largo del segmento que conecta pares de observaciones, para desplar la formulación a un espacio de dimensión univariante. Uno de los objetivos de dicho método es reducir la complejidad y el tiempo de computación de los métodos de agrupación basados en densidad.

A primera vista pudiera parecer que el método no es adecuado a los datos, pues estimar la densidad requiere de variables continuas y en este caso se dispone, mayormente, de variables factoriales. Sin embargo, para solucionar dicho problema, en Azzalini et al., 2016, se propone un enfoque basado en la identificación de los componentes continuos subyacentes a las variables no continuas.

Para realizar la clasificación de establecimientos se han utilizado las siguientes variables: `Ventas EUR`, `Ventas LTS`, `Ventas UDS INV`, `Local ID`, `Tipo de instalacion ID`, `Exclusiva establecimiento ID`, `Acuerdo promo ID` y `Provincia ID`. Adicionalmente, como ya se ha indicado en la Sección 1.3, haciendo uso de la variable `Marca desc` se ha creado una variable por cada uno de los 74 productos disponibles.

Pese a que uno de los objetivos de Azzalini y Menardi era reducir los tiempos de computación, realizar la agrupación con alrededor de 80 variables sigue requiriendo un esfuerzo computacional considerable. Esto, junto a las limitaciones derivadas del uso de un ordenador personal, han provocado que el método fuera descartado.

Capítulo 4

Conclusiones

El objetivo de la memoria de prácticas ha sido el estudio y desarrollo de sistemas de filtrado de información, concretamente de los sistemas de recomendación. En el Capítulo 2 se han revisado algunos de los sistemas de recomendación existentes en la literatura. Posteriormente, en el Capítulo 3 se ha chequeado el comportamiento práctico de un sistema de filtrado colaborativo basado en vecindarios y de dos sistemas de filtrado colaborativo basados en modelos, empleando la base de datos proporcionada por Estrella Galicia. En el primero de de los sistemas basados en modelos se utiliza la Descomposición en Valores Singulares (SVD) mientras que en el segundo se utiliza un método basado en clusters. Sin embargo, debido a su mal funcionamiento, el sistema basado en clusters ha sido descartado.

Es preciso destacar que otros modelos como el filtrado colaborativo utilizando datos binarios han sido considerados. Sin embargo, éstos han sido descartados y no han sido incluidos en la memoria porque su funcionamiento no es satisfactorio.

De todos los métodos desarrollados, el modelo de filtrado colaborativo utilizando SVD ha sido el único que presenta resultados aceptables. Es decir, el modelo genera adecuadamente las recomendaciones, cumple los requisitos de primicia y causalidad y se adecúa a los datos facilitados por Hijos de Rivera.

Sin embargo, a pesar de que los resultados obtenidos con el método de filtrado colaborativo utilizando descomposición matricial han sido satisfactorios, pueden considerarse distintas extensiones del problema para tener en cuenta en futuras líneas de trabajo y mejorar los resultados obtenidos.

En primer lugar, sería deseable poder desarrollar un sistema de recomendación híbrido, utilizando las características de los productos. Para ello sería necesario establecer nuevas variables que permitiesen identificar distintos atributos de los productos, puesto que las variables disponibles no son suficientes. En este aspecto sería adecuado considerar el modelo *LightFM* propuesto inicialmente por Hijos de Rivera. *LightFM* es un modelo de filtrado colaborativo híbrido de factorización matricial disponible en *python* y que combina características de usuarios con características de productos en forma de *embeddings* para combatir el problema de inicio en frío.

Por otro lado, dentro de los sistemas híbridos que sería interesante tener en consideración es importante destacar aquellos que combinan el filtrado colaborativo con la recomendación basada en conocimiento. A pesar de que un sistema basado en conocimiento puro no sería adecuado para solucionar el problema planteado por Hijos de Rivera, algunos de ellos permiten incorporar datos útiles como la estacionalidad de los productos.

Adicionalmente, en relación a los métodos de clasificación basados en clusters, sería interesante

volver a probar el sistema considerando otras técnicas distintas; *k-medias*, por ejemplo. Sin embargo, a la hora de contemplar los métodos basados en clusters es necesario tener en cuenta que la mayoría de variables disponibles son factoriales, hecho que puede dificultar la agrupación de las observaciones.

Asimismo, sería adecuado sopesar distintos aspectos en el planteamiento del problema que pueden afectar a los sistemas desarrollados. Algunos de los aspectos a considerar podrían ser la división de establecimientos atendiendo al canal de alimentación o agrupaciones distintas de productos.

Por último, reiterar que para el entrenamiento y evaluación del modelo han sido utilizados únicamente los datos correspondientes al ejercicio 2022. Dichos datos, a pesar de que a lo largo del trabajo se menciona que el sector está prácticamente recuperado, siguen estando débilmente influenciados por la COVID-19. Para mitigar este efecto sería interesante realizar un estudio detallado de los períodos de tiempo afectados.

Bibliografía

[1] Aggarwal C.C. (2014) Data classification: algorithms and applications. CRC Press.

[2] Aggarwal C.C. (2015) Data mining: the textbook. Springer, New York, 2015.

[3] Aggarwal C.C. (2016) Recommender Systems. New York, NY, USA: Springer

[4] Aggarwal P., Tomar V., Kathuria A. (2017) Comparing Content Based and Collaborative Filtering in Recommender Systems. En: International Journal of New Technology and Research (IJNTR). pp: 65-67.

[5] Azzalini A., Menardi G. (2014) An advancement in clustering via nonparametric density estimation. En: Statistics and Computing , Volume 24, Issue 5. pp: 753-767.

[6] Azzalini A., Menardi G. (2016) Density-based clustering with non-continuous data. En: Computational Statistics, Volume 31. pp: 771-798.

[7] Banirostam T., Javad M. S., Vahidy R. K. (2021) Resolving cold start and sparse data challenge in recommender systems using multi-level singular value decomposition. En: Computers & Electrical Engineering Vol 94.

[8] Basu C., Hirsh H., Cohen W. (1998) Recommendation as Classification: Using Social and Content-Based Information in Recommendation.

[9] Bellogín A., Cantador I., Díez F., Castells P., Chavarriaga E. (2013) An empirical comparison of social, collaborative filtering, and hybrid recommenders.En: ACM Transactions on Intelligent Systems and Technology Vol 4 Issue 1 Article No.: 14. pp: 1-29.

[10] Billsus D., J. Pazzani M. (2000) User Modeling for Adaptive News Access. En: User Modeling and User-Adapted Interaction 10, pp: 147-180.

[11] Borchers A., Herlocker J., Konstan J., Riedl J. (1999) An algorithmic framework for performing collaborative filtering. En: in Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, Berkeley, CA, USA. pp. 230-237.

[12] Breese S. J., Heckerman D., Kadie C. (2013) Empirical Analysis of Predictive Algorithms for Collaborative Filtering. En: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence.

[13] Brusilovsky P., Kobsa A., Nejdl W. (2007) The Adaptive Web - Methods and Strategies of Web Personalization.

[14] Burke R. (2000) Knowledge-based recommender systems Robin Burke. En: Encyclop. Lib. Inform. Syst. 69. pp: 180-200

[15] Burke R. (2002) Hybrid Recommender Systems: Survey and Experiments. En: User Modeling and User-Adapted Interaction. pp:331-370.

[16] Burke R., Christopher Y. B., Hammond K. (1997) The FindMe approach to assisted browsing. En:Institute of Electrical and Electronics Engineers Vol 12(4). pp: 32-40.

[17] Burke R. (2007) Hybrid Web Recommender Systems. En: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds) The Adaptive Web. Lecture Notes in Computer Science, vol 4321. pp: 377-408.

[18] Cerveceros de España (2023) Informe socioeconómico del sector de la cerveza en España 2022. Ministerio de Agricultura, Pesca y Alimentación.

[19] Cohen W. (2000) Learning Rules that Classify E-Mail. AT&T Laboratories, New Jersey, USA.

[20] Corporación Hijos de Rivera. Historia de Hijos de Rivera, fundadores y dueños de Estrella Galicia. Accedido el 11 de diciembre de 2023.

[21] Cotter P., Smyth B. (2000) PTV: Intelligent Personalised TV Guides. pp: 957-964.

[22] Deepjyoti R., Mala D. (2022) A systematic review and research perspective on recommender systems. En: Journal of Big Data 9-59.

[23] De Myttenaere A., Golden B., Le Grand B., Fabrice R. (2016) Mean Absolute Percentage Error for regression models. En: Neurocomputing, Elsevier, 2016, Advances in artificial neural networks, machine learning and computational intelligence - Selected papers from the 23rd European Symposium on Artificial Neural Networks (ESANN 2015), 192. pp: 38-48

[24] Deerwester S., Dumais S.T., Furnas G.W., Landauer T.K., Harshman R. (1990) Indexing by latent semantic analysis. En: Journal of the American Society for Information Science Vol 41 Issue 6. pp: 391-407

[25] Dietmar J., Zanker M., Felfernig A., Friedrich G. (2011) Recommender Systems An Introduction. Cambridge university press.

[26] Ekstrand D.M., Riedl J., Konstan J. (2011) Collaborative Filtering Recommender Systems. University of Minnesota, USA.

[27] Felfernig A., Burke R. (2008) Constraint-based recommender systems: technologies and research issues. In Proceedings of the 10th international conference on electronic commerce (3).

[28] Felfernig A., Friedrich G., Jannach D., Stumptner M. (2004) Consistency-based diagnosis of configuration knowledge bases. En: Artificial Intelligence Vol 152(2). pp: 213-234.

[29] Folajimi Y.O., Isinkaye F.O., Ojokoh B.A. (2015) review Recommendation systems: Principles, methods and evaluation. En: Egyptian Informatics Journal (2015) 16. pp: 261-273

[30] García Ropero J. (2023) Estrella Galicia reduce un 11,5 % su beneficio anual pese a lograr unos ingresos récord de 724 millones. Accedido el 11 de diciembre de 2023

[31] Good N., Schafer B., Konstan J., Borchers A., Sarwar B., Herlocker J., Riedl J. (1999) Combining Collaborative Filtering with Personal Agents for Better Recommendations. En: Conference: Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence. pp: 439-446.

[32] Gunawardana A., Shani G. (2009) A Survey of Accuracy Evaluation Metrics of Recommendation Tasks. En: The Journal of Machine Learning Research, Volume 10. pp 2935-2962

[33] Hahsler M. (2023) recommenderlab: Lab for Developing and Testing Recommender Algorithms. R package version 1.0.6, https://CRAN.R-project.org/package=recommenderlab. Accedido el 26 de septiembre de 2023.

[34] Herlocker L.J., Konstan A.J., Terveen G.L., Riedl J. (2004) Evaluating collaborative filtering recommender systems. En: ACM Transactions on Information Systems Vol 22 Issue 1. pp: 5-53.

[35] Herlocker L.J., McLaughlin M. R. (2004) A collaborative filtering algorithm and evaluation metric that accurately model the user experience. En: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '04).pp: 329-336.

[36] Huang Z., Chen H., Zeng D. (2004) Applying Associative Retrieval Techniques to Alleviate the Sparsity Problem in Collaborative Filtering. En: ACM Transactions on Information Systems, Vol. 22, No. 1, January 2004. pp: 116-142.

[37] Hyndman R., Koehler A. (2006) Another look at measures of forecast accuracy. En:International Journal of Forecasting, Volume 22, Issue 4. pp: 679-688.

[38] Jannach D. (2006) Finding Preferred Query Relaxations in Content-Based Recommenders. En: Intelligent Techniques and Tools for Novel System Architectures. pp:81-97.

[39] Kumar B., Sharma N. (2016) Approaches, Issues and Challenges in Recommender Systems: A Systematic Review. En: Indian Journal of Science and Technology, Vol 9(47).

[40] Lü L., Medo M., Ho Yeung C., Zhang Y., Zhang ZK., Zhou T. (2012) Recommender systems. En: Physics Reports 519. pp: 1-49.

[41] Macready W., Wolpert D. (1997) No free lunch theorems for optimization. En: IEEE Transactions on Evolutionary Computation, vol. 1, no. 1. pp. 67-82.

[42] Madaladipouya K., Chelliah S. (2017) A Literature Review on Recommender Systems Algorithms, Techniques and Evaluations. En:BRAIN: Broad Research in Artificial Intelligence and Neuroscience Volume 8, Issue 2.

[43] Melville P., Mooney J.R., Nagarajan R. (2002) Content-Boosted Collaborative Filtering for Improved Recommendations. En: Proceedings of the Eighteenth National Conference on Artificial Intelligence(AAAI-2002). pp: 187-192.

[44] Miyahara K., Pazzani M. (2002) Improvement of Collaborative Filtering with the Simple Bayesian Classifier. En: Journal of Information Processing (JIP) vol 43 No. 11.

[45] Nandan M.S., Chatterjee M.J., Jain S., Ahmed A.E., Gupta P. (2020) Recommender System with Machine Learning and Artificial Intelligence; Practical Tools and Applications in Medical, Agricultural and Other Industries, Scrivener Publishing, Beverly, USA.

[46] O'Donovan J., Dunnion J. (2002) A Comparison of Collaborative Recommendation Algorithms Over Diverse Data. En: The 14th Irish Conference on Artificial Intelligence & Cognitive Science. pp: 158-164.

[47] O'Mahony M., Hurley N., Silvestre G. (2003) An Evaluation of the Performance of Collaborative Filtering. En: Conference: Proceedings of the 14th Irish International Conference on Artificial Intelligence and Cognitive Science. pp: 164-168

[48] Orus A. (2022) La industria de la cerveza en España - Datos estadísticos. https://es.statista.com/temas/5410/la-industria-de-la-cerveza-en-espana/. Accedido el 11 de diciembre de 2023

[49] Poonam B.T., Goudar. R.M, Sunita S.B. (2015) Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System. En: International Journal of Computer Applications 110.

[50] Ricci F., Rokach L., Shapira B. (2022) Recommender Systems: Techniques, Applications, and Challenges. En: Ricci, F., Rokach, L., Shapira, B. (eds) Recommender Systems Handbook. Springer, New York, NY. pp:1-35

[51] Rodler P. (2022) Understanding the QuickXPlain Algorithm: Simple Explanation and Formal Proof. En: Artif Intell Rev 55. pp: 6185-6206.

[52] Salton G., Wong A., Yang C.S. (1975) A vector space model for automatic indexing. En: Communications of the ACM Vol 18 Issue 11. pp: 613-620. https://doi.org/10.1145/361219.361220

[53] Sarwar B., Karypis G., Konstan J., Riedl J. (2000) Application of Dimensionality Reduction in Recommender System - A Case Study. Obtenido de the University of Minnesota Digital Conservancy.

[54] Schein A., Popescul A., Ungar L., Pennock D. (2002) Methods and metrics for cold-start recommendations. En: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '02). Association for Computing Machinery, Nueva York, USA. pp: 253-260.

[55] Smyth B., McClave P. (2001) Similarity vs. Diversity. En: Proceedings of the 4th International Conference on Case-Based Reasoning: Case-Based Reasoning Research and Development. pp: 347-361

[56] Valdiviezo-Diaz P., Ortega F., Cobos E., Lara-Cabrera R. (2019) A Collaborative Filtering Approach Based on Naïve Bayes Classifier. En: IEEE Access, vol. 7. pp: 108581-108592.

[57] Van Wieringen W. (2023) Lecture notes on ridge regression.

[58] Wickham H., Bryan J. (2023) readxl: Read Excel Files. R package version 1.4.3. https://CRAN.R-project.org/package=readxl. Accedido el 26 de septiembre de 2023.

[59] Xuannhat L., Thuc V., Trongduc L., Anh D. (2008) Addressing cold-start problem in recommendation systems. En: Proceedings of the 2nd international conference on Ubiquitous information management and communication (ICUIMC '08). Association for Computing Machinery, Nueva York, USA. pp: 208-211

[60] Zanker M., Jessenitschnig M. (2009) Collaborative feature-combination recommender exploiting explicit and implicit user feedback. En: 11th IEEE Conference on Commerce and Enterprise Computing (CEC), Vienna, Austria, 2009. pp. 49-56.

[61] Zanker M., Jessenitschnig M. (2009) Case-studies on exploiting explicit customer requirements in recommender systems. En: User Model User-Adap Interaction. pp:133-166.

[62] Zhang M., Wang W., Li X. (2008). A paper recommender for scientific literatures based on semantic concept similarity.En: Universal and ubiquitous access to information. pp. 359-362.