2019-07

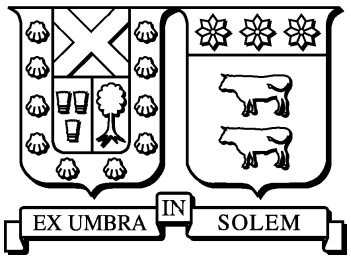
DISEÑO E IMPLEMENTACIÓN PARA UN SISTEMA RECOMENDADOR DE MARKETING BANCARIO

FERNÁNDEZ SOTO, EDUARDO ARIEL

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE INFORMÁTICA

SANTIAGO – CHILE



"DISEÑO E IMPLEMENTACIÓN PARA UN SISTEMA RECOMENDADOR DE MARKETING BANCARIO."

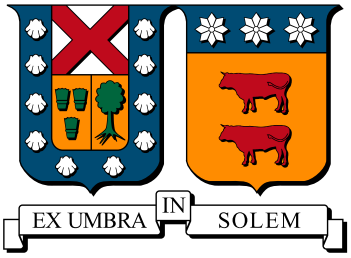
EDUARDO ARIEL FERNÁNDEZ SOTO

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL INFORMÁTICO

PROFESOR GUÍA: RICARDO ÑANCULEF

JULIO 2019

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO – CHILE



"DISEÑO E IMPLEMENTACIÓN PARA UN SISTEMA RECOMENDADOR DE MARKETING BANCARIO."

EDUARDO ARIEL FERNÁNDEZ SOTO

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO CIVIL INFORMÁTICO

PROFESOR GUÍA: RICARDO ÑANCULEF
PROFESOR CORREFERENTE: JOSÉ LUIS MARTI LARA

JULIO 2019

Agradecimientos

Lo que hagamos en la vida hará eco en la eternidad. Sin embargo, no se puede construir lo que tanto anhelamos de un día para otro. Es necesario trabajar duro, e incluso si hay tropiezos y caídas, es necesario superar los obstáculos, tener motivación, perseverancia e insistir. Como siempre los giros de la vida nos sorprenden, pero aquellos que gustan de lo que hacen y se sienten orgullosos del progreso diario, son los que logran continuar. La vida nos demanda coraje, valentía, creatividad y un indiscutible espíritu de lucha, sin embargo, todas estas cualidades no son siempre suficientes y es en esos frenéticos momentos de lucha donde aparecen los verdaderos amigos, nuestros verdaderos compañeros y es a éstos a quienes uno debe agradecer.

En primer lugar, debo no solo darle las gracias, sino que también hacer parte de esto a mi amado hijo Maximiliano, quién me otorga la motivación día a día para seguir, con quien incluso tuve que compartir clases en la universidad por no tener con quien dejarlo, quien fue mi compañero de estudio e incluso intento de profesor. Debo darles las gracias a mis papás que han sido y son el apoyo en este camino, quienes jamás me dejaron solo y estuvieron junto a mí cuando más lo necesite. También debo agradecer a Valeska, quien me otorgó su confianza, cariño y me alegró en el tramo final de esta aventura el cual, en ocasiones, fue el más complicado.

Gracias a Eric y Alfredo por depositar su confianza en mi trabajo. Gracias también a mis compañeros de trabajo quienes con su amistad y buen ánimo me brindaban energía para continuar día a día, incluso sin saber lo difícil que éstos eran en ocasiones. A Cata, Tere, Gonza,

Jose, Mosiah, Tomas, Consuelo, Vicente, Matías y en especial a Camilo, quien se daba el tiempo de ayudarme cuando lo necesitaba y aportar con sus ideas y conocimiento, incluso cuando tenía mucho trabajo por hacer.

Finalmente, no puedo dejar fuera de esto a quienes ya no están. A mí abuela María Guerra, quien siempre me amó y estuvo conmigo hasta sus últimos días, y a mi abuelo, quien soñaba con estar a mi lado al momento de mi titulación. Lamentablemente no están físicamente en este momento, pero tengan por seguro que están en mi corazón, los amo.

Resumen

Una de las problemáticas que se afrontan en el contexto del marketing bancario, es como ofrecer adecuadamente los productos que se poseen, es decir, la realización de una estrategia ad-hoc enfocada en los requerimientos de cada cliente, sin incurrir en el desagradable marketing viral. En este ámbito, se busca generar un modelo, que formule políticas que ayuden en la toma de decisiones, al momento de ofrecer los productos que posee la institución, de manera que esto se realice de forma efectiva, con el objetivo de maximizar los beneficios asociados a las políticas generadas por el método utilizado.

En base a esto, en esta memoria se propone abordar el problema anterior, diseñando e implementando un **sistema recomendador** que permita ofrecer productos enfocados en cada cliente. Dada las características específicas del negocio, es necesario considerar métodos alternativos a los sistemas recomendadores tradicionales para diversificar las soluciones de forma tal que se aprovechen las ventajas estratégicas que cada método a utilizar posee al momento de generar políticas de recomendación. Una de las principales problemáticas a considerar es la naturaleza cambiante en las condiciones económicas de cada persona, y los cambios de comportamiento que esto genera, lo que provoca que tanto el consumo y preferencias por cada producto, vayan variando en el tiempo. Por otra parte, se debe considerar la cardinalidad de usuarios en contraposición a los ítems ofrecidos, debido a que se posee una mayor cantidad de clientes que productos a ofrecer.

Finalmente se espera que el sistema ayude a generar un modelo capaz de crear un mapa de viajes de clientes. Así mismo, ayudará a definir una política de marketing que potencie la

venta cruzada de productos acorde a los cambios en las preferencias futuras que tendrá un determinado cliente.

Abstract

One of the problems that is faced in the area of bank marketing is how to adequetly offer the available products to clients, that is, the development of a specific strategy focused on the needs of each client without using the unpleasant viral marketing.To address this problem, it is useful to generate a model that formulates policies that help the bank to make decisions to offer products effectively in order to optimize the benefits of such institution.

In this thesis, the problem described above will be addressed by using a recommender system that will offer specific products to each client. Due to the nature of the problem, it is necessary to consider alternative methods to the traditional recommender systems in order to diversify the solutions and to exploit the strategic advantages that each one posseses when recommending policies are generated. On the one hand, one important issue to be considered is the changing nature in the economic conditions of each person and the corresponding changes that these trigger in their behaviour. On the other hand, there is a larger amount of clients than products to be offered which is not the case in traditional recommender systems.

Finally, it is expected that the recommender system will help to generate a model able to create a journey map. Additionally, the system will help to define a marketing policy to strengthen the cross-selling of products in relationship to the variations in future preferences of a specific client.

Índice de Contenidos

Índice de cuadros

Índice de figuras

Glosario

- **A**
  - **Algoritmo:** Conjunto de operaciones estructuradas que poseen como objetivo solucionar una problemática.
  - **Algoritmo Supervisado:** Técnica para encontrar relación existente entre variable de entrada y salida.
- **C**
  - **Cliente:** Corresponde a las personas u empresas con RUT personal o jurídico que utilizan productos bancarios.
  - **Cluster:** Agrupación de los datos acorde a un criterio definido.
  - **CRM:** Gestión de relación con clientes.
  - **Cross-Selling:** Proporcionar productos financieros adicionales a un cliente.
- **D**
  - **Data mining:** Proceso que busca descubrir patrones y extraer información relevante de los datos disponibles.
  - **Dataframe:** Conjunto de datos generado en python pandas.
  - **Dispersión:** Baja densidad de ratings.
- **M**
  - **Machine Learning:** Rama que explora la construcción de algoritmos capaces de aprender y realizar predicciones sobre datos.

- **Mapa de viaje del cliente:** Se trata de una herramienta que permite plasmar en un mapa, cada una de las etapas, interacciones, canales y elementos por los que atraviesa un cliente durante todo el ciclo de compra o adquisición de un producto.

- **R**

  - **Rating:** Valor numérico que representa la preferencia de un usuario sobre un determinado producto u objeto.

- **S**

  - **Similaridad:** Parecido de los daos según un criterio predefinido.

- **T**

  - **Testing:** Fase en que se evalúa un modelo predictivo.
  - **Training:** Fase en la que se construye un modelo predictivo.

- **U**

  - **Usuario:** Corresponde a la institución financiera que desea la generación de recomendaciones para generar campañas de marketing con dicha información.
  - **Up Selling:** Técnica de Marketing que consiste en los clientes aumenten el consumo de productos que ya utilizan y/o consumen productos de mayor valor.

- **V**

  - **Vecindario:** Grupo de clientes con gustos similares al cliente utilizado o activo.

Introducción

En los últimos años los avances en términos de digitalización de la información y la interacción a través de canales ha experimentado grandes cambios y la banca no es la excepción. Si bien se ha mejorado la interacción desarrollada durante las ventas y transacciones, aún existen grandes desafíos en cuanto a aumentar la personalización y la interacción de los consumidores con los servicios ofrecidos en la industria.

Uno de los desafíos que se enfrentan en el ámbito del marketing, es como ofrecer adecuadamente los productos que se poseen. En general, las instituciones financieras cuentan con una variedad de productos candidatos. Desafortunadamente se es incapaz de asignar adecuadamente cada uno de esos productos con cada cliente, ya que, esto puede ser muy costoso y consumir demasiado tiempo. O simplemente no se desea abrumar al cliente enviándole demasiadas ofertas, con lo cual cubrirían gran parte de la variedad de productos que le podría interesar a una persona.

En base a esto, se busca generar un modelo, que formule políticas que ayuden en la toma de decisiones, al momento de ofrecer productos. Por consiguiente, se diseñarán y compararán diversas propuestas para promover la venta cruzada de productos (Cross-Selling), es decir, ofrecer al cliente productos relacionados con los productos en lo que esta interesados actualmente.

Para realizar adecuadamente esto, se debe plantear como objetivo predecir qué productos

usarán los clientes actuales en función de su comportamiento anterior y de los clientes similares a ellos, en pro de satisfacer mejor las necesidades individuales de todos los clientes y aumentar su satisfacción.
Para esto se formulan 3 estrategias para abordar la problemática, las cuales consisten en:

- Sistemas Recomendadores (RecSys).
- Aprendizaje Supervisado (AA).
- Modelo Oculto de Markov (HMM).

Por otra parte, el contenido de este trabajo está conformado por 5 capítulos, el capítulo 1 refiere una descripción del tema a tratar y como se abordará el desafío principal establecido. En el segundo capítulo se realiza una investigación sobre el estado actual de sistemas recomendadores, aprendizaje supervisado y cadenas ocultas de Markov y cuál será la metodología a utilizar con cada uno de estos. Además, se desarrollará una visión más en profundidad de los métodos a utilizar para desarrollar la correcta solución de la problemática y establecer políticas adecuadas de recomendación.
En los Capítulos 3 y 4 se abordarán los métodos propuestos, indicando la estrategia de implementación según los datos y contexto de la problemática, por consiguiente se definirá el marco experimental y las métricas adecuadas para comparar los resultados obtenidos a partir de ellos.
Finalmente en el capítulo 5 se redactarán las respectivas conclusiones del trabajo con las recomendaciones a considerar para futuros trabajos en la materia.

Capítulo 1

Definición del Problema

Una de las principales competencias que se debe abordar para realizar un efectivo marketing bancario es el ofrecer de forma adecuada lo que requieren nuestros clientes, ya que para la banca es esencial poseer una adecuada estrategia de orientación al cliente.
Peter Drucker y Theodore Levitt [15] fueron los primeros en proponer la importancia de las relaciones con el cliente en pos de una adecuada rentabilidad de las organizaciones, destacando la importancia de utilizar y administrar de manera adecuada la información que se posee sobre éstos. Tomando esto en consideración es pertinente pensar, diseñar y probar diversas formas de realizar recomendaciones de manera que sepamos cual es la más adecuada a la realidad actual de la industria de modo que logremos ofrecerle al cliente el producto candidato que él necesita. Después de todo, fue el mismo Levitt en 1983 [15] quien señaló que "Los datos no dan información excepto con la intervención de la mente. La información no da sentido, excepto con la intervención de la imaginación."

Otro de los aspectos a tener en cuenta son los cambios de paradigmas que se han experimentado en cuanto a las relaciones con el cliente. En el área de la comunicación también se abandona la comunicación masiva y genérica para abordar estrategias de definición, de quien ha de ser el mejor receptor del mensaje emitido y así evaluar la rentabilidad de estos propósitos [17].

En todos los casos se pasa de pensar en el producto a pensar en el cliente. Cualquier análisis evoluciona cada vez más en ver que piensa, que quiere, qué necesita el consumidor final. Lo cual posee su lógica ya que es éste, el cliente, el que consuma el acto de adquirir un producto.

Es en este sentido que adquieren tanta relevancia estrategias como la de customer relationship management, más conocida por sus siglas CRM. Pero ¿qué es CRM? Según Martínez Sangil [17], CRM se puede definir como "la Filosofía empresarial, que toma como centro de gravedad de todos los procesos de la compañía, al cliente actual y potencial, con el objetivo final de adquirir clientes e incrementar su lealtad, mediante mecanismos técnicos, humanos y racionales que nos permitan conocer mejor al cliente".

Actualmente CRM posee módulos que proveen paquetes de información para 3 funciones específicas según Laudon & Laudon (2008) [16] las cuales son:

- Sales Force Automation (SFA), los cuales están enfocados en recolectar la mayor cantidad de información de manera se posea mayores argumentos de venta frente al cliente.
- Los módulos enfocados en servicio al cliente, enfocados en realizar una gestión de cuentas y clientes más efectiva, incorporando canales de información y relación como los Call Center o la Web.
- Pero sin duda el más relevante, así descrito por Laudon & Laudon, son todas aquellas aplicaciones orientadas a apoyar las campañas de marketing, es decir sistemas que utilicen la información del cliente, de modo que "Inteligencia de Clientes y Mercado" identifique los clientes con mayor propensión a realizar compras y aquellos que son más rentables.

El saber cuál es el producto que realmente requiere un cliente es uno de los puntos fundamentales el cuál describiremos como el ariete para atravesar la muralla invisible entre organización y consumidor, producto que no es suficiente con los módulos que posee actualmente CRM.

Y aún más allá, el conocer qué clientes realmente posee o poseerán la necesidad de adquirir un nuevo producto o servicio, permitirá que no se malgaste energía, tiempo y dinero de la organización en campañas innecesarias, permitiéndonos enfocar nuestros esfuerzos en un grupo específico de clientes a los cuales les ofreceremos lo que necesitan en el momento adecuado.

Por otra parte, no hay que dejarse engañar por la complejidad aparente de este problema. Después de todo este enfoque no es algo tan distinto a lo que ya realiza el vendedor de un almacén de barrio al atender a un vecino. Al ingresar al negocio, este generalmente ofrece justo lo que una persona necesita, haciendo uso de la larga experiencia y conocimiento acumulado sobre la persona y realizando un análisis rápido de que ofrecerle al cliente en base a la información que posee del comportamiento histórico que ha tenido el cliente. Por ejemplo la señora Juana, va a comprar todos los días de la semana a las 17:00 hrs pan. Además sabemos que el último mes ha tenido una marcada preferencia por el pan más tostado, basándose en esta información el vendedor al ver ingresar a su local a la señora Juana, realiza su recomendación y le ofrece llevar el pan tostado que necesitaba su cliente.

Replicando lo realizado por el vendedor del negocio de barrio, las grandes instituciones financieras como son los bancos, buscan establecer relaciones mucho más cercanas con sus clientes. En la actualidad existen diversas aplicaciones que usan información de clientes que están separadas en distintas plataformas, integrándose en bases de datos relacionales que brindan una visión única del cliente, de modo tal que se le puede ofrecer un servicio más ad-hoc a sus necesidades.

1.1. Características del problema a resolver

Para abordar de forma correcta la problemática planteada debemos conocer, de forma adecuada las características que posee nuestro problema, desde el contexto en que se desenvuelven nuestros productos, hasta las características propias de nuestros clientes.
Empecemos realizando un reconocimiento del terreno en que nos desenvolvemos.

1.1.1. Contexto global del problema

Actualmente las instituciones financieras son definidas por el artículo Nº 40 de la ley general de bancos. Esta definición señala que el giro básico es, captar dinero del público con el objeto de darlo en préstamo, descontar documentos, realizar inversiones, proceder a la intermediación financiera, hacer rentar esos dineros y, en general, realizar toda otra operación que la ley le permita. Además, la Ley enumera en su artículo Nº 69 otra serie de operaciones que pueden realizar los bancos en el país.
Los bancos también pueden desarrollar actividades complementarias y de apoyo a su giro, mediante sociedades. Algunas de estas sociedades quedan sujetas a la supervisión de la Comisión para el Mercado Financiero, en atención al tipo de actividad que desarrollan.
El desarrollo de estas actividades se enmarca en un creciente mercado nacional, debido que Chile ha desarrollado un eficiente sistema financiero, a la altura de las exigencias de su pujante economía. El empuje de la competencia ha demandado de los bancos mayor calidad y cantidad de servicios, esto producto de que en la actualidad operan en el país 25 bancos comerciales con evidente solvencia económica y que usan estrategias más innovadoras en son de tener mejores relaciones con los clientes.

Este trabajo de memoria se realizará para una entidad bancaria en particular. Las soluciones que se pretenden construir estarán enmarcadas en el contexto propio de esta entidad financiera. Pero, para poder entender de una manera adecuada el contexto en que se desenvolverá el sistema de recomendación, es necesario detallar que se generará con el objetivo de cubrir de una forma eficiente el marketing de productos bancarios, siempre teniendo presente

que se desenvolverá dentro una institución especialista en entregar soluciones financieras integrales, flexibles y convenientes, la cual tiene como objetivo mantener y mejorar la calidad de atención de un selecto grupo de clientes de los mercados de Grandes Corporaciones, Inversionistas Institucionales, Empresas y Personas. Debemos destacar que dentro de los principales pilares de la institución, se encuentra la orientación al cliente, la cual se describe como: “Ser un banco integrado por áreas de negocios especializadas, que nos permitan conocer al cliente, desarrollando relaciones de largo plazo y mutua conveniencia asegurando su lealtad”.

Finalmente, se debe tener en consideración que los sectores económicos dentro de los cuales la institución utilizada para esta memoria opera son:

- sector financiero.
- sector empresas del Estado.
- sector construcción.
- sector inmobiliario.
- sector minería.
- sector pesca.
- sector forestal.
- sector de la industria de la madera.
- sector agrícola y agro-industrial exportador.
- sector alimentos y bebidas.
- sector industrial y manufacturero.
- sector metal-mecánico.
- sector comercio y retail.
- sector automotriz.

- sector transporte.
- sector servicios básicos.
- sector hoteles, restaurantes y cines.
- sector servicios personales y a las empresas.
- sector educación.
- sector salud.
- sector organizaciones públicas y privadas.

Según el Panorama Bancario 2do Trimestre 2018 desarrollado por la Superintendencia de Bancos e Instituciones financieras, al cierre del primer semestre de 2018 las colocaciones del Sistema Bancario ascendieron a MM$ 168.153.105 (MMUSD 259.516), expandiéndose en doce meses en 6,56 %, por sobre la variación del cierre del trimestre pasado y de un año atrás. Por su parte, las principales carteras experimentaron una expansión en comparación a un año atrás. Se destacó el segmento de empresas que varió desde un 0,36 % en Junio del 2017 a un 6,58 % a Junio del 2018, (1,50 % Marzo del 2018), consumo alcanzó un 5,18 % (4,83 % Junio del 2017 y 3,97 % Marzo del 2018), mientras que vivienda se expandió un 8,74 %, dejando atrás el 7,19 % de Junio del 2017 y el 7,50 % de Marzo del 2018.

Pero, pese a estos montos que mueve la industria bancaria, podemos observar que aún se encuentra débil el sistema financiero en cuanto al concepto de bancarización, si consideramos que este involucra el establecimiento de relaciones de largo plazo entre usuarios e intermediarios financieros. En este sentido, no constituye bancarización el acceso puntual de un grupo de usuarios a un determinado tipo de servicios. En dicho contexto, la figura 1.1 muestra que un porcentaje muy acotado de los usuarios bancarios demanda la canasta completa de servicios financieros considerados. El grueso de ellos se relaciona con su entidad a través de un único producto, ante lo cual es claro que hay mucho que hacer en cuanto Cross-Selling se refiere.

	% de clientes con deuda durante cada año		
Tipo de crédito	**2003**	**2004**	**2005**
Sólo tarjetas de crédito	17%	10%	10%
Sólo créditos de consumo	37%	39%	38%
Sólo créditos para vivienda	14%	13%	12%
Tarjetas de crédito y créditos de consumo	19%	23%	25%
Tarjetas de crédito y créditos para vivienda	3%	2%	2%
Créditos de consumo y créditos para vivienda	5%	6%	6%
todos los anteriores	5%	7%	8%
Total Clientes (%)	**100%**	**100%**	**100%**

Cuadro 1.1: Distribución de los clientes bancarios según demanda de productos crediticios.
Fuente: Superintendencia de Bancos e instituciones financieras.

Por otra parte, podemos observar como el mercado bancario está adquiriendo una mayor relevancia en el uso de productos como las tarjetas de crédito, incluso provocando una disminución en la cantidad de tarjetas de casas comerciales, como se puede observar en la figura 1.2.

(en millones de tarjetas)	2004	2005
Número deTarjetas de Crédito Bancarias	2,7	3,8
Número de Tarjetas de Crédito Bancarias / Fuerza de Trabajo	0,3	0,4
Número de Tarjetas de Crédito Casas Comerciales	12,0	11,2
Número de Tarjetas de Crédito Casas Comerciales /Fuerza de Trabajo	1,2	1,1

Cuadro 1.2: Cantidad de tarjetas del mercado financiero nacional.
Fuente: Superintendencia de Bancos e instituciones financieras.

Hoy por hoy, la bancarización implica mucho más que el acceso al crédito. En efecto, para los usuarios comerciales el acceso a las cadenas de pago, al corretaje de seguros, a instrumentos de ahorro, a asesorías financieras y a operaciones de leasing, entre otras, es tan importante como el acceso al crédito. Tan relevante como los servicios financieros a los que pueda acceder, es el canal por el cual se desarrolle la comunicación entre el cliente y su institución. Durante los últimos años, han adquirido gran protagonismo los medios electrónicos, como ATM, Internet y los POS, los cuales dan cuenta de la aplicación, por parte de la banca, de tecnologías menos intensivas en mano de obra. Sin embargo, la banca se encuentra en deuda en cuanto a relaciones con el cliente se refiere y es en este sentido que se sustenta el acotado porcentaje de usuarios que demandan más productos bancarios, debido a que en la industria

nacional aún falta mucho trabajo por realizar en cuanto a Cross-Selling. Es por esto que se debe poder realizar una única referencia (artículo a recomendar) o un conjunto de ellas dependiendo del instante de vida del cliente bancario.

Por otra parte, se debe mencionar que el usuario que inyectará información al sistema es el cliente bancario, pero el usuario final del sistema recomendador es el banco o área encargada de realizar el marketing de los productos a la cual se le entregará el input de cada usuario, por lo que, a lo largo de este documento mencionaremos al cliente como la persona que inyectará información al sistema y en escasas ocasiones mencionaremos al usuario final del sistema.

Por otra parte, una de las principales características a tener en cuenta tiene que ver con la cardinalidad del problema, ya que en el caso bancario se poseen más usuarios (Y) que ítems (X), para cada periodo de tiempo.
Actualmente, el banco posee un total de 56.000 clientes, con un total de 39 productoscandidatos a ofrecer, sumado a que la cantidad de clientes aumenta a una razón mucho más elevada que la creación de nuevos productos.

Donde *i* denota los distintos tipos de clientes que posee el banco, mientras que *j* corresponde a los diversos productos ofrecidos por la institución.

1.1.2. Características del cliente bancario

Una de las principales consideraciones que debemos realizar es la naturaleza, cambiante y adaptativa de los clientes, que, lo largo de su vida, sufren transformaciones de naturaleza económica y social. Aunque estas transformaciones no ocurren constantemente, sí tienen un impacto fuerte en las características que describen al cliente en el momento en que ocurre, generando en ocasiones cambios radicales en éste. Un ejemplo de esto se produce cuando un cliente se gradúa y pasa al mundo laboral reflejando cambios sustanciales en su estado económico.

Por otra parte, la cartera de clientes está segmentada en:

- Personas Naturales, pertenecientes a comunas del sector oriente de Santiago de Chile.
- Corporaciones > UF. 4.000.000.
- Sector Financiero.
- Grandes Empresas Industriales y de Servicios [400.000, 4.000.000] UF.
- Grandes Empresas Agroindustriales y de Alimentos.
- Grandes Empresas Inmobiliarias y Constructoras.
- Empresas Medianas y Sucursales [5000, 400.000] UF.

Una vez desglosada la cartera de clientes, cabe señalar los segmentos a utilizar en esta memoria. Los sectores seleccionados para esta memoria, corresponde a las Personas Naturales y jurídicas, para las cuales se realiza un consideración simple para diferenciarlas. Esta consideración corresponde a tomar los RUT menores a 50 millones como personas naturales y los mayores a 50 millones como personas jurídicas, dentro de las cueles podemos encontrar los diversos sectores señalados con anterioridad.

Por otra parte, el comportamiento de los clientes no solo varía por condiciones económicas, también se puede apreciar comportamientos diversos según tramos de edad. Un ejemplo de esto es la participación en cuanto a deuda según edad como se muestra en la figura 1.1 donde observamos que los menores de 30 años y los mayores de 65 años deben cerca de $1,2 millones, mientras que los adultos entre 35 y 40 años deben cerca de $5,1 millones.

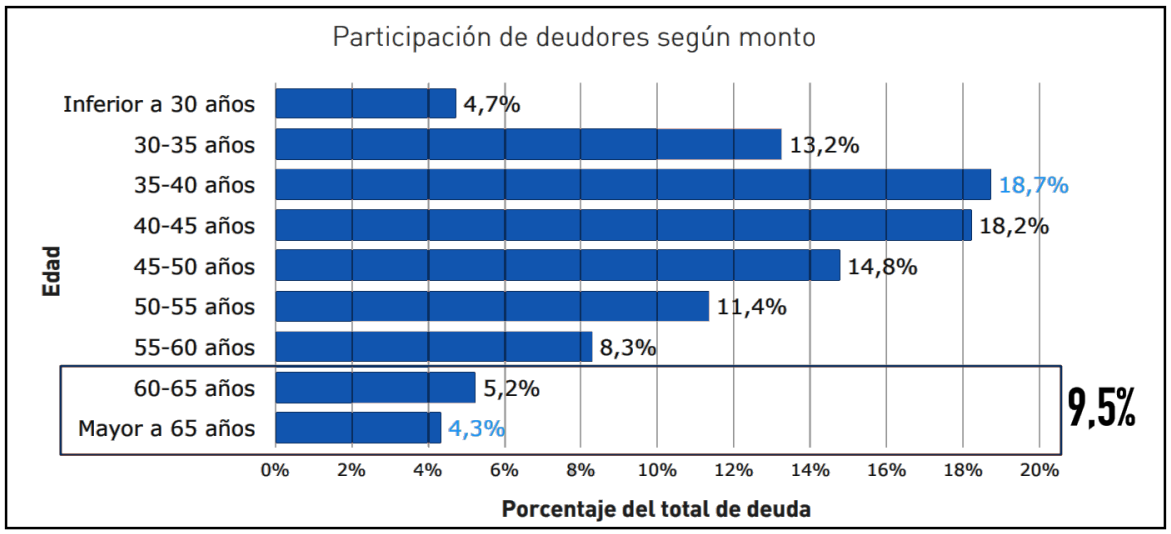


Figura 1.1: Porcentaje de participación por tramo de edad de los deudores. Según el monto total que poseen de deuda.
Fuente: Superintendencia de Bancos e instituciones financieras.

Además se debe destacar el comportamiento más arriesgado de los clientes más jóvenes a la hora de invertir versus un perfil más conservador al ir adquiriendo más edad. Es por esto que es clave realizar segmentaciones por tramos de edad.
La segmentación que se realiza en la institución bancaria de la cual provienen los datos es:

- Tramo 1: 18-24.
- Tramo 2: 25-34.
- Tramo 3: 35-44.
- Tramo 4: 45-54.
- Tramo 5: 55-64.
- Tramo 6: 65 o más.

1.1.3. Características de los productos

Dentro de los productos que nos podemos encontrar en el Banco tenemos los siguientes ítems, organizados en conjuntos según requisitos en común asociados a cada uno:

Créditos - Personas Naturales, es esta caso se requiere que la persona sea mayor de 22 años, ser profesional universitario, ingresos superiores a UF 120 o su equivalente en la moneda nacional.

- Productos asociados:
  - Cuenta Corriente.
  - Linea de Sobregiro.
  - Crédito de Consumo.
  - Tarjeta de Crédito.
  - Tarjetas de Créditos adicionales, vigentes y PLUS.
  - Linea de Crédito de protección.

Créditos Hipotecario, Se requiere que la persona posea una edad mínima de 25 años, ser profesional universitario o ejercer una actividad por mínimo 5 años e ingresos superiores a UF 120 o su equivalente en la moneda nacional. Además el valor mínimo a financiar es de inmuebles a financiar, será de UF6.500 tratándose de casas y de UF 4.500 tratándose de departamentos, oficinas y otros.

Persona Jurídica Consumidora, en este caso se hace referencia a las Micro y Pequeñas Empresas, según Ley 20.416, establecidas en Chile, las cuales deben registrar un patrimonio mínimo de UF 4.000.

- Productos asociados:
  - Cuenta Corriente.
  - Crédito de Consumo.

- Otras operaciones financieras.

Seguro de Desgravamen crédito en cuotas, en este caso se posee el requisito de edad máxima de cobertura equivalente a 80 años.

Seguro Contra Incendio + Sismo El monto asegurado debe ser menor a UF 25.000.

Seguro Contra Incendio Al igual que el indicado anteriormente el monto asegurado debe ser menor a UF 25.000.

Póliza de Seguros Colectiva de Desgravamen con Adicional de Invalidez Total y Permanente Dos Tercios, tiene como requisito edad máxima de ingreso de 59 años y 364 días y la edad máxima de cobertura de 64 años y 364 días.

Dentro de los instrumentos que son más interesantes o atractivos de adquirir por un cliente del banco una vez que posee una cuenta, tenemos:

- **BIA:** Es una línea en función de los activos financieros (acciones, fondos mutuos, depósitos a plazo, bonos, etc.), para libre disponibilidad (viajes, autos o proyectos) o para inversiones en instrumentos financieros en todo el mundo.
- **Factoring:** Es un contrato a plazo fijo, a través del cual la empresa de Factoring o factor, compra a sus clientes las cuentas por cobrar originadas por ventas a plazos, las cuales pueden estar constituidas por facturas, letras o cualquier otro título ejecutivo que implique una obligación que deba ser pagada en un plazo determinado.
- **Depósitos a Plazos:** Es una alternativa para que puedas invertir tu dinero, eligiendo aquella opción que más se adecue a tus necesidades:
  - Moneda nacional o extranjera.
  - Plazo fijo o renovable.
  - Plazos cortos, medianos y largos.
  - Depósitos acogidos a los Beneficios Tributarios.

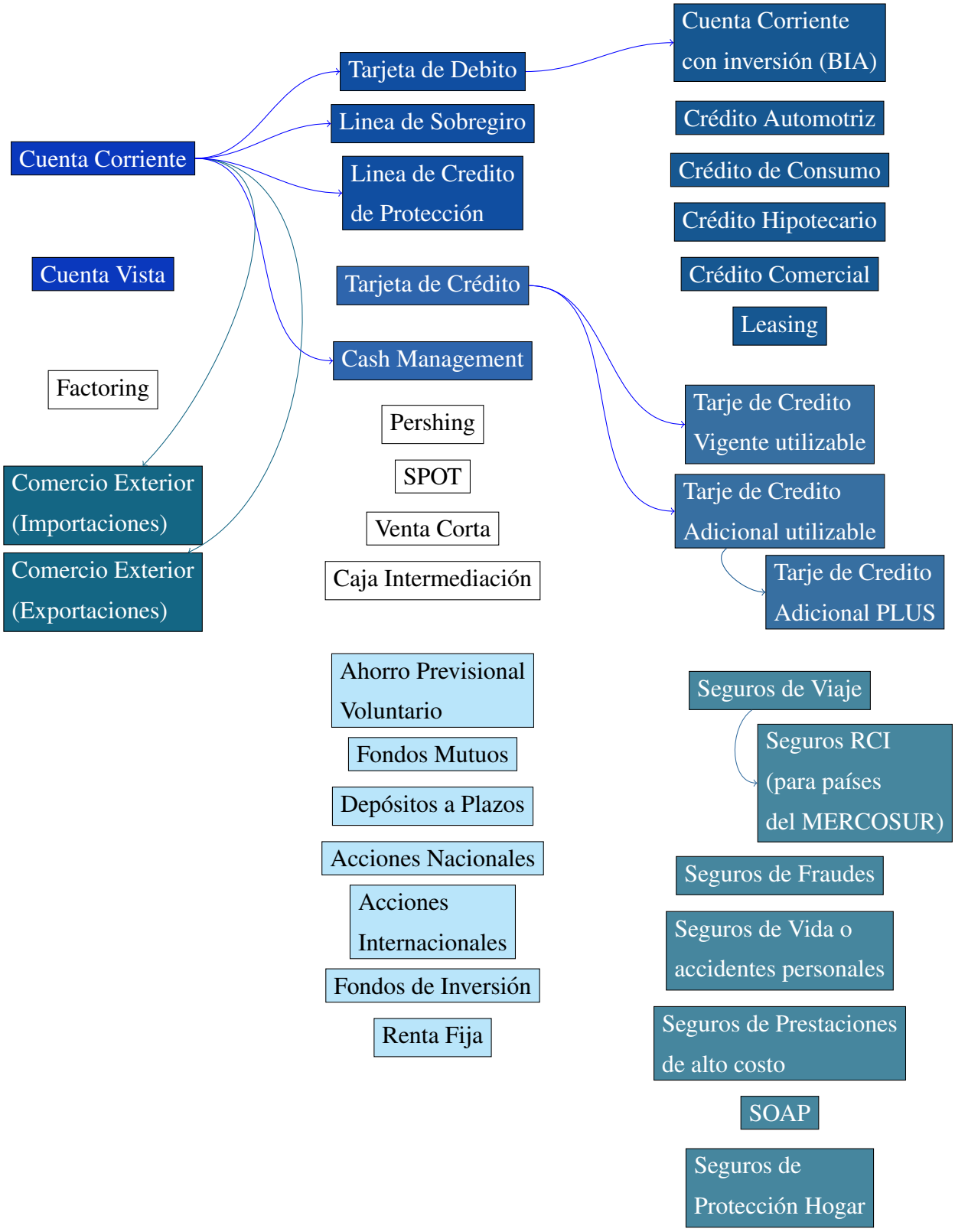


Figura 1.2: Grafo de productos bancarios, la flecha representa la dependencia de un producto respecto a otro.

Fuente: Elaboración propia.

1.2. Objetivos

En el presente documento, se propone la implementación y estudio de diversos métodos que generen sistemas recomendadores para productos pertenecientes al ámbito bancario. Utilizando estas implementaciones se realizarán pruebas en base a distintos parámetros y tipos de clientes para evaluar y poder realizar un análisis contrastante entre ellos, de modo que se logre determinar en qué situaciones son eficientes las nuevas implementaciones.

Por consiguiente, los objetivos para el presente trabajo se desglosan en:

1.2.1. General

- Diseñar e implementar a lo menos dos métodos de recomendación para productos bancarios utilizando y/o adaptando diferentes técnicas del estado del arte.

1.2.2. Específicos

- Determinar al menos dos métodos existentes en la literatura actual que permitan realizar recomendaciones acorde a las necesidades de la organización.
- Proveer al menos dos métodos que permitan resolver el problema considerando la eficacia y eficiencia de cada uno.
- Determinar en qué escenarios se desenvuelven de forma más eficiente las técnicas propuestas.

1.3. Observaciones técnicas

Todas las evaluaciones experimentales que son parte del trabajo presentado se han realizado en hardware local, con el uso de hasta 4 núcleos de procesamiento paralelo, para 8 subprocesos, con 16GB de memoria Ram.

Las pruebas e implementación se han realizado en Jupyter Notebook, con Python 3.7 de 64 bits y las bibliotecas ampliamente utilizadas han sido:

- Pandas (https://pandas.pydata.org/).
- NumPy (http://www.numpy.org/).
- scipy (https://www.scipy.org/).
- SQLalchemy (https://www.sqlalchemy.org/).
- Scikit-learn (https://scikit-learn.org/stable/).

Capítulo 2

Estado del Arte

En este capítulo se presentarán una serie de definiciones necesarias para así comprender el objetivo del trabajo propuesto, así como los métodos y técnicas empleadas.

2.1. Sistemas Recomendadores

Desde la década de 1990 los sistemas de recomendación se han convertido en un elemento relevante para diversas áreas de investigación, producto de sus abundantes aplicaciones prácticas para la industria. Su principal riqueza se basa en las recomendaciones personalizadas de los contenidos y servicios para el usuario. Un ejemplo clásico de esto es la recomendación de productos en los servicios de películas y series. Sin embargo, los actuales sistemas de recomendación no se adaptan a los diversos escenarios que posee la industria, producto de esto es necesario realizar mejoras y generar nuevas variantes en las técnicas conocidas, de modo que estas sean más efectivas. En este documento, se abordarán formas de ampliar la variedad de escenarios que abarcan los sistemas recomendadores, centrándose en las características del caso descrito con anterioridad, lo cual nos permite incrementar los beneficios de la institución por medio de Cross-Selling aumentando la retención de clientes:

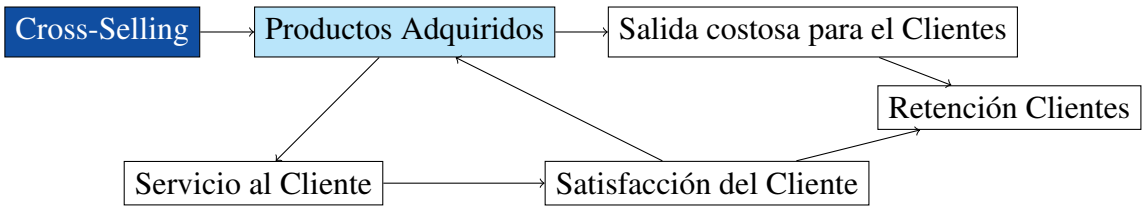


Figura 2.1: Grafo correspondiente al efecto del Cross-Selling en la retención del cliente. Fuente: Elaboración Propia.

2.2. Principales categorías de sistemas recomendadores

En su estado más simple, el problema de realizar una recomendación se reduce a estimar calificaciones para artículos que no se han visto o revisado por el usuario. Dicha estimación se basa en las calificaciones ya realizadas por el usuario, de esta forma se tiene una función que mide la utilidad de un artículo s a cierto usuario c.

De acuerdo a Gediminas Adomavicius y YoungOk Kwon [1], el problema de recomendación puede ser formalizado del siguiente modo.

Se tiene un conjunto de *usuarios* C que han expresado preferencias sobre determinados *ítems* calificables S. La preferencia expresada por un usuario sobre un ítem es llamada *rating* y se puede expresar como una función $u : C \times S \rightarrow \mathbb{R}$, denominada también función de utilidad. Algunos utilizan una escala de valores enteros o reales de 1 a 5 generalmente, como se puede apreciar en el clásico rating por estrellas. Además se utilizan rating con escala binaria, para afirmar si a un usuario le gusta (1) o no (0) un ítem.

La tríada usuario, ítem, rating se puede agrupar en una matriz denominada *matriz de ratings*, en la cual cada fila representa un usuario y las columnas un ítem, mientras que el valor es el rating respectivo. Si todas las entradas de esta matriz fuesen conocidas, el problema de recomendación se reduciría a encontrar

$$S'_c = argmax_{s \in S} u(c, s) \qquad \forall c \in C,$$

Sin embargo, como muchos ítems no han sido calificados por todos los usuarios o bien existen muchos usuarios que han calificado solo algunos ítems, la matriz de rating tiene muchas en entradas desconocidas. Por lo tanto, el objetivo de los sistemas de recomendación, se puede subdividir en dos. La **predicción** de ratings y la **recomendación** de ítems, con el fin de encontrar una lista de objetos que sean de gran interés para un usuario en particular.

De manera clásica se distinguen seis principales clases o tipos de sistemas de recomendación [3]:

- **Basados en el Contenido:** Generan recomendaciones centrados principalmente en la información que extraen a partir de los objetos. Por ejemplo el género o la categoría de un libro.
- **Filtrado Colaborativo:** Se utilizan usuarios que poseen gustos afines para poder estimar recomendaciones de objetos similares a partir de dichos gustos.
- **Basados en el Contexto:** Según las características generales de los usuarios se realizan recomendaciones. Por ejemplo se utiliza, la edad, género o características demográficas.
- **Basados en Conocimiento:** Se consideran las necesidades o intereses del usuario para realizar recomendaciones.
- **Basados en comunidades:** Se recomienda un ítem en función de los amigos que posee el usuario, ”Dime con quién andas y te diré quién eres”.
- **Híbridos:** Es la combinación de dos o más enfoques mencionados anteriormente, usualmente se utiliza el basado en contenidos en conjunto al colaborativo.

En este documento nos centraremos en tres categorías: las basadas en contenido, las colaborativas y las híbridas, que incluyen elementos de las dos primeras. En las secciones siguientes se profundiza el análisis de este tipo de sistemas de recomendación.

2.2.1. Recomendaciones basadas en el contenido

Este método se basa principalmente en que tan útil es cierto elemento para un usuario basado en el contenido que posee dicho elemento, por ejemplo para un libro el sistema buscará puntos en común entre los libros ya calificados por el usuario (personas, tema, editoriales, géneros, etc.) con lo cual se obtendrán libros con alto grado de similitud. Cabe destacar que el contenido generalmente se describe en palabras claves para estos casos.

Estos sistemas se basan principalmente en perfiles que poseen información sobre preferencias, gustos y necesidades del usuario, de modo que se pueda atribuir cierto artículo al usuario. Para esto es necesario poseer ponderaciones (pesos) para las palabras claves, para lo cual se pueden ponderar en torno a la frecuencia en el documento/artículo ($TF-IDF$). [AW 1998]

Por ejemplo consideremos a Valeska, quien posee preferencias a ciertos términos claves los cuales fueron obtenidos a partir de sus gustos registrados con anterioridad o en base a la información histórica que se posee sobre ella. Las palabras claves para Valeska son: libros, inglés, académicos, literatura. Esto se debe a que Valeska es una académica especialista en la enseñanza de la lengua inglesa. Teniendo en consideración estos términos se buscará recomendar el evento ideal para el perfil de ella. El primer evento corresponde a una charla sobre los principales libros en inglés. El segundo es una exposición de literatura contemporánea. Mientras que el tercer evento es una reunión de académicos. Una vez que poseemos una glosa de descripción de los eventos podemos realizar un conteo de la cantidad de veces que aparece una palabra clave de la académica en la descripción de cada evento, teniendo coincidencias en libros e inglés para el primer evento, literatura para el segundo evento y académicos para el tercer evento, ante lo cual podemos ver que el primer evento es quien posee más términos o palabras en común por lo cual es el evento a ser recomendado para Valeska.

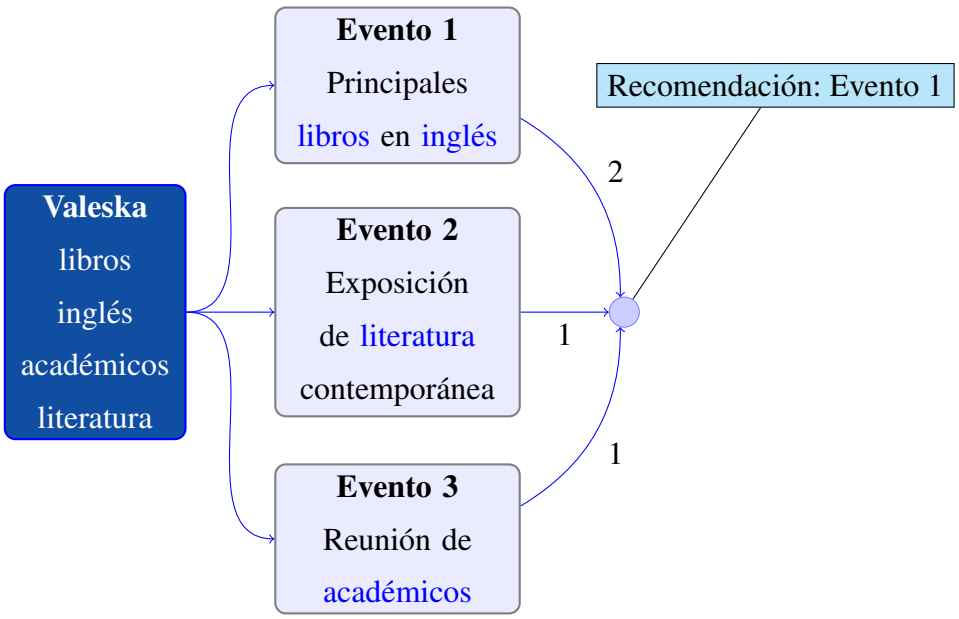


Figura 2.2: Funcionamiento de un Sistema Recomendador Basado en Contenido.
Fuente: Elaboración Propia.

Se deben considerar las limitaciones de estos sistemas. Las técnicas basadas en el contenido están limitadas por las características asociadas a los artículos, ya que, se debe poseer un conjunto vasto de características, las cuales se deben proporcionar automáticamente, lo cual posee un elevado costo computacional, o manualmente. Además, se pueden dar casos donde los elementos a analizar poseen características sumamente parecidas por lo que el sistema no logra distinguirlos como artículos distintos.

El enfoque basado en el contenido tiene sus raíces en la recuperación de información y el filtrado de información, los cuales se basan en el filtrado de agrupaciones de información, teniendo sus principales aplicaciones basadas en texto. La mejora que se realiza con respecto a estos enfoques es la incorporación de un perfil de usuario que contiene información sobre gustos y preferencias.

Según [2] “La información de perfil puede ser obtenido de los usuarios de manera explícita, por ejemplo, a través de cuestionarios, o implícitamente, aprendió de su comportamiento

transaccional con el tiempo".

Una de las problemáticas que surgen de las técnicas basadas en contenido es que están limitadas por las características que se asocian de forma explícita con los objetos que estos sistemas recomiendan. Por consiguiente para poseer un conjunto suficientemente nutrido de características, se debe utilizar formatos que nos permitan realizar un análisis automático por un computador (por ejemplo, texto) o de lo contrario dichas características deben ser asignadas a cada elemento manualmente. Otro problema que suele darse es que las características asociadas a artículos distintos pueden llevar al sistema a creer que se trata del mismo producto de la utilización de los mismos términos.

Por otra parte, se debe considerar la **superespecialización** de los usuarios, lo cual no nos permite diversificar los artículos o productos a recomendar, ya que, el usuario posee un universo acotado de productos calificados, con lo cual se hará imposible para el sistema recomendar artículos con otro tipo de característica que los ya calificados, convirtiéndose en uno de los principales problemas con los que se debe lidiar. La superespecialización, se da cuando un sistema solamente puede recomendar artículos que puntúan alto, lo cual esta descrito con mayor profundidad en el documento Toward the next generation of recommender system de Gediminas Adomavicius y Alexander Tuzhilin [2]. Esto se produce cuando el sistema se limita a recomendar al usuario elementos similares a los ya utilizados. Este es un problema clásico en algoritmos de inteligencia artificial, donde solo se realiza intensificación en una zona y jamás se diversifica la búsqueda a nuevas zonas a explorar, concentrándose el trabajo en una única ubicación, una analogía clásica de este problema es la de un minero buscando minerales en cierta zona, si este encuentra una veta de plata, este concentrara sus esfuerzos en esa zona para extraer tanta plata como sea posible, pero si este nunca más diversifica su búsqueda, jamás se dará cuenta que a su lado se encontraba una veta de oro.

Finalmente, nos encontramos con que se debe considerar que un usuario debe evaluar con anterioridad artículos para que el sistema pueda realizar recomendaciones, lo cual es una dificultad si dicho usuario es nuevo en el sistema, ya que, tendría muy pocas calificaciones

realizadas y por tanto no se pueden realizar recomendaciones precisas.

2.2.2. Recomendaciones colaborativos

Estos sistemas recomiendan en función de las calificaciones realizadas por otros usuarios, los cuales poseen características de perfil similares al usuario objetivo, es decir se busca la utilidad $u(c, s)$ de cierto artículo s para un usuario c en base a $u(c_j, s)$, el rating asignado por otro usuario c_j al artículo s. Por ejemplo, volviendo al caso de los libros, para usuarios con gustos similares en literatura se recomendaran los libros más "valorados" por sus pares.

Para esta categoría, nos centraremos principalmente en un enfoque probabilístico para el filtrado colaborativo, donde las calificaciones son enteros entre 0 y n, y la expresión de probabilidad corresponderá a la probabilidad de que un usuario c califique positivamente a un artículo s, reduciéndose el problema principalmente al cálculo de esa probabilidad.

Como ya mencionamos, los sistemas recomendadores con filtrado colaborativo podemos agruparlos 2 dos clases: Los basados en memoria y los basados en modelos. En la tabla 2.1 podemos ver sus características.

	Basados en Memoria	**Basados en Modelos**
Técnicas	**-** K vecinos más cercanos (kNN)	**-** Clustering **-** Factores Latentes **-** Métodos Espectrales
Ventajas	**-** Rápida implementación **-** Pérmiten agregar datos fácilmente y de manera incremental	**-** Mayor escalabilidad **-** Superean problemas de escasez de datos
Desventajas	**-** El desempeño decrece en matrices dispersas **-** No escalan bien en data sets a gran escala	**-** Construir el modelo es costoso **-** Puede existir perdida de información útil

Cuadro 2.1: Categorías de filtrado colaborativo.
Fuente: Memoria de titulación Nicolás Torres, Sistemas de recomendación basados en métodos de filtrado colaborativo.

En esta memoria se estudiarán los procedimientos basados en memoria. Revisando principalmente K-vecinos más cercanos (kNN).

Según Shardanand y Maes [24], existen tres fases principales en el funcionamiento de los sistemas colaborativos:

- El sistema crea un perfil de cada usuario con sus preferencias respectivas.
- Se mide el grado de similitud entre los distintos usuarios dentro del sistema y se crean grupos de usuarios con gustos parecidos.
- El sistema estima la preferencia de un usuario en base a la información recopilada previamente.

Estas fases las consideraremos para los algoritmos a describir a continuación.

Los algoritmos basados en memoria son heurísticas que realizan predicciones de calificaciones $r_{c,s} = u(c, s)$ desconocidas para un usuario c en base a las calificaciones previamente realizadas por otros usuarios c', que de alguna forma de estima que son ‘ similares” a c . Una forma de medir la similaridad es a través de la correlación de Pearson (PC) [23],

$$PC(c, c') = \frac{\sum_{i=1}^{n}(r_{c,i} - \overline{r}_c)(r_{c'i} - \overline{r}_{c'})}{\sqrt{\sum_{i=1}^{n}(r_{c,i} - \overline{r}_c)^2(r_{c',i} - \overline{r}_{c'})^2}}, \tag{2.1}$$

donde $r_{c,i} = u(c, s_i)$ y $\overline{r}_c$ es la media de los rating del usuario c. Un valor de $PC(c, c') = 1$ corresponde a una similaridad perfecta entre c y c' y un valor -1 a su complemento. En este contexto, podemos elegir solo correlaciones positivas para mejorar la predicción.

Una vez calculada la similaridad entre usuarios, el valor $r_{c,s} = u(c, s)$ se puede estimar con alguno de los métodos que se muestran a continuación [23]

$$\begin{aligned} &(a)\ r_{c,s} = \frac{1}{N}\sum_{c' \in C} r_{c,s} \\ &(b)\ r_{c,s} = k\sum_{c' \in C} PC(c, c') \cdot r_{c',s} \\ &(c)\ r_{c,s} = \overline{r}_{c,s} + k\sum_{c' \in C} PC(c, c') \cdot (r_{c',s} - \overline{r}_{c'}) \end{aligned} \tag{2.2}$$

donde multiplicador k sirve como un factor de normalización y normalmente se selecciona como $k = \frac{1}{\sum_{c' \in C} |PC(c,c')|}$.

Si bien esta implementación basada en usuarios (user-based) captura las recomendaciones y detecta patrones complejos, no logra incorporar un patrón único respecto de un ítem y, al ser dispersos los datos, pares de usuarios con pocos ratings son propensos a arrojar correlaciones sesgadas que pueden dominar el vecindario del usuario en cuestión.
El algoritmo puede ser implementado incluyendo a todos los usuarios del dataframe como vecinos de cada usuario, aunque se mejora la precisión y eficiencia. Aun así, su implementación es costosa ya que requiere comparar cada usuario con el dataframe completo, por lo

que el tiempo y memoria para procesamiento no escalan bien conforme aumentan los usuarios y los ratings. Hay técnicas que se emplean para reducir este costo: Subsampling es un ejemplo, donde se selecciona una muestra de la población antes de realizar el procesamiento y se encuentra de forma rápida los clusters de usuarios similares al target, entonces los vecinos cercanos pueden ser elegidos de los clusters más similares. Sin embargo resulta mucho más atractivo realizar el mismo análisis traspuesto, es decir mientras que los algoritmos user-based generan predicciones basadas en similaridades entre usuarios, los item-based lo hacen basándose en similaridades entre ítems, es decir, la predicción para un ítem se basa en ratings para ítems similares. Así, una predicción para un usuario c de un ítem s puede ser representada como la composición de sumas ponderadas de ratings del mismo usuario para los ítems más similares.

$$PC(s, s') = \frac{\sum_{c \subset RB_{s,s'}} (r_{cs} - \overline{r}_c)(r_{cs'} - \overline{r}_c)}{\sqrt{\sum_{c \subset RB_{s,s'}} (r_{c,s} - \overline{r}_c)^2} \sqrt{\sum_{c \subset RB_{s,s'}} (r_{c,s'} - \overline{r}_c)^2}}$$

Donde el set $RB_{s,s'}$ corresponde al set de usuarios que han dado rating tanto a s como a s' [23].

2.2.3. Enfoque híbrido

Este método se basa en la mezcla de los enfoques descritos con anterioridad, donde se busca generar una combinación entre las calificaciones generadas por el usuario basadas en el contenido y las probabilidad generada por un sistema colaborativo, tomando lo mejor de ambos mundos y generando una combinación, típicamente lineal, de ambos resultados.

2.3. Método de Domeniconi, utilizando técnicas de aprendizaje supervisado

Los sistemas de Aprendizaje Supervisado aprenden cómo combinar entradas para producir predicciones útiles sobre datos que no han sido revisados con anticipación, permitiendo que los computadores sean capaces de realizar tareas de alta complejidad, propias de los seres humanos.
Para lograr realizar dichas tareas es necesario, que los algoritmos sean capaces de clasificar un volumen de datos, pudiendo identificar patrones entre ellos, para esto se utilizan una serie de técnicas basadas en el paradigma de "aprender de la experiencia".

En este sentido, las técnicas computacionales capaces de predecir de forma fiable nuevas anotaciones con un determinado valor de probabilidad en base a los datos históricos que se poseen toman especial relevancia en diversos escenarios. Uno de ellos es el planteado por Giacomo Domeniconi en el 2016 en su tesis doctoral [4] en el cuál busca predecir nuevas anotaciones biomoleculares de forma que se descubran propiedades y funciones de los genes biológicos, sobre todo de aquellos organismos que se han estudiado recientemente.

Además, Domeniconi analiza varias técnicas planteadas previamente en esta materia entre ellas técnicas de clasificación como árboles de decisión y redes bayesianas [14], k-vecinos más cercanos (kNN) [26] y máquinas de vectores de soporte (SVM) [19], entre otros. Lamentablemente los resultados obtenidos mostraron una precisión limitada.

Por consiguiente, Domeniconi busca generar un modelo flexible, capaz de mejorar el rendimiento de predicción y de analizar una gran cantidad de datos disponibles, lo cual logra realizar gracias al aprendizaje supervisado.

Para entender mejor en que consiste el Aprendizaje supervisado, en primera instancia debemos saber diferenciar algunas terminologías claves, como etiquetas y atributos.

- **Etiquetas:** Una etiqueta es el valor que buscamos predecir, es decir, la variable. La

etiqueta podría ser el precio futuro del pan, la probabilidad de adquisición asociada a un determinado artículo, entre otros.

- **Atributos:** Corresponde a una variable de entrada, la cual suele ser representada como vectores. Donde cada dimensión/entrada de ese vector es un atributo o pieza de información que ayudará a predecir la etiqueta.

- **Modelos:** Los modelos establecen la relación existente entre los atributos y las etiquetas, para esto se realizan dos fases esenciales: El entrenamiento y la inferencia.

  Por una parte, el **entrenamiento** corresponde a la fase en que el modelo determinada la relación existente entre atributos y etiquetas gradualmente. Por otra parte, la **inferencia** corresponde a la etapa en que se le atribuyen las nuevas etiquetas a aquellos datos que aún no han sido clasificados o etiquetados, es decir se utiliza el modelo entrenado para realizar las predicciones correspondientes.

Una vez explicados algunos términos claves dentro del aprendizaje supervisado podemos ahondar más en el trabajo realizado por Domeniconi quien buscaba predecir si un gen g pertenecía o no a un término[1] t, para lo cual primero se establece una representación basada en:

$$M(i, j) = \begin{cases} \text{1 si el término t posee el gen g o si este gen está presente} \\ \text{en cualquiera de sus de sus descendientes} \\ \text{0 Si no} \end{cases}$$

Dicha tenencia es representada en una matriz binaria donde la clase a predecir es un término, es decir, la columna de la matriz que puede poseer como valor un 0 o 1 según la ausencia o presencia de un gen en el término correspondiente. En palabras más simples, si un término en específico poseía un gen determinado, esto es representado en la columna correspondiente a ese término y la fila correspondiente al gen con un 1 y en caso que dicho término no tenga a ese gen, se representa con un 0. De esta forma, para cada combinación de gen y término, tenemos anotaciones correspondientes a los valores 0 cuando no lo posee o 1 cuando si pertenece dicho gen al término. Si consideramos los cambios de esta matriz en el tiempo, tenemos una representación de términos que evolucionan o mutan en el tiempo. Por ejemplo,

[1] Secuencia alineada de genes, que sufre mutaciones en el tiempo.

el término $t12$ podría haber poseído al gen 048 pero, al sufrir una mutación y convertirse en el término $t13$, ya no posee dicho, lo que es representado en nuestra matriz de la siguiente forma:

Matriz de tenencia		
	t12	**t13**
Gen 048	1	0

Cuadro 2.2: Extracto de una matriz de tenencia para la representación genética de Domeniconi.
Fuente: Elaboración Propia.

En la matriz anterior observamos que $t12$ posee un 1 para el gen048 pero esto cambia a un 0 en el término $t13$ ya que no posee este gen. De igual manera, se hace esto para cada término con sus correspondientes descendientes construyendo así una matriz de anotaciones de Domeniconi.

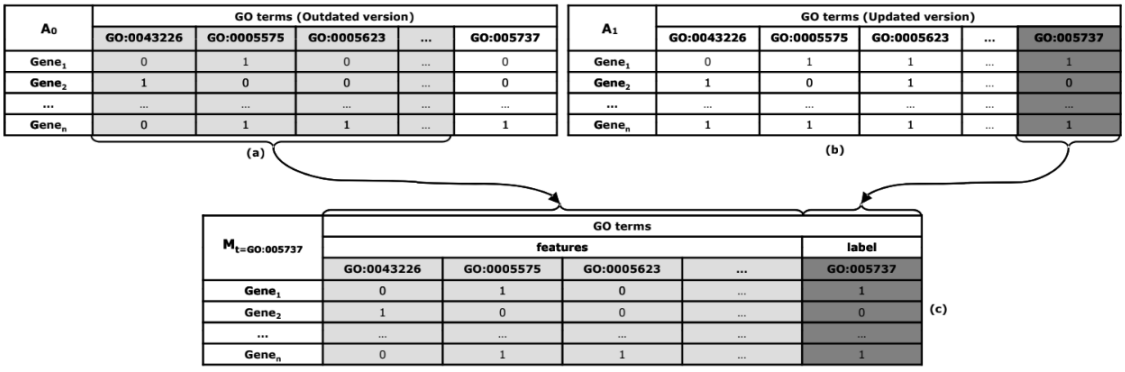


Figura 2.3: Diagrama ilustrativo de la representación de un conjunto de datos para el modelo de predicción de Domeniconi.
Fuente: Data and Text Mining Techniques for In-Domain and Cross-Domain Applications, Domeniconi.

Una vez teniendo las representaciones en forma matricial, se requiere dos versiones de la matriz de anotación para poder crear un modelo supervisado, debido a que dichas matrices

serán utilizadas como conjuntos de entrenamiento para el sistema. Por otra parte, se debe considerar que los biólogos almacenan únicamente la matriz de anotaciones más recientes o en otras palabras las informaciones más recientes de los genes, sin mantener versiones obsoletas de dicha información debido a razones de espacio, producto de la gran cantidad de datos que se requiere para todos los genes. Por lo tanto, para poseer dos versiones de matrices, se toma la matriz actual y se le realizan perturbaciones aleatorias, es decir de forma aleatoria se cambias valores de la matriz actual de modo que esta nueva matriz generada se asemeje a una versión más antigua de anotaciones, del mismo modo se genera una matriz futura de anotaciones o una matriz que representa los futuros cambios que sufrirá la matriz original de la cual se posee información.

Una vez que poseemos estas tres matrices:

- La matriz previa de perturbaciones (Un0).
- La matriz original de anotaciones.
- La matriz de perturbaciones futuras.

Son utilizadas para generar los conjuntos de entrenamiento y de validación, como se puede ver en la figura 2.4.

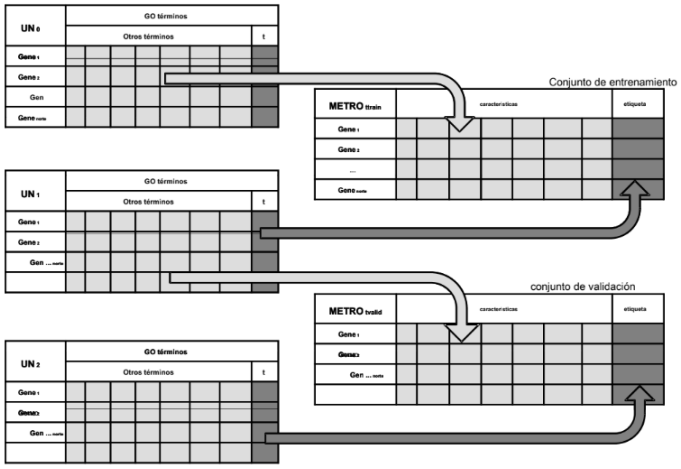


Figura 2.4: Diagrama ilustrativo de la representación de un conjunto de datos de entrenamiento y validación.
Fuente: Data and Text Mining Techniques for In-Domain and Cross-Domain Applications, Domeniconi.

El conjunto de entrenamiento se crea con una versión más antigua de la matriz de anotaciones denominada como $UN0$ por las características y una versión actual de anotaciones, llamada $UN1$ para generar las etiquetas. Del mismo modo, establece el conjunto de validación, el cual es creado utilizando la matriz $UN1$ y la matriz de anotación futuras $UN2$. Durante el experimento Domeniconi probó la eficiencia para descubrir nuevas anotaciones en los genes a partir de la información de anotaciones que los biólogos tenían disponibles utilizando diferentes modelos supervisados. Además, considerando que el método propuesto puede aplicar cualquier algoritmo supervisado que devuelva una distribución de probabilidad, Domenicoconi probó con diferentes algoritmos existentes con el fin de medir su eficiencia, en particular: Las máquinas de vectores de soporte, vecinos más cercanos, arboles de decisión y regresión logística.

Con el fin de determinar la calidad de los resultados, se realizan los siguientes procedimientos:

1. Se extrajeron las anotaciones de entrada desde una versión obsoleta de la matriz de

genes, excluyendo las anotaciones que son menos fiables.

2. De forma aleatoria se realizan perturbaciones en la matriz de anotaciones.
3. Mediante la utilización de un algoritmo de predicción se genera una lista de anotaciones ordenadas por su valor de confianza.
4. Se selecciona la parte superior de las predicciones (250) y se contaron el número de predicciones que se encontraron en la versión actualizada que poseían los biólogos.
5. Posteriormente se repiten los pasos 2, 3 y 4, 10 veces mediante la variación de la semilla aleatoria usada para la generación de las perturbaciones aleatorias.

Una vez explicados algunos términos claves y metodología utilizada por Domniconi es necesario aclarar algunos conceptos. Uno de los fundamentales es explicar en qué consisten las técnicas de clasificación y las de regresión.

2.3.1. Modelos de Clasificación

Como vimos anteriormente, diversos autores aplicaron modelos de clasificación en el problema de minería genética, pero ¿en qué consisten estos modelos? Los modelos de clasificación determinan valores discretos a predecir, por ejemplo si en una determinada imagen aparece, un perro, un gato o un ratón, o si un correo electrónico es o no es SPAM.

Dentro de los algoritmos de clasificación más comunes nos encontramos con:

- Regresión Logística.
- kNN.
- Redes Bayesianas.
- SVM.

- Arboles de Decisión.
- Bosques de Decisión.

En el caso de los **modelos de clasificación** nos enfocaremos en los K vecinos más cercanos, más conocido como **"k-Nearest Neighbor"(kNN)**, el cual busca en las observaciones más cercanas a la que se está tratando de predecir y clasifica el punto de interés basado en la mayoría de datos que le rodean.
kNN es ampliamente utilizado en la resolución de una multitud de problemas, como en sistemas de recomendación, búsqueda semántica y detección de anomalías. Dentro de las principales consideraciones que se debe tener al momento de utilizar kNN se encuentran como argumentos a favor su sencillez al momento de implementarlo y como contra que utiliza todo el dataset para entrenar "cada punto" y por consiguiente utiliza mucha memoria y recursos de procesamiento. Por estas razones kNN tiende a funcionar mejor en datasets pequeños y sin una cantidad enorme de features (las columnas).

2.3.2. Modelos de Regresión

Los modelos de regresión predicen valores continuos, por ejemplo establecer el valor de probabilidad que posee cierta ocurrencia. Entre las técnicas más aplicadas tenemos:

- Regresión Lineal Simple.
- Regresión Lineal Múltiple.
- Regresión Polinómica.

En este documento nos enfocaremos en explicar más en detalles solo algunas de estas técnicas. Empezaremos por la más conocida entre todas, la **Regresión Lineal Simple**.

La Regresión Lineal Simple consiste en utilizar diversos métodos para minimizar la distancia o error entre los valores y la función asociada, el método de mínimos cuadrados para encontrar la recta que resulta en la menor suma de errores al cuadrado (RMSE: Root Mean Square

Error). La palabra simple se refiere a que la variable respuesta solo depende de 1 variable independiente: Y = f(X). Por otra parte nos encontramos con la **Regresión Lineal Múltiple** la cual posee el mismo objetivo que la simple, pero considerando múltiples variables a utilizar como parámetros de entrada.

2.4. Cadena de Markov

Un proceso de Markov es un proceso estocástico que sirve para representar secuencias de variables aleatorias no independientes entre sí, pero donde se introduce el supuesto fundamental de que la probabilidad de observar un determinado esto en el futuro, sólo depende del estado actual.

Por ejemplo en el caso del Marketing Bancario, si la variable aleatoria consiste en saber que productos tendrá un determinado cliente, entonces saber que productos posee hoy puede servir para calcular cuales poseerá mañana y no es necesario saber cuáles poseía hace un mes o un año.

2.4.1. Propiedades de Markov

Sea $X = (X_1, ..., X_T)$ una secuencia de variables aleatorias que toman valores en un conjunto finito S = $(s_1, ..., s_N)$ que se llama el espacio de estados. Entonces, las propiedades de Markov son:

- Horizonte Limitado:
  $P(X_{t+1} = s_k|X_1, ..., X_t) = P(X_{t+1} = s_k|X_t)$
- Invariante en el tiempo:
  $P(X_{t+1} = s_k|X_1, ..., X_t) = P(X_2 = s_k|X_1)$

Si X cumple estas dos propiedades, se dice que X es una cadena de Markov o que tiene la propiedad de Markov.

2.4.2. Modelos Ocultos de Markov

Un modelo oculto de Markov (discreto) es una tupla $M = \{S, \sum, A, B, \prod\}$ donde:

- S es un conjunto finito de estados denominados ocultos.
- $\sum$ es un conjunto finito de estados denominados observables.
- A es la matriz de probabilidades de transición entre estados de S. A[i,j] es $P(X_{t+1} = X_t = s_i)$.
- B es la matriz de probabilidades de emisión de símbolos de $\sum B[j,k] = P(O_t = k|X_t = s_j)$.
- $\prod$ es el vector de probabilidades iniciales. $\prod[i] = P(x_1 = S_i)$.

Se denota una secuencia de estados $X = (x_1, ..., x_{T+1})$ donde $X_t : S \rightarrow \{1, ..., N\}$, donde N denota el último estado posible de la secuencia y una secuencia de observaciones $O = (o_1, ..., o_t)$ con $o_t \epsilon \sum$

2.4.3. Principales problemas que se presentan en HMM

1. Problema de evaluación de la probabilidad (o verosimilitud) de una secuencia de observaciones dado un HMM.
2. Problema de determinación de la secuencia más probable de estados.
3. Problema de ajuste de los parámetros del modelo para que den mejor cuenta de las señales observadas.

El problema de evaluación tiene particular interés cuando se desea elegir entre varios modelos posibles, ya que se puede elegir aquel que mejor explique una secuencia de observaciones. La solución directa consiste en extender todos los caminos de longitud T, calcular la probabilidad de cada uno y sumarlas. El costo es exponencial, porque en el peor caso desde cada estado se puede ir a N estados, es decir, que habría N^T caminos distintos y en cada camino se hacen 2T cálculos para obtener la probabilidad. Este costo no es aceptable en la práctica: con $N = 5$ y $T = 100$ el costo sería 5^{100} aproximadamente del orden de 10^{72}.

Existe una manera más eficiente de resolver el problema, que se basa en la definición de la variable forward [30]:

$$\alpha_t = P(O_1, ..., O_t, X_t = s_i|M)$$

La cual corresponde a la probabilidad de haber observado la secuencia parcial $(O_1, ..., O_t)$y encontrarse en el tiempo t en el estado i , dado el modelo M, tal como lo explica Yu, Shun-Zheng and Kobayashi en [30]. En pocas palabras este algoritmo busca resolver el problema que dada una secuencia de observación y un modelo, logremos encontrar la probabilidad de la secuencia con respecto al modelo. Para poder obtener la probabilidad asociada a esta frecuencia lo primero que uno se plantea es que este cálculo se puede realizar directamente mediante la fórmula de Bayes:

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

Sin embargo, el número de operaciones requeridas es del orden de N^T, lo que no resulta computacionalmente más eficiente. Sin embargo un método de menos complejidad que involucra la utilización de una variable auxiliar forward (α) [30]:

$$\alpha_t(i) = P(o_1, o_2, ..., o_t, q_t = 1|\lambda)$$

Utilizando recursividad podemos calcular esta variable:

$$\alpha_1(j) = \pi_j b_j(o_1), \forall j \in \{1, ..., N\}$$

$$\alpha_{t+1}(j) = b_j(o_{t+1}) \sum_{i=1}^{N} \alpha_T(i) a_{ij}; \forall j \in \{1, ..., N\}, \forall t \in \{1, ..., T-1\}$$

Las α'_T se pueden calcular utilizando la recursión. Así que la probabilidad requerida es dada por:

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

Este método es conocido comúnmente como forward algorithm.

Por otra parte, la variable backward ($\beta_t(i)$) se puede definir de manera muy similar [27]:

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, ..., o_T, q_t = 1|\lambda)$$

Como el estado actual es i, $\beta_t(i)$ es la probabilidad de tener en t el estado i sabiendo la historia futura $o_{t+1}, o_{t+2}, ..., o_T$.

De forma similar a la anterior, se deben plantear las siguientes formulas:

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) a_{ij} b_j(o_{t+1}); \forall i \in \{1, ..., N\}, \forall t \in \{1, ..., T-1\}$$

$$\beta_T(i) = 1, \forall i \in \{1, ..., N\}$$

Finalmente, al combinar forward y backward, obtenemos:

$$\alpha_t(i)\beta_t(i) = P(O, q_t = i|\lambda) \forall i \in \{1, ..., N\}, \forall t \in \{1, ..., T-1\}$$

Obteniendo:

$$P(O|\lambda) = \sum_{i=1}^{N} P(O, q_t = i|\lambda) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i)$$

El problema de determinar la secuencia más probable de estados recae en que quedarnos sólo con las secuencias de q_t más probables resulta en una secuencia de estados poco significativa. Para esto utilizamos el denominado algoritmo **Viterbi** [6], el cual permite encontrar el estado de secuencias que tiene mayor verosimilitud.

En este algoritmo se utiliza la variable auxiliar:

$$\delta_t(i) = \max_{q_1, q_2, ..., q_{t-1}} P(q_1, q_2, ..., q_{t-1}, q_t = i, o_1, o_2, ..., o_t|\lambda)$$

,

Es decir la probabilidad más alta de la secuencia parcial de estados y observaciones hasta t, dado el estado actual i. Para $t = 1$ se define como $\delta_1(j) = \pi_j b_j(o_1), \forall j \in \{1, ..., N\})$, mientras que para el resto de los términos debemos utilizar la fórmula de recursividad:

$$\delta_{t+1}(j) = b_j(o_{t+1})[\max_{1 \leq i \leq N} \delta_t(i)(a_{ij})] \forall i \in \{1, ..., N\}), \forall t \in \{1, ..., T-1\})$$

,

Lo cual denota que empezamos nuestro cálculo desde $\delta_T(j), \forall j \in \{1, ..., N\})$, manteniendo un puntero al estado "seleccionado"que será $j* = arg\,\text{máx}_{1 \leq i \leq N}\, \delta_T(j)$, para luego realizar backtrack en la secuencia redefinido $j*$ en cada paso, obteniendo de esta manera el conjunto de estados que se pide.

Finalmente, el tercer problema es de aprendizaje y se centra en cómo podemos ajustar los parámetros del HMM para que se ajusten de la mejor forma a las observaciones [11].

Uno de los criterios más utilizados es el de máxima verosimilitud. Dado el HMM de parámetros λ y las observaciones O, la verosimilitud puede ser expresada como:

$$L = P(O|\lambda)$$

Conocer el modelo que maximiza a λ es una operación imposible de resolver de forma analítica. Para esto existe el método iterativo, denominado Baum-Welch [27], o método basado en gradientes [11].

El algoritmo Baum-Welch. También conocido como Forward-Backward, ya que usa estas variables, utiliza una variable auxiliar $Q(\lambda|\lambda')$ para comparar dos modelos λ y λ'.

$$Q(\lambda, \lambda') = \sum_{q} P(q|O, \lambda) log[P(O, q, \lambda')]$$

También definimos unas nuevas variables a partir de las backward y forward.

La primera, referida a la probabilidad de estar en el estado i en el instante t dadas las observaciones O y el modelo λ. Se obtiene combinando α y β usando el Teorema de Bayes.

$$\gamma_t(i) = P(q_t(i)|O, \gamma) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)}$$

Observemos que, si sumamos cada $\gamma_t(i)$ en todos los instantes de tiempo, el resultado obtenido sera $i = \sum_{t=1}^{T-1} \gamma_t(i)$.

Por otra parte, la segunda variable también combina las variables backward y forward mediante:

$$\xi_t(i, j) = \frac{P(q_t = i, q_{t+1} = j, O|\lambda)}{P(0|\lambda)} = \frac{\alpha_i(t)a_{ij}\beta_j(t+1)b_j(y_{t+1})}{\sum_{k=1^N} \sum_{l=1}^{N} \alpha_k(t)a_{kl}\beta_l(t+1)b_l(y_{t+1})}$$

Al igual que antes si sumamos cada $\xi_t(i, j)$ para cada instante de tiempo, obtenemos el número esperado de transiciones desde el estado i al estado j en las observaciones O:

$$\sum_{t=1}^{T-1} \xi_t(i, j)$$

.

La relación entre $\gamma_t i$ y $\xi_t(i)$ se da mediante:

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i, j) \forall i \in \{1, ..., N\}, \forall t \in \{1, ..., M\}$$

.

Una vez que se han calculado estas variables, para resolver el problema de aprendizaje se procede a realizar las tres fases del algoritmo:

1. Determinar el modelo inicial, el cual se podría elegir aleatoriamente, sin embargo conviene realizar una aproximación apegada a la realidad, de modo que se favorezca la convergencia.

2. Realizar el cálculo de las transiciones y emisiones más probables del modelo inicial.

3. Redefinir los parámetros a partir de lo calculado en el paso anterior, para que se genere un nuevo modelo que mejore en verosimilitud el modelo inicial.

Con esos sencillos pasos definimos el algoritmo y poseeremos los cálculos y soluciones a los correspondientes problemas mencionados.

Capítulo 3

Métodos Propuestos

En este capítulo se abordara la implementación de tres métodos para realizar recomendaciones de productos bancarios. Para esto se adaptaran 3 técnicas descritas en el estado del arte, que se clasifican como filtrado colaborativo.

- Método basado en Memoria descrito en la Sección 2.2.2.
- Método de Domeniconi descrito en la Sección 2.3.
- HMM descrito en la sección 2.4.

A partir de la experiencia y tomando lo mejor de los procedimientos más exitosos que se ha tenido en los proyectos de datos dentro de la institución financiera, como también teniendo en cuenta los análisis que se han realizado acorde a los proyectos en el mercado [12], seleccionamos como metodología CRISP-DM.

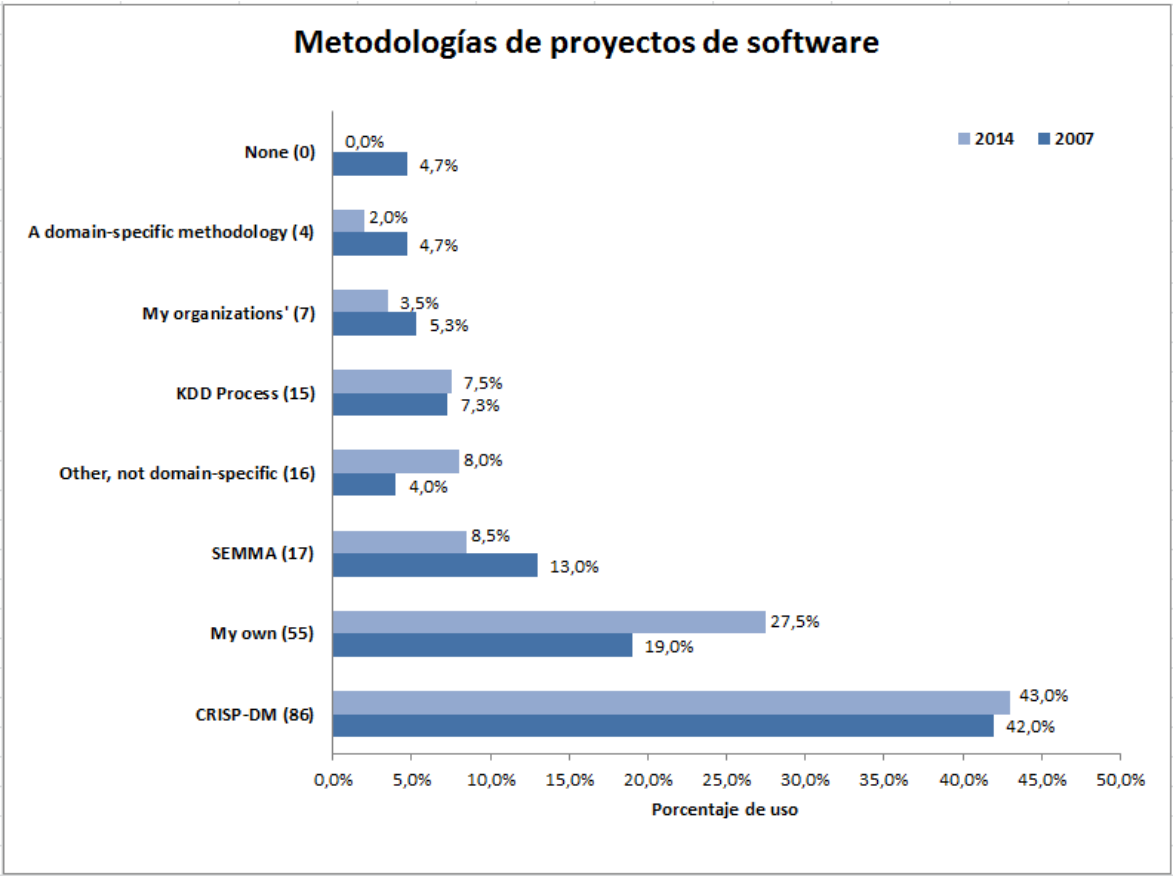


Figura 3.1: Principales metodologías utilizadas para proyectos de análisis, minería de datos o ciencia de datos [12].
Fuente: Elaboración propia.

3.1. Metodología

Para entender de una manera adecuada el proceso a seguir, utilizaremos una metodología denominada CRISP-DM [29] para el desarrollo de esta memoria, el cual es un modelo de procesos perteneciente al área de minería de datos, que posee varias etapas comunes y dividido en seis fases principales como fue descrito por Torres [28]:

- **Comprensión del tema:** Se centra principalmente en entender la problemática definiendo los objetivos para convertir este conocimiento en una definición formal con un

plan preliminar para alcanzar los objetivos.

  - Determinar objetivos.
  - Definir formalmente el problema.
  - Estudiar las investigaciones sobre el tema.

- **Comprensión de los datos:** Esta fase busca familiarizarse con los datos y descubrir los primeros patrones ocultos en los mismos.

  - Descripción inicial de los datos.
  - Análisis y exploración de los datos.

- **Preparación de los datos:** Esta fase cubre todas las actividades para la construcción del conjunto de datos, las tareas que son ejecutadas, incluyendo manipulación de tablas, registros, atributos y limpieza de datos para las herramientas de modelado.

- **Modelado:** En esta fase se seleccionan y aplican técnicas de modelado de datos y se calibran los parámetros para obtener resultados óptimos.

- **Evaluación:** Se analiza y evalúa la calidad del modelado.

  - Resultados.
  - Proceso.
  - Establecer pasos a seguir.

- **Despliegue:** Finalmente se realiza un reporte documentado de los resultados obtenidos.

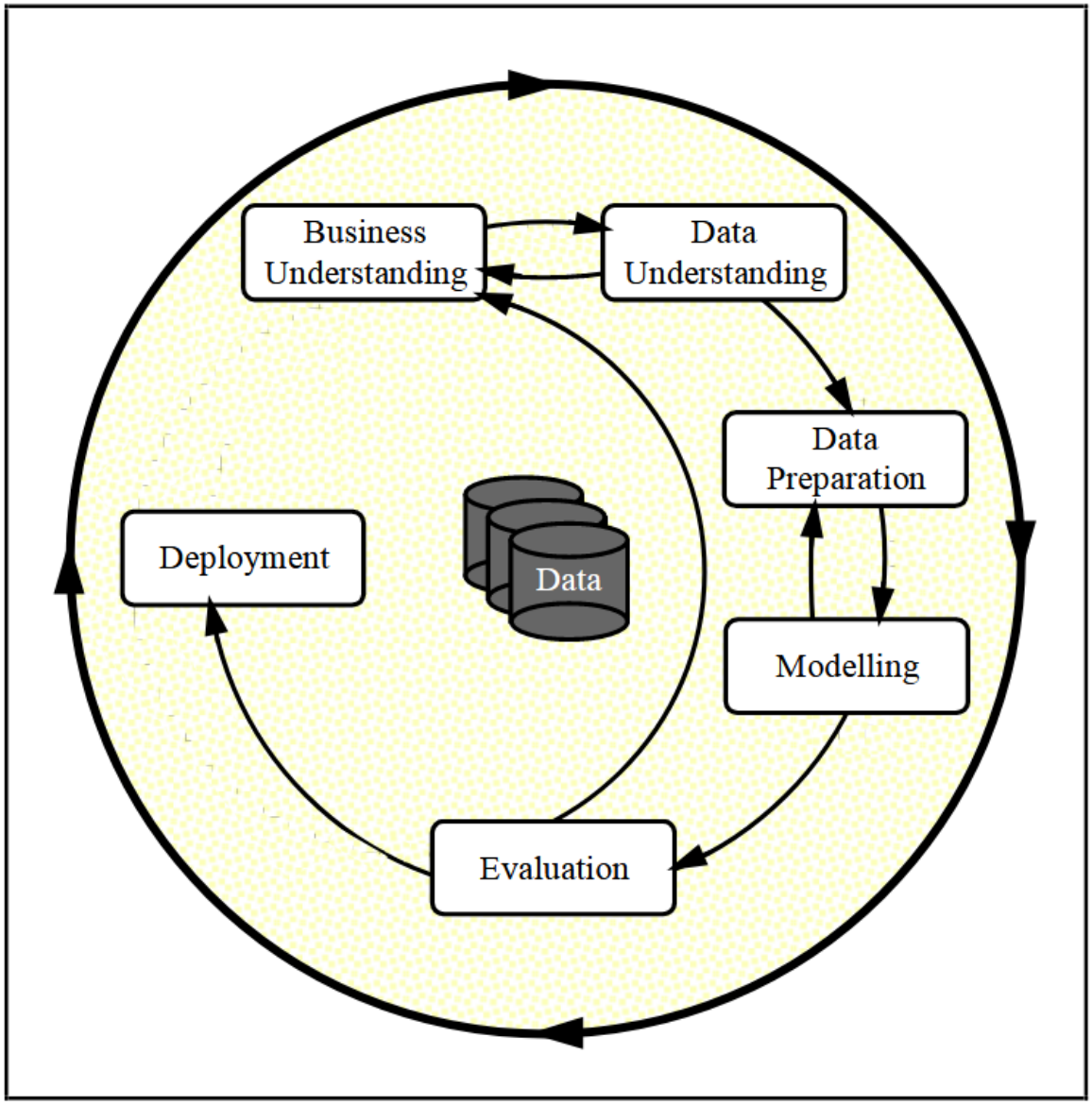


Figura 3.2: Fases del modelo de proceso CRISP-DM para minería de datos.
Fuente: Towards a standard process model for data mining [29].

3.2. Razones para preferir enfoques colaborativos

Previo a adentrarnos en el desarrollo de cada método, debemos realizar una planificación inicial, de forma que se posea una forma efectiva de abordar el problema con cada posible

escenario. Para esto debemos seleccionar los métodos a utilizar en esta memoria.

Primero, en el caso de los sistemas recomendadores, debemos verificar cuál de las alternativas existentes se adapta de una mejor manera a nuestra problemática y nos brinda tanto herramientas como procesos más adecuados. Como vimos previamente y ha sido descrito por Robin Burke [3] tenemos seis principales clases de sistemas recomendadores, Sin embargo para esta tesis se evaluará la implementación de métodos basados en contenido y métodos de filtrado colaborativo.

Los sistemas recomendadores basados en contenidos están centrados principalmente en la información extraída del objeto en análisis. Para nuestro escenario las características a extraer provienen desde los productos financieros que la institución le ofrece a los clientes o usuarios del sistema recomendador, Sin embargo poseemos la dificultad de identificar apropiadamente las características de cada producto, categorizarlos y ver posibles relaciones entre cada uno. En el caso de los productos bancarios podemos distinguir productos pertenecientes a la división personas, como tarjetas de crédito, tarjeta de débito, cuenta corriente, línea de sobregiro o créditos como el hipotecario, de consumo o automotriz. Además, podemos distinguir productos de inversiones como los fondos mutuos, compra y venta de acciones, depósitos a plazos, fondos de renta fija, etc. Mientras que en la línea de empresas y corporaciones podemos distinguir productos más de administración como lo son el leasing en el cual se genera un contrato de arriendo con opción de compra mediante el cual el Banco adquiere para el cliente, persona natural o jurídica, un bien para ser entregado en arrendamiento por un plazo determinado. También está el factoring que es un producto de la línea de empresas y sucursales en la cual se genera un contrato a plazo fijo, a través del cual la empresa de factoring, compra a sus clientes las cuentas por cobrar originadas por ventas a un plazo determinado. Como vemos las características de los productos son muy diversas entre categorías lo cual dificulta en gran manera el análisis y extracción de información en común, de forma que podamos indicar que producto es candidato debido a la selección previa de otro producto de similares características. Además, este enfoque posee otras fuertes desventajas. Una de ellas

es la poca diversificación de productos que esto nos provocaría, ya que nuestros clientes suelen ser de gustos muy parecidos, lo cual sumado a la escasa variedad de productos en el área bancaria provoca que estos cada vez sean mas sesgados a los de mejores calificaciones o mayor uso, dejando fuera productos menos utilizados. Un ejemplo de esto se ve reflejado en que si un grupo de clientes en particular usa productos que solo pertenecen al área crediticia estos poseerán una mayor correlación entre ellos, provocando al mismo tiempo que la correlación con productos de otras áreas como la de inversiones o seguros vayan disminuyendo, por lo que al ver la lista de productos que utiliza un cliente y ver que son solo de créditos se tenderá a recomendar solo productos de crédito, eliminando cada vez más las posibilidades de adquirir un producto de otra cartera. Sin duda la principal desventaja radica en los pocos productos que poseen los bancos, ya que al ser un número tan reducido no permitirían realizar un adecuado aprendizaje en base a éstos, ya que la información que se poseerá es mucho menor sobre la correlación entre un grupo pequeño de productos que al considerar la larga historia que poseen los clientes y el alto número de clientes que poseen las instituciones. En palabras más simples es mucho más rica la información que se puede obtener de 56.000 clientes que la obtenida de 30 productos.

3.2.1. Ventajas y Limitaciones de la segunda Propuesta

Los sistemas con un enfoque colaborativo nos permiten cubrir adecuadamente estas problemáticas, ya que nos permiten diversificar entre productos, explorando otras categorías según cada usuario y además sin duda los datos que más abundan en una institución financiera son los relacionados a los clientes. Sin embargo, la abundancia de datos sobre los clientes no nos garantiza su fácil utilización, ya que éstos deben ser buscados, tratados y ordenados de una manera adecuada para adaptarse a cada método, lo cual genera una complicación extra para la construcción del método a utilizar. Esta dificultad se debe principalmente a la naturaleza de los datos, debido que según Nicolás Torres [28] es necesario que cumplan las siguientes características:

- **Ítems Homogéneos:** Los ítems corresponderán únicamente a un tipo de dato; ya sea,

películas, canciones, libros o en nuestro caso productos bancarios. Si se mezclan entre sí se dificulta el análisis.

- **Rating Explícitos:** Los conjuntos de datos cuentan con calificaciones hechas por los usuarios directamente sobre los ítems, evitando de esta forma tener que inferir rating.

Para el caso de ítems homogéneos, debemos ser capaces de identificar características transversales entre todos los productos, para los clientes como lo son el uso de éstos, distinguibles mediante la frecuencia, la recencia o última utilización del producto por parte del cliente y el monto en dinero que éste genera como valor para la institución.

3.3. Primer método: Sistema Recomendador Colaborativo

3.3.1. Comprensión del tema

Como observamos lo referente a la comprensión del tema y del contexto de desarrollo se abordó ampliamente en el capítulo 1, por lo que se recomienda ante cualquier duda, volver a dicha sección.

3.3.2. Comprensión de los datos

Al trabajar con Sistemas Recomendadores es primordial tener claro cuál es el elemento a recomendar. Pese a parecer algo obvio e incluso básico el decidir el tipo de elemento candidato a recomendar, no es algo que se debe tomar a la ligera, ya que las perspectivas de recomendación pueden variar drásticamente según el enfoque a tomar. Por ejemplo, en el caso a desarrollar podemos tomar 2 visiones distintas: primero podríamos considerar que a un cliente le asignaremos un conjunto de productos que podrían interesarle, pero por otra parte, podríamos tomar un producto y asignarle clientes que se parecen entre sí. En primera instancia definamos los roles: Los clientes serán los considerados como usuarios y los productos serán nuestros ítems, debido a que poseemos como objetivo obtener aquellos productos que

irán adquiriendo los clientes a lo largo de su ciclo de vida, de forma que se pueda apreciar el comportamiento que estos tienen como un mapa de viaje de clientes, ya que al ir recomendando los productos al clientes, se va dejando un registro cronológico de la adquisición de estos, lo cual nos sirve para saber si al recomendar un determinado producto en un instante *t* especifico de tiempo este lo adquirió o por el contrario no le intereso dicha oferta, como podemos ver en la siguiente figura:

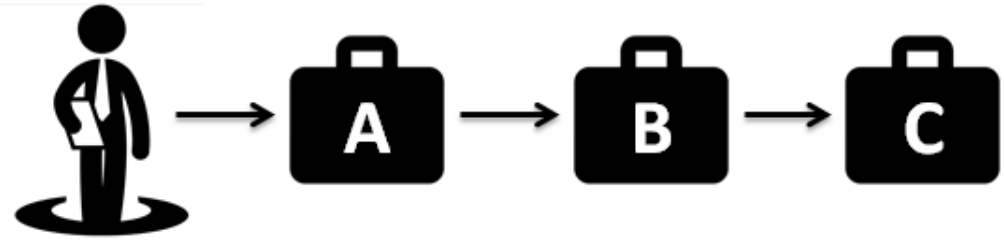


Figura 3.3: Diagrama ilustrativo que simplifica el concepto de mapa de viaje del cliente. Fuente: Elaboración propia.

En nuestro caso a los clientes les recomendamos productos en base las características o ratings en común con otros clientes, obtenidos en base a su historial de uso de cada producto, utilizando el método de filtrado colaborativo.

Por otra parte, es fundamental que consideremos los datos que tenemos a nuestra disposición. Una característica común en las instituciones financieras nacionales es el difícil y restringido acceso que se posee a los datos. Tanto la obtención como la normalización de estos suele resultar en un proceso largo y engorroso, por lo cual realizar una adecuada planificación y toma de requerimientos en este paso resulta sumamente útil para no gastar recursos innecesarios en datos que no aportarán una mayor utilidad a nuestro sistema. En primer lugar, debemos considerar la cardinalidad que posee el problema, ya que el número de productos que posee la banca es muy bajo en comparación a la cantidad de clientes, en nuestro caso poseemos

alrededor de cuarenta productos versus los sesenta mil clientes, lo cual es un caso de estudio poco común. Esta cardinalidad nos lleva casi naturalmente a pensar que lo mejor sería asociarle a este número limitado de productos los clientes que los consumirán, sin embargo aún nos queda considerar qué información asociada a los productos bancarios poseemos. Dentro de las instituciones financieras nos podemos encontrar con datos como la tenencia de productos, las fechas de activación y cierre asociadas a un cliente, los montos utilizados, ejecutivos asociados, sucursales en las que se atienden, entre otros. Se debe considerar, que se suele tener una base histórica de esta información por año-mes, por lo que la riqueza de datos que se posee para los productos resulta fundamental a la hora de inclinarnos por algún tipo de filtrado. Por otra parte, en el caso del cliente poseemos datos como edad, RUT, nombre e incluso a que segmento o cluster pertenece dicho cliente, basado en su comportamiento financiero.

Los productos son los que poseen un mayor valor en cuanto a datos para el sistema se refiere. Debemos saber distinguir adecuadamente cuál es la información de mayor utilidad que podemos asociar al producto. Se debe poseer alguna manera de clasificar a partir de características los productos, ¿pero qué características le asociaremos a dichos productos? Buscamos características medibles y transversales entre todos los productos, dentro de estas métricas una de las más comunes que se utilizan dentro de la institución en la cual se desarrolla esta memoria es la tenencia de productos o, en otras palabras, qué clientes posee vigente algún producto de las distintas líneas que ofrece el banco. Estos pueden ser de créditos, inversión o servicios para empresas entre otros. La información de tenencia de productos se encuentra disponible para su análisis, sin embargo una de las dificultades que se posee es la búsqueda de las bases de datos y tablas donde se almacena la información de cada producto. El difícil acceso es una característica en común entre todas las instituciones financieras, por lo cual es recomendable realizar con tiempo la búsqueda de datos y ver si la característica a utilizar para rankear los productos en el sistema recomendador está disponible para todos los productos. La tenencia es una de las principales: el saber si un determinado cliente posee o no un producto es fundamental en nuestro sistema y es una de las métricas básicas que podemos utilizar para realizar un rating.

Por otra parte, es lógico considerar cuál es el monto en dinero asociado a la adquisición o utilización de dicho producto. Otros factores a considerar y que suelen ser determinantes a la hora de ver la importancia de un producto es cuánto éste es utilizado y cuándo fue la última vez que se utilizó, ya que, puede que un producto tenga asociado grandes montos como es el caso de los créditos hipotecarios, pero se usa con poca frecuencia, o, por el contrario esté asociado a montos más bajos pero sea de uso frecuente como es el caso de las tarjetas de crédito. Sin embargo, puede que un cliente determinado posea un producto pero no lo use hace mucho tiempo por lo cual dejo de ser un producto determinante en su conducta como cliente. Por lo que, si queremos determinar cuándo un producto realmente es determinante e importante para un cliente, debemos considerar el monto que se gasta en éste, cuánto se utiliza y cuándo fue la última vez que se utilizó. A esta medida se le conoce como RFM [18] por las iniciales de las palabras en inglés Recency, Frecuency y Money.

3.3.3. Preparación de los datos

Luego de un exhaustivo proceso de búsqueda de cada producto, se utiliza la tecnología de consultas suministrada por SQLalchemy (https://www.sqlalchemy.org/) para consultar toda la información y al almacenarla en un único dataframe de pandas. Una vez definido que datos usaremos como métricas, debemos generar una adecuada estructuración, para esto generaremos como base un dataframe y una matriz de tenencia de productos, otra de la correlación entre productos y finalmente una base que posea el rating de los productos para cada usuario. El dataframe está conformado por las columnas “user id”que poseerá el identificador de cada cliente (en este caso corresponde al RUT), y también tendremos columnas con el ID asociado a cada producto y el nombre de dicho producto. Finalmente registramos el rating o calificación que cada usuario posee de dicho producto en base al modelo RFM [18].

El modelo RFM es una metodología con la que podemos segmentar clientes según características similares. Se creó hace más de 75 años, principalmente para los vendedores directos, para poder satisfacer a las finanzas mejorando los beneficios. Fue muy popular para

los pioneros del marketing de base de datos (Stan Rapp, Tom Collins, David Pastor, Arthur Hughes, etc). Este modelo contempla la Recencia, Frecuencia y Valor Monetario para cada cliente a partir del cual determinamos el comportamiento o evolución de compra de éstos. Lo gran ventaja es que es un modelo fácil de entender, de explicar e implementar.
El RFM sigue la premisa de que *"los más propensos a comprar son aquellos que han comprado más recientemente, con más frecuencia y gastan más dinero"* .

Se basa en la ley del 80/20 formulada por el economista italiano Vilfredo Pareto, en el siglo XIX. Esta ley anuncia que el 80 % de las compras las realizan el 20 % de los clientes.

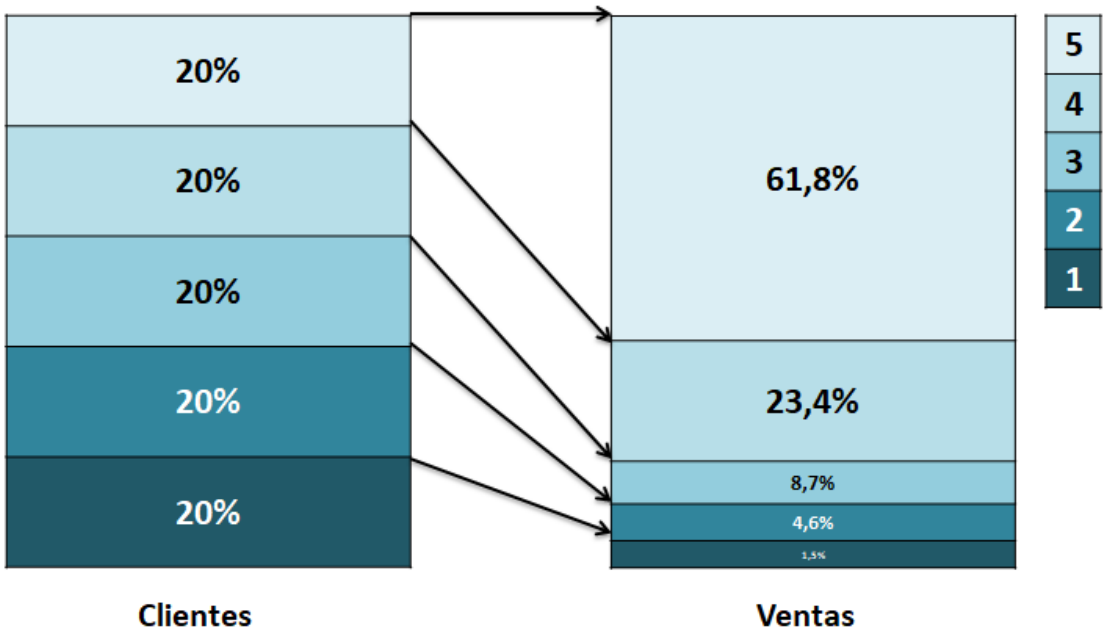


Figura 3.4: Distribución de clientes por valor monetario. 80/20 y RFM.
Fuente: Elaboración propia.

Como hemos comentado anteriormente:

- Con la Recencia (Recency) medimos los días que han pasado hoy a la fecha en que se utilizó por última vez un producto.
- Con la Frecuencia (Frequency) medimos el número de veces que el cliente ha utilizado

un producto cada cliente en total.

- El Valor Monetario (Monetary) es la suma total de cantidad de dinero que el cliente lleva utilizado en sus productos.

Con el RFM podemos construir escalas, basadas en estas tres variables, dando a cada cliente un valor según el percentil en el que se encuentre. Lo más común es escalar por quintiles, es decir, tanto a la Recencia como a la Frecuencia como al Valor Monetario les asignamos un valor del 1 al 5, siendo 1 la peor puntuación y 5 la mejor. El valor RFM es generado por la concatenación de Recency, Frecuency y Money, generando salidas del estilo "111", "235", "555", donde 1 equivale a la peor estadística y 5 a la más favorable.

De esta manera podemos afirmar que:

- Los clientes que compran de forma reciente son más favorables a comprar que aquellos que no lo han hecho últimamente.
- Los clientes que compran más frecuentemente están más dispuestos a comprar nuevamente que aquellos que han hecho una o dos compras.
- Los clientes que gastan más, están más dispuestos a comprar nuevamente.
- Los clientes más valiosos son aquellos que pueden llegar a hacerlo aún más.

Estas métricas son utilizadas en diferentes estudios, entre ellos podemos mencionar *Customer lifetime value (CLV) measurement based on RFM model* [25] o en *Customer's life–time value using the RFM model in the banking industry: a case study* [20] entre otros.

3.3.4. Modelado

Una vez generada nuestras bases con las que se trabajará, debemos definir el método y filtrado a utilizar en nuestro sistema.

Como se había mencionado anteriormente, uno de los algoritmos más apropiados son los de filtrado colaborativo, debido a sus métricas utilizadas a la hora de predecir, dándole mayor importancia a clientes con gustos similares.

Como es mencionado por Torres [28], el algoritmo de los K-vecinos más cercanos o kNN del inglés k Nearest Neighbors, es el primer sistema de filtrado colaborativo automatizado, presentado por GroupLens en su motor de recomendación Usenet [22]. Posteriormente se desarrollaron variantes para recomendaciones de música y vídeo.

Una de las ventajas que poseen los recomendadores basados en memoria es la utilización de toda la base de datos para poder generar predicciones. Esto se debe a que cada cliente será parte de un grupo de personas al cual denotaremos vecindario, los cuales serán utilizados para combinar preferencias de forma que podamos realizar las predicciones. Según sea el caso, los utilizaremos para encontrar los k clientes más cercanos o los k productos más cercanos, pero como habíamos mencionado con anterioridad, al poseer tan poca información respecto a los productos es preferible utilizar métricas de valor respecto a los clientes.

Tomando en cuenta la utilización de los k clientes más cercanos, realizaremos los siguientes pasos:

- Cálculo de la similaridad [ecuación 2.2]. Esta refleja la distancia o correlación entre clientes.
- Encontrar los k clientes o productos más cercanos al activo.
- Elaborar una predicción ponderando todos los ratings entre clientes afines.

Cálculo de la similaridad

Luego de realizados los cálculos de recencia, frecuencia y monto, para cada usuario obtendremos un gran dataframe en el que se registraran dichos valores por cliente y producto,

como se aprecia en la siguiente figura:

	user_id	producto_title	producto_id	Frecuency	Recency	Monetery
0	100001195	Cuenta_Corriente_MN	Cuenta_Corriente_MN	35.0	12.0	734815.0
1	100001195	Linea_de_Sobregiro_(Credito)	280	35.0	12.0	6500000.0
2	100001195	Linea_de_Sobregiro_(MN)	110	35.0	12.0	6500000.0
3	100001195	Tarjeta_de_Crédito	210	50.0	12.0	3000000.0
4	100001454	Tarjeta_de_Débito	130	7.0	531.0	0.0

Figura 3.5: Dataframe correspondiente al cálculo RFM por cada producto de un cliente. Fuente: Elaboración Propia.

Posteriormente, utilizaremos la función qcut() proporcionada por la librería pandas de python (https://pandas.pydata.org/), la cual genera una función de discretización basada en cuantiles. Esta discretiza las variables en fragmentos de igual tamaño según el rango o los cuantiles de muestra. Por ejemplo, 1000 valores para 10 cuantiles producirían un objeto categórico que indica la pertenencia a cuantil para cada punto de datos.

Listing 1 Algoritmo que genera los cortes para los límites de los cuantiles.

```
def pct_rank_qcut(series, n):
    edges = pd.Series([float(i) / n for i in range(n + 1)])
    f = lambda x: (edges >= x).idxmax()
    return series.rank(pct=1).apply(f)
```

Posteriormente se llama a la función anterior con el valor correspondiente de recencia, frecuencia y monto según corresponde, acorde a cada cliente, para luego sumar el valor obtenido con los demás, como se observa a continuación:

Listing 2 Algoritmo que genera el valor RFM para cada cliente.

```
data_RF['Recency']=data_RF['Recency'].astype(float)
data_RF['Frecuency']=data_RF['Frecuency'].astype(float)
data_RF['Monetery']=data_RF['Monetery'].astype(float)

# data_RF['Recency_q']=pct_rank_qcut(data_RF.Recency, 5)
data_RF['Recency_q']=pd.qcut(data_RF.Recency, 6, duplicates='drop',
    labels=False)
data_RF['Frecuency_q']=pd.qcut(data_RF.Frecuency, 6, duplicates='drop',
    labels=False)
data_RF['Monetery_q']=pd.qcut(data_RF.Monetery, 6, duplicates='drop',
    labels=False)

data_RF['RFM_Score']=data_RF.Recency_q.astype(str)+
    data_RF.Frecuency_q.astype(str) +
    data_RF.Monetery_q.astype(str)


df_RF=data_RF.drop('Frecuency', axis=1).
    drop('Recency', axis=1).drop('Monetery', axis=1).
    drop('Recency_q', axis=1).drop('Frecuency_q', axis=1).
    drop('Monetery_q', axis=1)
df_RF = df_RF.drop('producto_id', axis=1)
df_RF = df_RF.set_index(['user_id','producto_title'])
df_RF['RFM_Score'] = df_RF['RFM_Score'].astype(str)
#df_RF = df_RF.rename({'RFM_Score':''}, axis='columns')

df_RF_MTX=df_RF.reset_index().
    pivot_table(index='user_id',columns='producto_title',
    values='RFM_Score',aggfunc='max')
df_RF_MTX.fillna(0, inplace=True)
producto_index = df_RF_MTX.columns
# df_RF_MTX.to_csv(file_RFM, header=True, index=False) #GUARDAR DATAFRAME
df_RF_MTX.columns.get_level_values("producto_title")
```

Obteniendo finalmente un dataframe como el siguiente:

Acciones_Nacionales	Credito_Hipotecario	Crédito_Comercial_y_Consumo	Cuenta_Corriente_BIA	Cuenta_Corriente_MN	Cuenta_Inversiones	Cuenta_Vista
0.0	0.0	0.0	0.0	334.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	322.0	0.0	0.0	335.0	0.0	0.0
0.0	333.0	0.0	0.0	335.0	0.0	0.0
0.0	332.0	0.0	0.0	334.0	331.0	0.0

Figura 3.6: Fragmento de tabla descriptiva del valor RFM por producto para cada cliente. Fuente: Elaboración Propia.

Cálculo de los productos más cercanos al activo

Es fundamental no perder de vista que estamos en búsqueda de los productos con valores afines a los productos activos que posee el cliente. Para ello, se utiliza la correlación entre productos en función de la puntuación obtenida para los usuarios (valor RFM).

Una manera fácil de calcular la similaridad en python es usando la función numpy.corrcoef(), que calcula el coeficiente de correlación de Pearson(PMCC) entre cada pareja de ítems.

Este coeficiente mide el grado de relación lineal entre dos variables. Fue utilizado por GroupLens [22] y BellCore [8]. La correlación que existe entre el producto activo y otro tiene un valor entre -1 y 1 que mide cuán relacionadas están un par de variables cuantitativas.

Elaboración de una predicción ponderando todos los ratings entre clientes afines

Posteriormente realizaremos una ponderación de los valores obtenidos a través de la matriz de correlación, la cual es una matriz de tamaño m x m, donde el elemento M_{ij} representa la correlación entre el ítem i y el ítem j, considerando el coeficiente de correlación . Usamos la matriz traspuesta de ratings_mtx_df para que la función np.corrcoef nos devuelva la correlación entre productos. En caso de no hacerlo nos devolvería la correlación entre usuarios.

Para calcular la similaridad entre productos, usaremos la correlación entre ellas en función de

la puntuación obtenida para todos los usuarios. Para explicar cómo esto funciona lo ejemplificaremos con la frecuencia. Tomamos los valores de frecuencia que tiene un par de productos por ejemplo tarjeta de débito con la tarjeta de crédito y se obtiene el coeficiente de correlación entre este par de productos utilizando la formula 2.1. Este nos mostrara una correlación directa, donde podremos observar que en el caso de las tarjetas de débito y crédito están directamente correlacionadas. Ya que, al aumentar la frecuencia de uso de una, en la otra también lo hace. Este procedimiento utilizado en la función np.corrcoef() es aplicado para cada par de productos, obteniendo el coeficiente correspondiente. Algunas consideraciones a tener en cuenta para las variables cuantitativas a utilizar son:

- Para determinar el valor de la recency este se debe dar vuelta, debido que a mayor valor en tiempo de recency se considera peor como medida, ya que significa que el cliente no utiliza ese producto hace mucho tiempo. Por lo que, si tiene un mayor valor en tiempo se le asignara una menor medida RFM para que esté acorde a la frecuencia y monto, los cuales a mayor valor mejores son, debido a que correspondería que el cliente gasta más y utiliza más el producto.

Para un mayor detalle sobre np.corrcoef(), consultar la documentación en (www.docs.scipy.org).

Posteriormente se llama a la función anterior con el valor correspondiente de recencia, frecuencia y monto según corresponde, acorde a cada cliente, para luego sumar el valor obtenido con los demás, como se observa a continuación:

Listing 3 Algoritmo que genera la correlación entre productos.

```
1 corr_matrix = np.corrcoef(df_RF_MTX.T) #Matriz transpuesta
```

Finalmente para obtener la relación entre cada producto que se posea, dichos datos serán representados en un mapa de calor entre productos:

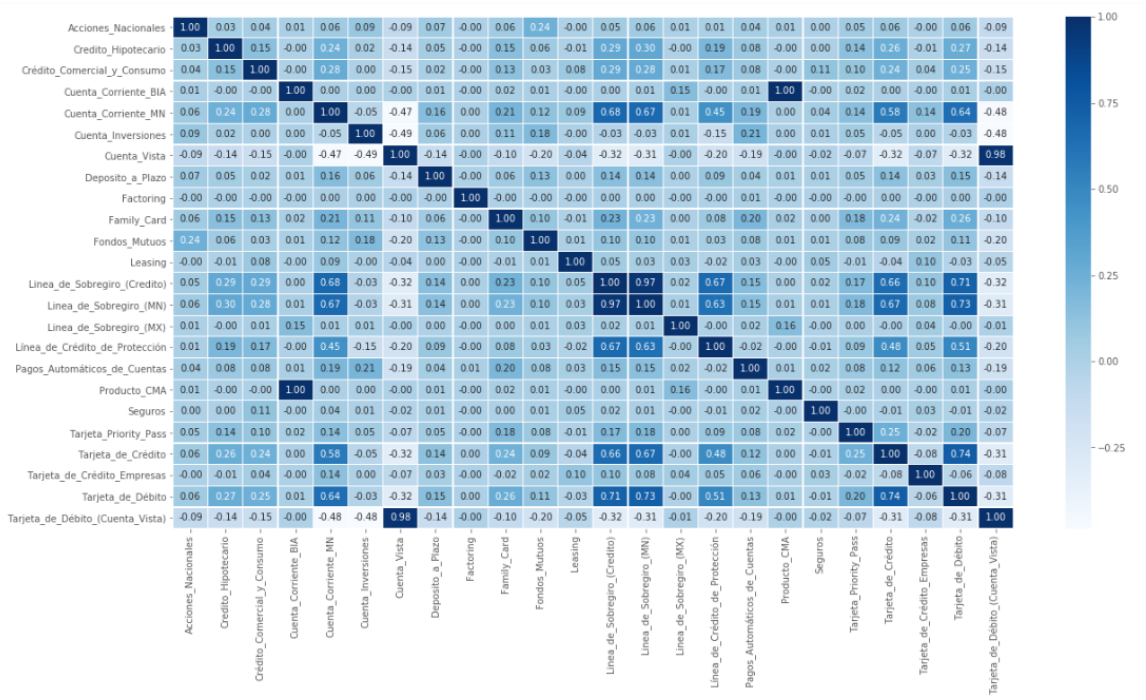


Figura 3.7: Mapa de calor, correspondiente a la correlación entre productos.
Fuente: Elaboración Propia.

Ahora, si queremos recomendar productos a un usuario, solo tenemos que conseguir la lista de productos que dicho usuario ha visto. Ahora, con dicha lista, podemos sumar las correlaciones de dichas productos con todas las demás y devolver los 3 productos con una mayor correlación total. Solo seleccionamos 3 productos debido que es el número máximo de productos que utilizan las instituciones financieras para realizar campañas de marketing. Esto se debe a que un mayor número de productos encarece los costos asociados a las campañas y genera mucho spam de parte de la institución hacia los clientes.

Listing 4 Algoritmo que devuelve el vector de correlación para un producto.

```
def get_producto_similarity(producto_title):
    "Devuelve el vector de correlación para un producto"
    producto_idx = list(producto_index).index(producto_title)
    return corr_matrix[producto_idx]
```

Listing 5 Algoritmo que dado un grupo de productos, devuelve los más similares.

```
def get_producto_recommendations(user_producto):
    '''Dado un grupo de productos, devolver los más similares'''
    producto_similarities = np.zeros(corr_matrix.shape[0])
    for producto_id in user_producto:
        producto_similarities = producto_similarities +
        get_producto_similarity(producto_id)
    similarities_df = pd.DataFrame({
        'producto_title': producto_index,
        'sum_similarity': producto_similarities
        })
    similarities_df = similarities_df[~(similarities_df.
        producto_title.isin(user_producto))]
    similarities_df = similarities_df.sort_values(by=['sum_similarity'],
        ascending=False)
    return similarities_df
```

Una vez que realizamos esta acción debemos proporcionar nuevas recomendaciones para dicho usuario teniendo en cuenta los productos que ha tenido como input para el sistema, obteniendo una recomendación como la que se puede observar en la figura 4.7.

3.3.5. Evaluación

Para evaluar el método se usaran métricas en común con las otras propuestas a realizar, por lo que se la evaluación será postergada para más adelante.

3.3.6. Despliegue

Al ir distinguiendo características propias de la naturaleza del problema, se va generando de a poco un ”pseudo-ordençon pasos para la generación de los métodos. Es en base a esto, que realizaremos una planificación previa, 3.8 que nos permitirá mantener un orden en los pasos a seguir tanto para preparar la data como en el desarrollo del método mismo. Como vemos está dividido en 2 secuencias: una correspondiente a la estructura más en detalle del sistema,

indicando elementos relevantes como el rating por producto, matrices necesarias e incluso el output del sistema. Por otra parte en el paso 2 se muestra una estructura más general de éste de forma que no se pierda de vista lo que se está realizando en 1.

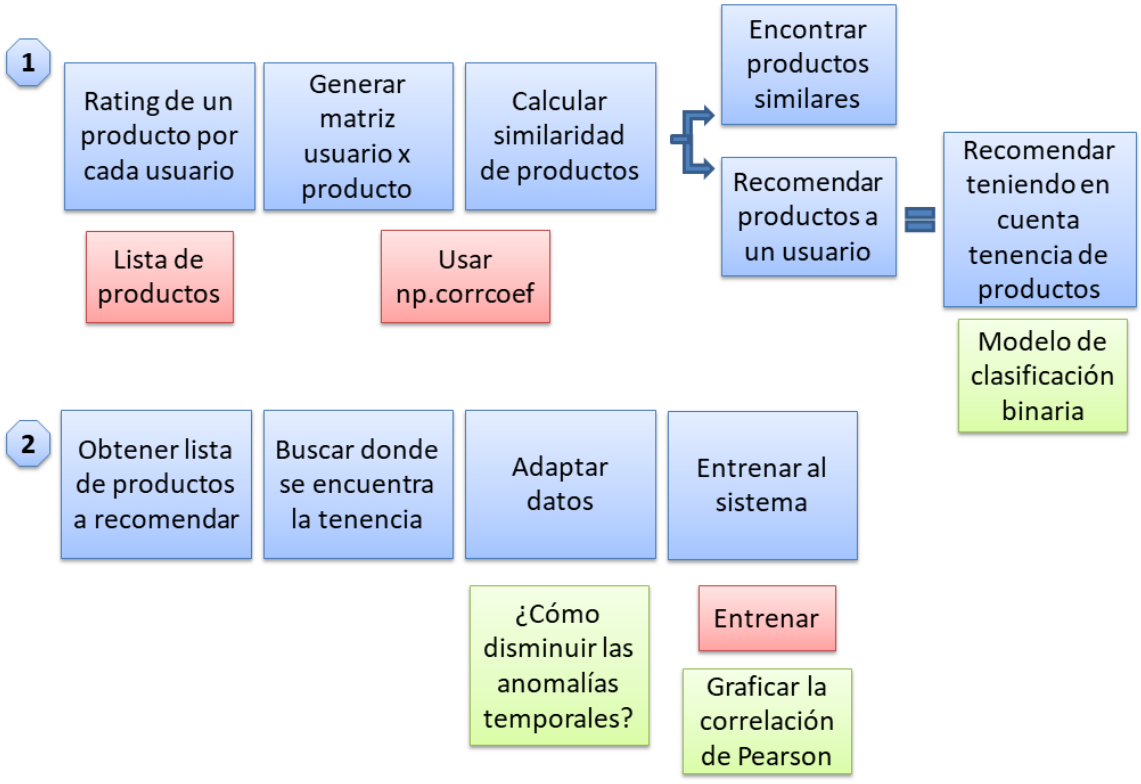


Figura 3.8: Descripción general de la implementación del sistema recomendador.
Fuente: Elaboración propia.

3.4. Recomendaciones utilizando Aprendizaje Supervisado

3.4.1. Comprensión del tema

Como no cambia el contexto del problema, no se realiza mayor desarrollo de esto, más allá de lo abordado en el capítulo 1, por lo que se recomienda ante cualquier duda, volver a dicha

sección.

3.4.2. Comprensión de los datos

Para este método tenemos como objetivo predecir el comportamiento de un segmento de clientes, para esto utilizaremos algoritmos de aprendizaje automático o aprendizaje de máquinas (*Machine Learning*) para que la computadora aprenda sobre dicho comportamiento. De forma más concreta, se crea un algoritmo capaz de generalizar comportamientos a partir de la información que le suministraremos de ellos. Para esto, se establecerá una correspondencia entre las entradas y las salidas deseadas del sistema, como es preestablecido en los algoritmos de aprendizaje supervisado, suministrando etiquetas adecuadas que sean de utilidad para el sistema con los datos que se poseen a disposición.
Dentro de los datos que poseemos nos encontramos con los datos utilizados para el Sistema Recomendador, pero esta vez debemos preparar dichos datos de una forma diferente. Primero utilizaremos la tenencia de productos, lo cual se está convirtiendo en un pivote fundamental para nuestros sistemas a utilizar. La base de datos posee diversas variables como el código de producto, el código del tipo de producto, la fecha de apertura, cierre y utilización, el monto utilizado y finalmente el identificador que distingue el cliente al cual corresponden dichos datos. Además utilizaremos una base compuesta por información del cliente, donde tenemos RUT, edad y género. Estos datos del cliente nos permiten realizar una clasificación entre ellos, de forma que al momento de entrenar nuestro sistema, los hagamos por grupos distintos de clientes, ya que, como mencionamos anteriormente, el comportamiento de un cliente cuyo tramo de edad corresponde a los 25 a 34 años es bastante distinto al correspondiente a 55 a 64 años (un perfil es mucho más arriesgado que el otro, sumado a que sus gustos y necesidades son completamente distintas). Cabe destacar que podríamos darle esta tarea de diferenciar entre clientes al mismo sistema de aprendizaje, pero no lo haremos debido al alto costo en procesamiento que implica realizar el entrenamiento con un conjunto de datos muy grande. Al hacer esto tenemos un costo sobre la exactitud (*accuracy*) pero ganamos eficiencia en procesamiento e incluso en precisión.

3.4.3. Preparación de los datos

Para visualizar cómo funcionan los algoritmos de aprendizaje automático, es mejor considerar datos de una o dos dimensiones, esto es datasets con solo una o dos características. Aunque, en la práctica los datasets tienen muchas más características, es difícil representar datos de alta dimensionalidad en pantallas 2D.
Para realizar una adecuada clasificación en una tarea supervisada, debemos verificar el rendimiento. Para esta verificación debemos dejar un conjunto de datos sin utilizar al momento de realizar el entrenamiento, por lo cual debemos dividir los datos en dos partes:

1. Un conjunto de entrenamiento que el algoritmo de aprendizaje utiliza para ajustar los parámetros del modelo.

2. Un conjunto de test para evaluar la capacidad de generalización del modelo.

El conjunto de entrenamiento es generado a partir de una representación matricial de la tenencia de productos, para esto se crea una matriz cuyas filas corresponden a los clientes, mientras que las columnas corresponden a un producto en específico, generando una representación binaria de la tenencia de un producto específico por un cliente en particular. Por otra parte, es sumamente importante determinar una manera adecuada de etiquetar cada conjunto de datos. Para esto nos basamos en el modelo generado por Giacomo Domeniconi [4].

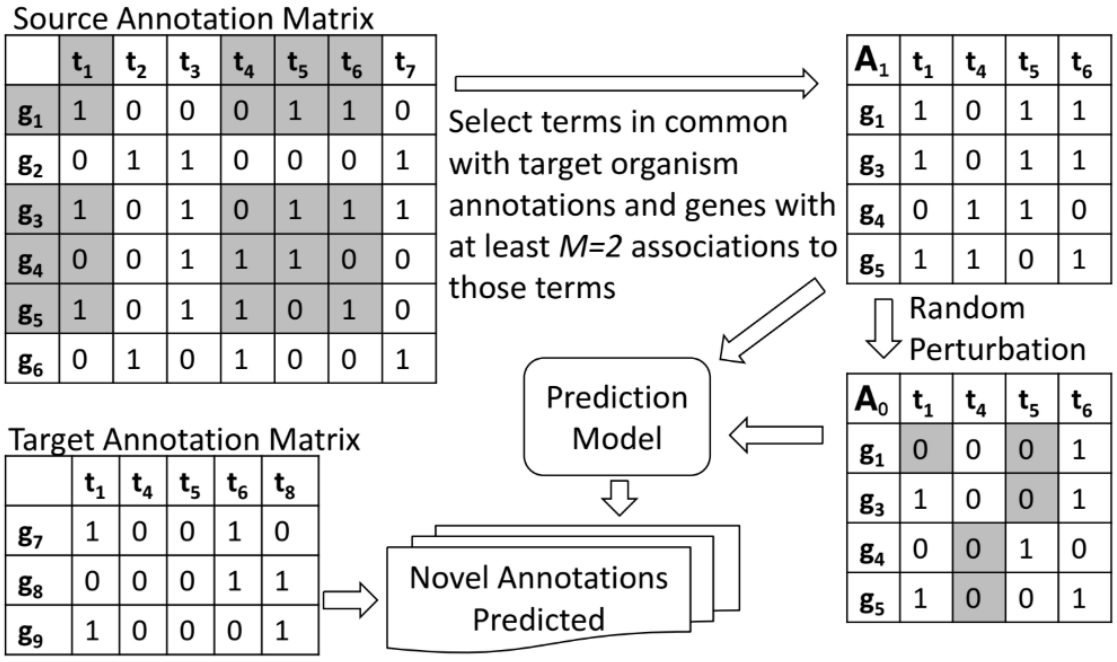


Figura 3.9: Flujo de trabajo para nuevas predicciones de anotación usando para un organismo.

Fuente: Data and Text Mining Techniques for In-Domain and Cross-Domain Applications.

En nuestro caso las etiquetas estarán conformadas por una cadena cuya posición en el arreglo corresponderá a un producto en específico y su valor binario representara la tenencia.

$$X(i, j) = \begin{cases} 1 \text{ Si el cliente i posee el producto j} \\ 0 \text{ Si no} \end{cases}$$

Como vemos cada fila de la matriz puede ser representada como una lista o arreglo con ceros y unos. Dicho arreglo lo utilizaremos como etiqueta para el cliente en dicho mes. De esta forma creamos un primer conjunto denominado conjunto de entrenamiento, el cual posee dos grandes partes: una corresponde a las características las cuales están tomadas desde la matriz correspondiente al tiempo 0, mientras que las etiquetas son obtenidas por la data correspondiente al siguiente mes o tiempo 1. Esta representación se aprecia mejor en la siguiente imagen:

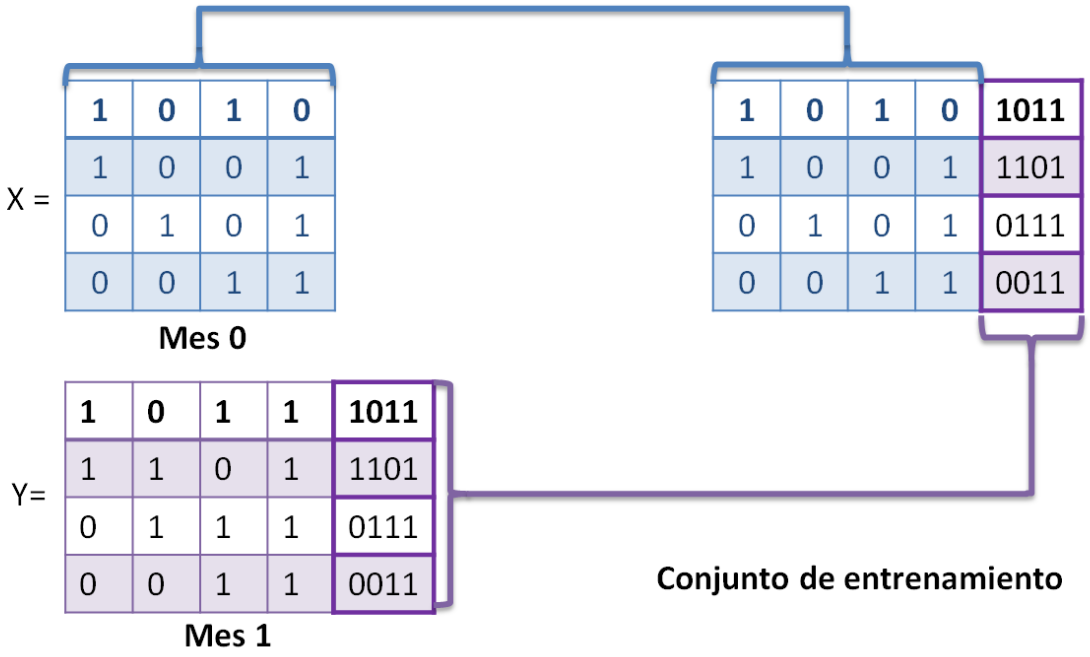


Figura 3.10: Matriz correspondiente al conjunto de entrenamiento.
Fuente: Elaboración propia.

Cabe destacar que a diferencia de la problemática enfrentada por Giacomo Domeniconi no necesitamos generar perturbaciones aleatorias para generar un nuevo conjunto o matriz de anotaciones, ya que, si poseemos los registros de cada mes. Utilizaremos esta ventaja a nuestro favor. Para esto realizaremos el mismo proceso descrito anteriormente para generar un conjunto de entrenamiento generando múltiples conjuntos de entrenamiento entre cada mes de modo que tendremos una arquitectura de datos con la siguiente forma:

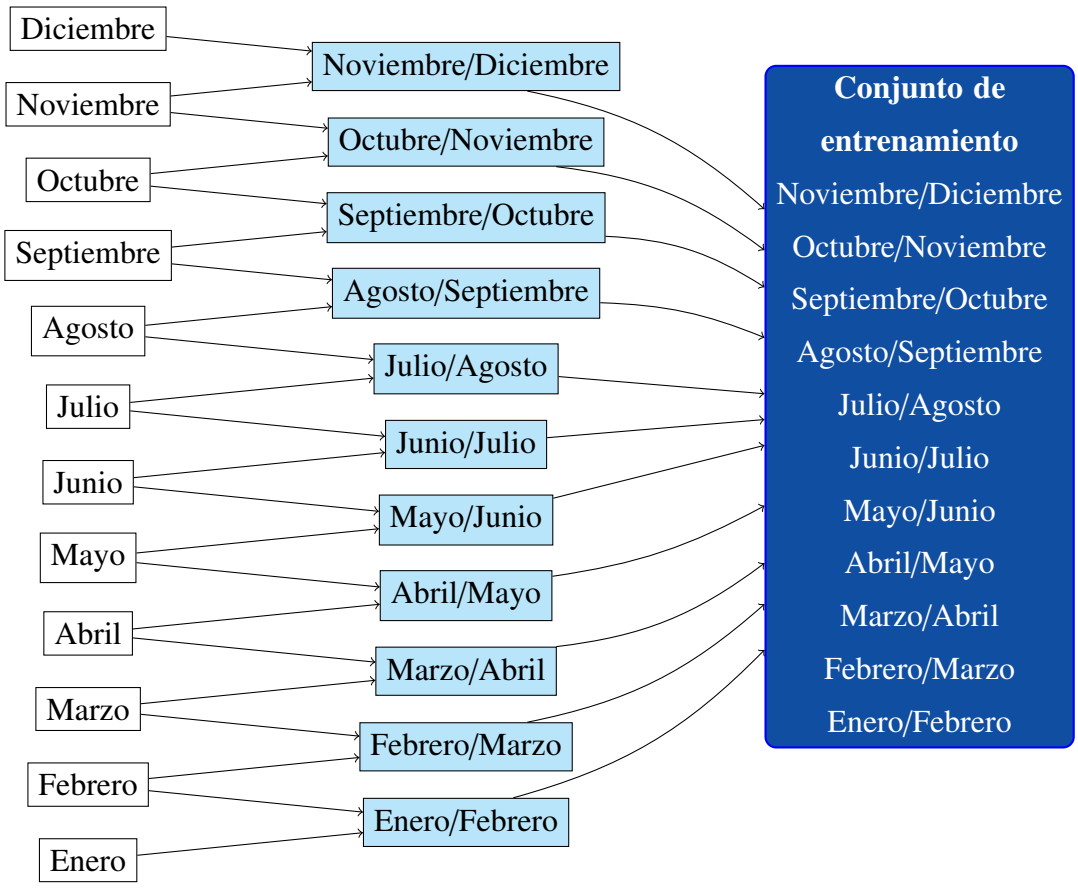


Figura 3.11: Representación del proceso de creación del conjunto de entrenamiento. Fuente: Elaboración propia.

Por otra parte, debemos generar el conjunto de test el cual posee la matriz del tiempo 1 y las etiquetas del tiempo 2, para poder realizar las verificaciones y correcciones al generar el conjunto de predicción.

3.4.4. Modelado

Para extraer los datos y utilizarlos de forma factible debemos seleccionar una herramienta adecuada que cumpla y optimice el desarrollo de nuestra tarea a realizar. Para esto utilizaremos Scikit-Learn (https://scikit-learn.org/) y Python (https://www.python.org/), los cuales

con su sintaxis nos permiten tener un código legible, favoreciendo la depuración, productividad, potencia y flexibilidad que ofrece el lenguaje, generando una curva de aprendizaje bastante suave, sin grandes costos al disponer de los datos de una manera adecuada. Scikit-Learn nos provee una variedad de algoritmos clasificación lo cual es bastante útil sumado a su compatibilidad con otras librerías como Pandas. Su variedad de módulos se convierte en una de las principales ventajas producto de la flexibilidad que esto nos ofrece. Esto debido a que la versatilidad y formación es la clave en el campo informático.
El estimador de Scikit-Learn funciona en base al siguiente modelo de su API:

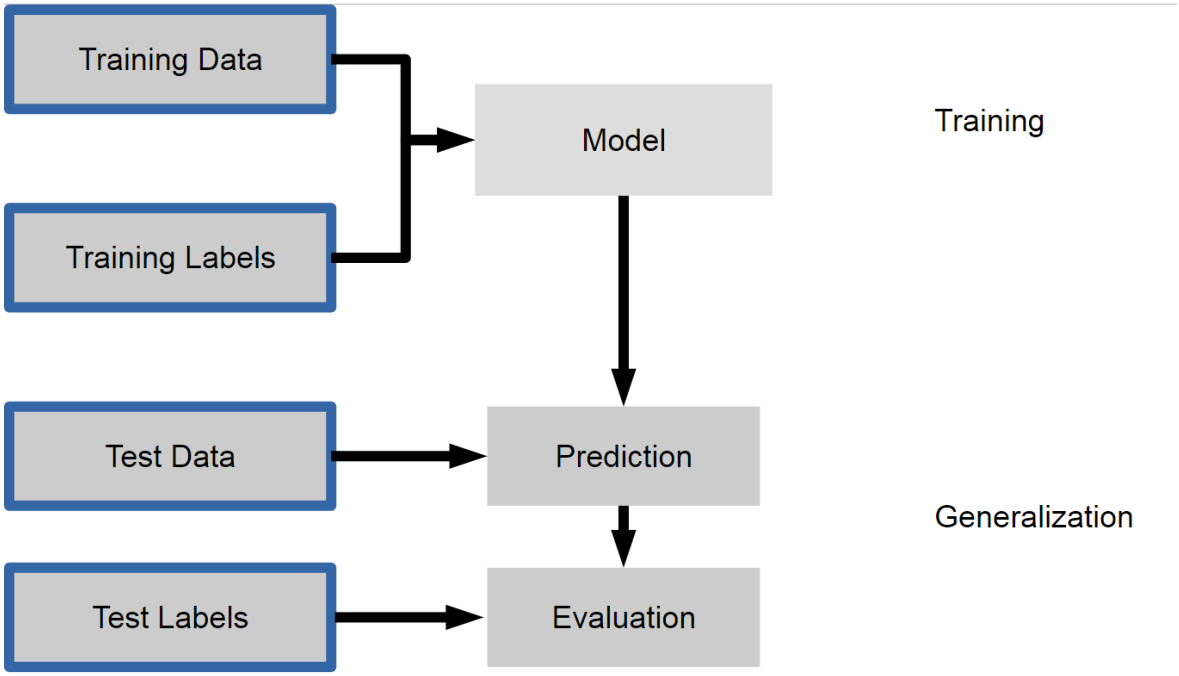


Figura 3.12: El API de un estimador de scikit-learn.
Fuente: scikit-learn documentation.

Cualquier algoritmo de scikit-learn, se maneja a través de una interfaz denominada ”Estimator” (una de las ventajas de scikit-learn es que todos los modelos y algoritmos tienen una interfaz consistente). Además, para construir el modelo a partir de nuestros datos, esto es, aprender a clasificar nuevos puntos, llamamos a la función “fit” pasándole los datos de entrenamiento, y las etiquetas correspondientes (la salida deseada para los datos de entrenamiento):

Listing 6 Algoritmo de Aprendizaje automático.

```
X_train = df_train[lista_final].values
X_test = df_validacion[lista_final].values
y_train = df_train.label
y_test = df_validacion.label

from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression()
clf = classifier.fit(X_train, y_train)
```

Por otra parte, además de la herramienta debemos preocuparnos de seleccionar el algoritmo de clasificación adecuado. En nuestro caso, seleccionaremos dos: uno de es la Regresión Logística [9] y el otro clasificador corresponde a K Nearest Neighbors [13]. Según las convenciones del aprendizaje automático, la regresión logística sigue estos pasos (Cabe destacar que por simplicidad tanto los pasos como ecuaciones correspondientes Regresión logística se definirán para el caso de 2 predictores):

1. Transformar la variable Y a predecir (que solo puede tomar dos valores: 0,1) en la probabilidad de $Y = 1$, (que es igual a P) usando la distribución binomial de bernoulli.

2. Calcular el Odds Ratio de la probabilidad P, que es igual a $P/(1-P) = OddsRatio$.

3. Calcula el logaritmo natural del Odds Ratio, es decir: hace el $\ln(P/(1-P))$ ó $\ln(OddsRatio)$. Esto también puede ser interpretado como la cantidad de éxitos dividido entre la cantidad de fracasos. Se utiliza logaritmo natural para luego poder hacer la inversa, que es el exponencial. El logaritmo del Odds Ratio ahora es la variable independiente a predecir, también llamado logit.

4. Estima la combinación lineal, o Regresión Lineal de las variables independientes, de la siguiente forma: $\ln(P/(1-P)) = \beta_0 + \beta_1 * X_1 + \beta_2 * X_2$. Al resultado obtenido también le dicen β_x.

5. Se transforma $\ln(P/(1-P))$ en $e^{\beta x}/(1+e^{\beta x})$. Esto también lo representan como: $1/(1+e^{-\beta x})$ donde e es la constante matemática igual a 2.718281828.

6. Se estiman los coeficientes (es decir los β) de la ecuación anterior, usando el método de máxima verosimilitud.

El otro método que utilizaremos será el de KNN, el cual consiste en realizar búsquedas en las observaciones más cercanas a la que se está tratando de predecir y clasificar el punto de interés basado en la mayoría de datos que le rodean. Como dijimos antes, es un algoritmo supervisado, es decir que tenemos etiquetado nuestro conjunto de datos de entrenamiento, con la clase o resultado esperado dada "una fila" de datos y además basado en instancias, ya que nuestro algoritmo no aprende explícitamente un modelo (como por ejemplo en Regresión Logística). En cambio memoriza las instancias de entrenamiento que son usadas como "base de conocimiento" para la fase de predicción.
Para utilizar este algoritmo debemos:

- Calcular la distancia entre el ítem a clasificar y el resto de ítems del dataset de entrenamiento.
- Seleccionar los "k" elementos más cercanos (con menor distancia, según la función que se use).
- Definir a qué grupo pertenecerán los puntos, sobre todo en las "fronteras" entre grupos.

En resumen, este clasificador popular y fácil de entender es el K Nearest Neighbors (KNN). Implementa una de las estrategias más simples de aprendizaje (de hecho, en realidad no aprende): dado un nuevo ejemplo desconocido, buscar en la base de datos de referencia (entrenamiento) aquellos ejemplos que tengan características más parecidas y le asigna la clase predominante.

La interfaz es exactamente la misma que para "LogisticRegression":

Listing 7 Código de Aprendizaje automático usando KNN.

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=10)
clf_knn = knn.fit(X_train, y_train)
prediction_knn = clf_knn.predict(X_test)
```

3.4.5. Evaluación

Al igual que en el método anterior, la evaluación se realizara en el capítulo 4, de forma que comparemos más en detalle el rendimiento entre cada método.

3.4.6. Despliegue

En el método 2, la planificación está centrada tanto en las decisiones a tomar en cada etapa como los pasos a realizar. Por ejemplo, se parte con la realización de la matriz binaria que indica la tenencia de productos por cada usuario, posteriormente se subdivide en 2 pasos la creación de un conjunto de entrenamiento y validación, para posteriormente decidir cuál es el algoritmo supervisado a utilizar para entrenar al sistema, para finalmente luego de una serie de pasos generar el output o lista de productos a recomendar. Cabe destacar que los cuadros verdes son notas que detallan más el paso que se encuentra en la parte superior de estas.

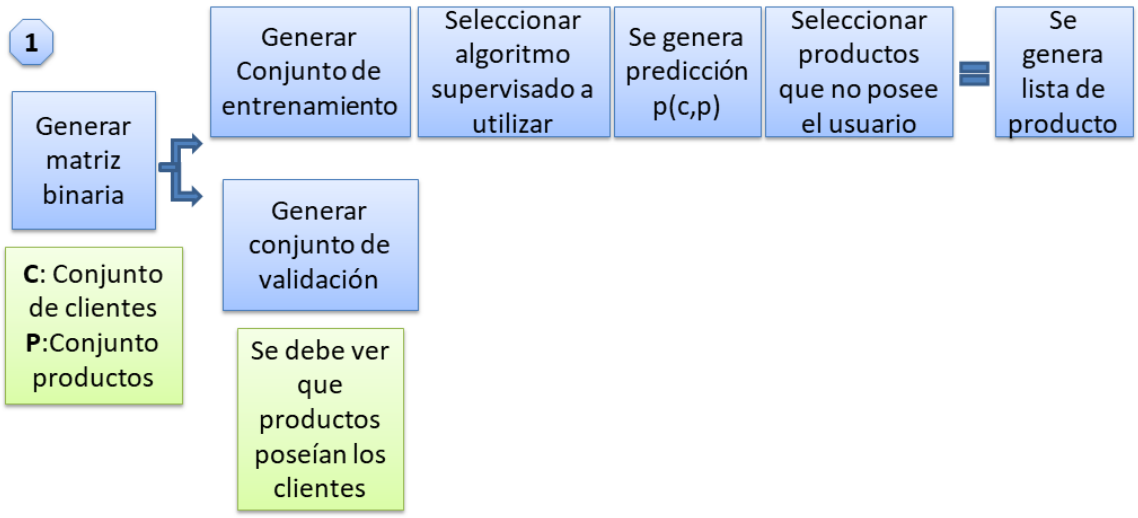


Figura 3.13: Pasos generalizados para la construcción del segundo método.
Fuente: Elaboración propia.

Por otra parte, es interesante documentar algunas características nuevas observadas para este método, sin embargo por motivos de orden estas serán postergadas hasta la sección 4. Algunas de las características que podremos encontrar son por ejemplo una identificación del comportamiento de clientes y el rendimiento del método en comparación a los otros métodos a seleccionar.

3.5. Recomendaciones utilizando Modelo Oculto de Markov

Un modelo oculto de markov se basa en una representación simplificada de un problema en forma de cadena de decisiones, donde representaremos el comportamiento secuencial de un cliente: cómo éste va adquiriendo nuevos productos, cuáles posee actualmente y cuáles son los productos más probables a adquirir.

3.5.1. Comprensión del tema

Al igual que en las secciones anteriores debemos señalar que la comprensión del tema fue abordado previamente en el capítulo 1, por lo que se recomienda ante cualquier duda, volver a dicha sección.

3.5.2. Comprensión de los datos

El modelo oculto de Markov puede ser representado por el siguiente esquema:

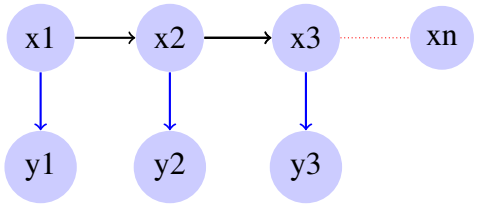


Figura 3.14: Grafo correspondiente a la representación de un modelo simplificado de Markov.
Fuente: Elaboración propia.

- Estados ocultos (X): Los estados ocultos representan los productos que adquiere un cliente en un tiempo (t) específico, y permitirán generar una secuencia de estados donde éste se moverá a través del tiempo. Por ejemplo adquiere primero una cuenta corriente, luego la tarjeta de débito asociada, junta a esta se le realiza la activación de la tarjeta de crédito, al mes siguiente solicita un crédito de consumo y un seguro asociado, etc. Cada estado representa el último producto adquirido por un cliente. Por ejemplo,

  $$X(201806) = \text{tarjeta de crédito},$$

  representa que el cliente especificado en Junio del 2018 adquirió el producto tarjeta de crédito.

- Observaciones (Y): Las salidas observables del problema representan la acción realizada por el cliente en un determinado instante de tiempo por ejemplo Y(201806) = 1 representa que el cliente especificado sí adquirió un producto en ese mes, lo cual nos genera dos posibles valores:
  - 1: Si el cliente adquiere un producto nuevo durante ese mes.
  - 0: Si el cliente NO adquirió un producto nuevo en ese mes.
- Arcos ($\rightarrow$) : Corresponde a la probabilidad asociada a cada transición y/o conexión.

Teniendo estos elementos principales claros es razonable preguntarse, ¿Cómo se establece la probabilidad de que ocurra cierta secuencia?

Las secuencias asociadas a una conducta determinada se pueden formalizar de la siguiente manera:

$$Y = y(0), y(1), \ldots, y(L-1)$$

Donde la secuencia es de tamaño L, y su probabilidad se puede obtener como

$$P(Y) = \sum P(Y|X)P(X)$$

Hacer el cálculo de la probabilidad de Y a partir de esta expresión es impráctico porque el número de posibles combinaciones de estados ocultos es muy grande, por lo que se utiliza el algoritmo *Forward – Backward* el cual disminuye considerablemente los costos de procesamiento y tiempos de cálculo.

El funcionamiento del algoritmo *Forward – Backward* se basa principalmente en 5 parámetros iniciales que sirven de imput para el algoritmo, los cuales son:

- P0 (estado inicial): Corresponde a un array con las probabilidades asociadas a la partida. En este caso se basa en el estado inicial que se encuentra el cliente, es decir que producto posee el cliente especifico a consultar.

  Por ejemplo, p(cuenta corriente) = 0.2 significa que la probabilidad de que el cliente parta adquiriendo una cuenta corriente en el banco es del 20 %.

- Matriz de emisión: Esta matriz representa las probabilidades de adquirir o no un determinado producto, es decir probabilidad de que el cliente X adquiera el producto “a” o la probabilidad que no lo adquiera.

Productos	P1	P2	P3
1	0,4	0,8	0,3
0	0,6	0,2	0,7

Figura 3.15: Matriz De Emisión en un modelo oculto de Markov.
Fuente: Elaboración propia.

- Matriz de Transición: Esta matriz representa las probabilidades por producto de pasar de un estado (producto) a otro, es decir dado que actualmente el cliente adquirió el producto “a” la probabilidad de que adquiera el “b”.

Estados	0	1	...	M
0	P00	p01	...	P0M
1	P10	P11	...	P1M
...	...	...	...	...
N	PN0	PN1	...	PNM

Figura 3.16: Matriz De Transición en un modelo oculto de Markov.
Fuente: Elaboración propia.

3.5.3. Preparación de los datos

Para poder realizar una adecuada definición de Estados y Secuencia de observaciones, es necesario explicar el análisis previo realizado sobre la data. Se debe tener una serie de consideraciones: primero que el comportamiento de los clientes es distinto según sus características. Además, la información disponible posee un gran volumen por lo que es impensable, realizar un análisis y ejecutar este algoritmo para los 56.000 clientes. Por otra parte, si solo se considerara el comportamiento histórico de 1 cliente para realizar la predicción del próximo producto, la información será demasiado limitada. Esto se debe a que solo se posee la secuencia e información de los productos con los cuales a interactuado el cliente hasta ese instante de tiempo, dejando fuera de esta lista a todos los productos que aún no ha consumido, por lo que un cliente que solo ha utilizado 3 productos durante toda su vida con el banco este no le dará posibilidad al algoritmo diversificar. Para poder realizar una diversificación de los productos es necesario poseer la información de otros clientes para poseer el máximo de productos como sea posible.

Por otra parte, realizaremos una segmentación adecuada de los clientes, de manera que se obtenga una muestra representativa de cada grupo para realizar el estudio de dicho segmento para aprender sobre el comportamiento de clientes parecidos.

El primer criterio a seleccionar es la Edad, ya que como ha sido mencionado previamente, cada segmento de edad, como se indica en la tabla 3.1 posee un comportamiento diferente producto del riesgo propio que están dispuestos a tomar las personas a diferentes tramos etarios.

Segmentación por tramo de Edad.		
Tramo Etario 0	0	18
Tramo Etario 1	18	24
Tramo Etario 2	25	34
Tramo Etario 3	35	44
Tramo Etario 4	45	54
Tramo Etario 5	55	64
Tramo Etario 6	65	...

Cuadro 3.1: Segmentación de los clientes bancarios según tramos de edad.
Fuente: Elaboración Propia.

El género o sexo de la persona es otro buen criterio de segmentación, ya que nos permite reducir nuestra población a la mitad generando solo 2 posibles combinaciones extras.

Un tercer criterio de segmentación de nuestros grupos es el Tramo de sueldo que poseen los clientes el cual definimos de la siguiente manera:

Segmentación por tramo de Sueldo.		
Tramo Sueldo 1	$0	$1.000.000
Tramo Sueldo 2	$1.000.000	$2.500.000
Tramo Sueldo 3	$2.500.000	$4.000.000
Tramo Sueldo 4	$4.000.000	$6.000.000
Tramo Sueldo 5	$6.000.000	$9.000.000
Tramo Sueldo 6	$9.000.000	...

Cuadro 3.2: Segmentación de los clientes bancarios según tramos de sueldos.
Fuente: Elaboración propia.

Finalmente, el último criterio utilizado para la segmentación es el modelo de clusterización

de clientes generado por el área de inteligencia de negocios de la institución financiera, la cual genero 6 grandes grupos, los cuales consisten en:

- CLUSTER1: Se identifican dos tipos de personas pertenecientes a este grupo, el primero es un **Cliente Joven**, poco involucrado con el banco, desinformado, el cual tiene como característica principal el poco manejo de productos bancarios. Por otra parte, nos encontramos con el **Cliente Decepcionado**, el cual se involucra poco con el banco y se alejó de éste producto de malas experiencias u otras variables.

- CLUSTER2: En esta agrupación nos encontramos con una persona austera, la cual se relaciona mucho más con productos de ahorro que con deuda, es ordenado en relación a sus gastos y no gasta más allá de sus ingresos, de aquí su definición de **Cliente Austero**.

- CLUSTER3: Son clientes intensos en la utilización de sus productos de financiamiento, se endeuda pero dentro de un rango acotado. Son laboralmente independientes, siempre están generando nuevas oportunidades de negocio, en resumen son **Clientes Arriesgados**.

- CLUSTER4: Profesionales independientes con ingresos variables, son **Clientes Endeudados**. Ocupan todos los productos de deuda que ofrece el banco para cubrir desfases en flujos de dinero.

- CLUSTER5: Son personas más permeables a ofertas de la competencia y crítico en relación a los productos que poseen en el banco. Es un **Cliente Infiel**, dispuesto a sacrificar una atención personalizada por una mejor oferta de productos.

- CLUSTER6: Es el **Cliente Fiel** al banco, le da una alta valoración a la atención personalizada ofrecida por el banco, además se encuentra conforme con los productos y servicios recibidos.

Con esto obtenemos 4 medidas para segmentar a cada cliente a las cuales les asignaremos valores numéricos enteros según tramo o cluster, por ejemplo el CLUSTER 5 será el valor 5, el tramo de edad 3, corresponderá al valor 3 obteniendo un valor entero con esta estructura:

Segmentación de clientes.			
Genero	**Cluster**	**Tramo Edad**	**Tramo Sueldo**
1: Femenino	1: CLUSTER 1	1: Tramo 1	1: Tramo 1
2: Masculino	2: CLUSTER 2	2: Tramo 2	2: Tramo 1
0: Null	3: CLUSTER 3	3: Tramo 3	3: Tramo 3
	4: CLUSTER 4	4: Tramo 4	4: Tramo 4
	5: CLUSTER 5	5: Tramo 5	5: Tramo 5
	6: CLUSTER 6	6: Tramo 6	6: Tramo 6
	0: Null	0: Null	0: Null

Cuadro 3.3: Valores numéricos correspondiente a la segmentación de clientes.
Fuente: Elaboración propia.

Con esta tabla se genera un valor numérico del estilo “2433” donde el número corresponderá a una persona de sexo Masculino, arriesgado como cliente, entre 35 y 44 años con un nivel de ingreso de 4,000,000 a 6,000,000.

Con esto se generan 1029 posibles etiquetas de segmentación de clientes, generando muestras bastante reducidas en cardinalidad pero lo suficientemente extensas para poder realizar un adecuado aprendizaje, además son bastante distintivas en cuando a comportamiento de estos, generando buenos grupos de experimentación.

3.5.4. Modelado

Una vez definido estos elementos, debemos asignarle los parámetros necesarios para realizar los cálculos con el algoritmo Forward Backward, pero nos encontramos con una nueva problemática, ya que, los parámetros para este problema son desconocidos y además son muchos, por lo cual recurriremos a nuestra información disponible.

Como sabemos por los otros dos métodos, poseemos información sobre el comportamiento de los clientes con cada producto mes a mes u nuestro principal objetivo es predecir que producto será el próximo en ser adquirido por un cliente en específico, para esto tomaremos uno de los grupos generados, por ejemplo el grupo experimental “2433” .

Listing 8 Código que genera secuencia de observaciones para un cliente en específico.

```
obs_seq=[]
states=[]
for f in range(Lista_fechas):
    print('Estoy en la fecha {} '.format(f))
    #Generamos una lista con todos los Rut de cada mes
    listarut = baser['num_iden'].
    loc[(baser['fec_fecha'] == FEC1[f])].
    values
    listarut = listarut[0:3]
    for rut in listarut:
        print('Estoy aprendiendo sobre el rut {} '.format(rut))
        arrayrut = []
        #Todos los productos en ese mes del RUT que está iterando
        lista1 = baser['cod_producto'].loc[(baser['fec_fecha'] == FEC1[f]) &
        (baser['num_iden'] == rut)].values
        for item in lista1:

            #Ahora vamos generando un Arrayrut que poseerá los productos que ya fueron
            #agregados por el RUT correspondiente de modo que si este lo vuelve a utilizar se
            #guardara un \0" en Obs Seq indicando que este cliente ya posee dicho producto,
            #en caso contrario si el cliente no lo posee se guarda un 1.

            if item in arrayrut:
                print('ya esta en arrayrut')
                states.append(item)
                obs_seq.append(0)

            else:
                states.append(item)
                obs_seq.append(1)
                arrayrut.append(item)
                print('Arrayrut tiene {} '.format(arrayrut))
```

De esta forma generamos una secuencia de observaciones con nuevos productos y productos ya adquiridos, mientras que en States tendremos una lista con los productos adquiridos en cada mes, teniendo lista nuestra base para el sistema predictivo o modelo oculto de Markov.

Una vez obtenidas estas secuencias necesitamos las probabilidades asociadas a las matrices de emisión y transición, para esto utilizaremos la técnica *Baum Welch Algorithm*. Para entrenar nuestro modelo con este algoritmo le suministramos parámetros iniciales generados de forma random. Esta probabilidad corresponde a la posibilidad de estar en el estado i en un momento determinado (t), utilizando la secuencia observada (Y) y el parámetro theta.
Luego de realizar estos cálculos actualizaremos la probabilidad del estado inicial $\pi_i^* = \gamma_i(1)$, para posteriormente actualizar la matriz de transición mediante:

Teniendo en cuenta que:

- $\sum_{t=1}^{T-1} \gamma_i(t)$ = número esperado de transiciones desde x_i.
- $\sum_{t=1}^{T-1} \xi_{ij}(t)$ = número esperado de transiciones de x_i a x_j.
- a_{ij}^* Corresponde al elemento de la fila i, columna j de la matriz de transición.

$$a_{ij}^* = \frac{\sum_{t=1}^{T-1} \xi_{ij}(t)}{\sum_{t=1}^{T-1} \gamma_i(t)}$$

Posteriormente se actualiza el número esperado de veces que las observaciones de salida han sido iguales a v_k mientras el estado sea i sobre el número total esperado de veces en el estado i, de esta forma actualizamos nuestra matriz de emisión usando:

$$b_i^*(v_k) = \frac{\sum_{t=1}^{T} 1_{y_t=v_k} \gamma_i(t)}{\sum_{t=1}^{T} \gamma_i(t)}$$

- Es posible sobre-ajustar un conjunto de datos en particular. Eso es $P(Y|\theta_{final}) > P(Y|\theta_{true})$.
- El algoritmo tampoco garantiza un máximo global.

Estas matrices y variables las vamos ajustando en una serie de pasos o etapas sucesivas que realiza el algoritmo hasta alcanzar una convergencia con el modelo, en ese momento se genera finalmente nuestro estado inicial (P_0), matriz de emisión, matriz de transición, los estados que posee nuestro modelo y las observaciones correspondientes a 1 si adquiere un producto y 0 si no.

3.5.5. Evaluación

La correspondiente comparación entre métodos será postergada al siguiente capítulo de "Experimentos y resultados".

3.5.6. Despliegue

Los algoritmos a utilizar una vez obtenido el estado inicial, serán:

Baum Welch Algorithm, para poder generar los parámetros para el HMM:

- Probabilidades de estado iniciales.
- Matriz de transición.
- Matriz de emisión.

Luego de tener el cálculo de los parámetros, se debe resolver el problema planteado y averiguar cuál es el posible estado en que se encuentra el sistema, para esto es necesario recurrir al algoritmo **Forward - Backward Algorithm**.

Por otra parte, al igual que con los métodos anteriores detallaremos en un diagrama ilustrativo algunos pasos a realizar para el método 3. Primero subdividiremos en 3 filas los pasos para poder mantener un orden en el desarrollo.
En el paso 1, realizaremos la preparación de los datos y definiremos los elementos necesarios para el método. En el paso 2 generamos parámetros, las funciones y guardamos los datos

necesarios para aplicar las Cadenas de Markov. Finalmente en 3 utilizaremos el algoritmo Run Forward Backward y le entregaremos los parámetros y variables generadas en 2, para obtener los cálculos y probabilidades de la cadena de Markov.

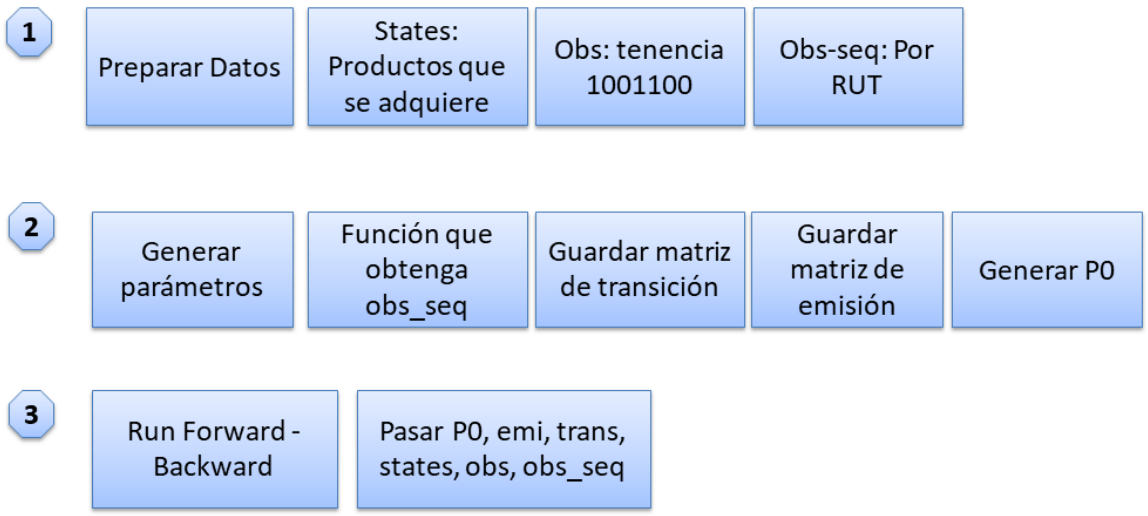


Figura 3.17: Descripción generalizada de los pasos para implementar el tercer método. Fuente: Elaboración propia.

Capítulo 4

Experimentos y resultados

Los tres métodos utilizados son muy diferentes entre sí, por lo cual se debe realizar los experimentos y el análisis por separado en cada uno. Sin embargo, hay métricas comunes entre estos, las cuáles serán consideradas como nuevas medidas de rendimiento como el tiempo empleado, la cantidad de datos que necesitaba y los recursos ocupados, estas medidas son empleadas para diferenciar resultados obtenidos entre los métodos.

- **Tiempo empleado en la carga de los datos:** Representa la cantidad de tiempo que utiliza la unidad central de procesamiento para evaluar las instrucciones del método, en oposición a la espera de las operaciones de entrada/salida que fueron proporcionadas. Este tiempo se basa principalmente en los segundos transcurridos para realizar acciones como *merge, join and concatenate* de tablas, en conjunto con otras operaciones como *group by, unstack, stack*, lectura de archivos y lectura de bases de datos, entre otros.

- **Tiempos de modelado y preparación de los datos:** Tanto los algoritmos de machine learning, los sistemas recomendadores y el método con modelos de Markov Ocultos realizan un proceso de aprendizaje a partir de los datos a ellos sometidos y, de esa manera, las máquinas son entrenadas para aprender a ejecutar diferentes tareas de forma autónoma. Luego, cuando son expuestas a nuevos datos, ellas se adaptan a partir de los cálculos anteriores y los patrones se moldean para ofrecer respuestas confiables.

Este tiempo de procesamiento utilizado es el que denominaremos “Tiempos de modelado”. Por otra parte, se debe considerar los tiempos necesarios para preparar los datos para cada método, este tiempo se considera en esta sección producto que se realizan diferentes operaciones para adaptar los datos a la estructura necesaria de cada método, como por ejemplo cálculo de valores RFM o similaridad entre productos.

- **Cantidad de datos utilizados:** Corresponde al tamaño de los archivos medidos en bytes. Es la cantidad real de espacio en disco consumida por los archivos generados por cada modelo, los cuales dependerán del sistema de archivos y varía según la cantidad de meses utilizados para seleccionar el tamaño y antigüedad de la muestra de datos, además de las variables empleadas por cada método para realizar sus respectivas predicciones.

Una vez definidas las métricas básicas de comparación a utilizar en cada método procedemos a realizar los respectivos experimentos para cada caso.

4.1. Tiempo empleado en la carga de los datos

Los tiempos de carga de datos están divididos en dos. Uno corresponde al tiempo empleado en la primera carga la cual suele ser mucho más costosa, debido a que debe realizar consultas SQL directo al servidor bancario, el cual está en constante interacción con diversos usuarios pertenecientes a la institución financiera, por lo cual estos tiempos suelen variar y tardan mucho más del tiempo normal que tomaría una consulta a un servidor que está siendo utilizado por un único usuario. Por otra parte, la segunda vez que se realiza una consulta de los datos requeridos, el tiempo baja drásticamente, debido a que dicha carga se realiza en base a un archivo CSV creado a partir de la primera consulta de datos realizada. En este archivo se guardan los datos con el objetivo de minimizar los tiempos necesarios para realizar predicciones en el futuro, por lo cual es recomendado utilizar archivos pickle o CSV para reducir estos tiempos.

En el caso de los archivos pickle o CSV, los tiempos de carga equivalen principalmente a un dataframe que contiene los datos de fecha de adquisición de productos, fecha de cierre del producto, montos y operaciones de cada cliente identificados con su respectivo RUT.

4.1.1. Tiempo empleado en la carga de los datos para el Sistema Recomendador clásico

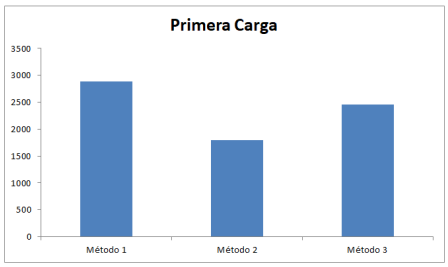
Al utilizar sistemas recomendadores clásicos, el tiempo de la primera carga corresponde a 48 minutos o 2880 segundos. Pero luego de guardar dichos datos, este se ve reducido drásticamente a solo 30.9 segundos debido a la facilidad que tiene la máquina para leer archivos CSV por sobre las consultas SQL. Cabe señalar que este tiempo solo corresponde a la operación de extraer la data necesaria desde las fuentes de información.

4.1.2. Tiempo empleado en la carga de los datos para Aprendizaje supervisado

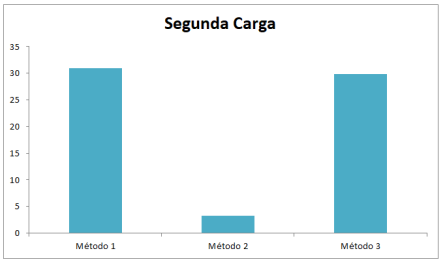
Los tiempos empleados tanto para la primera carga como para la carga desde archivos CSV son de 5 minutos al realizar cargas de solo 4 meses, mientras si se realiza una carga historia de 12 meses estos se elevan a 30 minutos o 1.800 segundos para la consulta SQL y 3,29 segundos en el caso de la carga de archivo CSV.

4.1.3. Tiempo empleado en la carga de los datos para el modelo oculto de Markov

En el caso del modelo oculto de Markov, los tiempos de carga se elevan a 41 minutos para la carga inicial equivalente a 2460 segundos, mientras que en la segunda carga equivale a 29,8 segundos.



(a)



(b)

Figura 4.1: Tiempos de carga de datos para cada método respectivamente, donde (a) corresponde a los de la primera carga y (b) a las cargas desde archivos CSV's.
Fuente: Elaboración propia.

4.2. Tiempos de modelado y preparación de los datos

Los Tiempos de modelado y preparación de los datos se componen principalmente de los segundos empleados en estructurar los datos de una forma adecuada para las funciones que realizaran los cálculos respectivos correspondientes a cada método, más el tiempo empleado en realizar dichos cálculos o predicciones.

4.2.1. Tiempos de modelado y preparación de los datos para el sistema recomendador

El Tiempo promedio de recomendación para el primer método se compone de 7 minutos 9 segundos para preparar la base para la recomendación, es decir emplea 429 segundos en realizar las operaciones correspondientes al cálculo de valores de recencia, frecuencia y monto por clientes. Además, tarda 20 segundos más para crear la recomendación para solo un cliente, lo cual nos suma un total de **449 segundos** para generar la recomendación de productos candidatos para un cliente en específico para el próximo mes.
Es esencial distinguir que la recomendación solo toma en cuenta a 1 cliente. Si se desea preparar una base de recomendaciones para muchos clientes al mismo tiempo, se necesita tener

en cuenta otras operaciones a aplicar en el código: Crear listas de RUT a recorrer, concatenar resultados, entre otros. Por tanto se requiere de más tiempo para el total de recomendaciones. Sin embargo, el tiempo medio baja debido a las operaciones en común entre los cálculos de cada cliente.

4.2.2. Tiempo de recomendación para aprendizaje supervisado

En este caso los tiempos se ven reducidos drásticamente, ya que para identificar cuál es la relación para una determinada clasificación de clientes se tarda en promedio 20 segundos en preparar la base para la recomendación, lo cual implica el tiempo necesario para entrenar el clasificador. A los 20 segundos anteriores debemos agregarle 14 segundos para generar la recomendación utilizando regresión logística como clasificador.
Al igual que en el método anterior se debe tener algunas consideraciones, como por ejemplo que estas recomendaciones son hechas para solo un segmento de clientes, el cual fue seleccionado previamente a partir de su rango etario, rango de sueldo, género y cluster bancario al que pertenecen. Tomando esto en cuenta el tiempo total de aprendizaje equivale a **34 segundos** en total.

4.2.3. Tiempos de recomendación para el modelo oculto de Markov

Para el método de Markov debemos tener en cuenta un promedio de 2 horas 56 min en preparar la base para la recomendación, lo cual implica realizar los cálculos de los valores correspondientes a las matrices de emisión y transición, es decir es el tiempo empleado en estimar los parámetros necesarios para la cadena de Markov. Sumado a esto, se adicionan 2 minutos con 17 segundos para crear la recomendación o un total de 137 segundos. Sin embargo para mantener las mismas métricas para la comparación con los otros métodos utilizados debemos sumar el tiempo para generar una recomendación, con el de preparación obteniendo un total de **10.687 segundos** para poder generar como resultado las recomendaciones de productos candidatos para un cliente en específico.

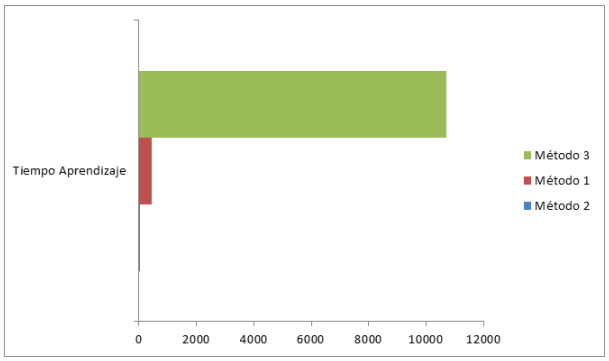


Figura 4.2: Gráfico correspondiente a los tiempos de aprendizaje para cada método.
Fuente: Elaboración propia.

4.3. Cantidad de datos utilizados

Por otra parte también debemos considerar la cantidad de datos utilizados, los cuales consideraremos como el tamaño total utilizado por los archivos QVD's generados luego de la primera carga realizada por cada método respectivamente.

4.3.1. Cantidad de datos empleados en el sistema recomendador

En base a 35 meses, el tamaño total equivale a 1740 Megabytes, mientras más meses utiliza, mejores son las predicciones realizadas.

4.3.2. Cantidad de datos empleados en el aprendizaje supervisado

En base a 4 meses el tamaño total del archivo equivale a 146 Megabytes, por otra parte al tomar los 12 meses para mejorar la tasa de aprendizaje el tamaño sube a 516 Megabytes.

4.3.3. Cantidad de datos empleados en el modelo oculto de Markov

En el caso de los modelos ocultos de Markov, en base a los últimos 24 meses el tamaño alcanzado por el archivo equivale a 893 Megabytes. Mientras más meses utiliza, mejores son las predicciones realizadas por este método, pero su tamaño aumenta drásticamente y la curva de aprendizaje es menor cada vez, es decir las mejoras que se obtenían al agregar un horizonte temporal de 12 meses ya no son las mismas, pese a mejorar las predicciones es muy leve la mejora en comparación a la cantidad de datos que se requiere.

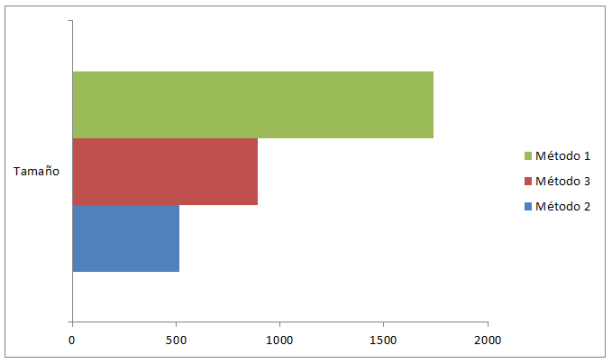


Figura 4.3: Gráfico correspondiente al tamaño de los archivos QVD's para cada método. Fuente: Elaboración Propia.

4.4. Similaridad entre los clientes y entre productos

Una vez que realizamos mediciones de estas tres métricas básicas de comparación, procederemos a realizar un análisis de la calidad de los métodos.

Para poder determinar clientes con gustos afines y poder realizar predicciones adecuadas, los algoritmos tradicionales utilizan métricas de similaridad. Sin embargo en algunos casos es más útil realizar cálculo de similitud entre ítems o en nuestro caso productos que comparar la similitud entre clientes. Para esto estableceremos algunos criterios de comparación entre los métodos, sin embargo evaluar un sistema recomendador no es tarea sencilla. Hasta la fecha,

no existe una métrica más importante que otra, todo depende del contexto.

A modo de lograr un escenario macro de lo realizado, se evalúa el comportamiento de los sistemas o métodos establecidos frente a indicadores como serendipia, exactitud, precisión, alcance y que el método sea explicable.

4.4.1. Serendipia

La serendipia es un descubrimiento o un hallazgo afortunado e inesperado que se produce cuando se está buscando una cosa distinta o se espera algo diferente. Este término también es conocido como **serendipity** introducido en el campo de los sistemas recomendadores por Herlocker [7] para referirse a los productos que no son utilizados habitualmente pero resultan de mucha utilidad.

En el caso de las predicciones basadas en **ítems similares**, no se permite recomendar diferentes productos para un mismo cliente. Ya que si usualmente un determinado tipo de cliente consume o utiliza habitualmente el mismo tipo de producto, el rating para este producto estará muy por sobre los otros, por lo que lo más probable es que a futuro se recomiende el mismo tipo de producto.

En cambio, en las de **clientes similares**, al relacionar los gustos entre clientes, hay mayor probabilidad de realizar recomendaciones diversas. Debido a que, aunque un cliente utilice constantemente los mismos tipos de productos, puede que alguno de estos productos se relacione con los productos utilizados por otro tipo de cliente.

En el caso de los **sistemas recomendadores**, no obtuvimos hallazgos afortunados o inesperados, en general los clientes se comportan de forma bastante estándar y los productos no salen de lo común. Por otra parte, se debe considerar que solo de mostraban los 3 principales productos a recomendar, esto debido a que no se suelen hacer campañas, ni ofrecimiento de más de 3 productos para un mismo cliente.

En el caso del método basado en **Domeniconi**, el hallazgo se basó en el comportamiento de los productos, debido a que no se esperaba obtener las fugas de productos o en otras palabras los productos que el cliente dejara de utilizar en el siguiente periodo. Algunos casos que resaltan son los productos de inversión como: Depósitos a plazo y acciones nacionales como se puede ver en la figura 4.4. Para estos productos se debe realizar un seguimiento constante para ver si el comportamiento de fuga es normal o está fuera de lo común.

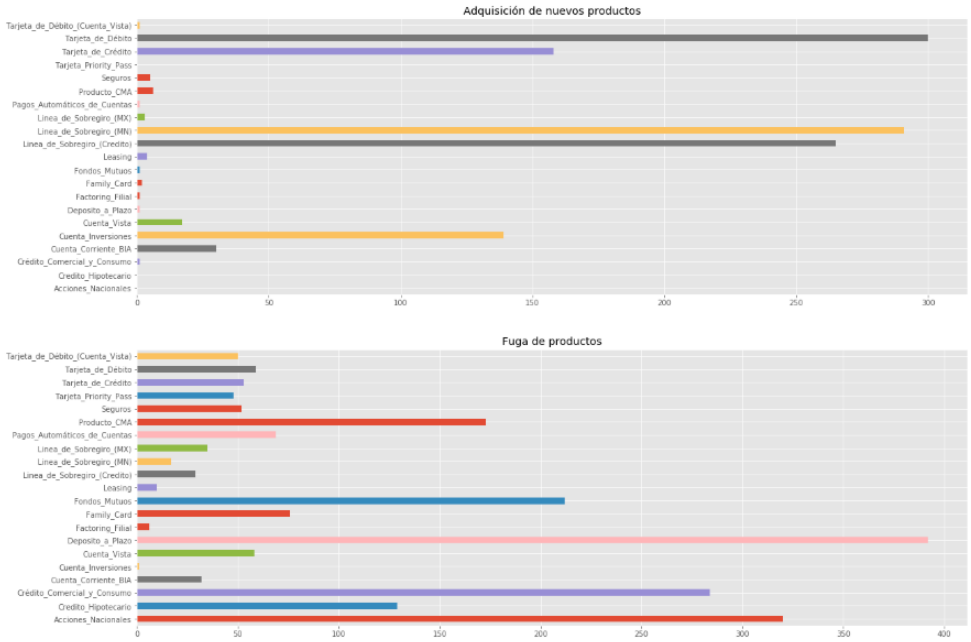


Figura 4.4: Gráfico correspondiente a la fuga y adquisición de productos, utilizando el método de Domeniconi.
Fuente: Elaboración Propia.

Finalmente, en el caso de las **Cadenas ocultas de Markov**, no se obtuvieron hallazgos que salieran de lo común y los clientes se comportaron acorde a la habitualidad esperada, donde los productos más utilizados son los que se adquieren al momento de abrir una cuenta, como: Tarjetas de débito y crédito, la cuenta corriente y las líneas de sobregiro.

4.4.2. Exactitud

En los métodos de recomendación, la exactitud, suele depender de la proporción que se posee entre clientes y productos.

Autores como Fouss, Pirotte, Renders y Saerens [21], señalan que en casos donde el número de usuarios, en nuestro caso clientes, es mucho mayor que el número de ítems o productos, es conveniente utilizar un enfoque basado en productos.

Esto se debe a que si m es el número de clientes, n el número de productos y R la cantidad de ratings por clientes, R/n corresponde al promedio de ratings por producto. De este modo, si hay más clientes que productos, el número promedio de ratings por productos es mayor, por lo que se tendrá a disposición una mayor cantidad de ratings.

$$m > n \Rightarrow \frac{R}{m} < \frac{R}{n}$$

4.4.3. Método explicable

Un aspecto a considerar en el contexto financiero es que se pueda **explicar** la recomendación, debido a que no es suficiente entregar una lista de productos para satisfacer al usuario bancario, también se requiere justificar las recomendaciones realizadas.

Una de las ventajas que posee el enfoque **basado en productos** es que permite explicar fácilmente una recomendación, ya que los ítems usados para elaborar la predicción pueden ser presentados al usuario argumentando las semejanzas. Sumado a esto se pudo realizar una representación visual de fácil entendimiento como lo es el mapa de calor (Figura [3.7]) dónde podemos apreciar la semejanza o correlación entre productos.

Por otra parte, el enfoque **basado en usuarios**, es menos manejable en este aspecto debido

a que no podemos entregar como justificación los gustos de usuarios que desconocemos. Es más, legalmente si quisiéramos presentarle a un cliente los gustos de un cliente similar en base al cual se le realizó la recomendación, estaríamos frente a una grave infracción en la ley de privacidad de datos.

4.5. Experimentos a realizar

En el caso del sistema recomendador, los experimentos consistirán en generar una muestra representativa de una segmentación de clientes. Posteriormente a esta muestra se le generar el correspondiente arreglo de productos a recomendar por cada cliente, basado en información histórica, teniendo como bases meses previos a la fecha de muestreo de manera que ya se contenga el comportamiento o adquisición de productos que realizó cada cliente al mes siguiente. Dichos productos se comparan con el valor real o los verdaderos productos que adquirieron cada cliente al mes posterior y se ve si estos coinciden o no acorde a medidas que determinaran la calidad de las recomendaciones.

En el caso de los sistemas de recomendación colaborativos, nos basaremos en dos métricas principales para realizar las pruebas y análisis de la calidad de las recomendaciones que éste genera. Estas métricas son [5]:

- **RECALL**: de todos los que me gustan, cuántos me ha mostrado el recomendador.
- **PRECISION**: de todos los que me ha mostrado el recomendador, cuántos me gustan.

Figura 4.5: Descripción del cálculo para Recall y Precisión.
Fuente: Macarena Estevéz, Junio 2016.

Este tipo de pruebas se hacen probando el recomendador con un conjunto de entrenamiento y comprobando con un conjunto de test. Volviendo a nuestro ejemplo, supongamos que a nosotros nos gustan los Chupetes (indicamos "Me gusta" con un *) y el recomendador nos muestra Toallitas y Chupetes (indicamos "Me recomiendan" con un +). Vemos los cálculos en la Figura 4.2.

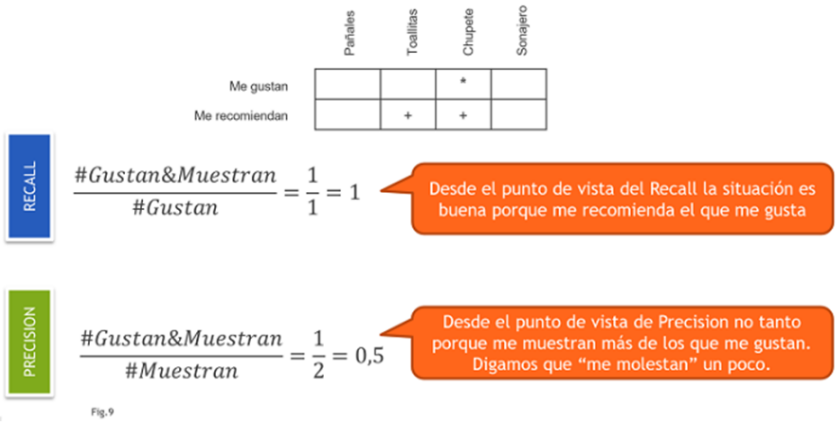


Figura 4.6: Ejemplo del cálculo para Recall y Precisión.
Fuente: Macarena Estevéz, Junio 2016.

Llevando esto al caso del sistema recomendador tomaremos un mes en particular en esta caso Noviembre del 2018 y realizaremos los cálculos de recomendaciones como si nos encontráramos en Octubre del 2018, es decir no sabemos nada sobre la adquisición de productos que realizarán los clientes bancarios en Noviembre y generaremos recomendaciones

para cada usuario que haya adquirido un producto en Noviembre del 2018. Es decir, realizaremos un cruce entre la información real de adquisición de productos en dicha fecha y las recomendaciones generadas para dichos clientes a partir de los datos que teníamos hasta Octubre del 2018, para posteriormente calcular Recall y Precisión del método.

Matriz de confusión.		
	Predicción: No cambia	**Predicción:** Cambia
Real: No cambia	Se predijo que no hay cambios en los productos de los clientes y realmente no los hubieron. **(TN)**.	Se predijo que se adquiría un producto nuevo, sin embargo el cliente no adquirió ningún producto. **(FP)**.
Real: No cambia	Se predijo que no se adquiría un producto pero si se adquirió producto **(FN)**.	Se predijo adquisición de un nuevo producto y el cliente adquirió nuevos productos **(TP)**.

Cuadro 4.1: Matriz de confusión, donde las clases predichas están representadas en las columnas de la matriz, mientras que las clases reales están en las filas de la matriz.
Fuente: Elaboración Propia.

Utilizando métodos provistos por scikit-learn, como:

- confusion_matrix(y_true, y_pred).
- accuracy_score(y_true, y_pred).
- precision_score(y_true, y_pred).

Podemos obtener las medidas descritas con anterioridad como el **recall** y **precisión**, en conjunto a la matriz de confusión, que nos indicara la cantidad de falsos negativos y verdaderos negativos de cada método.

A continuación, se mostraran los resultados correspondiente a las matrices de confusión de cada método:

Matriz de confusión Recomendador clásico.		
	Predicción: No cambia	**Predicción: Cambia**
Real: No Cambia	1252	987
Real: cambia	28	957

Cuadro 4.2: Valores numéricos correspondiente a la matriz de confusión del primer método. Fuente: Elaboración propia.

Matriz de confusión Domeniconi.		
	Predicción: No cambia	**Predicción: Cambia**
Real: No cambia	947	480
Real: Cambia	143	1654

Cuadro 4.3: Valores numéricos correspondiente a la matriz de confusión del segundo método. Fuente: Elaboración propia.

Matriz de confusión HMM.		
	Predicción: No cambia	**Predicción: Cambia**
Real: No cambia	982	445
Real: Cambia	271	1526

Cuadro 4.4: Valores numéricos correspondiente a la matriz de confusión del tercer método. Fuente: Elaboración propia.

Posteriormente podemos calcular el recall y precisión de cada método, tanto para el caso en que un cliente cambia su estado, es decir adquiere o se desase de un producto bancario, como también el caso en que no sufre cambio alguno. Cabe destacar que para el primer método solo se puede apreciar la adquisición de productos y no la fuga, debido a la naturaleza misma

del método, quedando representados en las siguientes tablas:

Recall y **Precision**.	
Recall: Cambia	**Recall: No cambia**
97,16 %	55,92 %
Precision: Cambia	**Precision: No cambia**
49 %	98 %

Cuadro 4.5: Valores porcentuales para el Recall y precisión en **Recomendador clásico**.
Fuente: Elaboración propia.

Recall y **Precision**.	
Recall: Cambia	**Recall: No cambia**
92,04 %	66,36 %
Precision: Cambia	**Precision: No cambia**
78 %	87 %

Cuadro 4.6: Valores porcentuales para el Recall y precisión en **Domeniconi**.
Fuente: Elaboración propia.

Recall y **Precision**.	
Recall: Cambia	**Recall: No cambia**
84,92 %	68,82 %
Precisión: Cambia	**Precisión: No cambia**
77 %	78 %

Cuadro 4.7: Valores porcentuales para el Recall y precisión en **HMM**.
Fuente: Elaboración propia.

En el caso del sistema recomendador clásico podemos apreciar en la matriz de confusión que

posee un elevado error de tipo 1, es decir se predice en muchos casos que un cliente adquirirá un producto, sin embargo este no lo adquiere, mientras que el error de tipo 2 es bastante bajo. Por otra parte, al analizar el recall y precisión por separado, podemos observar que la precisión al decir que se adquirirá un producto es bastante baja, de hecho menor a 50 %, es decir ni siquiera la mitad de los clientes que se creía que adquirirían un producto realmente lo hicieron.

Para el método basado en Domeniconi, podemos ver que la precisión de cambio aumenta hasta un 78 %, pese que el recall disminuye respecto al primer método, todos los indicadores están mucho más equilibrados, incluso bajo el error de tipo 1 o falsos positivos.

Por otra parte, en el caso del tercer método las métricas también se equilibran, sin embargo el método basado en Domeniconi posee un mejor desempeño en lo global.

Tomando en cuenta la matriz de confusión, el recall y precisión de cada método, es fácil determinar que el segundo y tercer método poseen un mejor desempeño en lo global, sin embargo, si se busca predecir, a partir de un conjunto de datos, si una campaña de marketing será exitosa, es preferible que el modelo tenga más errores de **Tipo I** que de **Tipo II**. Ya que esto implica que será mejor equivocarse al contactar a un cliente que no adquirirá algún producto, que equivocarse al perder una venta por no haber contactado a una persona que sí lo compraría, pero el modelo la clasificó en sentido contrario. Es en base a esto que podemos determinar que el segundo método basado en lo planteado por Domeniconi es el que obtiene la ventaja sobre los otros.

Otro análisis necesario es ver el comportamiento de los métodos recomendadores en la práctica, más allá de los números, para esto en primer lugar empecemos viendo algunos resultados generales obtenidos con cada método, para algún usuario o cliente bancario en particular. Para esto tomamos un sujeto candidato e ingresaremos su número identificador o RUT y veremos qué predicciones realiza cada uno de los métodos mencionados para él en un instante de tiempo en común.

En el caso del sistema recomendador, podemos apreciar según la figura 4.7 que se recomienda ofrecerle Tarjeta de Débito dentro de los principales productos.

	producto_title
22	Tarjeta_de_Débito
14	Linea_de_Sobregiro_(MN)
13	Linea_de_Sobregiro_(Credito)
10	Family_Card
4	Cuenta_Corriente_MN

Figura 4.7: Recomendación generada por el método 1 para un cliente bancario.
Fuente: Elaboración propia.

Al utilizar el método 2 el producto recomendado es la Tarjeta de Débito como se aprecia en la figura 4.8.

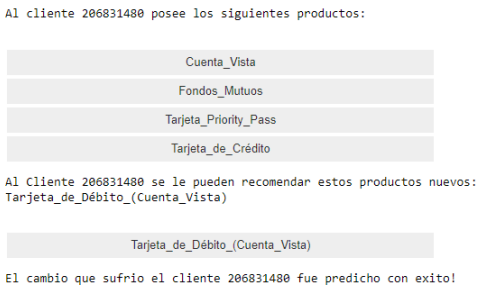


Figura 4.8: Recomendación generada a partir del método 2 utilizando Regresión logística.
Fuente: Elaboración Propia.

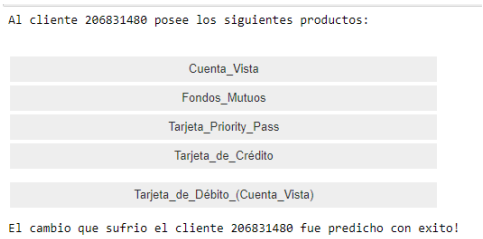


Figura 4.9: Recomendación generada a partir del método 2 utilizando kNN.
Fuente: Elaboración Propia.

Mientras que en el caso de las cadenas ocultas de Markov la acción con mayor probabilidad de ocurrencia corresponde efectivamente a la tarjeta de débito.

Este mismo fenómeno se repitió en los clientes que sufrieron cambios durante los meses utilizados para la prueba, ya sea en adquisición como fuga de productos, obteniendo mejores predicciones a partir del método 2 que logró identificar correctamente un 92 % de los casos al utilizar KNN (considerando 3.224 resultados), cabe destacar que este rendimiento es más bajo que el 98 % (considerando 370.152 resultados) aproximado de accuracy mostrado anteriormente, debido a que solo toma en cuenta casos donde el cliente sufre cambios (fuga o adquisición de productos).
Por otra parte el método 1 solo identifica la adquisición de productos. En el caso del método 3 logra identificar un 84 % de los casos de adquisición y fuga de productos adecuadamente, mientras que el método 1 logro identificar un 97 % de los casos que representan la adquisición de nuevos productos.

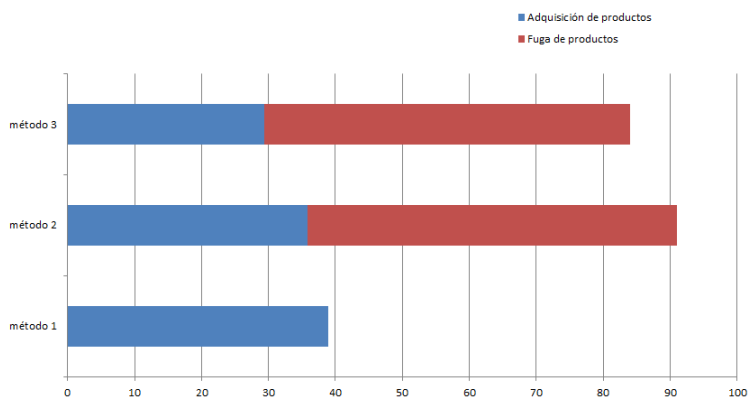


Figura 4.10: Rendimiento logrado por cada método considerando el total de cambios ocurridos en el mes de prueba.
Fuente: Elaboración Propia.

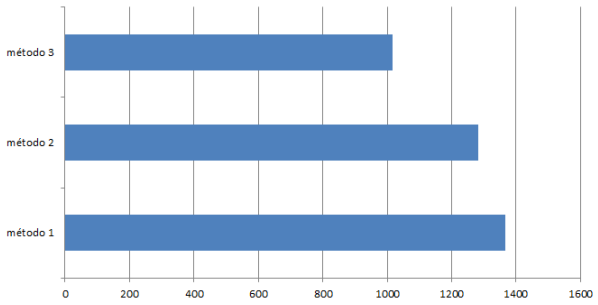


Figura 4.11: Rendimiento logrado por cada método considerando tomando como muestra solo la adquisición de productos.

Fuente: Elaboración Propia.

Conclusiones

Conclusiones de la investigación

Se logró implementar tres Sistemas Recomendadores para productos bancarios en base a:

1. Un método clásico de filtrado colaborativo.
2. Una propuesta reciente de la literatura (Domeniconi).
3. Un método clásico de aprendizaje no supervisado pero no tradicionalmente utilizado para recomendaciones.

Para esto, se realizó un arduo análisis previo para determinar cuál era el tipo de filtro más adecuado, sobre todo considerando la naturaleza inusual del problema, ya que, como se mencionó, la cardinalidad de los productos era mucho menor que la del universo de clientes que poseen las instituciones bancarias. Sin embargo, pese al desafío de estar en un escenario desconocido para el filtrado que realizan estos sistemas, nos encontramos con resultados buenos en cuanto a recomendaciones para clientes.

El método que obtuvo un rendimiento destacado tanto en tiempo como recursos fue el método 2 adaptado a partir de las técnicas empleadas por Domeniconi para las mutaciones en cadenas de genomas. Este rendimiento sobresaliente no es casualidad. El mismo Domeniconi señala que debe lidiar con la falta de información del mes anterior para su experimento, debido a que estos no podían ser almacenados mes a mes debido a su gran volumen. Esto

lo obligó a generar un método capaz de adaptarse a estas adversidades, lo cual es replicado en nuestro enfoque derivando resultados sobresalientes por sobre los otros dos métodos propuestos en cuanto a rendimiento se refiere.
Por otra parte, este método no solo logra predecir adquisiciones de productos candidatos, si no, que también logra identificar clientes que poseen una tendencia a deshacerse de algunos productos. Esto se debe a que este método no genera una recomendación como tal, más bien genera una predicción del comportamiento de los clientes bancarios, basado en sus características. Esta característica genera una gran ventaja por sobre los otros métodos debido a que permite orientar las campañas de Marketing a un grupo específico de clientes ahorrando recursos tanto en tiempo como monetarios. Lo anterior se debe principalmente a que, al tener seleccionado el grupo de clientes a los que les interesara un producto, solo debemos enviar la publicidad respectiva a este grupo, en vez de a todo los clientes, ahorrando costos monetarios de envió de correos y SMS.

Uno de los objetivos propuestos era determinar en qué escenarios es conveniente el uso de las diferentes técnicas propuestas. Para esto es necesario señalar que fue evidente que la utilización de un sistema recomendador, es mucho más complicado de utilizar para una masa de clientes. Mientras que el método de Domeniconi o aprendizaje supervisado es el ideal para esta tarea. Sin embargo, si se trata de una consulta por un cliente en particular, ambos métodos proporcionan respuestas adecuadas en cuanto a tiempos de procesamiento y accuracy del método. Esta ventaja que proporciona a la industria, además de su eficiencia computacional, lo hace sumamente atractivo para áreas específicas de la banca como inteligencias de negocios o Marketing, debido a que les proporciona una herramienta que no solo les entrega una respuesta adecuada a sus necesidades para realizar cross-selling, sino que también genera dicha respuesta en un tiempo bastante adecuado para la industria. Todo esto hace que el método 2 se convierta en una alternativa práctica y competitiva para el marketing bancario.

Además debemos mencionar que fue importante considerar nuevas medidas de rendimiento, como fue el tiempo empleado, la cantidad de datos que necesitaba y los recursos ocupados para el procesamiento, los cuales se convirtieron en nuestras principales herramientas para

diferencias resultados obtenidos entre métodos. A partir de estas métricas podemos identificar que el primer método de recomendadores clásicos es el que obtiene el segundo mejor rendimiento.

Finalmente, se crearon cadenas de Markov ocultas, las cuales nos dan más seguridad al momento de calcular probabilidades, sin embargo debido a su gran tiempo de procesamiento y alto costo en recursos, es inviable para una institución bancaria que requiere de la generación de campañas en un tiempo acotado, generalmente de un par de horas. Por este motivo, pese a ser un método con resultados bastante parecidos a los otros, queda descartado para su utilización en instituciones que requieren procesos mucho más rápidos. Además, se debe mencionar que es necesario estar realizando entrenamientos constantemente debido a la naturaleza cambiante de los mercados y las características dinámicas que poseen los clientes en el ámbito económico. De otra forma el método sería más rápido que KNN, ya que, una vez entrenado sólo deberíamos predecir, en cambio KNN busca similaridad en toda la base de datos.

Finalmente, con la respuesta entregada por el segundo método, podemos generar una política adecuada capaz de determinar de forma correcta un mapa de viajes de cada cliente, ya que poseeremos una respuesta adecuada sobre qué hacer con las campañas para dicho cliente: El sistema determinará su estado actual y si su tendencia es a permanecer igual (Cliente conforme), adquirir un nuevo producto (Cliente a gusto e interesado) o deshacerse de un producto (disconformidad).

Trabajo A Futuro

A futuro algunos de los trabajos a realizar son los relacionados con experimentos y la utilización de estas mismas técnicas con algunas modificaciones.
Por ejemplo, sería bastante interesante probar cada método con ponderaciones de las medidas

RFM utilizadas en el sistema recomendador tradicional del método 1. Otra de las modificaciones interesantes a realizar sería desarrollar estos métodos en otros lenguajes de programación como C y C+ para ver cómo mejora el rendimiento de cada método y si por motivos de lenguaje los tiempos de procesamiento y carga mejoran o incluso cambian de lo obtenido en este documento.

Resultaría también interesante realizar un análisis luego de un tiempo de utilización de los métodos. Por ejemplo para el primer método sería interesante generar encuestas que midan la satisfacción del cliente al ofrecerle una cierta gama de productos generados por el sistema recomendador y en base a esto asignarle un puntaje o calificación a los productos. Esto permitiría que se arme una base con calificaciones generadas por los mismos clientes y no a partir de sus comportamiento, de modo que se pueda comparar un sistema recomendador colaborativo basado en comportamiento (RFM) en contraste a un sistema recomedador que use calificaciones reales generadas por los usuarios a los productos que posee la institución.

Mientras que para el segundo método, debemos verificar en base a la historia generada luego de meses de su utilización, datos cuantificables que nos muestren los beneficios que ha obtenido la institución respecto al año anterior en que no se utilizaba dicho método.

Finalmente, teniendo definidas las nuevas implementaciones debemos ser capaces de generar valor para los clientes a partir de ellas, para esto se propone analizar el ciclo de compra o de adquisición de productos desde la perspectiva del cliente, teniendo en en cuenta cómo este se siente, qué expectativas tiene o qué idea sobre la institución se lleva luego de finalizado el ciclo. Permitiendo que se plasme en un mapa más conocido como mapa de viaje de cliente [10], cada una de las etapas, interacciones, canales y elementos con los que interactúa un cliente.

Capítulo 5

Anexos

5.0.1. Imágenes con tiempos de carga y procesamiento

Figura 5.1: Recorte del tiempo tomado para realizar la primera carga de datos necesarios para el método 1.
Fuente: Elaboración Propia.

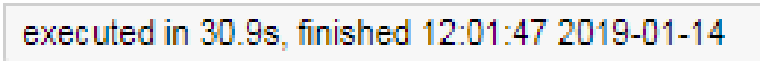


Figura 5.2: Se muestra el tiempo utilizado al cargar los datos para el método 1 desde un archivo CSV.
Fuente: Elaboración Propia.

Figura 5.3: Tiempo de carga de datos del primer método. primer método.
Fuente: Elaboración Propia.

Figura 5.4: Tiempo de carga del primer método considerando algunos datos extra como fecha de apertura de productos y de cierre.
Fuente: Elaboración Propia.

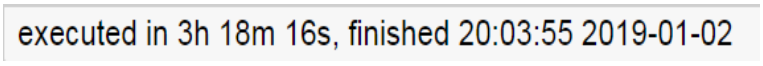


Figura 5.5: Tiempo de ejecución del algoritmo Forward-Backward correspondiente al tercer método.
Fuente: Elaboración Propia.

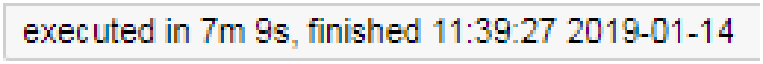


Figura 5.6: Tiempo de cómputo de recencia, frecuencia y monto para cada usuario, sumado a almacenar dicho valor RFM en una matriz.
Fuente: Elaboración Propia.

Bibliografía

[1] Gediminas Adomavicius and YoungOk Kwon. New recommendation techniques for multicriteria rating systems. *IEEE Intelligent Systems*, 22(3), 2007.

[2] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge & Data Engineering*, (6):734–749, 2005.

[3] Robin Burke. The adaptive web. *Berlin, Heidelberg: Springer-Verlag*, pages 377–408, 2007.

[4] Giacomo Domeniconi. *Data and Text Mining Techniques for In-Domain and Cross-Domain Applications*. PhD thesis, alma, 2016.

[5] Macarena Estevéz. Macarena estevéz: Un acercamiento a los sistemas de recomendaciÓn. *PLoS One*, 2016.

[6] G David Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[7] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53, 2004.

[8] Will Hill, Larry Stead, Mark Rosenstein, and George Furnas. Recommending and evaluating choices in a virtual community of use. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 194–201. ACM Press/Addison-Wesley Publishing Co., 1995.

[9] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.

[10] Tharon Howard. Journey mapping: A brief overview. *Communication Design Quarterly Review*, 2(3):10–13, 2014.

[11] Biing Hwang Juang and Laurence R Rabiner. Hidden markov models for speech recognition. *Technometrics*, 33(3):251–272, 1991.

[12] kdnuggets. CRISP-DM, still the top methodology for analytics, data mining, or data science projects, 2019.

[13] James M Keller, Michael R Gray, and James A Givens. A fuzzy k-nearest neighbor algorithm. *IEEE transactions on systems, man, and cybernetics*, (4):580–585, 1985.

[14] Oliver D King, Rebecca E Foulger, Selina S Dwight, James V White, and Frederick P Roth. Predicting gene function from patterns of annotation. *Genome research*, 13(5):896–904, 2003.

[15] Theodore Levitt. The globalization of markets. *Readings in international business: a decision approach*, 249, 1983.

[16] Kenneth C Loudon, Jane P Loudon, and R Dass. Management information systems. *Prentice Hall Int, Inc*, 2008.

[17] Sangil Martínez and A Jordi. Crm¿ filosofía o tecnología? mitos y realidades de a orientación al cliente. 2008.

[18] John Miglautsch. Application of rfm principles: What to do with 1–1–1 customers? *Journal of Database Marketing & Customer Strategy Management*, 9(4):319–324, 2002.

[19] Federico Minneci, Damiano Piovesan, Domenico Cozzetto, and David T Jones. Ffpred 2.0: improved homology-independent prediction of gene ontology terms for eukaryotic protein sequences. *PLoS One*, 8(5):e63754, 2013.

[20] Esmaeil Nikumanesh and Amir Albadvi. Customer's life–time value using the rfm model in the banking industry: a case study. *International Journal of Electronic Customer Relationship Management*, 8(1-3):15–30, 2014.

[21] Alain Pirotte, Jean-Michel Renders, Marco Saerens, et al. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on Knowledge & Data Engineering*, (3):355–369, 2007.

[22] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work*, pages 175–186. ACM, 1994.

[23] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. Collaborative filtering recommender systems. In *The adaptive web*, pages 291–324. Springer, 2007.

[24] Upendra Shardanand and Pattie Maes. Social information filtering: Algorithms for automating"word of mouth". In *Chi*, volume 95, pages 210–217. Citeseer, 1995.

[25] Babak Sohrabi and Amir Khanlari. Customer lifetime value (clv) measurement based on rfm model. 2007.

[26] Ying Tao, Lee Sam, Jianrong Li, Carol Friedman, and Yves A Lussier. Information theory applied to the sparse gene ontology annotation network to predict novel gene function. *Bioinformatics*, 23(13):i529–i538, 2007.

[27] Juan Tornero Lucas et al. Machine learning: modelos ocultos de markov (hmm) y redes neuronales artificiales (ann). 2017.

[28] Nicolás Torres. Sistemas de recomendación basados en métodos de filtrado colaborativo, 2015.

[29] Rüdiger Wirth and Jochen Hipp. Crisp-dm: Towards a standard process model for data mining. In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, pages 29–39. Citeseer, 2000.

[30] Shun-Zheng Yu and Hisashi Kobayashi. An efficient forward-backward algorithm for an explicit-duration hidden markov model. *IEEE signal processing letters*, 10(1):11–14, 2003.