

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE CIENCIAS

Venta Cruzada de Productos Financieros: una aplicación de los sistemas recomendadores para la generación de acciones personalizadas en un portafolio de clientes heterogéneo

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO
MATEMÁTICO**

por

DAVID GONZALO YÁNEZ PETER
dyanezd@gmail.com

DIRECTOR: Ing. DIEGO ROLANDO MALDONADO GUERRERO, Msc.
drmaldon@bgr.com.ec

CODIRECTOR: Dra. SANDRA GUTIÉRREZ
sandra.gutierrez@epn.edu.ec

Quito, Septiembre de 2012

DECLARACIÓN

Yo DAVID GONZALO YÁNEZ PETER, declaro que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional, puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

DAVID GONZALO YÁNEZ PETER

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por DAVID GONZALO YÁNEZ PETER bajo mi supervisión.

Ing. Diego Maldonado, Msc.
Director del Proyecto

Ing. Sandra Gutiérrez, Ph.D.
Codirector del Proyecto

AGRADECIMIENTOS

Me gustaría que estas líneas sirvieran para expresar mi más profundo y sincero agradecimiento a todas aquellas personas que con su ayuda han colaborado en la realización del presente trabajo, en especial al Ing. Diego Maldonado, director de esta investigación, por la orientación, el seguimiento y la supervisión continúa de la misma, pero sobre todo por la motivación y el apoyo recibido a lo largo de estos años. Un agradecimiento muy especial merece la comprensión, paciencia y el ánimo recibidos de mi familia, amigos y compañeros.

A todos ellos Muchas gracias.

DEDICATORIA

Para mi padre y hermanos

RESUMEN

El desafío de la venta cruzada es determinar que producto ofrecer a un determinado cliente. En general los bancos cuentan con varios productos financieros candidatos. Desafortunadamente, el banco es incapaz de ubicar todos y cada uno de aquellos productos a cada cliente. Esto puede ser muy costoso, consumir demasiado tiempo o inefectivo. O simplemente, no desea que el cliente abandone el banco al sentirse abrumado por demasiadas ofertas. En ese sentido, en este trabajo se diseña y se evalúa un sistema recomendador para promover la venta cruzada de productos financieros puntualizando los elementos necesarios para una adecuada aplicación. El documento se encuentra conformado de la siguiente manera: en el Capítulo 1 se hace una introducción a la problemática planteada (estrategia de venta cruzada de productos financieros), mientras que en el Capítulo 2 se expone el contexto en el cual se abordará el tratamiento de dicha problemática (Sistemas Recomendadores). En el Capítulo 3 se realiza una segmentación previa a la aplicación del sistema recomendador con el fin que las recomendaciones sean más precisas. En el Capítulo 4 se expone la teoría concerniente a los Sistemas recomendadores. En el Capítulo 5 se aplica y se evalúa los algoritmos recomendadores para cada grupo determinado en el Capítulo 3, así como también sus resultados. Finalmente en el Capítulo 6 se encuentran las conclusiones y recomendaciones.

ABSTRACT

The challenge of cross-selling is to determine which product to offer a particular customer. Generally, banks have several financial products candidates. Unfortunately, the bank is unable to locate each and every one of those products to each customer. This can be costly, time consuming and ineffective. Or maybe just do not want the customer leaves the bank to get overwhelmed by too many offers. In that sense, this paper designs and evaluates a recommender system to promote cross-selling of financial products emphasizing the elements necessary for proper implementation.

CONTENIDO

Figuras	xi
Cuadros	xii
1. Introducción	1
2. CRM (Gestión de Relación con Clientes) Analítico	5
2.1. Gestión de Relación con el Cliente (CRM)	5
2.2. CRM Analítico y Venta Cruzada	6
2.3. Modelos Analíticos de Venta Cruzada y Sistemas Recomendadores	9
3. Segmentación de Clientes	14
3.1. Segmentación de Clientes Basada en el Valor y el Riesgo	15
3.1.1. Análisis RFM-Risk	16
3.1.2. Análisis de Conglomerados	19
3.2. Partición Alrededor de Medoids (PAM): descripción del algoritmo de segmentación	26
3.2.1. Función pam en R	31
3.3. Segmentación de Clientes Activos para la Cartera de una Institución Financiera	33

4. Sistemas Recomendadores	37
4.1. Introducción a los Sistemas Recomendadores	37
4.2. El Problema de Recomendación	39
4.2.1. Definición del problema de recomendación	41
4.3. El Enfoque de Recomendación Basado en Filtros Colaborativos	42
4.3.1. Recomendación Basada en el Usuario	45
4.3.2. Recomendación Basada en Productos	47
4.4. Evaluación de Sistemas Recomendadores	60
4.4.1. Midiendo la Precisión de los Ratings Predichos	61
4.4.2. Midiendo el Uso de las Predicciones	63
4.5. El paquete Recommenderlab en R	65
4.5.1. Infraestructura de Recommenderlab	66
5. Resultados	70
5.1. Análisis del Conjunto de Datos Transaccional de la Cartera	71
5.2. Calibración de la Matriz Usuario-Producto	75
5.3. Aplicación, Selección y Evaluación del Algoritmo Recomendador	78
5.3.1. Selección del algoritmo recomendador	79
5.3.2. Evaluación Capacidad de Predicción	83
6. Conclusiones Y Recomendaciones	87
6.1. Conclusiones	87
6.2. Recomendaciones	90
Anexos	92

A. Algoritmo PAM	93
B. Segmentación de Clientes	97
C. Calibración Matriz Usuario–Producto (Primera parte)	100
D. Calibración Matriz Usuario–Producto (Segunda parte)	101
E. Calibración Matriz Usuario–Producto (Tercera parte)	102
F. Selección Algoritmo Recomendador	103
G. Evaluación Capacidad Predicción Sistema Recomendador	104
Referencias Bibliográficas	109

FIGURAS

2.1. Diagrama de Retención de Clientes	10
3.1. Asignación del score RFM-Risk.	18
3.2. Un gráfico de 8 objetos.	20
3.3. Una partición con $n = 18$ y $k = 3$	24
3.4. Distinción entre técnicas aglomerativas y divisivas.	25
3.5. Un ejemplo bidimensional con 10 objetos.	27
3.6. Ejecución del algoritmo PAM	30
3.7. Determinación de $k = 8$	34
3.8. Visualización de las características de los clientes.	36
4.1. Metodología a utilizar para la venta cruzada de productos financieros	40
5.1. Diccionario Campos Base Transaccional.	72
5.2. Distribución conjunta del monto y plazo concedido para CON_1	77
5.3. Resultado del esquema de evaluación para los grupos 1–4 con $\mathcal{I} = \mathcal{I}_2$	79
5.4. Resultado del esquema de evaluación para los grupos 5–8 con $\mathcal{I} = \mathcal{I}_2$	80
5.5. Resultado del esquema de evaluación para los grupos 1–4 con $\mathcal{I} = \mathcal{I}_1$	81
5.6. Resultado del esquema de evaluación para los grupos 5–8 con $\mathcal{I} = \mathcal{I}_1$	82

CUADROS

3.1. Un ejemplo de un conjunto de datos transaccionales	17
3.2. Coordenadas de los objetos del ejemplo de la Figura 3.5	27
3.3. Evaluación de cada uno de los objetos a dos objetos representativos . .	29
3.4. Evaluación de cada uno de los objetos a otros dos objetos representativos	29
3.5. Segmentación de clientes generada por pam en base a los valores RFM–Risk.	35
4.1. Clasificación de los posibles resultados de una recomendación de un pro- ducto a un usuario	63
5.1. Resumen: Tipo de cuenta	72
5.2. Resumen: Tipo de Operación	73
5.3. Resumen: Tipo de Institución	74
5.4. Resumen: Estado Cartera	74
5.5. Características de los grupos para la cartera de consumo	76
5.6. Total de productos determinados	77
5.7. Productos Adquiridos por 180xxxx364 del grupo 8	84
5.8. Tasa de predicciones acertadas de productos financieros	85
5.9. Tasa de predicciones acertadas del tipo de productos financieros	85

CAPÍTULO 1

Introducción

Antecedentes

En una economía basada en servicios, las compañías se esfuerzan en incrementar sus ingresos al crear y fomentar relaciones a largo plazo con sus clientes. Un caso puntual es el de la banca minorista, donde el valor del cliente es de suma importancia para la institución. En efecto, Richardson [1992] menciona que los banqueros están obligados a iniciar negocios en pro de una *venta cruzada*¹ y desempeñarse como asesores de sus clientes si esperan lograr una participación justa o desproporcionadamente alta del mercado. Además, Richardson [1992] también habla del enfoque de ventas que debe asumir un banco, la llamada “banca consultiva”. La banca consultiva es un enfoque orientado a las ventas que proporciona a los banqueros las destrezas necesarias para aumentar la rentabilidad y desarrollar relaciones a largo plazo en un ambiente altamente competitivo. La banca consultiva se centra en los procesos de comunicación de ventas entre el banco y sus clientes, ayudando a los banqueros a encontrar productos desde el punto de vista del cliente. Por lo tanto, las necesidades individuales de los clientes son el punto central de la venta. En este contexto, los banqueros, en general, deberían tomar en cuenta enfoques similares al de la banca consultiva de manera que

¹Venta Cruzada (*Cross-selling*), significa proporcionar productos financieros adicionales a un cliente existente

éstas ayuden a construir relaciones a largo plazo que sean beneficiosas para ambos: clientes y el banco.

En el entorno sumamente competitivo de nuestros días, los bancos aprovechan las bases de datos de sus clientes para realizar actividades, en torno a la venta de productos, que generen no solo mayores beneficios para los banqueros, sino que también que sean del mayor interés para el cliente. Tal es el caso de la venta cruzada de productos financieros. La venta cruzada también puede verse como una estrategia de marketing mediante la cual se ofrece a los clientes de la compañía productos diferentes de los que ya posee². La idea detrás de la venta cruzada es capturar una mayor cuota del mercado de consumo mediante el cumplimiento de varias de las necesidades y deseos de cada cliente Ritter [1993]. La experiencia de muchos vendedores da cuenta que el costo de vender un producto adicional a un cliente actual representa un costo inferior de vender el mismo producto a un nuevo cliente Harding [2002]. Además, bajo determinadas circunstancias adversas al negocio, el incorporar nuevos clientes al banco se vuelve cada vez más difícil y costoso en una industria de consumo masivo como es la financiera, debido a que la adquisición de nuevos clientes se produce principalmente a expensas de los competidores Kamakura et al. [2003]. Para que la actividad de venta cruzada tenga éxito, es trascendental que los bancos adopten la potencia del análisis analítico (técnicas estadísticas, minería de datos, entre otras) para conseguir mejores perspectivas, evaluar apropiadamente el riesgo y crear un entorno más efectivo a la hora de tomar decisiones en la búsqueda de mejorar los beneficios de la compañía. Los bancos también necesitan

²A menudo la venta cruzada consiste en ofrecer al cliente los productos que complementan la compra original de alguna manera

asegurar que sus clientes existentes permanezcan satisfechos con la calidad del servicio que se les ofrece, sin olvidar también que el costo de la adquisición de un cliente nuevo es mucho más alto que el de retener uno existente. Considerando estos hechos, es esencial que los bancos opten por un modelo analítico con el fin de implementar de forma técnica una estrategia de venta cruzada de productos financieros con el fin de mejorar el valor del cliente y maximizar los beneficios que éstos generan al banco.

De la intuición al conocimiento

Durante décadas el conocimiento de clientes se basaba en la experiencia y en la intuición que los directivos de banca tenían sobre la posibilidad de adquisición de productos financieros de su clientela. Mientras se manejaban grupos de clientes sobre los que se podía obtener información de forma directa y asequible, el sistema funcionaba a la perfección. Al comienzo de la década de los noventa se empezaron a utilizar técnicas de análisis de la información de los clientes con el objetivo de conocer sus posibilidades comerciales. Eran técnicas con resultados limitados, que sólo podían actuar sobre bases de datos muy reducidas y cuyas conclusiones no podían extrapolarse a grandes masas de clientes. Pero el almacenamiento inteligente de la información, el desarrollo de aplicaciones de gestión de clientes y la posibilidad de analizar información con técnicas estadísticas avanzadas hace que se cuente con un complemento extraordinario a la experiencia, la ciencia. Por lo tanto, si para la organización el determinante del valor de cliente es lo que compra a lo largo de su vida de relación, su estrategia la determinarían los modelos de venta cruzada y de *up selling*³; mientras que para otras,

³Técnica de marketing que consiste en conseguir que los clientes aumenten el consumo de productos que ya utilizan y/o consuman productos de mayor valor

la capacidad de retención de clientes es lo que prima, los modelos de predicción de abandono serían su preocupación esencial.

CAPÍTULO 2

CRM (Gestión de Relación con Clientes) Analítico

La estrategia de venta cruzada puede abordarse como un objetivo más del CRM dirigido a mejorar y facilitar la labor comercial al integrar al cliente en la configuración de vender productos a la medida de sus posibilidades y gustos. CRM es básicamente la respuesta de la tecnología a la creciente necesidad de las empresas de fortalecer las relaciones con sus clientes Swift [2001].

§ 2.1 Gestión de Relación con el Cliente (CRM)

Para Valcárcel [2001] el CRM es la forma que tienen las empresas de interactuar con sus clientes, y destaca que lo que realmente ha otorgado protagonismo a este concepto desde hace ya algunos años es la capacidad que ofrecen las tecnologías de la información actuales para poder relacionar individualmente a las empresas con cada cliente. Pero para Díez [2004] CRM es algo más que relaciones, según el autor, CRM es el báculo donde el marketing apoya y orienta sus decisiones y su conocimiento. CRM, manifiesta él, no es un invento contemporáneo. Es lo que ha venido practicando cualquier persona que se ha relacionado con sus clientes de manera intensa. ¿Quién no ha observado o incluso apreciado el CRM que practica cualquier tendero de un mercado de abastos? o ¿y el CRM del comerciante de la esquina del barrio? Estos segmentos de empresas a priori poco tienen que hacer contra las multinacionales de la distribución y su concepto

de áreas de consumo. Ha sido precisamente este concepto de proximidad y conocimiento del cliente el que está posibilitando su supervivencia. De manera que lo nuevo del CRM no es el concepto o la filosofía, es simplemente la tecnología y la herramienta con las que se cuentan hoy en día para gestionar relaciones con los clientes. Son las nuevas herramientas tecnológicas las que han permitido tener mucha más información genérica del cliente y de su comportamiento en particular. La conclusión es rápida e inmediata, si se conoce al cliente, sus hábitos, sus consumos, sus preferencias, se puede dirigirse a él de manera mucho más personalizada, abriéndose así una puerta para la generación de venta cruzada, por ejemplo. Además, Los clientes son el activo más importante de una organización. No pueden haber perspectivas de negocio sin clientes satisfechos que permanezcan leales y que desarrollen relaciones con la organización. Es por esto que una organización debería planear y emplear una estrategia clara para tratar a los clientes. Por lo tanto CRM no es más que la estrategia de construir, administrar y fomentar lealtad y relaciones a largo plazo con el cliente.

§ 2.2 CRM Analítico y Venta Cruzada

El CRM debe ser un enfoque centrado en el cliente. Su alcance debería ser el “tratamiento” personalizado de los clientes como entidades distintas a través de la identificación y entendimiento de sus diferentes necesidades, preferencias y comportamientos. El CRM tiene dos objetivos principales [Tsipsis and Chorianopoulos \[2011\]](#):

1. Retención del cliente a través de la satisfacción del mismo.
 2. El desarrollo del cliente a través del conocimiento del mismo.
-

La importancia del primer objetivo es obvia. Siempre será más difícil reemplazar clientes existentes con nuevos provenientes de la competencia. Respecto al segundo objetivo, el punto clave es que no existe el comúnmente llamado cliente promedio. La base de clientes está compuesta por diferentes personas, con diferentes necesidades, comportamientos y potencialidades que debería ser administradas con responsabilidad. Varios programas de CRM están disponibles hoy en día y son usados ya que ofrecen un apoyo a los procesos del negocio de la organización, como por ejemplo, ventas, comercialización, etc.. Las interacciones con los clientes se almacenan generalmente en algún historial de contacto, y el personal puede recuperar información de los clientes según sea necesario. Este historial proporciona a los miembros del personal acceso inmediato a información importante sobre el cliente (productos de propiedad, el apoyo antes de llamadas, etc.), eliminando la necesidad de obtener esta información de forma individual, directamente con el cliente. Aquellos sistemas son conocidos como sistemas CRM operacionales. Sin embargo, éstos son simplemente herramientas que deberían ser usadas para soportar la estrategia de una efectiva administración de clientes. Para lograr el éxito del CRM y por tanto alcanzar los objetivos antesmencionados, las organizaciones necesitan conocer a sus clientes, sus necesidades y gustos a través del análisis de datos.

El CRM analítico consiste en analizar la información del cliente para atender de mejor manera los objetivos del CRM y entregar el mensaje correcto al cliente correcto. El CRM Analítico involucra el uso de técnicas de minería de datos¹ con el fin de evaluar el valor

¹Turban et al. [2010] define *Minería de Datos* como “el proceso que usa técnicas y métodos estadísticos, matemáticos, inteligencia artificial y aprendizaje de máquina para extraer e identificar información útil y posteriormente conseguir conocimiento, a partir de un gran conjunto de bases de

de los clientes, entenderlos y predecir su comportamiento. En definitiva, CRM Analítico trata sobre el análisis de patrones en los datos con el fin de extraer conocimiento para la optimización de las relaciones con los clientes. Por ejemplo, la minería de datos puede ayudar en la retención de clientes, ya que permite la identificación oportuna de aquellos más valiosos con una mayor posibilidad de que abandonen la organización, permitiendo a tiempo la focalización de campañas de retención. También puede ayudar a revelar distintos segmentos de clientes, facilitando el desarrollo de nuevos productos, y en la oferta de productos que mejor se ajusten a las preferencias y prioridades específicas de los clientes.

La venta cruzada es una aplicación CRM cuyo objetivo estratégico es el de incrementar los beneficios de la empresa. Lejos de buscar oportunidades de negocio, inversiones en inteligencia y desarrollo y gastos en recursos comerciales, se puede sacar gran provecho del negocio simplemente en la venta cruzada de productos entre los clientes de la organización. La venta cruzada aporta diferentes beneficios de oportunidad Díez [2004]:

- Ahorro en el costo de adquisición de un nuevo cliente.
- Ahorro en una venta con menor costo estructural.
- Aumento de la fidelidad de la cartera de clientes.
- Mejoramiento de la posición competitiva en el mercado.
- Reforzar el producto estrella de la empresa.

De manera que conocer las necesidades, hábitos y atributos de valoración en el proceso de compra de los clientes facilitará el poder presentar ofertas de venta cruzada. Es datos”.

decir, haciendo uso del CRM analítico se puede gestar una estrategia técnica de venta cruzada de productos.

§ 2.3 Modelos Analíticos de Venta Cruzada y Sistemas Recomendadores

La venta cruzada es una estrategia de marketing importante que trata de incrementar los beneficios de la institución por medio de los clientes actuales. La venta cruzada se da cuando una compañía vende diferentes productos a sus clientes. Los potenciales beneficios que se conseguirían al tomar en cuenta esta estrategia es ilustrado por el modelo de retención simple del valor de tiempo de vida [Blattberg et al. \[2008\]](#):

$$\text{LTV} = \sum_{t=1}^n \frac{m_t r^{t-1}}{(1 + \delta)^{t-1}},$$

donde LTV^2 representa el valor del tiempo de vida del cliente para la empresa en n períodos, m_t es la contribución de un cliente a los beneficios generados en el período t , r la tasa de retención y δ es alguna tasa de descuento. Primero, la venta cruzada puede generar altas ventas en el período actual (m_1 es mayor debido a que en este mes, además de la carga mensual normal, el cliente ha comprado un producto nuevo). Segundo, la venta cruzada puede incrementar ingresos futuros (mayor m_t para $t > 1$). Tercero, y menos obvia, la venta cruzada puede incrementar la tasa de retención (r). La Figura 2.1 muestra porque la venta cruzada puede incrementar la retención. La venta cruzada da como resultado que el cliente posea más productos de la compañía. A medida que el cliente adquiera más productos, mejor atendido será éste. Esto in-

²LTV es un procedimiento que permite calcular el valor presente de un determinado número de flujos de rentabilidad

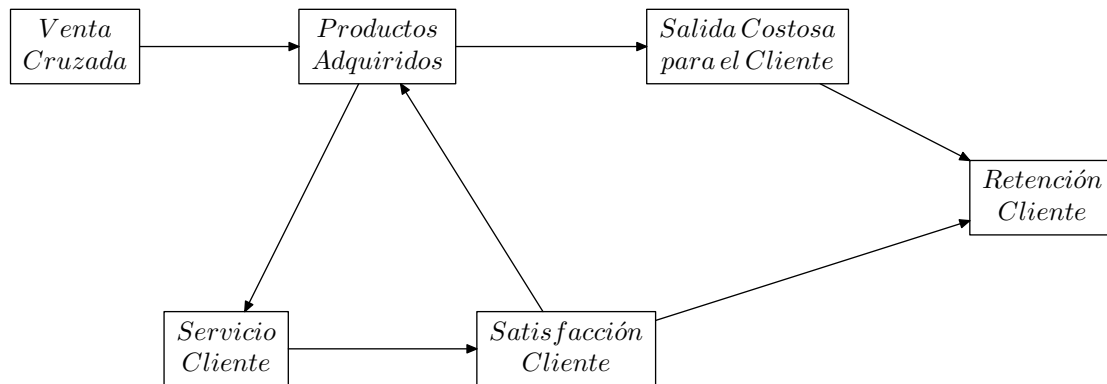


Figura 2.1: Diagrama que muestra cómo la venta cruzada puede incrementar la retención de clientes. Fuente: [Blattberg et al. \[2008\]](#).

crementa la satisfacción del cliente y a su vez producir retención. Además, la mayor satisfacción del cliente alienta a éste a comprar más productos de la compañía, lo que refuerza este ciclo. Por otro lado, a mayor cantidad de productos posea el cliente, más alto será el costo para el cliente si cambia de empresa, y esto aumenta la retención. La pregunta clave es: *¿qué productos debería el banco vender en forma cruzada y a quienes?* Tres tipos de modelos predictivos han sido desarrollados para solventar esta tarea:

1. Modelos que se centran en predecir cuál es el producto más probable que compraría el cliente.
2. Modelos que también consideran cuando probablemente se efectuó la compra.
3. Modelos que también consideran que tan probable el cliente responda a la oferta de venta cruzada.

Los dos primeros tipos tratan de inferir que producto el cliente necesita. El tercer

tipo se centra en cómo el cliente responderá a los esfuerzos de marketing de venta cruzada. Naturalmente, en el primer conjunto de modelos se encuentran los métodos más básicos para efectuar una estrategia de venta cruzada. La idea detrás de aquellos tipos de modelos es la siguiente: indagar, para clientes con perfil similar al cliente objetivo, que productos (productos adquiridos actualmente, características personales) compraron después, y asumir que esto es lo que necesita el cliente objetivo. Uno de aquellos tipos de modelos es el *Filtro Colaborativo*.

El filtro colaborativo es considerada como la técnica más popular y ampliamente implementada en *Sistemas Recomendadores*. Hay dos clases de sistemas recomendadores del tipo filtro colaborativo, *basado en el usuario* y *basado en el producto*. Los sistemas basados en el usuario encuentran clientes que tienen gustos similares al cliente objetivo, y busca dentro de aquellos clientes, el o los productos a recomendar. Los sistemas basados en productos parten con el conjunto de productos que el cliente objetivo ya ha comprado; la suposición detrás de este enfoque es que los usuarios preferirán productos similares a los productos que les gusta. El núcleo de los sistemas recomendadores basados en filtro colaborativo, es entonces, predecir preferencias. Los sistemas recomendadores aplican técnicas estadísticas y de minería de datos para hacer recomendaciones de productos en base a datos de compras del cliente, previamente registrados. De esa manera, los sistemas recomendadores pueden ayudar a promover una venta cruzada de productos financieros de forma efectiva al sugerir productos adicionales, y además puede mejorar la lealtad del cliente al crear relaciones en base al valor.

El problema central en aplicar una estrategia de venta cruzada de productos financieros, bajo el contexto del marketing analítico, es ubicar el o los productos correctos al cliente

correcto. Sin embargo una implementación práctica y efectiva de esta meta no es tan fácil de conseguir. Con el objetivo de dar respuesta a este problema se plantea la utilización de un sistema recomendador. A continuación se exponen varias razones por las cuales optar por este tipo de modelos es adecuada para afrontar el problema planteado Ricci et al. [2011]:

- **Incrementa el número de productos vendidos.** Éste es posiblemente la función más importante para un sistema recomendador, es decir, ser capaz de vender un conjunto adicional de productos comparados a aquellos que usualmente se vendería sin algún tipo de recomendación. Esta meta se puede lograr porque los productos recomendados posiblemente se ajusten a las necesidades y preferencias del usuario.
 - **Vender productos más diversos.** otra función importante de un sistema recomendador es que permite al usuario seleccionar productos que podría ser difícil encontrar sin una precisa recomendación.
 - **Incrementar la satisfacción del usuario.** Un sistema recomendador correctamente diseñado puede también mejorar la experiencia del usuario con el sitio o la aplicación. El usuario encontrará las recomendaciones interesantes y posiblemente relevantes. La combinación de efectividad en las recomendaciones, y una conveniente interfaz incrementará la evaluación subjetiva del usuario hacia el sistema. Esto dará como resultado que se incremente el uso del sistema y posiblemente que las recomendaciones sean aceptadas.
 - **Incrementar la fidelidad del usuario.** Un usuario debería ser leal a un sitio
-

en el cual, cuando lo viste, reconozca al cliente y lo trate como un visitante de valor. Esta es una característica normal de un sistema recomendador dado que muchos de estos sistemas calculan recomendaciones aprovechando la información obtenida de los demás usuarios en interacciones anteriores, dando como resultado que las recomendaciones sean más eficaces a medida que las recomendaciones anteriores coincida con las preferencias del usuario.

- **Mejor entendimiento de lo que quiere el usuario.** Otra función importante de un sistema recomendador es la descripción de las preferencias del usuario. La institución puede entonces decidir usar este conocimiento para otras metas.

CAPÍTULO 3

Segmentación de Clientes

Los bancos poseen varios conjuntos de bases de datos, que por lo general, contienen un volumen bastante grande de registros. Entre aquellos registros se puede encontrar información relacionada a datos transaccionales de los clientes. Como es natural, existe información valiosa que puede ser extraída de aquellos bancos de datos, en particular si el interés del banco recae en la construcción de diferentes grupos de clientes con el fin de desarrollar estrategias de marketing diferenciadas de acuerdo a las características presentes en cada grupo. Entre una de aquellas estrategias se encuentra la de realizar recomendaciones personalizadas de productos con el fin de fomentar la venta cruzada. La segmentación es el proceso de dividir la base de datos de los clientes dentro de distintos grupos internamente homogéneos. Hay muchos tipos diferentes de segmentación. Todas estas basadas en algún criterio específico o atributos usados para la segmentación. Por ejemplo, en la segmentación *basada en el comportamiento*, los clientes son agrupados de acuerdo al comportamiento y características de uso presentes en cada uno de ellos. Aunque ésta segmentación puede ser creada con reglas determinadas por el negocio, este enfoque tiene inherentes desventajas:

1. Puede ser manejado eficientemente únicamente con pocos campos de segmentación.
2. Es cuestionable, ya que se basa en percepciones personales del analista.

La minería de datos, por otro lado, puede crear segmentaciones basadas en el comportamiento deducidas a partir de los datos. Los algoritmos de clustering¹ pueden analizar datos de comportamiento, identificar agrupaciones naturales de clientes y proponer una solución fundamentada en los patrones de datos observados. Al igual que la segmentación basada en el comportamiento, la minería de datos también puede ser usada para el desarrollo de un esquema de segmentación basada en el valor esperado/estimado del cliente. Aquellas segmentaciones son necesarias para el banco a fin de priorizar el tratamiento del cliente y realizar intervenciones de marketing de acuerdo a la importancia de cada uno de los clientes.

§ 3.1 Segmentación de Clientes Basada en el Valor y el Riesgo

Como se menciona anteriormente la *segmentación basada en el valor* es un esquema que permite agrupar a los clientes y de esa manera identificar a los clientes de alto valor y priorizar su tratamiento de acuerdo a su importancia medida. Pero bajo qué criterios el banco podría definir a un cliente como de alto valor. La respuesta se encuentra en el análisis de los hábitos de consumo del cliente. Los académicos del marketing directo han encontrado que la respuesta de compra del cliente puede ser predicha usando el historial de sus compras previas Blattberg et al. [2008]. Las tres variables más importantes que resumen el historial de compras del cliente son: recency, frequency y monetary. La medida *recency* (R) indica el tiempo transcurrido desde la última compra del cliente. *Frequency* (F) denota el número de compras que ha hecho el cliente. *Monetary* (M) indica el valor gastado en aquellas compras. El análisis RFM

¹El *Análisis Cluster* es una técnica estadística que es usada para identificar un conjunto de grupos tal que se minimize la distancia entre el grupo y a la vez se maximize la distancia entre grupos.

puede ser usado para identificar buenos clientes (quienes posean los mejores scores de acuerdo a estas medidas), quienes generalmente tienden a ser buenos prospectos para realizar compras adicionales [Tsiptsis and Chorianopoulos \[2011\]](#). Sin embargo, al tratar de aplicar este análisis a la industria bancaria hay que tener presente que los productos que allí se ofertan difieren a los de cualquier otra empresa de retail, el riesgo. Por lo tanto, un cliente de alto valor para el banco no es simplemente aquel que ha comprado recientemente, que lo hace con alta frecuencia y que gasta bastante, sino que también el que tenga menos posibilidad de incumplir sus obligaciones. A esta cuarta medida se le llamará *Risk*.

§ § 3.1.1 Análisis RFM-Risk

El concepto de RFM fue introducido por [Bult and Wansbeek \[1995\]](#) y posteriormente aplicado por [Blattberg et al. \[2008\]](#) al marketing. El análisis RFM-Risk depende de las medidas de Recency (R), Frequency (F), Monetary (M) y Risk. Aquellas cuatro son las variables más importantes relacionadas al comportamiento de compras de los clientes. A fin de demostrar el Análisis RFM-Risk, en el Cuadro [3.1](#) se muestra un extracto de un conjunto de datos transaccionales. La segmentación empieza ordenando, primeramente, la información de los clientes en base de la variable recency, es decir, el tiempo transcurrido desde la última compra efectuada, en orden ascendente (las compras más recientes en el top de la tabla). Luego, los clientes son divididos en quintiles. Al primer quintil, el top 20% en recency, se le asigna un score de 5, al siguiente quintil un score de 4, al tercer quintil, 3, al cuarto, 2, y finalmente al quinto quintil, 1. Los clientes nuevamente son ordenados, esta vez por la variable frequency, de forma descendente. Al igual que la anterior variable, se asigna al primer quintil

Cliente	Recency (días)	Frequency (#)	Monetary (\$)	Risk (%)
1	3	6	540	0
2	6	10	940	0.5
3	45	1	30	1
4	21	2	64	2
5	14	4	169	8.5
6	32	2	55	7.5
7	5	3	130	5
8	50	1	950	0
9	33	15	2430	3
10	10	5	190	1.5
11	5	8	840	10
12	1	9	1410	3
13	24	3	54	0
14	17	2	44	2.5
15	4	1	32	4

Cuadro 3.1: Un ejemplo de un conjunto de datos transaccionales. Elaboración Propia.

un score de 5 y a los quintiles menos frecuentes un score de 4, 3, 2 y 1. Este proceso también se lleva a cabo de igual manera para la variable monetary; en el caso de la variable risk, el proceso es similar al de la variable recency. Finalmente, todos los clientes son ranqueados al concatenar los scores correspondientes a los valores R, F, M y Risk, dando como resultado el valor RFM-Risk. Los clientes con los valores más altos de RFM-Risk serán aquellos cuyos valores, para cada una de las cuatro variables, sean de 5, es decir, 5555. De manera similar, los clientes con recency promedio (quintil 3), con la más alta frecuencia de compra (quintil 5), con los valores monetarios más bajos (quintil 1) y que presentan el menor riesgo posible (quintil 5) tendrán un score de 3515.

El procedimiento para construir el score RFM-Risk se ilustra en la Figura 3.1. Cuando se agrupan a los clientes en quintiles (grupos de 20%), el método resulta en un total de

$5 \times 5 \times 5 \times 5 = 625$ valores posibles de RFM-Risk. Esta combinación de los componentes R, F, M y Risk es ampliamente usada, a pesar de que ésta tiene una cierta desventaja. El gran número de valores RFM-Risk derivados hacen que el proceso sea un tanto difícil de manejar e interpretar. Un método alternativo para la segmentación de clientes de acuerdo a sus patrones RFM-Risk es el uso de los distintos componentes como *inputs* en un modelo clustering y dejar que el algoritmo revele las agrupaciones naturales subyacentes de clientes. Por ejemplo, [Hosseini et al. \[2010\]](#) combina un modelo RFM ponderado² junto con el algoritmo *k*-medias para mejorar el CRM para empresas. [Wu et al. \[2009\]](#) aplicaron el modelo RFM y el método *k*-medias en el análisis del valor del cliente para la base de datos de un proveedor en Taiwan para establecer fuertes relaciones con los clientes y naturalmente consolidar su lealtad.

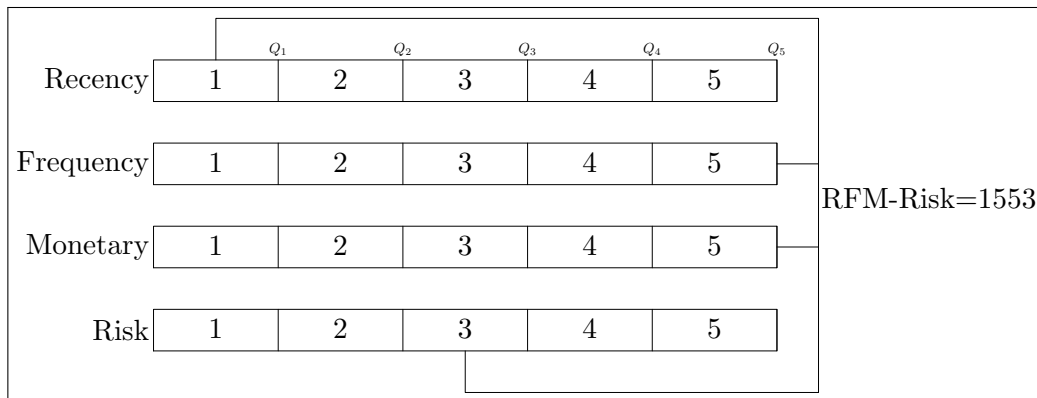


Figura 3.1: Asignación del score RFM-Risk. Elaboración Propia.

²*RFM Ponderado*, es una versión del análisis RFM cuyos valores R, F y M son multiplicados por determinados pesos w_R , w_F y w_M , respectivamente, de acuerdo a su importancia relativa, generalmente basadas en juicios intuitivos.

§ § 3.1.2 Análisis de Conglomerados

Kaufman and Rousseeuw [2005] define al análisis de conglomerados o cluster como el arte de encontrar grupos en los datos. Para ver que significa esto, considérese la Figura 3.2. En ésta se muestran ocho observaciones, en las cuales se miden dos variables: peso y altura de una persona. En este pequeño conjunto de datos, claramente se observa dos grupos distintos de personas: {TIN, TAL, KIM, ILA} y {LIE, JAC, PET, LEO}. Tales grupos son llamados *clusters*, y descubrirlos es el objetivo del análisis cluster. Básicamente, lo que se desea es formar grupos de tal manera que los objetos en el mismo grupo sean similares unos a otros, mientras que los objetos en diferentes grupos sean lo más disímiles³ posible.

Usualmente, se requiere que tal medida tenga ciertas propiedades, dependiendo del problema específico al cual es aplicado. En principio, una medida de similaridad es una función $d : D \times D \rightarrow \mathbb{R}^+$ aplicado en un conjunto de objetos D y que tiene ciertas propiedades específicas. La medida de similaridad, al ser una función de distancia, posee las propiedades básicas de una métrica, esto es, para dos objetos A y B :

- $d(A, B) \geq 0, d(A, B) = 0 \Leftrightarrow A = B$;
- $d(A, B) = d(B, A)$;
- $d(A, B) + d(B, C) \geq d(A, C)$.

A continuación se presentan las medidas de similaridad más populares, las cuales son aplicadas en la mayoría de los casos Gorunescu [2011]. Para medir la similaridad

³Con frecuencia, en lugar de usar la medida de similaridad, se puede considerar la disimilaridad, dado que es más apropiada la idea de medir la distancia entre objetos

entre dos objetos, hay que considerar a estos como dos vectores: $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$, $\mathbf{y} = (y_1, y_2, \dots, y_n)^T$, que tienen por simplicidad, la misma dimensión n .

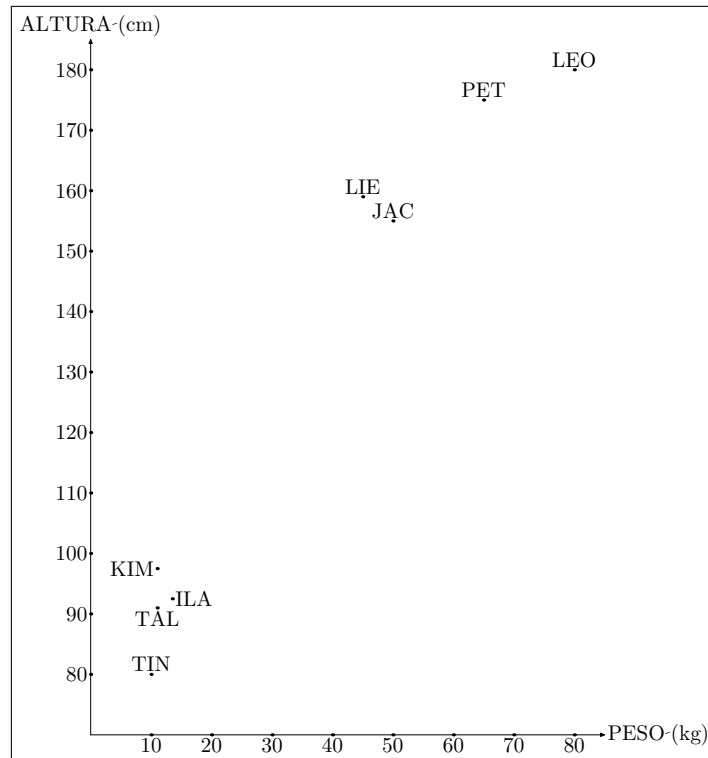


Figura 3.2: Un gráfico de 8 objetos. Fuente: [Kaufman and Rousseeuw \[2005\]](#).

Distancia de Minkowski

La distancia de Minkowski está dada por la siguiente expresión:

$$d_p(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{1/p}, \quad p \in \mathbb{N}. \quad (3.1)$$

Para $p = 1$ en la Ecuación 3.1 se obtiene la *distancia* L_1 (o *distancia de Manhattan*):

$$d_M(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n |x_i - y_i|. \quad (3.2)$$

En cambio para $p = 2$ en la Ecuación 3.1 se obtiene la bien conocida *distancia* L_2 (o *distancia Euclidiana*):

$$d_E(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^n |x_i - y_i|^2 \right)^{1/2}. \quad (3.3)$$

Distancia del Coseno

Esta medida está dada por el coseno del ángulo entre dos vectores \mathbf{x} y \mathbf{y} , es decir,

$$d_c(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^T \cdot \mathbf{y}}{\|\mathbf{x}\|_E \cdot \|\mathbf{y}\|_E}, \quad (3.4)$$

donde (\cdot) denota el producto punto entre dos vectores y $\|\cdot\|_E$ representa la norma euclidiana.

Distancia de Tanimoto

Esta medida está dada por:

$$d_T(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^T \cdot \mathbf{y}}{\mathbf{x}^T \cdot \mathbf{x} + \mathbf{y}^T \cdot \mathbf{y} - \mathbf{x}^T \cdot \mathbf{y}}. \quad (3.5)$$

Distancia de Jaccard

También conocido como índice de Jaccard. Esta medida se usa en estadística para medir la similaridad/disimilaridad entre dos conjuntos de muestras. Así, el índice de Jaccard entre dos conjuntos de muestras, $J(A, B)$, es la relación de la cardinalidad de su intersección y la cardinalidad de su unión, es decir,

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}. \quad (3.6)$$

Por lo tanto, la distancia de disimilaridad de Jaccard entre A y B está dada por $1 - J(A, B)$.

Distancia de Pearson

También conocida como *coeficiente de correlación*. Esta medida está da por:

$$r(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}. \quad (3.7)$$

Distancia de Mahalanobis

La medida de Mahalanobis generalmente está dada por:

$$d_{MH}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T B (\mathbf{x} - \mathbf{y})}, \quad (3.8)$$

donde B es una matriz simétrica definida positiva. Además, para dos vectores aleatorios \mathbf{x} y \mathbf{y} , con la misma distribución y matriz de covarianza $\text{Cov}(\mathbf{x}, \mathbf{y})$, la medida de Mahalanobis está dada por:

$$d_{MH}(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T \text{Cov}(\mathbf{x}, \mathbf{y}) (\mathbf{x} - \mathbf{y})}. \quad (3.9)$$

Si la matriz de covarianza es la matriz identidad, la distancia de Mahalanobis se reduce a la distancia Euclidiana (Ecuación 3.3).

Una vez que se ha seleccionado la distancia de similaridad, basada en los objetos a comparar, el segundo paso consiste en particionar el conjunto de datos al usar una metodología particular. La elección de un algoritmo clustering depende de dos cosas, del tipo de datos disponibles y del propósito particular. Es permisible tratar varios algoritmos en el mismo conjunto de datos, ya que al análisis cluster es principalmente usado como una herramienta descriptiva o exploratoria, en contraste con las pruebas estadísticas las cuales son llevadas a cabo para propósitos inferenciales o confirmatorios. Es decir, no se desea probar (o desaprobar) una hipótesis preconcebida; simplemente

se quiere observar lo que los datos nos quieren contar. [Kaufman and Rousseeuw \[2005\]](#) considera dos tipos de algoritmos clustering, los *métodos jerárquicos y de partición*. A continuación se da una breve descripción de aquellos métodos.

Métodos de Partición

Un método de partición construye k grupos. Es decir, el método clasifica los datos en k grupos, que satisfacen los siguientes requerimientos de partición:

- Cada grupo debe contener al menos un objeto.
- Cada objeto pertenece a un solo grupo.

Aquellas condiciones implican que hay a lo más tantos grupos como objetos, es decir, $k \leq n$, donde n es el número de objetos. La segunda condición dice que dos diferentes clusters no pueden tener un objeto en común y que la unión de los k grupos da como resultado el conjunto de datos, es decir, una partición. La [Figura 3.3](#) muestra un ejemplo de una partición de 18 puntos en 3 clusters. Hay que notar sin embargo, que k es dado por el usuario. Realmente el algoritmo construirá una partición con tantos clusters como se desee. Desde luego, no todos los valores de k conducen a clustering “naturales” de modo que sería saludable ejecutar el algoritmo varias veces con diferentes valores de k y seleccionar aquel k^* para el cual se cumplan ciertas características, o que conduzcan a la mejor interpretación. También es posible que esta decisión se la realice de forma automática⁴, es decir, dejar al computador que evalúe todos (o muchos de)

⁴En [Brock et al. \[2008\]](#) se describe el paquete de R, `clValid`, el cual contiene funciones para validar los resultados de un análisis cluster para varios algoritmos. El usuario puede simultáneamente seleccionar múltiples algoritmos, medidas de validación y el número de clusters con el llamado de una

los valores posibles de k y escoger el mejor de acuerdo a algún criterio numérico.

Los métodos de partición son aplicados si lo que se desea es clasificar los objetos en k grupos, donde k es fijo. En general el algoritmo trata de encontrar una “buena” partición en el sentido que los objetos del mismo grupo deben ser cercanos o relacionados unos a otros, mientras que los objetos de diferentes grupos deben estar lo más lejos posible o ser muy diferentes. El objetivo es por lo general descubrir una estructura que está presente en los datos, pero algunas veces el algoritmo es usado para imponer una nueva.

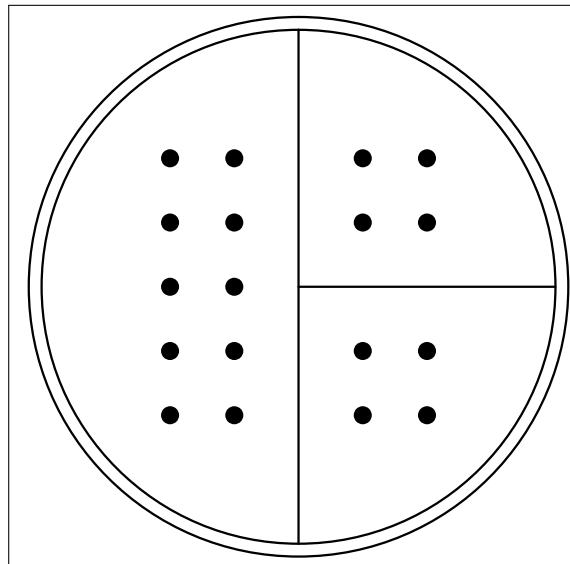


Figura 3.3: Una partición con $n = 18$ y $k = 3$. Fuente: [Kaufman and Rousseeuw \[2005\]](#).

sola función, para determinar el método más apropiado y el número óptimo de grupos para el conjunto de datos.

Métodos Jerárquicos

Los algoritmos jerárquicos no construyen una única partición con k grupos, sino más bien evalúan todos los valores de k en una misma ejecución. Es decir, la partición con $k = 1$ (todos los objetos están juntos en un mismo grupo) es parte del resultado, y también el caso $k = n$ (cada objeto forma un grupo separado con únicamente un solo elemento). El resto de valores de $k = 2, 3, \dots, n - 1$ son cubiertos en un tipo de transición gradual: la única diferencia entre $k = r$ y $k = r + 1$ es que uno de los r grupos se divide a fin de obtener $r + 1$ grupos (o, desde otra perspectiva, dos de los $r + 1$ grupos se combinan para formar r grupos).

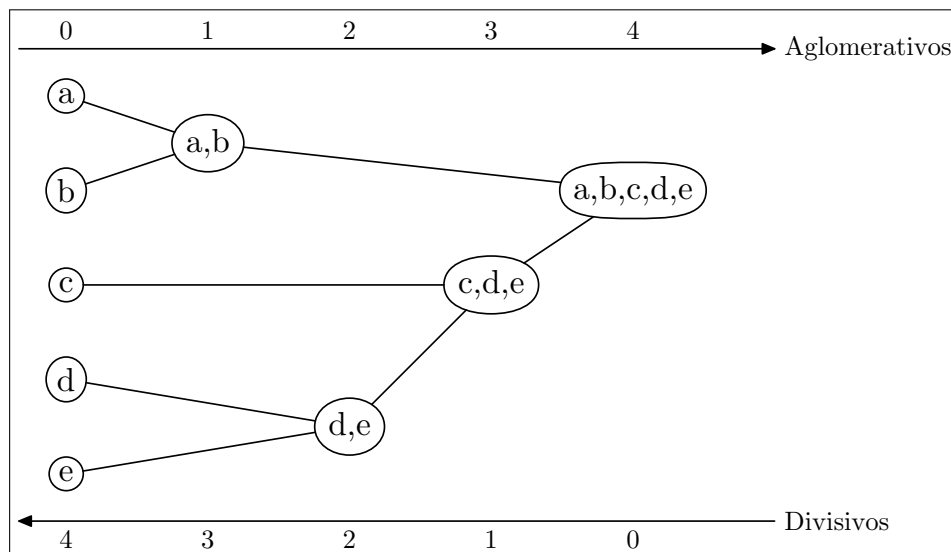


Figura 3.4: Distinción entre técnicas aglomerativas y divisivas. Fuente: [Kaufman and Rousseeuw \[2005\]](#).

Hay dos tipos de técnicas jerárquicas: los *aglomerativos* y los *divisivos*. Aquellos construyen su jerarquía en dirección opuesta, posiblemente llegando a resultados diferentes. La Figura 3.4 muestra lo que sucede con un conjunto de datos con 5 objetos.

Los métodos aglomerativos empiezan cuando todos los objetos son disjuntos. Luego, en cada paso dos grupos son unen, hasta que solo quede uno. Por otro lado, los métodos divisivos empiezan cuando todos los objetos están juntos (es decir, en el paso 0 hay un solo grupo) y cada paso subsiguiente se produce la división de un grupo, hasta que hayan n de ellos. En este ejemplo, los 2 métodos coinciden, pero usualmente sus resultados son diferentes. Los métodos jerárquicos adolecen del defecto de que no se puede reparar lo que se hizo en pasos anteriores. De hecho, una vez que el algoritmo aglomerativo ha unido dos objetos, éstos no pueden separarse más. También, todo lo que el algoritmo divisivo ha separado no puede ser reunido. Por otro lado, las técnicas jerárquicas en realidad no compiten con los métodos de partición, ya que no persiguen el mismo objetivo, éstos tratan de describir los datos de una manera totalmente diferente.

§ 3.2 Partición Alrededor de Medoids (PAM): descripción del algoritmo de segmentación

Cuando se particiona un conjunto de objetos dentro de k grupos, el principal objetivo es agrupar los objetos que muestran un alto grado de similaridad, mientras que los objetos que pertenecen a grupos diferentes sean los más disimilares posible. Desde luego, existen muchos métodos que tratan de resolver esta tarea. El algoritmo usado en el método PAM se basa en la búsqueda de k *objetos representativos* entre los objetos del conjunto de datos. Como evoca su nombre, aquellos objetos deberían representar varios aspectos de la estructura de los datos. En el algoritmo PAM los objetos representativos son los así llamados *medoids* de los grupos.

Número	Coordenada x	Coordenada y
1	1	4
2	5	1
3	5	2
4	5	4
5	10	4
6	25	4
7	25	6
8	25	7
9	25	8
10	29	7

Cuadro 3.2: Coordenadas de los objetos del ejemplo de la Figura 3.5. Elaboración propia.

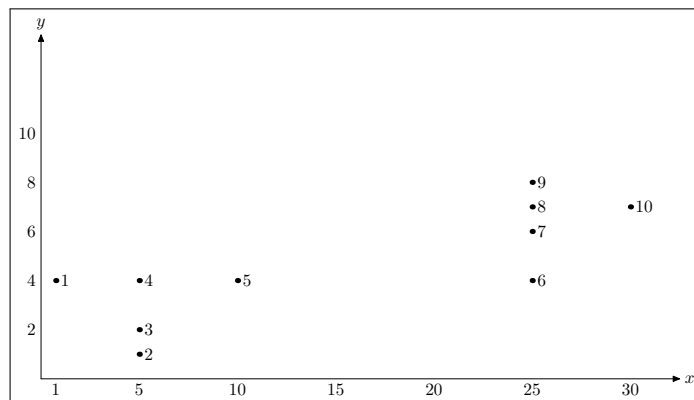


Figura 3.5: Un ejemplo bidimensional con 10 objetos. Fuente: Kaufman and Rousseeuw [2005].

Luego de encontrar un conjunto de k objetos representativos, se construyen los k grupos al asignar cada objeto del conjunto de datos a los objetos representativos más cercanos. Para ilustrar el algoritmo PAM considérese el conjunto de datos representados en la Figura 3.5. Este conjunto de datos contiene 10 objetos ($n = 10$) cada uno caracterizado por dos variables ($p = 2$), llamados x y y . Los valores x y y están dados en el Cuadro 3.2. Supongamos que los datos deben ser divididos en dos subconjuntos ($k = 2$). El algoritmo, en primera instancia, debe escoger dos objetos representativos y luego construir los grupos alrededor de aquellos objetos. Por ejemplo, supongamos que los objetos 1 y 5 son los objetos representativos. En el Cuadro 3.3 las disimilaridades de cada uno de los objetos a los objetos seleccionados son registrados, así como también la más pequeña de aquellas medidas y el correspondiente objeto representativo⁵. La disimilaridad promedio es 9.37. Este valor mide la rigurosidad de los grupos y por lo tanto la calidad de los grupos. En el Cuadro 3.4 se lleva a cabo la evaluación para el caso de los objetos 4 y 8 que son seleccionados como objetos representativos. Las agrupaciones asociadas con aquellos dos pares de objetos representativos se muestran en la Figura 3.6. La disimilaridad promedio para el caso de los objetos 4 y 8 seleccionados es de 2.3, cuyo valor es considerablemente menor al valor de 9.37 que se encontró cuando los objetos 1 y 5 fueron los objetos representativos. En el Apéndice A se describe el algoritmo PAM, el cual para un k dado se es posible seleccionar k objetos representativos que producen una disimilaridad promedio muy baja, y por lo tanto una “buena” partición.

⁵En este ejemplo se ha usado la distancia Euclidiana dada por la Ecuación 3.3

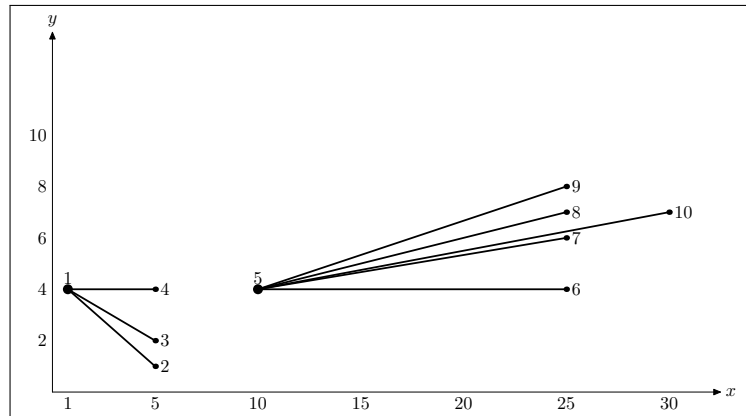
Objeto	$d_E(x_i, 1)$	$d_E(x_i, 5)$	$\min \{d_E(x_i, 1), d_E(x_i, 5)\}$	Objeto Repre.
1	0	9	0	1
2	5	5.83	0	1
3	4.47	5.39	5	1
4	4	5	4.47	1
5	9	0	4	5
6	24	15	0	5
7	24.08	15.13	15	5
8	24.19	15.30	15.13	5
9	24.33	15.52	15.30	5
10	28.16	19.24	15.52	5
Promedio			9.37	

Cuadro 3.3: Evaluación de cada uno de los objetos a dos objetos representativos

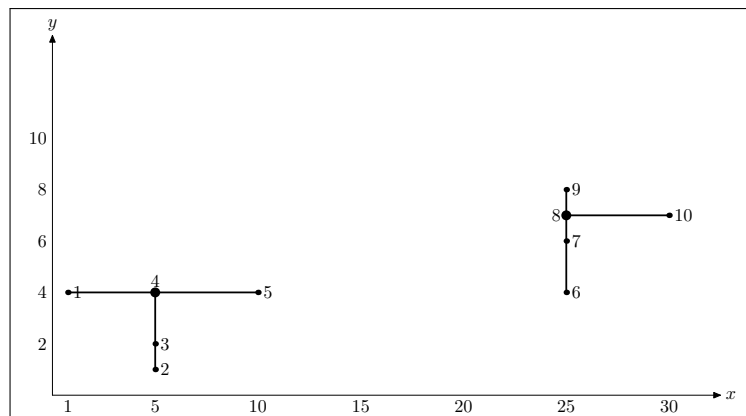
Objeto	$d_E(x_i, 4)$	$d_E(x_i, 8)$	$\min \{d_E(x_i, 1), d_E(x_i, 5)\}$	Objeto Repre.
1	4	24.19	4	4
2	3	20.88	3	4
3	2	20.62	2	4
4	0	20.22	0	4
5	5	15.30	5	4
6	20	3	3	8
7	20.1	1	1	8
8	20.22	0	0	8
9	20.40	1	1	8
10	24.19	4	4	8
Promedio			2.3	

Cuadro 3.4: Evaluación de cada uno de los objetos a otros dos objetos representativos

El clustering encontrado con este algoritmo es el mismo presentado en el Cuadro 3.4, pero los dos objetos representativos son 3 y 8 y por lo tanto la disimilaridad promedio es 2.19. En el método usado por el algoritmo PAM los objetos representativos de un grupo es su *medoid*, el cual se define como el objeto del grupo para el cual la disimilaridad promedio a todos los demás objetos dentro del grupo es mínima. Como el objetivo es encontrar k de tales objetos, a este método se lo conoce como *método k -medoid*.



(a) Grupos correspondientes a las selecciones descritas en el Cuadro 3.3



(b) Grupos correspondientes a las selecciones descritas en el Cuadro 3.4

Figura 3.6: Ejecución del algoritmo PAM

§ § 3.2.1 Función pam en R

Tanto en R como en S-PLUS el algoritmo PAM está implementado en la función `pam`. Ésta función se basa en la búsqueda de k objetos representativos, que como se mencionó, son llamados *medoids*, entre los objetos de un conjunto de datos. Recordemos que aquellos medoids son seleccionados de tal forma que la matriz de disimilaridad total de todos los objetos hacia su medoid más cercano sea mínima, es decir, la meta es encontrar un subconjunto $\{m_1, \dots, m_k\} \subset \{1, \dots, n\}$ que minimicen la función objetivo

$$\sum_{i=1}^n \min_{t=1, \dots, k} d(i, m_t). \quad (3.10)$$

Cada objeto es entonces asignado al cluster correspondiente al medoid más cercano. Es decir, un objeto i es designado dentro del cluster v_i cuando el medoid m_{v_i} está más cerca a i que a cualquier otro medoid m_w , o

$$d(i, m_{v_i}) \leq d(i, m_w), \quad \text{para todo } w = 1, \dots, k.$$

El algoritmo actual de `pam` procede en dos pasos:

1. PASO DE CONSTRUCCIÓN

Se determinan los medoids iniciales:

- m_1 es el objeto con el más pequeño valor de $\sum_{i=1}^n d(i, m_1)$
 - m_2 disminuye el valor del objetivo 3.10 lo más posible
 - \vdots
 - m_k disminuye el valor del objetivo 3.10 lo más posible
-

2. PASO DE INTERCAMBIO

Se repite hasta llegar a la convergencia. Considerar todos los pares de objetos (i, j) con

$$i \in \{m_1, \dots, m_k\} \quad \text{y} \quad j \notin \{m_1, \dots, m_k\},$$

y hacer el intercambio de i por j para el cual el objetivo 3.10 disminuya.

Dado que la función objetivo 3.10 únicamente depende de las disimilaridades entre los objetos, la función `pam` solo necesita una matriz de disimilaridad⁶. El método PAM puede ser comparado con el bien conocido método de k -medias cuya implementación en R se da por medio de la función `kmeans` usando el algoritmo de Hartigan y Wong. En el método de k -medias el centro de cada cluster está definido como la media de todos los objetos del cluster. La meta de `kmeans` es minimizar la suma de las distancias euclidianas al cuadrado. La función `pam` es más robusta porque ésta minimiza una suma de disimilaridades no cuadradas. Además, a diferencia de `kmeans`, `pam` no necesita de una elección aleatoria de puntos iniciales para el centro del cluster.

La sintaxis para llamar a la función `pam` es la siguiente:

```
pam(x, k, diss=F, metric="euclidean", stand=F)
```

Los argumentos de la función tienen los siguientes significados:

- `x` y `diss`: el argumento `x` contiene el input de datos. `diss` le indica a `pam` cuando `x` debe ser interpretado como una matriz de datos (`diss=F`) o como un vector que representa una matriz de disimilaridad (`diss=T`), siendo la opción por

⁶Cuando el input consta de una matriz de datos $n \times p$, `pam` primero la convertirá en una matriz de disimilaridad

defecto la matriz de datos. En este caso, todas las variables deben ser numéricas, de lo contrario la función `daisy` debe ser usada para calcular las disimilaridades. Cuando una matriz es dada como input (`diss=F`), éste es tratado como un objeto de la clase `dissimilarity`. Sin embargo, `pam` también aceptará disimilaridades producidas con la función `dist`, o con cualquier otra que pueda ser interpretada como una matriz de disimilaridad.

- `k`: el número de cluster deseado.
- `metric` y `stand`: estos argumentos son utilizados únicamente cuando `diss=F`. `metric` es una cadena de caracteres que especifica la métrica a ser utilizada para calcular las disimilaridades. Las opciones disponibles son “euclidean” y “manhattan”. Cuando (`stand=T`), los valores en `x` son estandarizados antes de calcular las disimilaridades.

§ 3.3 Segmentación de Clientes Activos para la Cartera de una Institución Financiera

En esta sección se expondrán los resultados del análisis cluster empleando el método *k-medoids*, mismo que es ejecutado a través del algoritmo PAM en la plataforma R por medio de la función `pam`, descrito en la Sección 3.2 para determinar grupos de clientes con similares valores RFM-Risk. Los clientes fueron segmentados en ocho grupos en términos del período desde la última transacción (Recency), frecuencia de adquisición (Frequency), el total del monto adquirido (Monetary) y el nivel de riesgo que presenta (Risk). Se escogieron $k = 8$ grupos dado que hay $2 \times 2 \times 2$ posibles combinaciones de los input (RFM) que se pueden obtener al asignar como $-$ o como $+$, a cada una de

las variables de acuerdo a los valores promedios de cada grupo como se ilustra en la Figura 3.7. Si el valor de R (F,M,Risk) de un grupo excede el promedio total, entonces el signo + es incluido, caso contrario se incluye el signo -. Por ejemplo, R+F-M-Risk+ representa que los valores promedio tanto de Recency como de Risk, para un grupo particular, están por encima del promedio total, mientras que los valores promedio de Frequency y Monetary están por debajo del promedio total. Aquellos ocho grupos de clientes incluyen a los mejores clientes (los de mayor valor), clientes valiables, compradores, clientes primerizos, clientes desertores, clientes frecuentes, derrochadores y clientes inciertos (los de menor valor).

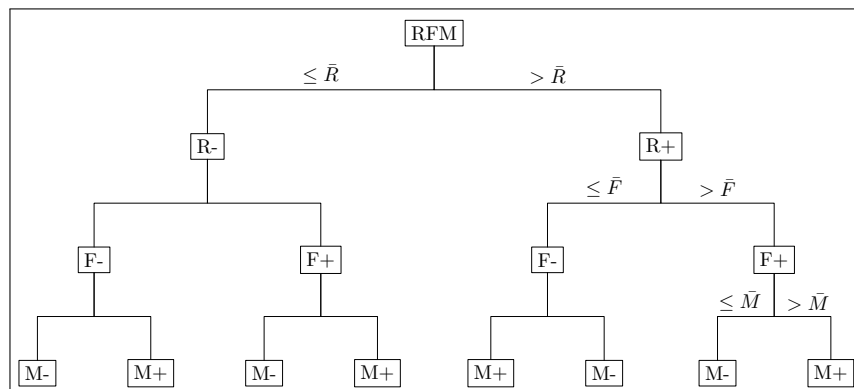


Figura 3.7: Determinación de $k = 8$. Elaboración Propia.

En primer lugar se deben determinar las cuatro variables de interés que conforman el análisis RFM-Risk ⁷ (ver Subsección 3.1.1) a partir del conjunto de datos transaccional de los 19266 clientes. Estos cálculos se llevan a cabo mediante las instrucciones dadas en el Apéndice C. En el Cuadro 3.5 se exponen los resultados para cada uno de los ocho grupos, con el número correspondiente de clientes y su promedio actual de los

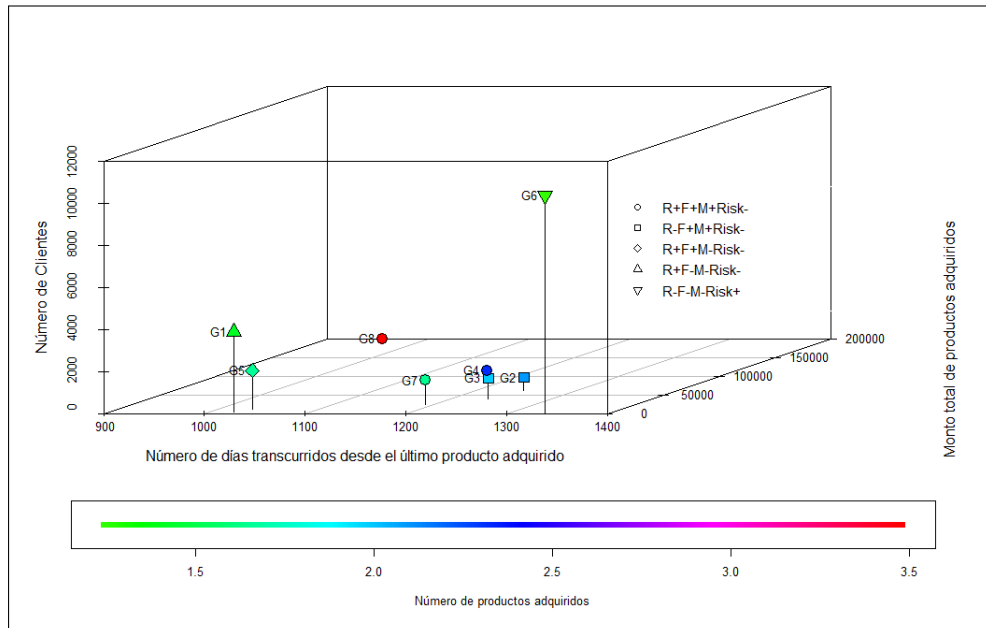
⁷La variable Risk se calculó como la razón de operaciones en mora sobre el total de operaciones adquiridas

Grupo	Tamaño	Recency	Frequency	Monetary	Risk	Patrón
G1	3788	1024.74	1.43	4516.84	25.54%	R+ F- M- Risk-
G2	617	1248.64	2.10	61574.69	22.59%	R- F+ M+ Risk-
G3	969	1237.28	1.99	40163.66	18.57%	R- F+ M+ Risk-
G4	365	1175.25	2.33	94810.97	21.97%	R+ F+ M+ Risk-
G5	1864	1035.78	1.66	10683.60	23.14%	R+ F+ M- Risk-
G6	10354	1337.15	1.24	910.51	55.18%	R- F- M- Risk+
G7	1160	1193.80	1.66	23242.76	20.97%	R+ F+ M+ Risk-
G8	149	964.17	3.48	191707.76	26.73%	R+ F+ M+ Risk-
TOTAL	19266	1224.13	1.45	11081.37	40.46%	

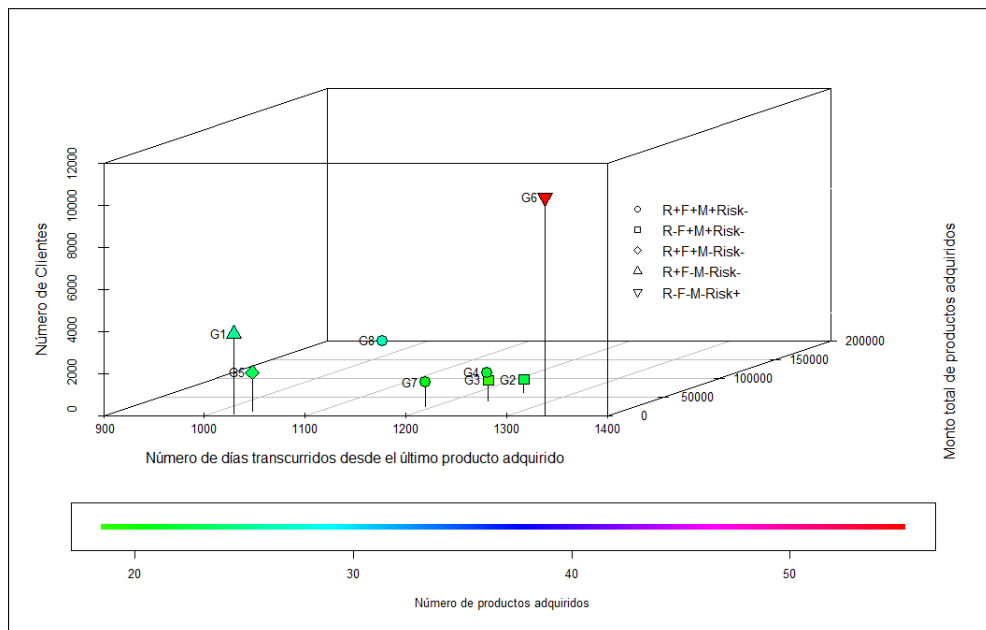
Cuadro 3.5: Segmentación de clientes generada por pam en base a los valores RFM–Risk.

Elaboración Propia

valores R, F, M y Risk. La última fila también muestra el promedio total para todos los clientes. La última columna del Cuadro 3.5 muestra el patrón RFM–Risk para cada cluster. Mientras el grupo G6 contiene al máximo número de clientes (10354 clientes, 53.7%), G8 incluye el mínimo, únicamente 149 clientes (menos del 1%). La Figura 3.8 representa un completo resumen de las características en términos del análisis RFM–Risk que presentan los clientes en la institución financiera. El segmento de clientes G8 contiene a los clientes de mayor valor, porque éste consta de clientes quienes han registrado recientemente operaciones financieras con la mayor frecuencia, el mayor monto y un riesgo bajo. El siguiente grupo de más valor sería el G4, seguido del G7, todos ellos etiquetados como R+ F+ M+ Risk-. El siguiente bloque de grupos los conforman los segmentos G2 y G3, en ese orden etiquetados como R- F+ M+ Risk-. Continuando con el descenso de los segmentos de valor, se tiene el grupo G5 etiquetado como R+ F+ M- Risk-. Y a la cola de la escala de valor, se encuentran los segmentos G1 y G6 en aquel orden, cuyas etiquetas son R+ F- M- Risk- y R- F- M- Risk+, respectivamente. En el siguiente capítulo se expondrá y se aplicará una metodología para recomendar productos financieros a cada uno de los 8 segmentos de la cartera.



(a) Características de R, F y M para los segmentos dados en el Cuadro 3.5



(b) Características de R, M y Risk para los segmentos dados en el Cuadro 3.5

Figura 3.8: Visualización de las características de los clientes. Elaboración Propia.

CAPÍTULO 4

Sistemas Recomendadores

§ 4.1 Introducción a los Sistemas Recomendadores

La venta cruzada es la táctica mediante la cual un vendedor intenta vender productos complementarios a los que consume o pretende consumir un cliente. Pero de ¿qué manera se debe proceder? Lo que se podría hacer es tratar de vender todos los productos para cada cliente. Esta es sin duda una estrategia, pero no necesariamente una efectiva, ya que incluso se estaría incurriendo en un riesgo atribuido a un pobre conocimiento acerca del comportamiento de compra y pago del cliente. En lugar de aquello, se puede inferir y aprender a partir de transacciones previas de todos los clientes y proponer el conjunto correcto de productos que podrían ser bien recibidos y comprados por los clientes. Ahora, la pregunta es ¿de qué manera se debe proceder para promover los productos que puedan ser adquiridos por los clientes? La metodología presentada a continuación es una descripción general sobre el enfoque por el que se optará para responder esta pregunta.

Segmentación de clientes

El sistema recomendador arrojará como resultado el (o los) productos a recomendar a cada usuario de forma personalizada. Las recomendaciones serán realizadas para cada uno de los segmentos de clientes de manera que las preferencias sean lo más

homogéneo posible. Para llevar a cabo la segmentación se utilizó el algoritmo de Partición Alrededor de Medoids a partir de atributos basados en RFM-Risk que proporciona un conocimiento más actual del comportamiento del cliente. Es un método útil para mejorar la segmentación al dividir a los clientes en varios grupos para hacer la recomendación personalizada. Además, la segmentación permitió identificar a los clientes con mayor valor para el banco.

Definición de la estructura para aplicar el sistema recomendador

Esto quiere decir, adecuar las bases de datos disponibles de manera tal que se pueda aplicar el algoritmo recomendador, esto incluye: definir el conjunto de productos a ser expuesto a la estrategia de venta cruzada (por ejemplo, diferentes cupos de tarjetas de crédito, definirán productos distintos).

Seleccionar el sistema recomendador a utilizar

Como se verá más adelante, los sistemas recomendadores basados en filtros colaborativos (y sus variantes) son las técnicas más ampliamente utilizadas en la industria. Los algoritmos para aquellos filtros colaborativos se dividen en dos grupos: los basados en usuarios y los basados en productos. Para el primero de ellos, la suposición es que los usuarios con preferencias similares raquearan (elegirían) productos similares. Mientras que en el segundo algoritmo, la suposición detrás de este enfoque es que el usuario preferirá productos que son similares a los que les gusta. Lógicamente, las recomendaciones que arrojen ambos algoritmos serán distintas. Por lo tanto, la elección de uno de ellos dependerá de la estructura de la información, de la evaluación de rendimiento

de cada uno de ellos y del juicio del analista.

Evaluación del sistema recomendador

El objetivo de la evaluación recae en conocer el futuro rendimiento en un nuevo conjunto de datos. La metodología básicamente consiste en un análisis estadístico y partición de las datos en dos conjuntos: uno de entrenamiento (para aplicar el modelo) y otro de prueba (para evaluar los resultados). Las medidas más habituales para evaluar algoritmos de este tipo son aquellas que miden cuánto la predicción p se acerca al valor observado (en el caso de ratings). En cambio las métricas de clasificación clasifican las recomendaciones como: a) verdaderos positivos, b) verdaderos negativos, c) falsos positivos y d) falsos negativos. A partir de aquellas tasas se generan nuevas medidas tales como: precisión, recall y AUC, para evaluar una lista de recomendación.

En la Figura 4.1 se presenta un esquema del procedimiento de recomendación de productos financieros descrito anteriormente. Todos los procesos que se desarrollan son llevados a cabo en R por su alta versatilidad y facilidad para manipular grandes volúmenes de información. Además en R se encuentra disponible la librería `recommenderlab` el cual proporciona varios algoritmos de recomendación y métricas para su evaluación.

§ 4.2 El Problema de Recomendación

El problema concerniente a la recomendación puede definirse como la estimación de la respuesta de un usuario en relación a nuevos productos, basados en información histórica almacenada en algún sistema, y sugerir a este usuario productos que no posee para el cual la respuesta predicha sea alta. El tipo de respuesta usuario-producto varía

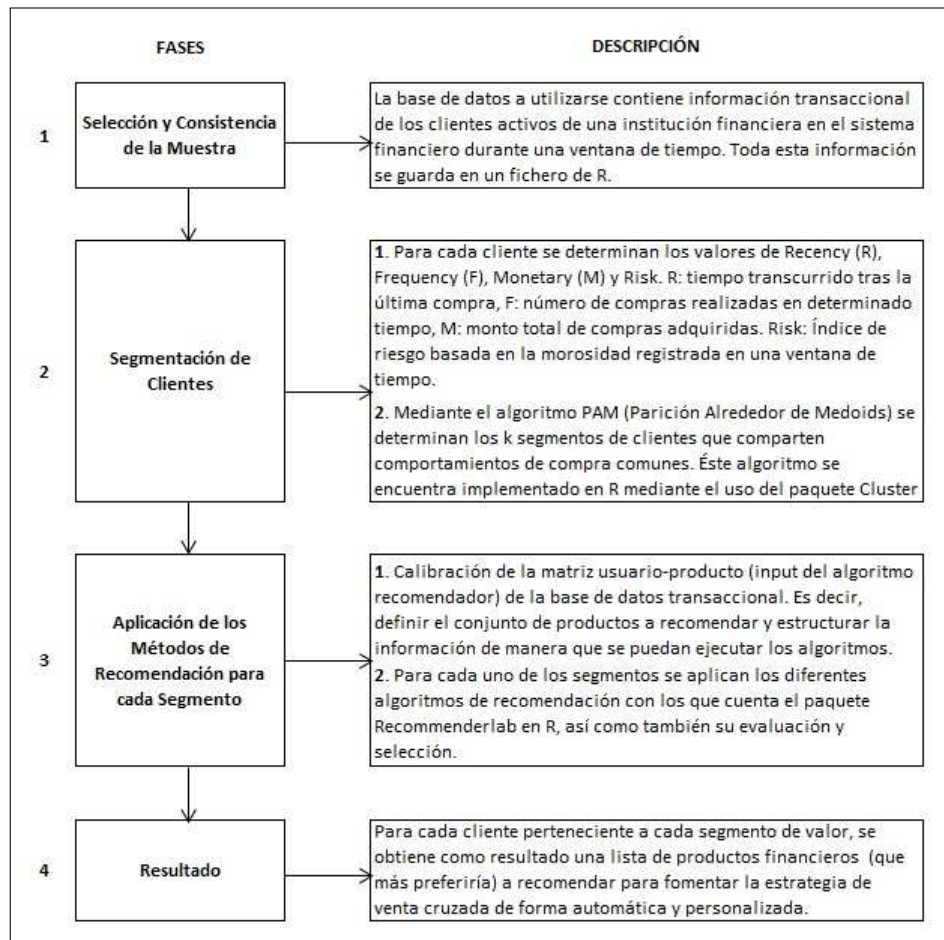


Figura 4.1: Metodología a utilizar para la venta cruzada de productos financieros. Elaboración Propia.

de una aplicación a otra, y recae en una de las tres categorías: escalar, binaria y unaria. Las respuestas escalares, también conocidas como ratings, son numéricas (por ejemplo, 1-5 estrellas) o de valor ordinal (por ejemplo, muy de acuerdo, de acuerdo, neutral, en desacuerdo, muy en desacuerdo) que representa los posibles niveles de apreciación de los usuarios acerca de los productos. Las respuestas binarias, por otro lado, únicamente tiene dos posibles valores codificadas en niveles opuestos de apreciación (por ejemplo, agrado/desagrado o interesado/no interesado). Finalmente, las respuestas unarias

capturan la interacción de un usuario con un producto (por ejemplo, compra, acceso online, etc.) sin dar información explícita sobre la apreciación del usuario para con el producto. Dado que la mayoría de usuarios tienden a interactuar con productos que ellos encuentran interesantes, la respuesta unaria todavía proporciona información útil acerca de las preferencias del usuario.

§ § 4.2.1 Definición del problema de recomendación

El conjunto de usuarios en el sistema será denotado por U y al conjunto de productos por I . Además, se denota por R al conjunto de ratings registrados en el sistema, y por S al conjunto de posibles valores que pueden tomar los ratings (por ejemplo, $S = [1, 5]$ o $S = \{\text{agrado, desagrado}\}$). También, se supone que un usuario $u \in U$ solo puede registrar más que un rating para un producto particular $i \in I$, y este rating es denotado por r_{ui} . Para identificar el subconjunto de usuarios que han ranqueado un producto i , se usa la notación U_i . De igual forma, I_u representa el subconjunto de productos que han sido ranqueados por un usuario u . Finalmente, los productos que han sido ranqueados por dos usuarios u y v es denotado por I_{uv} . De forma similar, U_{ij} denota el conjunto de usuarios que han ranqueado los productos i y j .

Dos de los problemas más importantes asociados con sistemas recomendadores son los problemas de recomendar el mejor producto o recomendar una lista con los mejores N productos. El primer problema consiste en encontrar, para un usuario particular, el nuevo producto $i \in I \setminus I_u$ para el cual el usuario u probablemente esté más interesado. Cuando los ratings están disponibles, esta tarea es comúnmente definida como un problema de clasificación o regresión donde la meta es construir una función $f : U \times I \rightarrow$

S que prediga el rating $f(u, i)$ de un usuario u para un nuevo producto i . Entonces, ésta función es usada para recomendar al usuario objetivo u_a un producto i^* para el cual el rating estimado tiene el valor más alto:

$$i^* = \max_{j \in I \setminus I_u} f(u_a, j).$$

En determinadas ocasiones los ratings no se encuentran disponibles, como es el caso donde únicamente se conoce la lista de productos comprados por cada usuario. En tales circunstancias, el problema de encontrar el mejor producto es usualmente transformado en la tarea de recomendar a un usuario objetivo u_a una lista $L(u_a)$ que contiene N items probablemente de interés para el usuario.

§ 4.3 El Enfoque de Recomendación Basado en Filtros Colaborativos

El enfoque de filtros colaborativos a diferencia de otros métodos como el basado en contenidos (véase Ricci et al. [2011]), usa los ratings del usuario u así como también los ratings de otros usuarios en el sistema para realizar una recomendación. La idea detrás de los algoritmos es que el rating de u para un nuevo producto i sea probablemente similar a la de otro usuario v , si u y v han ranqueado otros productos de manera similar. De igual manera, u posiblemente ranquee dos productos i y j de manera similar, si otros usuarios tienen ratings similares a aquellos dos productos.

Los métodos de filtrado colaborativo pueden ser agrupados en dos clases generales: los métodos basados en *vecindades* y los basados en *modelos*. En el método basado en vecindades, los ratings almacenados en el sistema son usados directamente para predecir los ratings de nuevos productos. Esto puede hacerse de dos maneras, conocidas

como recomendación basada en *usuarios* y basada en *productos*. Los sistemas basados en usuarios evalúan el interés del usuario u para un producto i usando los ratings para éste producto de otros usuarios, llamados *vecinos*, que tienen patrones de ratings similares. Los vecinos del usuario u son los usuarios v cuyos ratings en los productos ranqueados por u y v (es decir, I_{uv}), están más correlacionados a aquellos de u . Por otro lado, el enfoque basado en productos predice el rating de un usuario u para un producto i basado en los ratings de u para productos similares a i . En tal enfoque, dos productos son similares si varios usuarios del sistema han ranqueado aquellos productos de manera similar.

En contraste a los sistemas basados en vecindades, los cuales usan directamente los ratings almacenados para la predicción, el enfoque basado en modelos usan aquellos ratings para aprender un modelo predictivo. La idea general es modelar las interacciones usuario-producto con factores que representen las características latentes de los usuarios y productos en el sistema, como la clase de preferencia de los usuarios y la clase de categorías de los productos. Este modelo es entonces entrenado usando la data disponible, y luego usado para predecir los ratings de usuarios para nuevos productos.

Ventajas del Enfoque de Vecindades

Las principales ventajas de los métodos basados en vecindades son:

- **Simplicidad** Los métodos basados en vecindades son intuitivos y relativamente simples de implementar. En su forma más simple, únicamente se requiere un parámetro (el número de vecinos usados en la predicción).
 - **Justificación** Tales métodos también proporcionan una concisa e intuitiva jus-
-

tificación para el cálculo de las predicciones. Por ejemplo, en una recomendación basada en artículos, la lista de los artículos vecinos, así como también los ratings dados por el usuario para aquellos artículos, pueden ser presentados al usuario como una justificación para la recomendación. Esto puede ayudar al usuario a entender mejor la recomendación y su relevancia, y puede servir como base para un sistema interactivo donde los usuarios pueden seleccionar los vecinos para los cuales una gran importancia debe ser dada en la recomendación.

- **Eficiencia** Uno de los puntos fuertes de los sistemas basados en vecindades es su eficiencia. A diferencia de los sistemas basados en modelos, ellos no requieren una costosa fase de entrenamiento, las cuales necesitan ser llevadas a cabo a intervalos frecuentes en grandes aplicaciones comerciales. Mientras que la fase de recomendación es usualmente más costosa para los métodos basados en modelos, los vecinos más cercanos pueden ser pre calculados fuera de línea, proporcionando recomendaciones casi instantáneas. Además, el almacenar aquellos vecinos más cercanos requiere muy poca memoria, haciendo que tales enfoques sean aplicables a gran escala.
 - **Estabilidad** Otra propiedad útil de un sistema recomendador basado en este enfoque es que ellos no son afectados por la inclusión constante de usuarios, artículos y ratings, los cuales son típicamente observados en grandes aplicaciones comerciales. Por ejemplo, una vez que las similitudes de los artículos han sido calculadas, un sistema basado en artículos puede ser leído para hacer las recomendaciones a los nuevos usuarios sin tener que re entrenar el sistema. Además, una vez que se hayan ingresado los ratings para nuevos artículos, solo las similitudes
-

entre estos artículos y los ya almacenados en el sistema necesitan ser calculados.

§ § 4.3.1 Recomendación Basada en el Usuario

Los métodos de recomendación de vecindades basado en el usuario predicen el rating r_{ui} de un usuario u para un nuevo artículo i usando los ratings dados para i por los usuarios más similares a u , llamados los vecinos más cercanos. Supongamos que tenemos para cada usuario $v \neq u$ un valor w_{uv} que representa la similaridad de preferencia entre u y v . Los k vecinos más cercanos de u , denotado por $\mathcal{N}(u)$, son los k usuarios v con la similaridad más alta w_{uv} a u . Sin embargo, únicamente los usuarios quienes a ranqueado el artículo i pueden ser usados en la predicción de r_{ui} , y en realidad solo se consideran los k usuarios más similares a u que *han ranqueado* i . Al conjunto de vecinos se lo denotará como $\mathcal{N}_i(u)$. El rating r_{ui} puede ser estimado como el rating promedio dado a i por aquellos vecinos:

$$\hat{r}_{ui} = \frac{1}{|\mathcal{N}_i(u)|} \sum_{v \in \mathcal{N}_i(u)} r_{vi}. \quad (4.1)$$

Uno de los problemas de la estimación de los rating con la Ecuación 4.1 es que no toma en cuenta el hecho que los vecinos pueden tener diferentes niveles de similaridad. una solución común para este problema es ponderar la contribución de cada vecino por su similaridad a u . Sin embargo, si aquellas ponderaciones no suman uno, los ratings predichos podrían estar fuera del rango de los valores permitidos. Consecuentemente, es recomendable normalizar aquellos pesos, tal que los ratings predichos se conviertan en

$$\hat{r}_{ui} = \frac{\sum_{v \in \mathcal{N}_i(u)} w_{uv} r_{vi}}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|}. \quad (4.2)$$

En el denominador de la Ecuación 4.2, $|w_{uv}|$ es usado en lugar de w_{uv} porque las ponderaciones negativas pueden producir ratings fuera del rango permitido. También, w_{uv} puede ser reemplazado por w_{uv}^α , donde $\alpha > 0$ es un factor de amplificación. Cuando $\alpha > 1$, como en la mayoría de los casos empleados, una gran importancia es dada a los vecinos que son los más cercanos a u . La Ecuación 4.2 también tiene un defecto importante, ésta no considera el hecho que aquellos usuarios pueden usar diferentes valores de ratings para cuantificar el mismo nivel de apreciación para un item. Por ejemplo, un usuario puede dar el valor de rating más alto solo para unos pocos artículos destacados, mientras que es menos difícil que pueda dar este valor a la mayoría de los items que le gusta. Este problema es usualmente resuelto al convertir los ratings de los vecinos r_{vi} a los normalizados $h(r_{vi})$, dando como resultado la siguiente predicción:

$$\hat{r}_{ui} = h^{-1} \left(\frac{\sum_{\mathcal{N}_i(u)} w_{uv} h(r_{vi})}{\sum_{\mathcal{N}_i(u)} |w_{uv}|} \right). \quad (4.3)$$

Nótese que los ratings predichos pueden volver a la escala original, por eso la presencia de h^{-1} en la Ecuación 4.3.

Clasificación Basada en el Usuario

El enfoque de predicción descrito anteriormente, donde los ratings predichos son calculados como un promedio ponderado de los ratings de los vecinos, esencialmente resuelve un problema de *regresión*. La *clasificación* basada en vecindades, por otro lado, encuentra el rating más probable dado por el usuario u al item i , al tener los vecinos más cercanos de u un voto en este valor. El voto v_{ir} dado por los k vecinos más cercanos de u para el rating $r \in \mathcal{S}$ puede ser obtenido como la suma de los pesos de similaridad

de los vecinos que han dado este rating a i :

$$v_{ir} = \sum_{v \in \mathcal{N}_i(u)} \delta(r_{vi} = r) w_{uv}, \quad (4.4)$$

donde $\delta(r_{vi} = r)$ es 1 si $r_{vi} = r$ y 0 caso contrario. Una vez que esto se ha calculado para todo valor posible de rating, el rating predicho es simplemente el valor r para el cual v_{ir} es el más grande.

Un método de clasificación que considera ratings normalizados también puede ser definido. Sea \mathcal{S}' el conjunto de los valores normalizados posibles (que puede requerir discretización), el rating predicho se obtiene como:

$$\hat{r}_{ui} = h^{-1} \left(\max_{r \in \mathcal{S}'} \sum_{v \in \mathcal{N}_i(u)} \delta(r_{vi} = r) w_{uv} \right). \quad (4.5)$$

§ § 4.3.2 Recomendación Basada en Productos

Mientras que los métodos basados en el usuario cuentan con la opinión del gusto de los usuarios para predecir el rating, los enfoques basados en productos buscan en los ratings dados por productos similares. Esta idea puede ser formalizada de la siguiente manera. Sea $\mathcal{N}_u(i)$ los productos ranqueados por el usuario u hacia los productos más similares a i . El rating predicho de u para i se calcula como un promedio ponderado de los ratings dado por u hacia los productos de $\mathcal{N}_u(i)$:

$$\hat{r}_{ui} = \frac{\sum_{i \in \mathcal{N}_u(i)} w_{ij} r_{uj}}{\sum_{j \in \mathcal{N}_u(i)} |w_{ij}|}. \quad (4.6)$$

De nuevo, las diferencias en las escalas de ratings individuales puede ser considerada por la normalización de ratings con un h :

$$\hat{r}_{ui} = h^{-1} \left(\frac{\sum_{i \in \mathcal{N}_u(i)} w_{ij} h(r_{uj})}{\sum_{j \in \mathcal{N}_u(i)} |w_{ij}|} \right). \quad (4.7)$$

Además, se puede definir un enfoque de clasificación basado en productos. En este caso, los productos j ranqueados por el voto del usuario u para el rating es dado para el nuevo producto i , y aquellos votos son ponderados por la similaridad entre i y j . La versión normalizada de este enfoque se expresa de la siguiente manera:

$$\hat{r}_{ui} = h^{-1} \left(\max_{r \in \mathcal{S}'} \sum_{i \in \mathcal{N}_u(i) \delta(h(r_{uj})=r)} w_{ij} \right). \quad (4.8)$$

Componentes de los Métodos Colaborativos

En las secciones previas se mencionó que al decidir entre el método de predicción de rating de regresión y uno de clasificación, así como también el de escoger un enfoque de recomendación basado en el usuario o en el producto, se puede tener un impacto en la precisión, eficiencia y calidad total del sistema recomendador. Adicionalmente a aquellos atributos cruciales, existen tres consideraciones muy importantes en la implementación de un sistema recomendador [Ricci et al. \[2011\]](#):

1. Normalización de los ratings
2. El cálculo de las ponderaciones de similaridad
3. La selección de los vecinos

Normalización de los Ratings

Cuando se asigna un rating a un item, cada usuario tiene su propia escala personal. Incluso si una definición explícita de cada uno de los posibles ratings es proporcionado (por ejemplo, 1=muy de acuerdo, 2=de acuerdo, 3=neutral, etc.), algunos usuarios pueden ser reacios a dar altos/bajos scores a los productos que les agrada/desagrada.

Dos de los esquemas más populares de normalización de rating que han sido propuestos para convertir ratings individuales a una escala más universal, estos son: *mean-centering* y *Z-score*.

Mean-centering

La idea de mean-centering es determinar cuando un rating es positivo o negativo al compararlo al rating medio. En la recomendación basada en el usuario, un rating r_{ui} es transformado a $h(r_{ui})$ al sustraer de r_{ui} el promedio \bar{r}_u de los ratings dados por el usuario u a los productos en \mathcal{I}_u :

$$h(r_{ui}) = r_{ui} - \bar{r}_u.$$

Usando este enfoque, la predicción basada en el usuario de un rating r_{ui} se obtiene como

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in \mathcal{N}_i(u)} w_{uv}(r_{vi} - \bar{r}_v)}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|}. \quad (4.9)$$

De la misma forma, la normalización centrada en la media por producto de r_{ui} está dado por

$$h(r_{ui}) - \bar{r}_i,$$

donde \bar{r}_i corresponde al rating promedio dado al producto i por el usuario en \mathcal{U}_i . Esta técnica de normalización es usada con más frecuencia en la recomendación basada en productos, donde un rating r_{ui} es predicho como:

$$\hat{r}_{ui} = \bar{r}_i + \frac{\sum_{j \in \mathcal{N}_u(i)} w_{ij}(r_{uj} - \bar{r}_j)}{\sum_{j \in \mathcal{N}_u(i)} |w_{ij}|}. \quad (4.10)$$

Una propiedad interesante de mean-centering es que se puede observar de manera correcta si la apreciación de un usuario respecto a algún producto es positiva o negativa al examinar el signo del rating normalizado. Además, el módulo de este rating da el nivel de preferencia del usuario hacia el producto.

Normalización Z-score

Si se consideran dos usuarios, A y B , ambos con un rating promedio de 3. Además, supongamos que los ratings de A alternan entre 1 y 5, mientras que los de B son siempre 3. Un rating de 5 dado a un producto por B es más excepcional que el mismo rating dado por A , y así, refleja una apreciación mucho más grande para este producto. Mientras mean-centering elimina la compensación causada por las percepciones diferentes de un rating promedio, la normalización Z-score también considera el spread en las escalas de rating individual. En los métodos basados en el usuario, la normalización de un rating r_{ui} divide el rating centrado respecto a la media del usuario por la desviación estándar σ_u de los ratings dados por el usuario u :

$$h(r_{ui}) = \frac{r_{ui} - \bar{r}_u}{\sigma_u}.$$

Una predicción, basada en el usuario, del rating r_{ui} usando este enfoque de normalización se obtiene como

$$\hat{r}_{ui} = \bar{r}_u + \sigma_u \frac{\sum_{v \in \mathcal{N}_i(u)} w_{uv} (r_{vi} - \bar{r}_v) / \sigma_u}{\sum_{v \in \mathcal{N}_i(u)} |w_{uv}|}. \quad (4.11)$$

De igual forma, la normalización z-score de r_{ui} en los métodos basados en productos dividen el rating centrado en la media por producto por la desviación estándar de los

ratings dados al producto i :

$$h(r_{ui}) = \frac{r_{ui} - \bar{r}_i}{\sigma_i}.$$

La predicción basada en productos del rating r_{ui} es

$$\hat{r}_{ui} = \bar{r}_i + \sigma_i \frac{\sum_{j \in \mathcal{N}_u(i)} w_{ij} (r_{uj} - \bar{r}_j) / \sigma_j}{\sum_{j \in \mathcal{N}_u(i)} |w_{ij}|}. \quad (4.12)$$

Cálculos de los pesos de similaridad

Los pesos de similaridad juegan un doble papel en los métodos de recomendación colaborativos: 1) permiten la selección de los verdaderos vecinos cuyos ratings serán usados en la predicción, y 2) proporcionan los medios para dar más o menos importancia a aquellos vecinos en la predicción. El cálculo de los pesos de similaridad es uno de los aspectos más cruciales en la construcción de un sistema recomendador basado en filtros colaborativos, ya que esto puede tener un impacto significativo en su precisión y rendimiento.

Similaridad Basada en la Correlación

Una medida de similaridad entre dos objetos a y b , consiste en la representación de aquellos objetos en la forma de dos vectores \mathbf{x}_a y \mathbf{x}_b y calcular el *coseno* entre aquellos vectores:

$$\cos(\mathbf{x}_a, \mathbf{x}_b) = \frac{\mathbf{x}_a \mathbf{x}_b}{\|\mathbf{x}_a\| \|\mathbf{x}_b\|}.$$

En el contexto de la recomendación por productos, esta medida puede ser empleada para calcular la similaridades de los usuarios al considerar un usuario u como un vector

$\mathbf{x}_u \in \mathbb{R}^{|I|}$, donde $\mathbf{x}_{ui} = r_{ui}$ si el usuario u ha ranqueado el producto i , y 0 caso contrario.

La similaridad entre dos usuarios u y v puede ser calculado como

$$CV(u, v) = \cos(\mathbf{x}_u, \mathbf{x}_v) = \frac{\sum_{i \in \mathcal{I}_{uv}} r_{ui} r_{vi}}{\sqrt{\sum_{i \in \mathcal{I}_u} r_{ui}^2 \sum_{j \in \mathcal{I}_v} r_{vj}^2}}, \quad (4.13)$$

donde \mathcal{I}_{uv} , una vez más, denota los productos ranqueados tanto por u como de v .

Un problema con esta medida es que ésta no considera las diferencias en la media y varianza de los ratings hechos por el usuario u y v . Una medida popular que compara los ratings donde los efectos de la media y varianza han sido extraídas es la similaridad de *Pearson*:

$$PC(u, v) = \frac{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \bar{u})(r_{vi} - \bar{v})}{\sqrt{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - \bar{u})^2 \sum_{i \in \mathcal{I}_{uv}} (r_{vi} - \bar{v})^2}}. \quad (4.14)$$

La misma idea puede ser usada para obtener similaridades entre dos productos i y j , esta vez al comparar los ratings hechos por los usuarios que han ranqueado ambos productos:

$$PC(i, j) = \frac{\sum_{u \in \mathcal{U}_{ij}} (r_{ui} - \bar{i})(r_{uj} - \bar{j})}{\sqrt{\sum_{u \in \mathcal{U}_{ij}} (r_{ui} - \bar{i})^2 \sum_{u \in \mathcal{U}_{ij}} (r_{uj} - \bar{j})^2}}. \quad (4.15)$$

Mientras el signo de una similaridad indica cuando la correlación es directa o inversa, su magnitud representa la fuerza de la correlación.

Otras medidas de similaridad

Otras medidas han sido propuestas para calcular la similaridad entre los usuarios o productos. Uno de ellos es la *Diferencia en Media Cuadrática* (DMC), la cual evalúa la similaridad entre los usuarios u y v como la inversa de la diferencia cuadrada promedio

entre los ratings dados por u y v en los mismos productos:

$$\text{DMC}(u, v) = \frac{|\mathcal{I}_{uv}|}{\sum_{i \in \mathcal{I}_{uv}} (r_{ui} - r_{vi})^2}. \quad (4.16)$$

Una desventaja de la similaridad DMC, comparada a la similaridad PC, es que ésta no captura las correlaciones negativas entre las preferencias de los usuarios o la apreciación de productos diferentes. Otra bien conocida medida de similaridad es la *Correlación de Spearman* (SC). Mientras PC usa los valores de los ratings directamente, SC en cambio considera el ranking de aquellos ratings. La similaridad SC entre dos usuarios u y v es evaluada como

$$\text{SC}(u, v) = \frac{\sum_{i \in \mathcal{I}_{uv}} (k_{ui} - \bar{k}_u)(k_{vi} - \bar{k}_v)}{\sum_{i \in \mathcal{I}_{uv}} (k_{ui} - \bar{k}_u)^2 \sum_{i \in \mathcal{I}_{uv}} (k_{vi} - \bar{k}_v)^2}, \quad (4.17)$$

donde \bar{k}_u es el rango promedio de productos ranqueados por u (el cual puede diferir de $|\mathcal{I}_u| + 1$ si hay ratings empatados). La ventaja principal de SC es que esta evita el problema de la normalización de los ratings al usar rangos. Por otro lado esta medida no será la mejor cuando el rango de los ratings toma únicamente unos pocos valores posibles, dado que se pueden crear un gran número de ratings empatados. Además, esta medida es usualmente más costosa que PC cuando los ratings necesitan ser ordenados con el fin de calcular su rango.

Selección de los Vecinos

El número de vecinos más cercanos a seleccionar y el criterio usado para esta selección puede también tener un serio impacto en la calidad del sistema recomendador. La selección de los vecinos usados en la recomendación de los productos es realizada normalmente en dos pasos: 1) un filtro global donde únicamente los candidatos más

probables son retenidos, y 2) una pre predicción en la cual se escoge a los mejores candidatos para esta predicción.

Pre Filtro de vecinos

En grandes sistemas recomendadores que tienen millones de usuarios y productos, usualmente no es posible almacenar las similitudes (diferentes de cero) entre cada par de usuarios o productos, debido a la limitación de memoria. El pre filtrado de los vecinos es un paso esencial que hace a los enfoques basados en filtros colaborativos reduzcan el monto de las similitudes a almacenar, y limitar el número de vecinos candidatos a considerar en las predicciones. Hay varias maneras en las cuales esto puede llevarse a cabo:

- **Filtro Top- N .** Para cada usuario o producto, solo una lista de los N vecinos más cercanos y sus respectivas similitudes son retenidas. Para evitar problemas con la eficiencia o precisión, N debe ser escogido cuidadosamente. Así, si N es demasiado grande, un monto excesivo de memoria sería requerido para almacenar la lista de vecinos y la predicción de los ratings sería lenta. Por otro lado, seleccionar un valor demasiado pequeño para N puede reducir la convergencia del método de recomendación, lo cual causa que algunos productos nunca sean recomendados.
 - **Umbral del Filtro.** En lugar de retener un número fijo de los vecinos más cercanos, este enfoque retiene todos los vecinos cuya similitud tiene una magnitud más grande que un umbral dado w_{\min} . Mientras que esto es más flexible que la técnica de filtro previa, como solamente los vecinos más significantes son
-

retenidos, el valor correcto de w_{\min} puede ser difícil de determinar.

- **Filtro Negativo.** En general, las correlaciones de rating negativas son menos fiables que las positivas. Intuitivamente esto se debe a que una correlación fuerte y positiva entre dos usuarios es un buen indicador de que pertenecen a un grupo común. Sin embargo, aunque la correlación negativa puede indicar pertenencia a grupos diferentes, ésta no nos dice que tan diferentes son aquellos grupos, o cuando aquellos grupos son compatibles para otras categorías de productos.

Vecinos en las predicciones

Una vez que se ha calculado una lista de vecinos candidatos para cada usuario o producto, la predicción de los nuevos ratings se la hace normalmente con los k vecinos más cercanos, es decir, los k vecinos cuya similaridad tienen la magnitud más grande. De manera que surge una pregunta importante, qué valor usar para k ? Como se menciona en Ricci et al. [2011]), la precisión de la predicción, para valores crecientes de k , típicamente sigue una función cóncava. Así, cuando el número de vecinos es restringido al usar un k pequeño (por ejemplo, $k < 20$), la precisión de la predicción es normalmente baja. Cuando k se incrementa, más vecinos contribuyen a la predicción y la varianza introducida por vecinos individuales es promediada. Como resultado, la precisión de la predicción mejora. Finalmente, la precisión usualmente decae cuando muchos vecinos son usados en la predicción (por ejemplo, $k > 50$), debido al hecho que las pocas relaciones locales fuertes son diluidas por muchas débiles. Aunque un número de vecinos entre 20 a 50 es más a menudo descrito en la literatura, el valor óptimo de k debería ser determinado por validación cruzada.

Métodos para Reducción de la Dimensionalidad

Los métodos de reducción de la dimensión tratan los problemas de la limitada cobertura y esparcidad al proyectar a los usuarios y productos dentro de un espacio reducido que capture la mayoría de sus características. Como los usuarios y productos son comparados en este subespacio denso de características de alto nivel, en lugar del espacio de ratings, se podrían descubrir relaciones más significativas. En particular, se puede encontrar una relación entre dos usuarios, incluso cuando aquellos usuarios han ranqueado productos diferentes. Como resultado, tales métodos son generalmente menos sensitivos a datos sparse. Hay esencialmente dos maneras por las cuales la reducción de la dimensión puede ser usada para mejorar los sistemas recomendadores: 1) descomposición de una matriz de *rating* usuario producto y 2) descomposición de una matriz de *similaridad* sparse.

Descomposición de la Matriz de Rating

Un enfoque popular para llevar a cabo la reducción de la dimensionalidad para la recomendación de productos es el *Índice Semántico Latente* (ISL). En este enfoque, la matriz de rating usuario producto R , $|\mathcal{U}| \times |\mathcal{I}|$, de rango n es aproximada por la matriz $\hat{R} = PQ^T$ de rango $k < n$, donde P es una matriz $|\mathcal{U}| \times k$ de factores de *usuarios* y Q una matriz $|\mathcal{I}| \times k$ de factores de *productos*. Intuitivamente, la u -ésima fila de P , $\mathbf{p}_u \in \mathbb{R}^k$, representa las coordenadas del usuario u proyectado en el espacio latente de dimensión k . De igual manera, la i -ésima fila de Q , $\mathbf{q}_i \in \mathbb{R}^k$ puede verse como las coordenadas del producto i en este espacio latente. Las matrices P y Q se encuentran normalmente al minimizar el error de reconstrucción definido con la norma

de Frobenius al cuadrado:

$$\begin{aligned} \text{err}(P, Q) &= \|R - PQ^T\|_F^2 \\ &= \sum_{u,i} \left(r_{ui} - \mathbf{p}_u \mathbf{q}_i^T \right)^2. \end{aligned}$$

Minimizar este error es equivalente a encontrar la Descomposición de Valor Singular (SVD) de R :

$$R = U\Sigma V^T,$$

donde U es la matriz $|\mathcal{U}| \times n$ de vectores singulares izquierdos, V es la matriz $|\mathcal{I}| \times n$ de vectores singulares derechos y Σ es la matriz diagonal $n \times n$ de valores singulares. Si se denota por Σ_k , U_k y V_k las matrices obtenidas al seleccionar el subconjunto que contiene los k valores singulares más altos y sus correspondientes vectores singulares, entonces las matrices de factores tanto para usuarios como productos corresponden a $P = U_k \Sigma_k^{1/2}$ y $Q = V_k \Sigma_k^{1/2}$ respectivamente. Una vez que se han obtenido P y Q , la predicción de un rating r_{ui} es:

$$r_{ui} = \mathbf{p}_u \mathbf{q}_i^T.$$

Hay, sin embargo, un problema mayor con la aplicación de SVD a la matriz de rating R : la mayoría de valores de r_{ui} de R están indefinidos, dado que podría no haber un rating dado a i por u . Aunque es posible asignar un valor por defecto a r_{ui} , como se mencionó anteriormente, esto podría introducir un sesgo en los datos. Más importante aun, esto podría hacer que una gran matriz R sea densa y, consecuentemente, la descomposición SVD de R no sería posible. La solución común a este problema es que P y Q usen

únicamente los ratings conocidos:

$$\text{err}(P, Q) = \sum_{r_{ui} \in R} (r_{ui} - \mathbf{p}_u \mathbf{q}_i^T)^2 + \lambda (\|\mathbf{p}_u\|^2 + \|\mathbf{q}_i\|^2), \quad (4.18)$$

donde λ es un parámetro que controla el nivel de regularización. En recomendadores basados en filtros colaborativos, se puede usar el mismo principio para calcular la similitud entre usuarios o productos en el espacio latente. Específicamente esto se hace al resolver el siguiente problema:

$$\text{err}(P, Q) = \sum_{r_{ui} \in R} (z_{ui} - \mathbf{p}_u \mathbf{q}_i^T)^2 \quad (4.19)$$

$$\text{sujeto a:} \quad (4.20)$$

$$\|\mathbf{p}_u\| = 1, \quad \forall u \in \mathcal{U}, \quad \|\mathbf{q}_i\| = 1, \quad \forall i \in \mathcal{I}, \quad (4.21)$$

donde z_{ui} es el rating centrado en la media r_{ui} normalizado al rango $[-1, 1]$. Por ejemplo, si r_{\min} y r_{\max} son los valores más altos y más pequeños en los ratings originales,

$$z_{ui} = \frac{r_{ui} - \bar{r}_u}{r_{\max} - r_{\min}}.$$

Este problema corresponde a encontrar, para cada usuario u y producto i , las coordenadas en la superficie de la esfera unitaria de dimensión k tal que u de un alto rating a i si sus coordenadas son lo suficientemente cercanas a la superficie. Si los usuarios u y v están cercanos en la superficie, entonces ellos darán ratings similares a los mismos productos, y así, la similitud entre aquellos usuarios puede ser calculada como

$$w_{uv} = \mathbf{p}_u \mathbf{p}_v^T.$$

De forma similar, la similitud entre dos productos i y j se obtiene como

$$w_{ij} = \mathbf{q}_i \mathbf{q}_j^T.$$

Descomposición de la Matriz de Similaridad

El principio para este segundo enfoque de reducción de la dimensión es el mismo que el analizado previamente: descomponer una matriz en sus factores principales que representan la proyección de los usuarios o productos en el espacio latente. Sin embargo, en lugar de descomponer la matriz de rating, una matriz de similaridad sparse es descompuesta. Sea W una matriz simétrica de rango n que representa ya sea a las similaridades entre usuarios o productos. Para simplificar la representación, consideremos el primer caso. De nuevo, lo que se desea es aproximar W con una matriz $\hat{W} = PP^T$ de rango $k < n$ al minimizar el siguiente objetivo:

$$\begin{aligned} \text{err}(P) &= \|R - PP^T\|_F^2 \\ &= \sum_{u,v} (w_{uv} - \mathbf{p}_u \mathbf{p}_v^T)^2. \end{aligned}$$

La matriz \hat{W} puede verse como una versión compuesta de W la cual es menos sparse que W . Como antes, encontrar la matriz factor P es equivalente a calcular la descomposición de autovalores de W :

$$W = V\Lambda V^T,$$

donde Λ es una matriz diagonal que contiene los $|\mathcal{U}|$ autovalores de W y V es una matriz ortogonal $|\mathcal{U}| \times |\mathcal{U}|$ que contiene los correspondientes autovectores. Sea V_k una matriz formada por los k principales autovectores (normalizados) de W , los cuales corresponden a los ejes del subespacio latente de dimensión k . Las coordenadas $\mathbf{p}_u \in \mathbb{R}^k$ de un usuario u en este subespacio está dado por la u -ésima fila de la matriz $P = V_k \Lambda_k^{1/2}$. Además, las similaridades del usuario calculados en este subespacio latente están dadas

por la matriz

$$W' = PP^T \quad (4.22)$$

$$= V_k \Lambda_k V_k^T. \quad (4.23)$$

Este enfoque fue usado para un sistema recomendador. En este sistema, una matriz W contiene las similaridades PC (Ecuación 4.15) entre un par de productos es descompuesta para obtener el subespacio latente definido por los dos autovectores principales de W . Sea V_2 la matriz que contiene aquellos autovectores. Un usuario u , representado por la u -ésima fila \mathbf{r}_u de la matriz de rating R , es proyectada en el plano definido por V_2 :

$$\mathbf{r}'_u = \mathbf{r}_u V_2.$$

En un paso fuera de línea, los usuarios del sistema son agrupados en el plano usando una técnica de subdivisión recursiva. Entonces, el rating del usuario u para un producto i es evaluado como el rating medio para i hecho por los usuarios que están en el mismo grupo que u .

§ 4.4 Evaluación de Sistemas Recomendadores

La precisión en las predicciones es un factor muy importante, y de hecho, es la propiedad más discutida en la literatura de sistemas recomendadores. Como la base de la vasta mayoría de sistemas recomendadores yace en un motor de predicción, este motor puede predecir las opiniones de los usuarios acerca de productos o la probabilidad de uso. Una suposición básica en un sistema recomendador es que un sistema que proporciona predicciones más precisas serán preferidas por el usuario. Así, muchos investigadores

se dedican a la tarea de encontrar algoritmos que proporcionen mejores predicciones. A continuación se tratarán tres clases de medidas de precisión de las predicciones: 1) medición de la precisión de los ratings predichos, 2) medición de la precisión del uso de las predicciones, y 3) medición de la precisión de los rankings de productos.

§ § 4.4.1 Midiendo la Precisión de los Ratings Predichos

En algunas aplicaciones, lo que se desea es predecir el rating que un usuario podría dar a un producto. En tales casos, lo que se desea es medir la precisión de los ratings predichos por el sistema.

La Raíz del Error en Media Cuadrática (REMC) es quizás la métrica más popular usada para evaluar la precisión de los ratings predichos. El sistema genera los ratings predichos \hat{r}_{ui} para un conjunto de prueba \mathcal{T} del par usuario-item (u, i) para los cuales los verdaderos ratings r_{ui} son desconocidos. Típicamente, r_{ui} son desconocidos porque ellos están ocultos en un experimento fuera de línea, o porque ellos fueron obtenidos a través de un experimento en línea. El REMC entre los ratings predichos y los reales está dado por:

$$\text{REMC} = \sqrt{\frac{1}{|\mathcal{T}|} \sum_{(u,i) \in \mathcal{T}} (\hat{r}_{ui} - r_{ui})^2}. \quad (4.24)$$

Error Absoluto Medio (EAM) es una alternativa popular, dada por

$$\text{EAM} = \sqrt{\frac{1}{|\mathcal{T}|} \sum_{(u,i) \in \mathcal{T}} |\hat{r}_{ui} - r_{ui}|}. \quad (4.25)$$

Comparado a EAM, REMC penaliza desproporcionadamente grandes errores, así que, dado un conjunto de prueba para cuatro productos ocultos, REMC podría preferir un sistema que cometa un error de 2 en tres ratings y 0 en los cuatro a uno que cometa

un error de 3 en un rating y 0 en los otros tres, mientras EAM preferiría el segundo sistema.

REMC Normalizado (NREMC) y **EAM Normalizado** (NEAM) son versiones de REMC y EAM que han sido normalizadas por el rango de los ratings (es decir, $r_{\max} - r_{\min}$). Dado que éstas son simplemente versiones escaladas de REMC y EAM, el ranking resultante de los algoritmos es la misma que el ranking dado por las mediadas no normalizadas.

REMC Promedio y **EAM Promedio** hacen un ajuste para conjuntos de prueba desbalanceados. Por ejemplo, si el conjunto de prueba tiene una distribución desbalanceada de productos, el REMC o el EAM obtenido a partir de éste estar fuertemente influenciado por el error o por muy pocos productos frecuentes. Si se necesita una medida que sea representativa del error de predicción para cualquier producto, es preferible calcular el EAM o el REMC por separado para cada producto y entonces tomar el promedio sobre todos los productos. De manera similar, se puede calcular un promedio REMC/EAM por usuario si el conjunto de prueba tiene una distribución de usuarios desbalanceada y se desea entender el error de predicción para un usuario seleccionado aleatoriamente. REMC y EAM dependen solamente de la magnitud de los errores. En algunas aplicaciones, la semántica de los ratings pueden ser tales que el impacto de un error de predicción no dependa únicamente de su magnitud. En tales dominios puede ser preferible usar una medida de distorsión conveniente $d(\hat{r}, r)$ en lugar de las diferencias cuadradas o absolutas. Por ejemplo, en una aplicación con un sistema de rating de 3, donde 1 significa desagrado, 2 neutral y 3 agrado, y donde se recomendará un producto al usuario que le desagrada es peor que no recomendar

un producto al usuario que si le agrada, una medida de distorsión con $d(3, 1) = 5$, $d(2, 1) = 3$, $d(3, 2) = 3$, $d(1, 2) = 1$, $d(2, 3) = 1$ y $d(1, 3) = 2$ puede ser razonable.

§ § 4.4.2 Midiendo el Uso de las Predicciones

En muchas aplicaciones los sistemas recomendadores no predicen la preferencia de los usuarios sobre ciertos productos, tales como los ratings de las películas, sino más bien tratan de recomendar al usuario productos que ellos podrían usar. Por ejemplo, cuando las películas son añadidas a la cola, Netflix sugiere un conjunto de películas que también pueden ser de interés para el usuario, dado que la película fue incorporada. En este caso, el interés no se centra en conocer cuando el sistema predice apropiadamente los ratings de aquellas películas sino más bien cuando el sistema predice que el usuario añadirá aquellas películas a la cola (uso de los productos).

En una evaluación fuera de línea del uso de la predicción, típicamente se tiene un conjunto de datos que consta de los productos que cada uno de los usuarios han usado. Entonces se selecciona un usuario de prueba, se ocultan algunas de sus selecciones, y se le consulta al recomendador para predecir un conjunto de productos que el usuario usará. Naturalmente se tendrán cuatro posibles resultados para los productos tanto recomendados como los reservados, como se muestra en el Cuadro 4.1.

	Recomendado	No Recomendado
Usado	Verdadero Positivo (vp)	Falso Negativo (fn)
No usado	Falso Positivo (fp)	Verdadero Negativo (vn)

Cuadro 4.1: Clasificación de los posibles resultados de una recomendación de un producto a un usuario

En el caso fuera de línea, dado que los datos generalmente no son agrupados usando

el sistema recomendador bajo evaluación, estamos forzados a asumir que los productos no usados podrían no haberse usados incluso si ellos han sido recomendados, es decir, que éstos son de poco interés o de poca utilidad para el usuario. Desde luego, esta suposición puede ser falsa, tal es el caso cuando el conjunto de productos no usados contiene algún producto interesante que el usuario no seleccionó. Por ejemplo, un usuario no podría haber usado o adquirido un producto porque éste desconocía de su existencia, pero después que la recomendación exponga aquel producto el usuario puede decidir si lo selecciona. En este caso el número de falsos positivos está sobre estimado. Se puede contabilizar el número de ejemplos que caen en cada celda en el Cuadro 4.1 y calcular las siguientes cantidades:

$$\text{Precision} = \frac{\text{Número vp}}{\text{Número vp} + \text{Número fp}}. \quad (4.26)$$

$$\text{Recall (Tasa de Verdaderos Positivos)} = \frac{\text{Número vp}}{\text{Número vp} + \text{Número fn}}. \quad (4.27)$$

$$\text{Falsos Positivos (1-Especificidad)} = \frac{\text{Número fp}}{\text{Número fp} + \text{Número vn}}. \quad (4.28)$$

Generalmente podemos esperar un equilibrio entre aquellas cantidades. Mientras más grande sea una lista de recomendación, se mejorará el recall, pero esto probablemente también reduzca la precisión. En aplicaciones donde el número de recomendaciones que se pueden presentar al usuario están pre ordenadas, la medida de interés más útil es la precision.

En otras aplicaciones donde el número de recomendaciones que son presentadas al usuario no está preordenada, es preferible evaluar los algoritmos sobre un rango de di-

versas listas de recomendación. Así, se pueden construir curvas al comparar la precisión versus el recall, o la tasa de verdaderos positivos versus la tasa de falsos positivos. Las curvas del primer tipo son conocidas simplemente como curvas precisión-recall, mientras que las del segundo tipo son conocidas como curvas ROC. Las curvas precisión-recall enfatizan la proporción de productos recomendados que son preferidos mientras que las curvas ROC enfatizan la proporción de productos que nos son preferidos y que al final fueron recomendados. Se debería seleccionar cuando usar precisión-recall o ROC basándose en la propiedades del dominio y la meta de la aplicación. Además, dado dos algoritmos, se puede calcular un par de tales curvas, una para cada algoritmo. Si una curva domina completamente a la otra curva, la decisión sobre que algoritmo elegir es fácil. Sin embargo, cuando las curvas se intersecan, la decisión es menos obvia, y dependerá de la aplicación e ne cuestión. Medidas que resumen la precisión de la curva ROC tales como la medida F y el área bajo la curva ROC (AUC) son útiles para comparar independientemente de la aplicación, pero cuando se selecciona un algoritmo para usarlo en una tarea particular, es preferible tomar la elección basada en una medida que refleje las necesidades específicas.

§ 4.5 El paquete **Recommenderlab** en R

La librería **Recommenderlab** de R proporciona una infraestructura para desarrollar y probar algoritmos recomendadores para un conjunto de datos que contienen rating de productos o para aquellos que solo representan uso o adquisición de productos (datos binarios). Este paquete tiene incorporado algunos algoritmos básicos y permite la opción que el usuario desarrolle y use sus propios algoritmos.

§ § 4.5.1 Infraestructura de **Recommenderlab**

El paquete usa ciertas estructuras de resumen arrojadas por `ratingMatrix` para proporcionar una interface común para los ratings. `ratingMatrix` implementa muchos métodos que generalmente están disponibles para objetos similares a las matrices. Por ejemplo, se encuentran a disposición las siguientes funciones: `dim()`, `dimnames()`, `colCounts()`, `rowCounts`, `colMeans`, `rowMeans()`, `colSums()` y `rowSums()`. Además, se puede usar la función `sample()` para extraer una muestra de usuarios (filas de la matriz de datos usuario-producto).

En `ratingMatrix` existen dos implementaciones según el tipo de estructura de la matriz de datos usuario-producto, éstas son `realRatingMatrix` y `binaryRatingMatrix`. `realRatingMatrix` implementa una matriz de ratings con elementos numéricos almacenados en un formato sparse definido en el paquete `Matrix`. En cambio `binaryRatingMatrix` implementa una matriz de rating binaria (0-1) usando la implementación de `itemMatrix` definida en el paquete `Arules`. `itemMatrix` únicamente almacena los elementos unitarios e internamente usa una representación sparse a partir del paquete `Matrix`. Con esta clase de estructura, `Recommenderlab` puede ser extendido a otras formas de matrices de rating con diferentes conceptos para un almacenamiento eficiente en el futuro.

La función `Recommender` implementa la estructura de datos para almacenar los modelos de recomendación. La sintaxis de la función es la siguiente:

```
Recommender(data, method, parameter=NULL)
```

Como input toma la matriz de datos usuario-producto almacenado como un objeto de la clase `ratingMatrix` y retorna un objeto de la clase `Recommender`. Una vez que se obtiene un objeto `recommender`, se puede predecir las recomendaciones para un usuario

al usar la siguiente función:

```
predict(object, newdata, n=10, type=c("topNList", "ratings"), ...)
```

`Predict` puede retornar ya sea una lista $\text{top-}N$ (argumento por defecto) o predecir ratings. El parámetro `object` es el objeto `recommender`, `newdata` es el conjunto de datos para el usuario activo (usuario al cual se le está haciendo la recomendación). Para una lista $\text{top-}N$, `n` es el número máximo de productos recomendados en cada lista. En definitiva, la función `predict()` retornará un objeto de clase `topNList` el cual contiene una lista $\text{top-}N$ para cada usuario activo. Para los ratings `n` es ignorado y un objeto de `realRatingMatrix` es retornado. Cada fila contiene los ratings predichos para un usuario activo. Los productos para los cuales existe un rating en `newdata` tiene un NA en lugar de un rating predicho.

Evaluación de una lista $\text{top-}N$

Dada un matriz de rating R , los algoritmos recomendadores son evaluados primariamente al particionar a los usuarios en R dentro de dos conjuntos $\mathcal{U}_e \cup \mathcal{U}_p = \mathcal{U}$. Los usuarios de entrenamiento de \mathcal{U}_e son usados para ensayar el modelo recomendador. Luego cada usuario $u_a \in \mathcal{U}_p$ es tomado como un usuario activo, sin embargo, antes de crear recomendaciones, algunos productos deben ser reservados del perfil r_{u_a} . Al final se mide que tan bien aquellos productos eliminados son predichos por el algoritmo recomendador en su lista $\text{top-}N$. Para determinar como dividir \mathcal{U} en los conjuntos \mathcal{U}_e y \mathcal{U}_p se pueden usar los siguientes enfoques:

- **Splitting** De manera aleatoria se asigna una proporción predefinida de los usuarios al conjunto de entrenamiento y el resto al conjunto de pruebas.
-

- **Bootstrap** Realiza un muestreo con reemplazamientos para crear el conjunto de entrenamiento y entonces usar los usuarios que no están en el conjunto de entrenamiento como el conjunto de prueba.
- **k -fold cross-validation** Se divide \mathcal{U} en k conjuntos (llamados folds) de aproximadamente el mismo tamaño. Entonces se evalúa k veces, siempre usando un campo para la prueba y el resto de campos para el entrenamiento. Los k resultados se promedian. Este enfoque asegura que cada usuario está al menos una vez en el conjunto de prueba y el promedio produce resultados más robustos y con menor error.

Los resultados de la predicción para todos los usuarios de prueba \mathcal{U} pueden ser agregados dentro de una matriz de confusión y deducirse las medidas descritas en la Subsección 4.4.2.

Para evaluar los algoritmos recomendadores el paquete `Recommenderlab` proporciona la infraestructura para crear y mantener esquemas de evaluación almacenados como un objeto de la clase `evaluationScheme` a partir de la matriz de datos de ratings. La siguiente función:

```
evaluationScheme(data, method="split", train=0.9, k=10, given=3)
```

crea el esquema de evaluación a partir de un conjunto de datos usando un método específico. Luego, la función `evaluate()` es usada para evaluar varios algoritmos recomendadores haciendo uso de un esquema de evaluación que da como resultado una lista de evaluación (`evaluationResultsList`) con una entrada (`evaluationResult`) por algoritmo. Cada objeto de `evaluationResult` contiene uno o varios objetos de `confusionMatrix` dependiendo del número de observaciones especificadas en

`evaluationScheme` (por ejemplo, k para la validación cruzada de k campos). Con esta infraestructura varios algoritmos recomendadores pueden ser comparados para un mismo conjunto de datos con una sola línea de código.

CAPÍTULO 5

Resultados

En las circunstancias actuales, donde el comercio y la competencia crece y se acentúa cada día, se hace cada vez más importante considerar las necesidades y deseos de los clientes. Como ya se ha mencionado anteriormente, la tarea de atraer clientes en la industria bancaria es mucho más difícil y costosa que la de mantener a los ya existentes. Por lo tanto, la aplicación de la estrategia de venta de cruzada de productos financieros debe estar enmarcada dentro de la ejecución de un CRM, es decir, implementar un modelo de negocio que vincule las estrategias de venta y promoción de los productos financieros con las necesidades y preferencias de los clientes. De ese modo, se presenta en las secciones siguientes la aplicación y evaluación del sistema recomendador como modelo seleccionado para cada uno de los grupos segmentados en la Sección 3.3, ya que diferentes grupos de clientes prefieren productos diferentes y hasta especiales. Como se expone en el Capítulo 3, el reconocimiento del tipo del cliente es uno de los principales aspectos del negocio y de un CRM. Los algoritmos a implementar y evaluar son los de tipo colaborativo: basados en usuarios y basados en productos. Como se verá más adelante los dos algoritmos presentan resultados similares y son superiores a otros métodos presentes en el paquete `recommenderlab`.

§ 5.1 Análisis del Conjunto de Datos Transaccional de la Cartera

Los sistemas recomendadores son tecnologías que ayudan al negocio a implementar estrategias de marketing uno a uno (personalizadas), entre ellas, la venta cruzada. Éstos “revisan” la historia de las compras realizadas por los clientes (*usuarios*, como se los definió en el Capítulo 4) e identifican los productos deseados por los clientes. El uso de los sistemas recomendadores llevan a ambos, cliente y banco, a la obtención de beneficios. El primer paso para la recomendación de productos es la de considerar la información acerca de la adquisición pasada de productos financieros. Esta información contiene todos los datos en relación a cualquier combinación de productos que han sido adquiridos por cada uno de los grupos de clientes. Con esta información, el sistema recomendador extrae las reglas útiles y puede proponer los productos correctos a los clientes.

El conjunto de datos usado corresponde al historial de productos adquiridos en el sistema financiero para la cartera de clientes perteneciente a la institución financiera y que han sido segmentados en 8 grupos. La información fue adquirida por la institución al buró de crédito con corte Julio de 2011. Esta base describe los productos adquiridos por los clientes en las carteras de Consumo, Tarjetas de Crédito, Vivienda y Microcrédito a partir de Julio 2009, dando un total de 154 tipos distintos de productos (ver Cuadro 5.6) de acuerdo al monto y plazo concedido. Este conjunto de datos transaccional consta de 99104 transacciones registradas con 19266 clientes. El sistema recomendador arrojará como resultado el (o los) productos a recomendar a cada usuario de forma personalizada. Las recomendaciones serán realizadas para cada

uno de los segmentos de clientes de manera que las preferencias sean lo más homogéneo posible. La descripción de los campos de la base de datos se encuentra en la Figura 5.1. Los siguientes cuadros resumen algunos de los campos más importantes descritos en la Figura 5.1.

Tipo de Cuenta

La distribución de este campo se muestra en el siguiente cuadro, donde se aprecia que los créditos destinados al consumo representan casi el 90% de las operaciones registradas.

Comercial	Consumo	Vivienda	Microcrédito	Prendario
3.65%	87.88%	2.84%	5.08%	0.55%

Cuadro 5.1: Resumen: Tipo de cuenta. Elaboración Propia.

NOMBRE CAMPO	DESCRIPCION
bd_fecha	Fecha de corte; hace referencia a la fecha en la cual se está realizando la consulta
bd_cod_tipo_id	(c) Cédula, (R) RUC, (P) Pasaporte
bd_cod_numero_id	10 dígitos para la cédula, 13 para el RUC y para Pasaporte cualquier número de dígitos
bd_des_nombre_cliente	Nombres completos de la persona natural o jurídica
bd_tipo_cuenta	Tipo del producto financiero que tiene el cliente
bd_tipo_operacion	Código del tipo del operación financiera que tiene el cliente
cb_des_tipoperacion_buro	Nombre del tipo del operación financiera que tiene el cliente
bd_sector1	Tipo de la institución que emitió el producto financiero, (R) Real, (IF) Institución Financiera
bd_sector2	Tipo de la institución que emitió el producto financiero, (R) Regulada, (NR) No regulada
bd_sector3	Tipo de la institución que emitió el producto financiero
bd_nombre_entidad	Nombre de la institución que emitió el producto financiero
bd_tipo_riesgo	Tipo de relación que tiene el cliente con el producto financiero, que puede ser TITULAR, GARANTE o CODEUDOR
bd_md_valor_saldo	El valor del saldo del producto financiero a la fecha de corte; para el producto financiero cancelado colocar el valor de cero
bd_md_valor_vencido_mas_nodevenga	El valor del saldo vencido y no devenga intereses del producto financiero a la fecha de corte; para el producto financiero cancelado colocar el valor de cero
bd_md_valor_inicial	El valor de saldo inicial del producto financiero
bd_md_cuota_estimada	El valor de la Cuota que tiene que pagar el cliente cada mes del producto financiero; para el producto financiero cancelado colocar el valor de cero
bd_fecha_inicial	Fecha en la que se abrió el producto financiero
bd_fecha_vencimiento	Fecha en la que vence el producto financiero
bd_fecha_cancelacion	Fecha en la que canceló el producto, en caso de que la operación haya sido cancelado, si el producto sigue vigente debe venir vacío
bd_estado_actual	Estado actual del producto financiero, que puede ser AL DIA o EN MORA; para el producto financiero cancelado colocar NINGUNO

Figura 5.1: Diccionario Campos Base Transaccional. Elaboración Propia

Tipo de Operación

El siguiente cuadro muestra la información desagregada del Cuadro 5.1, es decir, se muestra la distribución de las diferentes operaciones registradas. En cuanto a las operaciones financieras se trata, se observa que las carteras de Tarjetas de Crédito y de Consumo son las más frecuentes, dejando relegadas a las carteras de Microcrédito, Comercial y Vivienda por debajo de operaciones correspondientes al sector real de la economía.

Tipo Operación	Frecuencia
Tarjeta de Crédito	18.8%
Cartera de Consumo	17.9%
Cartera Telefonía Celular	12.7%
Cartera Vestuario	12.3%
Cartera Electrodomésticos	10.6%
Cartera de Comunicaciones	8.8%
Cartera de Microcrédito	3.8%
Cartera Comercial Bancaria	3.5%
Cartera de Vivienda	2.6%
Carteras Restantes	9.0%

Cuadro 5.2: Resumen: Tipo de Operación. Elaboración Propia.

Tipo de Institución

En el siguiente cuadro se muestra la distribución del tipo de institución que emitió el producto financiero y no financiero. En relación a los productos financieros, se observa que la mayoría de éstos han sido otorgados por parte de los Bancos Privados con cerca del 40%, seguido muy de lejos por Cooperativas y demás sociedades financieras.

Tipo Institución	Frecuencia
Bancos Privados Nacionales	39.4%
Cartera Telefonía Celular	12.7%
Cartera Vestuario	12.3%
Cartera Electrodomésticos	10.6%
Cartera de Comunicaciones	8.8%
Cooperativa	3.1%
Sociedades Financieras	2.5%
Instituciones Restantes	10.6%

Cuadro 5.3: Resumen: Tipo de Institución. Elaboración Propia.

Estado de la Cartera

En este último cuadro se muestra el estado de la cartera a la fecha de corte (Julio de 2011). El aspecto rescatable de este cuadro es el hecho que la mayoría de las operaciones ya han sido canceladas y las operaciones financieras que todavía están vigentes no presentan morosidad en su gran mayoría.

Estado	Frecuencia
Canceladas	72.43%
Al día	11.4%
Cartera Castigada	3.12%
En Mora 120 días	1.77%
En Mora 60 días	0.55%
En Mora 30 días	0.40%
En Mora 90 días	0.28%
Dudoso Recaudo	0.05%

Cuadro 5.4: Resumen: Estado Cartera. Elaboración Propia.

A forma de resumen se puede establecer que las principales carteras financieras que prefieren adquirir los clientes de la institución financiera están directamente relacionadas a satisfacer las necesidades de consumo (entre ellas tarjetas de crédito), microcrédito

y vivienda, principalmente a partir de la banca privada. En la siguiente sección se construirá la matriz de productos a recomendar a partir de estas 3 carteras.

§ 5.2 Calibración de la Matriz Usuario-Producto

El proceso para la creación de recomendaciones utilizando `recommenderlab` en R comienza por establecer la matriz de rating usuario producto R . Para el caso presente, se plantearán dos formas distintas de generar la matriz R . Esto se debe a que los datos con los que se dispone provienen de muchas fuentes de información (instituciones financieras) que emiten a su vez muchos productos financieros con distintos montos otorgados y diferentes horizontes de plazo (o vigencia, en el caso de tarjetas de crédito) y su correspondiente cuota de pago¹. Así que una forma de generar el conjunto \mathcal{I} de productos sería a partir de éstas tres variables.

Para generar cada uno de los productos en cada cartera primero se determinan grupos que contengan características de monto, plazo y cuota similares. El código de instrucción descrito en el Anexo C genera los diez grupos² para la cartera de Consumo (el proceso es similar para el resto de carteras).

¹Cuota de pago estimada por parte del buró de crédito

²Se escogió $k = 10$ ya que con este número de grupos se tiene una distribución más uniforme en relación al número de individuos que pertenecen a cada grupo

Producto	Tamaño	Monto	Plazo	Cuota
CON_5	13464	558.72	0.3686582	2.26160
CON_3	9628	1746.59	1.1824069	16.30975
CON_4	7865	3766.45	2.0272220	42.83200
CON_2	5903	7515.06	2.5470402	90.84981
CON_1	4044	12500.00	3.1527147	150.89386
CON_6	1414	17685.00	3.5707834	201.30584
CON_8	1098	35149.96	3.5386381	287.70977
CON_7	167	200000.00	3.4002625	1010.14000
CON_9	4	1701415.00	3.8000000	407.48000
CON_10	2	45000000.00	0.3260274	0.00000

Cuadro 5.5: Características de los grupos para la cartera de consumo. Elaboración Propia.

Luego, para cada grupo (CON_ i) se analiza la distribución conjunta del monto y el plazo concedido en base a la Figura 5.2, por ejemplo se puede concebir como un producto los créditos concedidos con monto entre 8000 y 10000 dólares a con plazos de 3 y 4 y años, es decir, se tendrían 4 productos: 1) crédito de consumo de 8000 a 3 años plazo, 2) crédito de consumo de 8000 a 4 años plazo, 3) crédito de consumo de 10000 a 3 años plazo y 4) crédito de consumo de 10000 a 4 años plazo. El código que genera esta figura para (CON_1) se presenta en el Anexo D.

Realizando el mismo análisis para el resto de grupos CON_ i se determinaron en total 55 productos destinados a la cartera de consumo. De igual forma se determinan los demás productos para el resto de carteras³, el Cuadro 5.6 resume el número de grupos por cartera. A este conjunto de productos se le designa como \mathcal{I}_1 .

³para el caso de tarjetas de crédito, solo se analizó el cupo utilizado

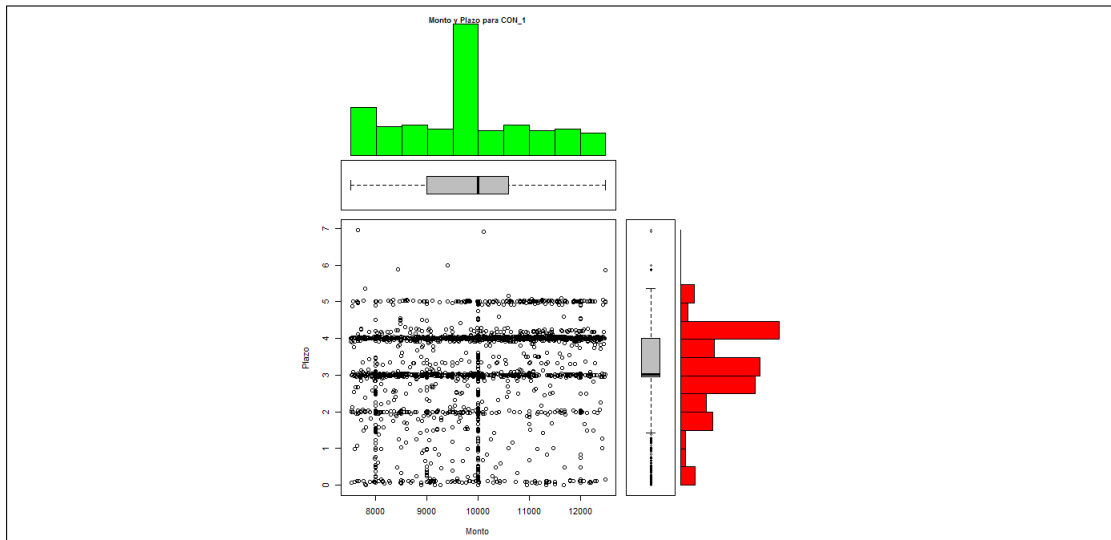


Figura 5.2: Distribución conjunta del monto y plazo concedido para CON_1. Elaboración Propia

Cartera	Número de Productos
Consumo	55
Tarjetas de crédito	31
Microcrédito	51
Vivienda	17
TOTAL	154

Cuadro 5.6: Total de productos determinados. Elaboración Propia.

La segunda forma de estructurar el conjunto de productos es mediante la discretización del monto otorgado en cada cartera por medio de una partición en deciles⁴ mediante la ejecución del código descrito en el Anexo E.

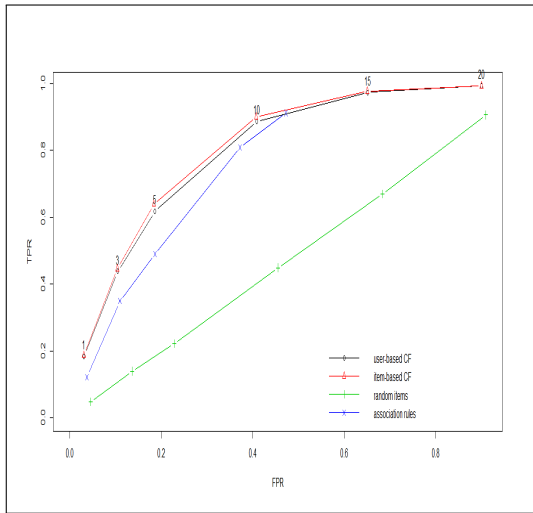
Como resultado se obtuvo un conjunto de productos \mathcal{I}_2 con 25 elementos correspondientes a las carteras de consumo, tarjeta de crédito y de vivienda.

⁴en el caso de tarjetas de crédito se calcularon los quintiles

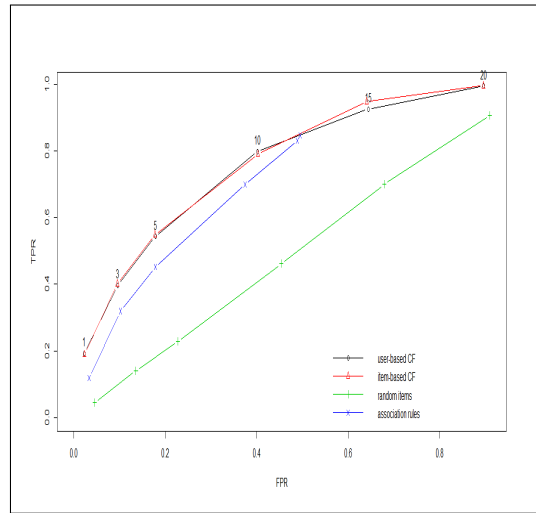
§ 5.3 Aplicación, Selección y Evaluación del Algoritmo Recomendador

Dado que ya se cuenta con el conjunto de productos \mathcal{I} y cada usuario (cliente) está asignado a un grupo, se procede a entrenar y usar el algoritmo recomendador para cada uno de los grupos. El código del Anexo F muestra las instrucciones necesarias para convertir la base de datos Transaccional en la matriz de rating R para los $i \in \mathcal{U}$ clientes (usuarios) y $j \in \mathcal{I}$ productos así como también la ejecución del esquema de evaluación (ver Sección 4.5.1) de los algoritmos aplicados. Este esquema de evaluación tiene como objetivo seleccionar el mejor algoritmo a ser implementado para cada grupo.

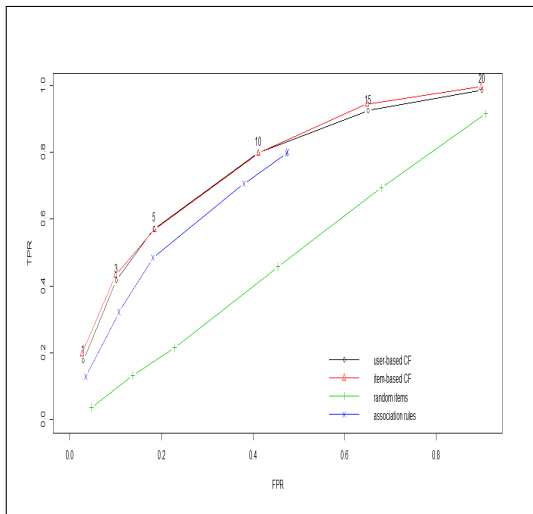
§ § 5.3.1 Selección del algoritmo recomendador



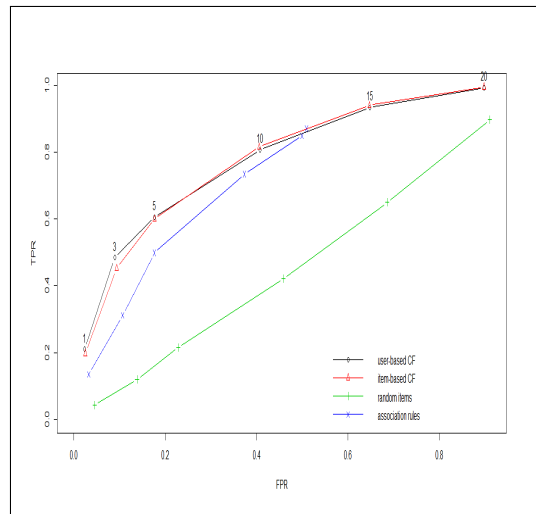
(a) Curva ROC para el grupo 1



(b) Curva ROC para el grupo 2



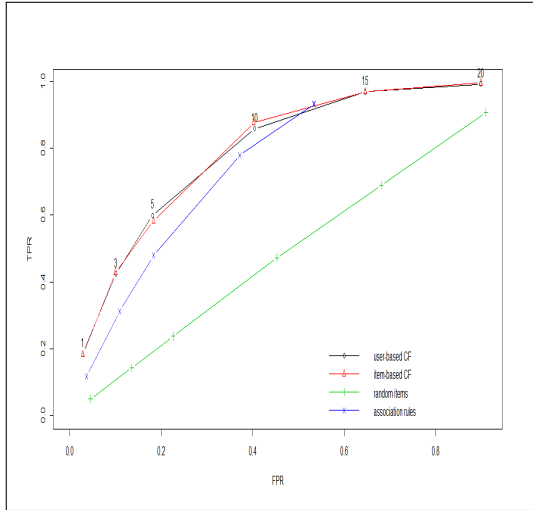
(c) Curva ROC para el grupo 3



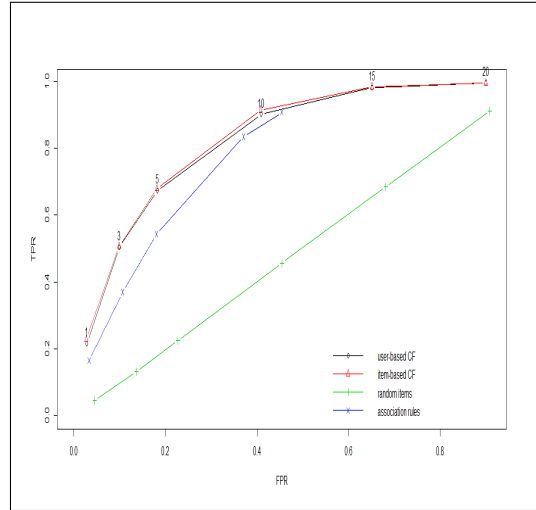
(d) Curva ROC para el grupo 4

Figura 5.3: Resultado del esquema de evaluación para los grupos 1–4 con $\mathcal{I} = \mathcal{I}_2$.

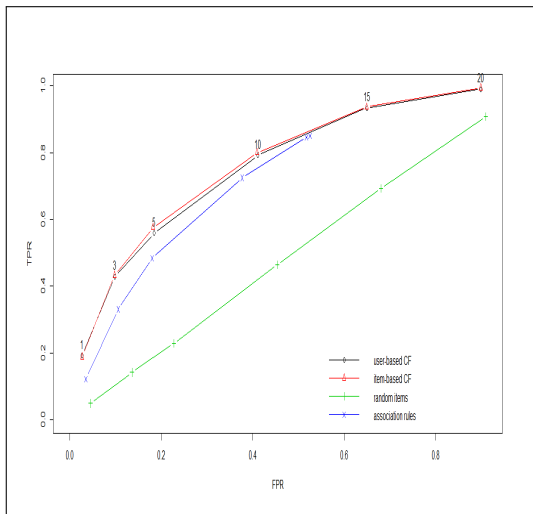
Elaboración Propia.



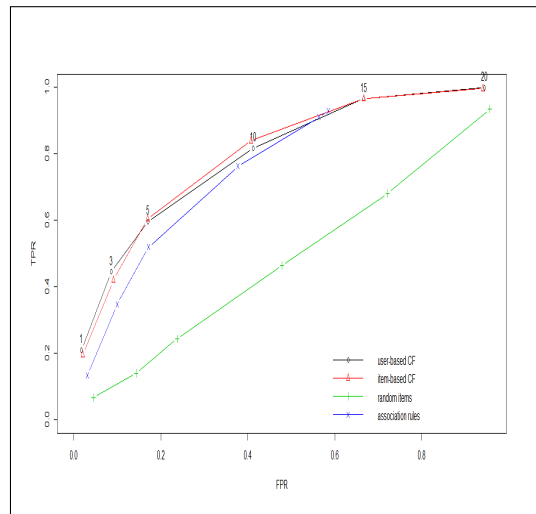
(a) Curva ROC para el grupo 5



(b) Curva ROC para el grupo 6



(c) Curva ROC para el grupo 7

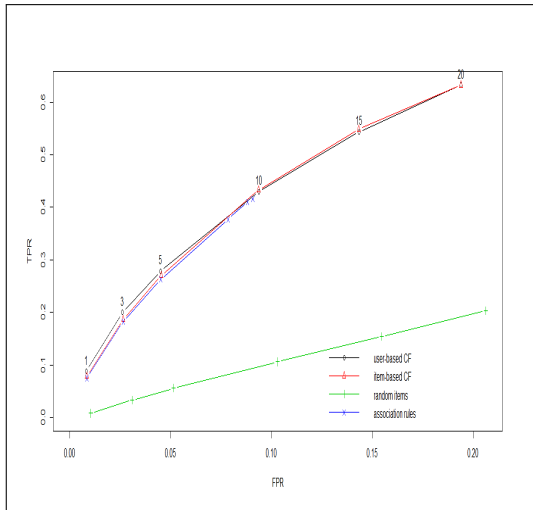


(d) Curva ROC para el grupo 8

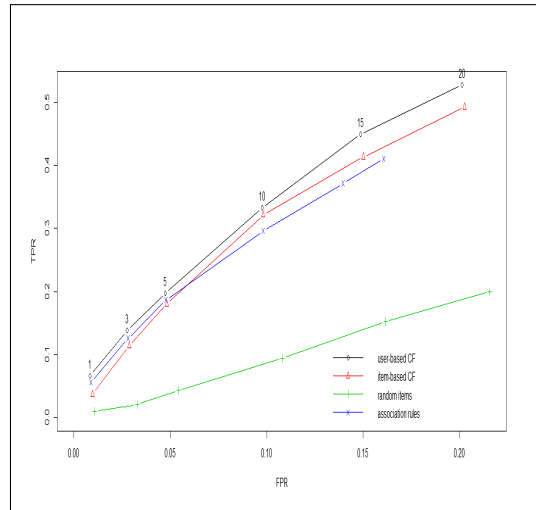
Figura 5.4: Resultado del esquema de evaluación para los grupos 5–8 con $\mathcal{I} = \mathcal{I}_2$.
Elaboración Propia.

Las Figuras (5.3–5.4) muestran las gráficas ROC para el esquema de evaluación descrito haciendo uso del conjunto de productos \mathcal{I}_2 . A partir de éstas se observa que para todos los grupos los métodos basados en filtro colaborativo son superiores al algoritmo basado

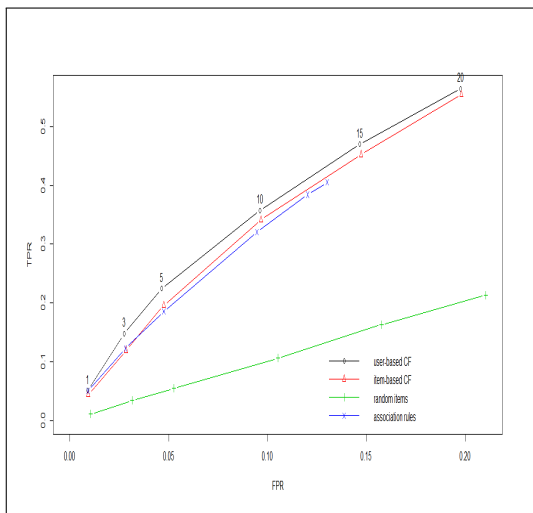
en reglas de asociación y a la recomendación aleatoria, así como también es notoria que la eficiencia de los primeros es bastante significativa, cerca del 80% para todos los casos.



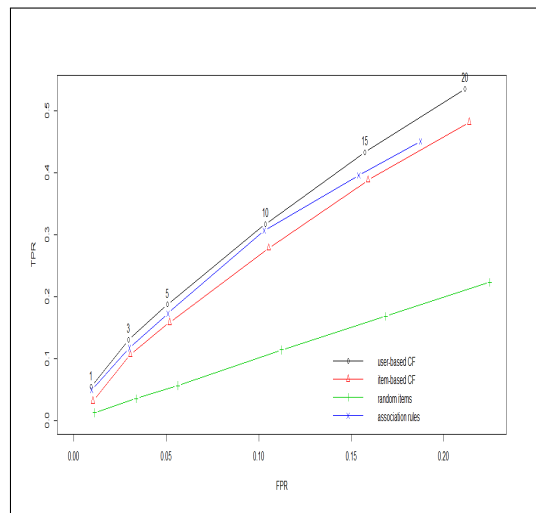
(a) Curva ROC para el grupo 1



(b) Curva ROC para el grupo 2



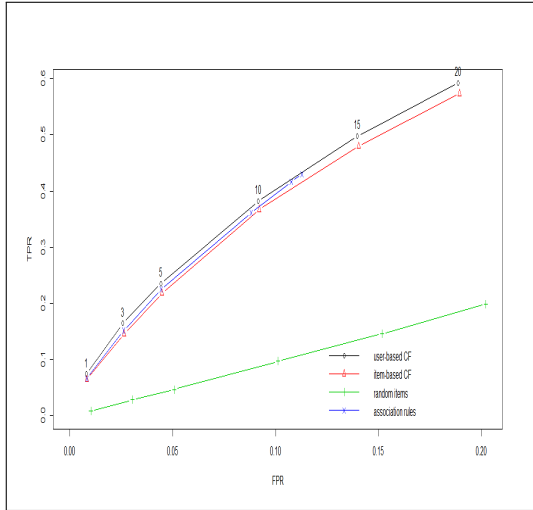
(c) Curva ROC para el grupo 3



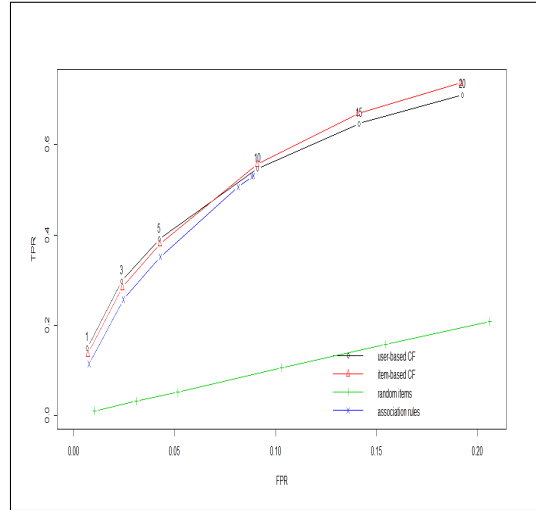
(d) Curva ROC para el grupo 4

Figura 5.5: Resultado del esquema de evaluación para los grupos 1–4 con $\mathcal{I} = \mathcal{I}_1$.

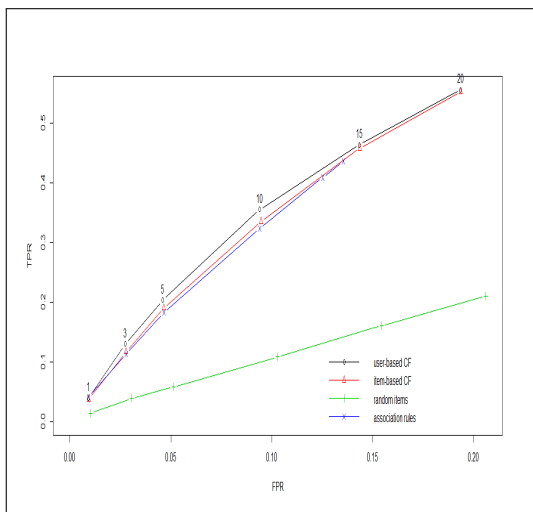
Elaboración Propia.



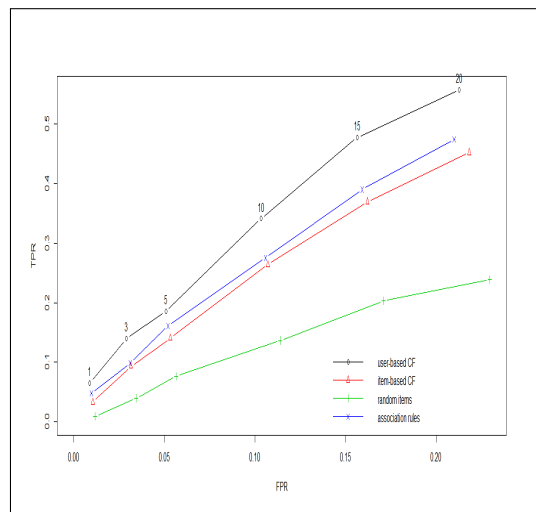
(a) Curva ROC para el grupo 5



(b) Curva ROC para el grupo 6



(c) Curva ROC para el grupo 7



(d) Curva ROC para el grupo 8

Figura 5.6: Resultado del esquema de evaluación para los grupos 5–8 con $\mathcal{I} = \mathcal{I}_2$.
Elaboración Propia.

De igual forma en las Figuras (5.5–5.6) se muestran los mismos resultados pero esta vez para \mathcal{I}_1 . Una vez más los métodos de filtros colaborativos se muestran superior, pero el rendimiento es mucho menor que usar como \mathcal{I} a \mathcal{I}_1 . Por lo tanto, en los análisis

posteriores se tomará $\mathcal{I} = \mathcal{I}_2$ como el único conjunto de productos.

§ § 5.3.2 Evaluación Capacidad de Predicción

En la sección anterior se evaluaron no solamente varios algoritmos de recomendación sino también cual es la estructura del conjunto de productos más conveniente para la base de datos transaccional con la que se cuenta. Esta evaluación se la realizó bajo el esquema que ofrece `recommenderlab`. La deficiencia de este enfoque es la de escoger aleatoriamente los elementos que pertenecerán a los conjuntos de prueba (\mathcal{U}_p) y de entrenamiento (\mathcal{U}_e), de manera que no se podría determinar si la recomendación que se hace a cierto instante será posteriormente acogida. Por consiguiente, resulta de gran interés conocer la capacidad que tiene el algoritmo recomendador para adelantarse a la adquisición de los productos por parte de los usuarios antes que éstos las hagan. Por ejemplo, en el Cuadro 5.7 se describe la adquisición de productos financieros por parte de un individuo perteneciente al grupo 8; éste cliente hasta Diciembre de 2010 ha adquirido al menos un producto de CAV10, TDC4, CON1, TDC5, y TDC3, mientras que en el 2011 adquirió por lo menos los siguientes productos: CON3 y CON4. Utilizando únicamente información correspondiente al grupo 8 y hasta Diciembre de 2010 se entrena el algoritmo recomendador; luego, mediante la siguiente instrucción se determina la lista de productos a recomendar para el cliente del Cuadro 5.7. Por lo tanto, dos de los cuatro productos de la lista de recomendación⁵ han sido adquirido por el cliente en el año 2011, de manera que la predicción del sistema es del 100 por ciento para este caso.

⁵El tamaño de la lista de productos a recomendar (pp) se tomó como el promedio de productos adquiridos por los clientes del grupo hasta Diciembre de 2010

```

r=Recommender(TRANS, method="UBCF")
recom=predict(r,TRANS[TRANS@data@transactionInfo
               $transactionID=="180xxxx364",], n=pp)
prod_recom=as(recom, "list")[[1]]
prod_recom
[1] "CON3" "CON10" "CON2" "CON4"

```

bd_cod_numero_id	bd_fecha_inicial	bd_valor_inicial	PERC
180xxxx364	2003 – 09 – 30	1820.00	TDC3
180xxxx364	2005 – 03 – 23	2500.00	TDC4
180xxxx364	2008 – 01 – 01	4830.00	TDC5
180xxxx364	2008 – 10 – 09	2500.00	TDC4
180xxxx364	2009 – 02 – 20	102000.00	CAV10
180xxxx364	2010 – 01 – 29	98.39	CON1
180xxxx364	2010 – 02 – 22	66.82	CON1
180xxxx364	2010 – 03 – 01	1500.00	TDC3
180xxxx364	2011 – 04 – 26	798.28	CON4
180xxxx364	2011 – 05 – 30	468.37	CON3
180xxxx364	2011 – 06 – 23	475.01	CON3

Cuadro 5.7: Productos Adquiridos por 180xxxx364 del grupo 8. Elaboración Propia.

La función descrita en el Anexo G establece este esquema de evaluación temporal, dando como resultado tanto la tasa de aciertos de productos (productos recomendados que han sido adquiridos), como la de tipo de producto, para cada uno de los 8 grupos. Los resultados se presentan a continuación.

Grupo	UBCF	RANDOM
1	47,49%	31,85%
2	59,01%	31,06%
3	63,12%	28,72%
4	66,39%	34,43%
5	63,23%	30,76%
6	45,61%	30,01%
7	64,01%	30,09%
8	69,84%	36,51%

Cuadro 5.8: Tasa de predicciones acertadas de productos financieros. Elaboración Propia.

Grupo	UBCF	RANDOM
1	62,28%	35,74%
2	52,80%	39,13%
3	54,26%	36,88%
4	49,18%	35,25%
5	57,80%	32,19%
6	54,87%	33,16%
7	56,03%	39,23%
8	46,03%	36,51%

Cuadro 5.9: Tasa de predicciones acertadas del tipo de productos financieros. Elaboración Propia.

Los Cuadros 5.8 y 5.9 describen la tasa de aciertos de los productos y tipos de productos que han sido recomendados utilizando la información disponible hasta el año 2010 y que han sido adquiridos (información no utilizada por el algoritmo) en el siguiente año. Además, se comparan el método de filtro colaborativo (UBCF) frente a una recomendación hecha al azar. Claramente se observa que ésta última recomendación, tanto para los productos como para el tipo de producto, es muy inferior a la del filtro colaborativo que en cambio si presenta mejores resultados en la predicción de los

productos. Por ejemplo, las recomendaciones realizadas para el grupo 8 desembocaron en una adquisición de casi el 70%, lo que se traduce en una muy buena oportunidad de generar beneficios tanto para el banco como para el cliente, ofreciéndole a éste último productos de su preferencia y a su medida, mientras que la institución esperaría una respuesta positiva a esta oferta en el 70% de los casos sin incurrir en costos adicionales por tratar de vender otro producto al cliente.

CAPÍTULO 6

Conclusiones Y Recomendaciones

En esta investigación se presentó una manera de llevar a cabo la estrategia de venta de cruzada de productos financieros en una institución a través de un CRM. Esta estrategia de CRM involucra la adquisición, el análisis y el uso del conocimiento de los clientes con la finalidad de vender más productos y hacerlo de forma más eficiente.

Como paso fundamental de esta implementación, un CRM debe partir, fundamentalmente, de una segmentación de clientes, para que la organización sepa donde se encuentra, hacia donde quiere ir y que pasos (estrategias) debe tomar para alcanzar el objetivo. Dado ese punto de partida, la estrategia de venta cruzada de productos financieros ha sido abordada bajo el modelo de los *Sistemas Recomendadores* que capturan las preferencias de los clientes que pertenecen a un conglomerado común y realiza recomendaciones de productos que posiblemente sea de su agrado y lo compre.

§ 6.1 Conclusiones

- A pesar de que el concepto de CRM no es del todo nuevo pues se ha efectuado de forma empírica por los empresarios, es necesario llevar esta herramienta a un entorno más formal. Esto se debe a que se debe tener un control pleno de los procesos internos de la organización, y que además apoya al correcto flujo de información y la correcta comunicación interdepartamental. Por otro lado, un

CRM no es una iniciativa de un departamento, sino más bien un enfoque global de la compañía

- Actualmente, la tecnología es muy importante para optimizar las relaciones con los clientes. Por medio de la tecnología de la información se pueden rediseñar los procesos de negocios, facilitando los cambios en las prácticas de trabajo y el establecimiento de métodos innovadores que unan a la compañía con los clientes, y que ésta se convierta en el punto de partida del replanteamiento y renovación de los procesos de negocio, en una constante búsqueda del perfeccionamiento de los mismos.
 - Dada la potencia de capacidad de análisis de datos con lo que se cuenta hoy en día y la versatilidad a la hora de capturar datos de muy diversas fuentes, se pueden generar una gran variedad de modelos predictivos. Como se mencionó anteriormente, una de las primeras necesidades en torno a la implementación de un CRM es la planificación del conocimiento del cliente. La necesidad de conocer a los clientes por parte de la organización fue solventada por medio del diseño de una segmentación utilizando cuatro variables esenciales: *Recency*, *Frecuency*, *Monetary* y *Risk*. Como resultado, se determinaron que los segmentos G8, G4 y G7 son clientes de muy alto valor para la institución; de manera que se deberían estructurar ofertas diferenciadas orientadas a mantener a aquellos clientes que en definitiva son los más rentables. Por otro lado, para los grupo G1 y G6 que son los clientes de menor valor para el banco, se deberían tomar estrategias destinadas a evitar la deserción paulatina de la institución. Para ambos casos, la recomendación de productos financieros, es una estrategia que puede ayudar a
-

cumplir aquellos objetivos sin incurrir en mayores gastos.

- Un aspecto importante al momento de decidir implementar un *Sistema Recomendador* es el de escoger la mejor técnica-algoritmo o combinación de algoritmos (sistemas híbridos) que se ajusten a las necesidades de factibilidad de la organización y que naturalmente presenten los mejores resultados. En ese sentido, en el Capítulo 4 se describió brevemente acerca de los métodos basados en vecindades llamados *Filtros Colaborativos* y que fueron aplicados y evaluados en la Sección 5.3. Analizando la gráfica ROC del esquema de evaluación implementado para cada grupo, se concluye que ambos algoritmos colaborativos (basado en el usuario y basado en el producto) tienen una eficiencia casi similar, es decir, se puede optar por cualquier algoritmo al momento de implementar el sistema.
 - El índice AUC (Área Bajo la Curva ROC) que mide la eficiencia con la cual se desenvuelve el algoritmo recomendador es alto, cerca del 80%, lo cual indica un buen rendimiento del algoritmo en cuanto a la certeza con la que las recomendaciones han sido acogidas. Además, la tasa de aciertos de productos recomendados que han sido adquiridos resulta ser mucho mayor (casi el doble para el grupo 8) que si se hubiese hecho una recomendación al azar (es decir, una campaña masiva y común). Cabe mencionar que los grupos con mayores tasa de aciertos son los grupos 8, 4 y 7, que precisamente son los conglomerados que agrupan a los clientes de más alto valor para la institución; de esa manera se concluye que estos grupos a más de ser los más rentables para el banco, son los que presentan mayor posibilidad de adquirir el producto dada una lista de productos a recomendar, específicamente se esperaría una respuesta positiva a la estrategia de venta cruzada
-

en el 69.84% (Grupo 8), 66.39% (Grupo 4) y 66.39% (Grupo 7) de los casos.

- Un aspecto importante que vale la pena mencionar es que los grupos con mayor número de individuos (G1 y G6) a más de ser los conglomerados de menor valor para la institución, son los que registran la menor tasa de aciertos de productos, pero en cambio su tasa de acierto de tipo de producto es de las más altas. Esto se podría explicar al hecho que al presentarse mayor concentración de clientes, la heterogeneidad del grupo se acentúa mucho más y por lo tanto la recomendación que esta basada en las *preferencias de usuarios similares* no resulta ser muy efectiva. Por otro lado, al tener esta alta concentración de tipos de productos en estos grupos, que son tres en total, hacen que el conjunto de productos se distribuya de manera casi uniforme, de manera que no es sorprendente que se presente mejores tasas de aciertos, ya que no hay muchas opciones para recomendar. Por lo tanto se puede concluir que es necesario contar con un conjunto suficientemente amplio de productos y usuarios lo más similares posibles para una efectiva recomendación.

§ 6.2 Recomendaciones

- Para implementar una estrategia CRM no sólo se debe entender claramente todo lo que esto conlleva, como es la infraestructura, desarrollo y demás, sino tener claro el objetivo por el cual se lo va a llevar cabo. Estos objetivos deber ser lo suficientemente claros y precisos.
 - En la etapa del descubrimiento del conocimiento (ver Sección 2.2), el cliente es analizado para identificar oportunidades de mercado y posibles estrategias de
-

inversión. Es en esta etapa donde la identificación, segmentación y predicción del cliente es hecha; todo aquello encaminado a la mejor toma de decisiones. Por lo tanto, es necesario (aunque no indispensable) requerir al Data Warehouse para comenzar a entender el comportamiento del cliente.

- Al momento de poner en producción un sistema recomendador, hay que tomar en cuenta que éste es por naturaleza parte de algún sistema más grande. Un recomendador es una de las características de una aplicación general. Éste puede ser una característica menor o un punto importante para la venta del producto; la aplicación puede ya estar establecida o construirse junto con el recomendador, pero en cualquier caso el diseño del sistema recomendador tiene que ser integrado dentro del diseño de la aplicación el host (core bancario).

El satisfacer a los clientes en muchas ocasiones es algo difícil de lograr, ya que al ser algo subjetivo, no se puede tener la misma percepción de lo que el cliente busca como un factor de satisfacción. Sin embargo, sí se puede tener el cuidado y la intención de acercarse lo más posible a ese factor. Esto se puede lograr si se aprovechan plenamente los recursos con los que cuenta la organización (humanos y tecnológicos), además de ser combinados con estrategias adecuadas para lograr un entendimiento pleno con el cliente (es decir, un CRM efectivo).

Anexos

ANEXO A

Algoritmo PAM

El algoritmo PAM fue desarrollado por [Kaufman and Rousseeuw \[2005\]](#), y una descripción del mismo fue dada en la sección 3.2. También se mencionó que en el método k -medoid se busca minimizar la disimilaridad *promedio* de los objetos hacia los objetos representativos más cercanos. Sin embargo, a la hora de programar el algoritmo es preferible minimizar la suma de aquellas disimilaridades, lo cual es matemáticamente equivalente pero como resultado se tienen cálculos más precisos.

El algoritmo para ejecutar PAM consta de dos fases. En la primera, llamada *Construcción*, se obtiene un clustering inicial por la selección sucesiva de objetos representativos hasta que se encuentran k de ellos. El primer objeto es aquel para el cual la suma de las disimilaridades hacia el resto de objetos es la más pequeña posible. Este objeto es el más cercano al centro del conjunto de objetos. Subsecuentemente, en cada paso otro objeto es seleccionado. Este objeto en cambio es aquel para el cual la función objetivo decrece tanto como sea posible. Para encontrar este objeto, los siguientes pasos son llevados a cabo.

- Considerar un objeto i el cual todavía no se haya seleccionado.
- Considerar un objeto no seleccionado j y calcular la diferencia entre su disimilaridad D_j con los objetos más similares previamente seleccionados, y su disimi-

laridad $d(j, i)$ con el objeto i .

- Si la diferencia es positiva, el objeto j contribuirá a la decisión de seleccionar el objeto i . Por lo tanto se calcula

$$C_{ji} = \max \{D_j - d(j, i), 0\}.$$

- Calcular la ganancia total obtenida al seleccionar el objeto i :

$$\sum_j C_{ji}.$$

- Escoger el objeto i que aun no ha sido seleccionado tal que maximice

$$\sum_j C_{ji}.$$

El proceso continua hasta que k objetos hallan sido encontrados. En la segunda fase del algoritmo (llamada *Intercambio*), se intenta mejorar el conjunto de objetos representativos y por lo tanto también mejorar el clustering producido por este conjunto. esto se hace al considerar todos los pares de objetos (i, h) para el cual *el objeto i ha sido seleccionado y el objeto h no*. Se determina que efecto se obtiene en el valor¹ del clustering cuando se produce un intercambio, es decir, cuando el objeto i no ha sido seleccionado como un objeto representativo pero el objeto h si. Para calcular el efecto de un intercambio entre i y h en el valor del clustering, los siguientes cálculos son llevados a cabo.

1. Considerar un objeto no seleccionado j y calcular su contribución C_{jih} al intercambio:

¹El valor de un clustering determinado por los k objetos representativos se define como la suma de las disimilaridades entre cada objeto y los objetos representativos más similares.

- (a) Si j es el objeto más distante tanto de i como de h que de otros objetos representativos, C_{jih} es cero.
- (b) Si j no está lo suficientemente lejos de i que de algún otro objeto representativo seleccionado ($d(j, i) = D_j$), se deben considerar dos situaciones:

- i. j está mas cerca de h que al segundo objeto representativo más cercano,

$$d(j, h) < E_j,$$

donde E_j es la disimilaridad entre j y el segundo objeto representativo más similar. En este caso la contribución del objeto j al intercambio entre los objetos i y h es

$$C_{jih} = d(j, h) - d(j, i)$$

- ii. j está al menos tan distante de h que del segundo objeto representativo más cercano,

$$d(j, h) \geq E_j.$$

En este caso la contribución del objeto j al intercambio es

$$C_{jih} = E_j - D_j.$$

- (c) j está más distante del objeto i que de al menos uno de los otros objetos representativos pero más cercano a h que a cualquier objeto representativo. En este caso la contribución de j al intercambio es

$$C_{jih} = d(j, h) - D_j.$$

2. Calcular el resultado total de un intercambio al añadir las contribuciones C_{jih} :

$$T_{ih} = \sum_j C_{jih}.$$

3. Seleccionar el par (i, h) para el cual se minimiza T_{ih} .
4. Si el mínimo T_{ih} es negativo, el intercambio se lleva a cabo y el algoritmo retorna al paso 1. Si el mínimo T_{ih} es positivo o 0, el valor del objetivo ya no puede decrecer al producirse un intercambio y el algoritmo se detiene.
-

ANEXO B

Segmentación de Clientes

```
#BASES Y LIBRERÍAS NECESARIAS
setwd("C:/Users/David/Desktop/Recomendador_Final")
load("Comportamiento.RData");load("Transaccional.RData")
library(gdata);library(cluster);library(recommenderlab)

#GENERACIÓN DE LAS VARIABLES R, F y M
#Formato correcto de fachas
Transaccional$bd_fecha=as.Date(Transaccional$bd_fecha)
Transaccional$bd_fecha_inicial=as.Date(Transaccional
$bd_fecha_inicial)
Transaccional$bd_fecha_vencimiento=as.Date(Transaccional
$bd_fecha_vencimiento)
Transaccional$bd_fecha_cancelacion=as.Date(Transaccional
$bd_fecha_cancelacion)
#Filtrar solo operaciones de Entidad F
BANCO=Transaccional[Transaccional$bd_nombre_entidad=="Entidad F",]
BANCO=BANCO[BANCO$bd_cod_tipo_id!="R",]
BANCO$bd_tipo_cuenta=trim(BANCO$bd_tipo_cuenta)
BANCO=BANCO[BANCO$bd_tipo_cuenta!="COMERCIAL",]
BANCO$Cont=1
BANCO$Dif_Fecha=BANCO$bd_fecha-BANCO$bd_fecha_inicial
#Realiza cálculos y guarda resultados
TablaA=aggregate(bd_valor_inicial~bd_cod_numero_id,data=BANCO,sum)
TablaB=aggregate(Cont~bd_cod_numero_id,data=BANCO,sum)
TablaC=aggregate(Dif_Fecha~bd_cod_numero_id,data=BANCO,
function(x) min(x,na.rm=FALSE))
```

```
TABLA_RFM=cbind(TablaA,TablaB[,2],TablaC[,2])
names(TABLA_RFM)=c("ID","Monetary","Frequency","Recency")
save(TABLA_RFM,file="RFM.RData")
```

#GENERACIÓN DE LA VARIABLE Risk

```
TablaD_1=Comportamiento[,c("Cadiasmora","CODIGO_CEDULA","Catpoper")]
TablaD_2=Comportamiento[,c("Camonto","CODIGO_CEDULA","Catpoper")]
TablaD1=tapply(TablaD_1$Cadiasmora,TablaD_1[-1],function(x)
               max(x,na.rm=FALSE))
TablaD1_N=TablaD1;TablaD1_N[is.na(TablaD1_N)==FALSE]=1
TablaD1_N[is.na(TablaD1_N)==TRUE]=0
TablaD1_M=TablaD1;TablaD1_M[TablaD1_M<30]=0;TablaD1_M[TablaD1_M>=30]=1
TablaD1_N=data.frame(TablaD1_N);TablaD1_M=data.frame(TablaD1_M)
TablaD1_N$ID=substr(row.names(TablaD1_N),1,10)
TablaD1_N$Productos=rowSums(TablaD1_N[,-143],na.rm=TRUE)
TABLA_E1=aggregate(Productos~ID,TablaD1_N,sum)
TablaD1_M$ID=substr(row.names(TablaD1_M),1,10)
TablaD1_M$Productos=rowSums(TablaD1_M[,-143],na.rm=TRUE)
TABLA_E2=aggregate(Productos~ID,TablaD1_M,sum)
TABLA_F=cbind(TABLA_E1,TABLA_E2[,2])
TABLA_F$Risk=TABLA_F[,3]/TABLA_F[,2]
TABLA_Risk=merge(TABLA_RFM,TABLA_F[,c(-2,-3)],by.x="ID",
                 by.y="ID",all.x=TRUE)
TABLA_Risk=TABLA_Risk[is.na(TABLA_Risk$Risk)==FALSE,]
TABLA_Risk=TABLA_Risk[order(TABLA_Risk$Risk),]
save(TABLA_Risk,file="RFM_Risk.RData")
```

#DETERMINACIÓN DE LOS GRUPOS

```
fit=pam(TABLA_Risk[,c("Monetary","Frequency","Recency","Risk")],8,
        diss=FALSE,metric="euclidean",stand=FALSE)
TABLA_Risk=data.frame(TABLA_Risk, fit$cluster)
TABLA_Risk$Grupos=TABLA_Risk$fit.cluster
TABLA_Risk[["fit.cluster"]]=NULL
```

```

save(TABLA_Grupos,file="Grupos.RData")
#Resumen de los grupos
TABLA_Risk=TABLA_Grupos
TABLA_Risk$Tamaño=1
A1=aggregate(Tamaño~Grupos,data=TABLA_Risk,sum)
A2=aggregate(Recency~Grupos,data=TABLA_Risk,mean)
A3=aggregate(Frequency~Grupos,data=TABLA_Risk,mean)
A4=aggregate(Monetary~Grupos,data=TABLA_Risk,mean)
A5=aggregate(Risk~Grupos,data=TABLA_Risk,mean)
GRUPOS=cbind(A1,round(A2[,2],2),round(A3[,2],2),
             round(A4[,2],2),round(100*A5[,2],2))
GRUPOS[,1]=as.factor(GRUPOS[,1])
levels(GRUPOS[,1])=c("G1","G2","G3","G4","G5","G6","G7","G8")
r=round(mean(TABLA_Risk[,4]),2);f=round(mean(TABLA_Risk[,3]),2)
m=round(mean(TABLA_Risk[,2]),2);ri=round(100*mean(TABLA_Risk[,5]),2)
for (i in 1:8)
{
  if (GRUPOS[i,3]<=r) {Patron_R=c("R+")} else{Patron_R=c("R-")}
  if (GRUPOS[i,4]>=f) {Patron_F=c("F+")} else{Patron_F=c("F-")}
  if (GRUPOS[i,5]>=m) {Patron_M=c("M+")} else{Patron_M=c("M-")}
  if (GRUPOS[i,6]<=ri) {Patron_Ri=c("Risk-")} else{Patron_Ri=
    c("Risk+")}
  Patron=paste(Patron_R,Patron_F,Patron_M,Patron_Ri,sep="")
  if (i==1){PatronT=Patron} else {PatronT=rbind(PatronT,Patron)}
}
A6=as.vector(rep(0,8));for (i in 1:8) { A6[i]=PatronT[i][1] }
A7=cbind("TOTAL",sum(TABLA_Risk[,7]),r,f,m,ri,"")
A7=data.frame(A7)
names(A7)=c("Grupos","Tamaño","Recency","Frequency",
            "Monetary","Risk","Patrón")
GRUPOS=cbind(GRUPOS,A6)
names(GRUPOS)=c("Grupos","Tamaño","Recency","Frequency",
                "Monetary","Risk","Patrón")
GRUPOS=rbind(GRUPOS,A7)

```

ANEXO C

Calibración Matriz Usuario–Producto (Primera parte)

```
BASE_GRUPOS=Transaccional[,c("bd_cod_numero_id","bd_tipo_operacion",
"bd_fecha_inicial","bd_valor_inicial","plazo","bd_tipo_riesgo",
"bd_md_cuota_estimada")]
BASE_GRUPOS=BASE_GRUPOS[BASE_GRUPOS$plazo>0,]
BASE_GRUPOS=BASE_GRUPOS[BASE_GRUPOS$bd_tipo_riesgo=="TITULAR",]
BASE_GRUPOS=na.omit(BASE_GRUPOS)

BASE_CON=BASE_GRUPOS[BASE_GRUPOS$bd_tipo_operacion=="CON",]
agn1=clara(BASE_CON[,c(4,5,7)],sampsiz=1000,samples=20,10)
BASE_CON$Prod=agn1$clustering
BASE_CON$Prod=as.factor(BASE_CON$Prod)
levels(BASE_CON$Prod)=paste("CON_",c(1:10),sep="")
BASE_CON$Tamaño=1
A1=aggregate(Tamaño~Prod,data=BASE_CON,sum)
A2=aggregate(bd_valor_inicial~Prod,data=BASE_CON,max)
A3=aggregate(plazo~Prod,data=BASE_CON,mean)
A4=aggregate(bd_md_cuota_estimada~Prod,data=BASE_CON,mean)
GRUPOS_CON=cbind(A1,A2[,2],A3[,2],A4[,2])
names(GRUPOS_CON)=c("Producto","Tamaño","Monto","Plazo","Cuota")
```


ANEXO D

Calibración Matriz Usuario–Producto (Segunda parte)

```
j=1
d=matrix(c(3,2,1,0,0,4,0,0,5),nrow=3,ncol=3)
x=BASE_CON$bd_valor_inicial[BASE_CON$Prod==paste("CON_",j,sep="")
                             & BASE_CON$plazo<7]
y=BASE_CON$plazo[BASE_CON$Prod==paste("CON_",j,sep="")
                 & BASE_CON$plazo<7]
nf=layout(d,widths=c(3,0.5,1.3),heights=c(1.3,0.5,3),respect=T)
xhist=hist(x,plot=FALSE);yhist=hist(y,plot=FALSE)
top=max(c(xhist$counts,yhist$counts))
xrange=c(min(x),max(x));yrange=c(min(y),max(y))
par(mar=c(4,4,1,1))
plot(x,y,xlim=xrange,ylim=yrange,xlab="Monto",ylab="Plazo")
par(mar=c(0,4,0.5,1))
boxplot(x,horizontal=TRUE,xaxt='n',col="gray")
par(mar=c(0,4,1,1))
barplot(xhist$counts,axes=FALSE,ylim=c(0,top),space=0,col="green")
title(main=paste("Monto y Plazo para CON_",j,sep=""),
      cex.main=0.9,font=2)
par(mar=c(4,0,1,0.5))
boxplot(y,yaxt='n',col="gray")
par(mar=c(4,0,1,1))
barplot(yhist$counts,axes=FALSE,xlim=c(0,top),space=0,horiz=TRUE,
      col="red")
```

ANEXO E

Calibración Matriz Usuario–Producto (Tercera parte)

```
# Cartera de Consumo
BASE_CON=BASE_GRUPOS[BASE_GRUPOS$bd_tipo_operacion=="CON",]
param=10
cuan=quantile(BASE_CON$bd_valor_inicial,c(0:param)/param)
BASE_CON$PERC=cut(BASE_CON$bd_valor_inicial,breaks=cuan)
aa=seq(1,param)
for (i in 1:param)
{
  aa[i]=paste("CON",i,sep="")
}
levels(BASE_CON$PERC)=aa
```

ANEXO F

Selección Algoritmo Recomendador

```
BASE_P=rbind(BASE_CON, BASE_TDC, BASE_CAV)
BASE_P=merge(BASE_P, TABLA_Grupos, by.x="bd_cod_numero_id", by.y="ID")
BASE_PE=unique(BASE_P[, c(1, 2, 8, 13)])
EVALUACION=function(G, np){
  MATRIX=BASE_PE[BASE_PE$Grupos==G,]
  TRANS=as(split(MATRIX[, "PERC"], MATRIX[, "bd_cod_numero_id"]),
            "transactions")
  TRANS=new("binaryRatingMatrix", data=TRANS)
  TRANS_np=TRANS[rowCounts(TRANS)>np,]
  Esquema=evaluationScheme(TRANS_np, method="cross-validation",
                           k=10, given=np)
  Algoritmos=list('user-based CF'=list(name="UBCF", param=list(method=
    "Jaccard", nn=50)), 'item-based CF'=list(names="IBCF", param=list(method=
    "Jaccard", k=50)), 'random items'=list(name="RANDOM", param=NULL),
    'association rules'=list(name="AR", param=list(supp=0.001,
    conf=0.2, maxlen=2)))
  Resultados=evaluate(Esquema, Algoritmos, n=c(1, 3, 5, 10, 15, 20))
  plot(Resultados, annotate=TRUE)
  avg(Resultados)
}
```

ANEXO G

Evaluación Capacidad Predicción Sistema Recomendador

```
BASE_P_2008_2010=BASE_P[BASE_P$bd_fecha_inicial>=
as.Date("2008/01/01", "%Y/%m/%d") & BASE_P
$bd_fecha_inicial<=as.Date("2010/12/31", "%Y/%m/%d"),]
BASE_P_2011_hoy=BASE_P[BASE_P$bd_fecha_inicial>=
as.Date("2011/01/01", "%Y/%m/%d") & BASE_P
$bd_fecha_inicial<=as.Date("2011/12/31", "%Y/%m/%d"),]
BASE_P_2008_2010=unique(BASE_P_2008_2010[,c(1,2,8,13)])
BASE_P_2011_hoy=unique(BASE_P_2011_hoy[,c(1,2,8,13)])
PREDICCION=function(G,TP){
INFO1=BASE_P_2008_2010[BASE_P_2008_2010$Grupos==G,]
INFO2=BASE_P_2011_hoy[BASE_P_2011_hoy$Grupos==G,]
cedulas_1=unique(INFO1[,1])
cedulas_2=unique(INFO2[,1])
cedulas_T=cedulas_1[cedulas_1 %in% cedulas_2]
AA=INFO1[INFO1[,1] %in% cedulas_T,]
BB=INFO2[INFO2[,1] %in% cedulas_T,]
TRANS=as(split(AA[, "PERC"], AA[, "bd_cod_numero_id"]), "transactions")
TRANS=new("binaryRatingMatrix", data=TRANS)
#Número promedio de productos adquiridos desde 2008 hasta 2010
AA$count=1
aux=aggregate(count~bd_cod_numero_id, data=AA, sum)
pp=round(mean(aux[, 2]), 0)
if (TP=="producto"){
#Predicción de productos adquiridos en 2011
for (i in 1:length(cedulas_T))
```

```

{
  r=Recommender(TRANS, method="UBCF")
  recom=predict(r, TRANS[TRANS@data@transactionInfo$
    transactionID==cedulas_T[i],], n=pp)
  prod_recom=as(recom, "list")[[1]]
  for (j in 1:pp)
  {
    prod=prod_recom[j]
    a1=as.numeric(substr(prod,4,6))
    a2=substr(prod,1,3)
    margen1=paste(a2,a1-1,sep="")
    margen2=paste(a2,a1+1,sep="")
    posible_prod=cbind(margen1,prod,margen2)
    if (j==1){posible_prodT=posible_prod} else
    {posible_prodT=rbind(posible_prodT,posible_prod)}
  }
  posible_prodT=as.vector(posible_prodT)
  a3=posible_prodT %in% as.character(BB$PERC[BB
    $bd_cod_numero_id==cedulas_T[i]])
  a3=data.frame(a3);a3$count=1
  a4=aggregate(count~a3,data=a3,sum)
  if (a4[1,2]==length(posible_prodT)) {a5=0} else {a5=1}
  if (i==1){a5T=a5} else {a5T=rbind(a5T,a5)}
}
a6=round(100*sum(a5T)/length(a5T),2)
Tasa_Acierto=paste(a6,"%",sep="")
print(Tasa_Acierto)
}
else{
#Predicción tipo de Producto Adquirido en 2011
for (i in 1:length(cedulas_T))
{
  r=Recommender(TRANS, method="UBCF")
  recom=predict(r, TRANS[TRANS@data@transactionInfo

```

```
      $transactionID==cedulas_T[i],], n=1)
prod_recom=as(recom, "list")[[1]]
first_prod=prod_recom[1]
a7=substr(first_prod,1,3)
a8=a7 %in% as.character(BB$bd_tipo_operacion[BB
  $bd_cod_numero_id==cedulas_T[i]])
a9=0 # 1:Acierto
if (a8[1]==TRUE) {a9=1}
if (i==1){a9T=a9} else {a9T=rbind(a9T,a9)}
}
a10=round(100*sum(a9T)/length(a9T),2)
Tasa_Acierto=paste(a10,"%",sep="")
print(Tasa_Acierto)
}}
```

Referencias Bibliográficas

- R. Blattberg, P. Kim, B. Kim, and S. Neslin. *Database marketing: analyzing and managing customers*. International Series in Quantitative Marketing. Springer, 2008. ISBN 9780387725789.
- G. Brock, V. Pihur, S. Datta, and S. Datta. clvalid: An R package for cluster validation. *Journal of Statistical Software*, 25(4):1–22, 3 2008. ISSN 1548-7660.
- J. R. Bult and T. Wansbeek. Optimal selection for direct mail. *Marketing Science*, 14(4):378–394, 1995.
- I. Díez. *Cómo conquistar el mercado con una estrategia CRM*. Fundación Confemetal, 2004. ISBN 9788496169388.
- F. Gorunescu. *Data Mining: Concepts, Models and Techniques*. Intelligent Systems Reference Library. Springer, 2011. ISBN 9783642197208.
- F. Harding. *Cross-selling success: a rainmaker's guide to professional account development*. Adams Media Corp., 2002. ISBN 9781580627054.
- S. M. S. Hosseini, A. Maleki, and M. R. Gholamian. Cluster analysis using data mining approach to develop crm methodology to assess the customer loyalty. *Expert Syst. Appl.*, 37:5259–5264, July 2010. ISSN 0957-4174.

- W. A. Kamakura, M. Wedel, F. de Rosa, and J. A. Mazzon. Cross-selling through database marketing: a mixed data factor analyzer for data augmentation and prediction. *International Journal of Research in Marketing*, 20(1):45–65, 2003. ISSN 0167–8116.
- L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley-Interscience, 2005. ISBN 0471735787.
- F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011. ISBN 978-0-387-85819-7.
- L. Richardson. *Bankers in the Selling Role: A Consultative Guide to Cross-Selling Financial Services*. John Wiley & Sons, 1992. ISBN 9780471572657.
- D. Ritter. *Relationship banking: cross-selling the bank's products & services to meet your customer's every financial need*. Bankers Publishing, 1993. ISBN 9781557383815.
- R. Swift. *Accelerating customer relationships: using CRM and relationship technologies*. Prentice Hall PTR, 2001. ISBN 9780130889843.
- K. Tsipstsis and A. Chorianopoulos. *Data Mining Techniques in CRM: Inside Customer Segmentation*. John Wiley & Sons, 2011. ISBN 9781119965459.
- E. Turban, R. Sharda, D. Delen, J. Aronson, T. Liang, and D. King. *Decision Support and Business Intelligence Systems*. Prentice Hall, 2010. ISBN 9780136107293.
- I. Valcárcel. *CRM. Gestión de la relación con los clientes*. Fundación Confemetal, 2001. ISBN 9788495428394.
-

H.-H. Wu, E.-C. Chang, and C.-F. Lo. Applying RFM model and k -means method in customer value analysis of an outfitter. In S.-Y. Chou, A. Trappey, J. Pokojski, and S. Smith, editors, *Global Perspective for Competitive Enterprise, Economy and Ecology*, Advanced Concurrent Engineering, pages 665–672. Springer London, 2009. ISBN 978-1-84882-762-2. 10.1007/978 - 1 - 84882 - 762 - 2_63.
